

ИНФО

ИНФОРМАТИКА И ОБРАЗОВАНИЕ

2 1995

ИЦИСО

**НАУЧНЫЙ ЦЕНТР
ПРОГРАММНЫХ
СРЕДСТВ ОБУЧЕНИЯ**

Российская Федерация
Москва,
Б. Коммунистическая, 9-а

Телефон: 272 26 71,
214 46 49, 214 77 84
214 54,33
Факс: 271 04 28

«Вы любите сказки?»

Что учителя говорят о компьютерах Макинтош компании Apple Computer



«Макинтош устранил барьеры, которые все эти годы, несмотря на шум о компьютеризации, отделяли учителей начальной школы, биологии, математики, истории от богатства образов, информационных массивов и моделей, которые теперь становятся доступными учащимся... Мы увидели, что объем образовательного программного обеспечения, разработанного для компьютера Макинтош, намного больше, чем для других компьютеров, например, IBM-совместимых...»

И.Н. Бахметьева, шк. №57, г. Москва



«Вы любите сказки? Так вот, такой волшебной сказкой стал Макинтош в нашей школе. Занимаетесь ли вы музыкой, рисуете или пытаетесь решить более серьезные задачи вашей школьной жизни, лучшего друга и помощника, чем Мак, вам не найти...»

Мы долго решали на какие компьютеры просить денег у Отдела народного образования Южного округа. И вот теперь, поработав на Макинтоше два месяца, мы поняли, что не обманулись в своих ожиданиях... Да, Мак—это сказка!»

Т.В. Ерохова,
методист Южного округа по информатике,
учитель шк. № 550 г. Москвы



«Я год проработала на IBM PC, но Макинтош изменил моё представление о компьютерах вообще. Возможности, появляющиеся здесь у учителя и ученика просто поразительны. Единственное моё желание теперь—иметь эти компьютеры у себя в школе.»

И.А. Макарова, гимназия №2, г. Саранск

Apple ™

Научно-методический журнал
Учрежден Министерством
образования РФ
и коллективом редакции

Издается с августа 1986 г.
Выходит шесть раз в год

СОДЕРЖАНИЕ

10 ЛЕТ ШКОЛЬНОЙ ИНФОРМАТИКЕ

- | | |
|---|---|
| Уваров А. Ю. Вступая в век информации | 4 |
| Гейн А. Г., Шолохович В. Ф. Десять лет спустя | 7 |

ОБЩИЕ ВОПРОСЫ

- | | |
|--|----|
| Конкурс на разработку федеральных компонентов государственных образовательных стандартов | 13 |
| Базлов И. Ф., Шляго А. Н. Экзаменационные материалы по информатике (К разработке образовательных стандартов Санкт-Петербургской школы) | 18 |
| Рекомендации международной научно-практической конференции «Организация единого информационного образовательного пространства средствами телекоммуникаций» (г. Пермь, 6—9 декабря 1994 г.) | 29 |
| Информационное письмо МО РФ | 32 |

МЕТОДИКА

- | | |
|---|----|
| Семакин И. Г. Базовый курс ОИВТ: «Пермская версия» (часть 5) | 33 |
| Исаков В. Н., Исакова В. В. Алгоритмизация и программирование: методические аспекты | 44 |
| Пронина С. Е. Лого в школьном курсе информатики | 49 |
| Юдина А. Г. Информатика на Лого для старшекласников (окончание) | 54 |

ЗАДАЧИ

- | | |
|---|----|
| Сычев Н. А. Задания для вступительных экзаменов по информатике в НГУ | 65 |
| Нечаев В. М. Одна задача — три языка (окончание) | 73 |
| Софронова Н. В. Решение задач на массивы с помощью электронных таблиц | 85 |

Главный редактор
академик
БЕЛОЦЕРКОВСКИЙ О.М.

РЕДАКЦИОННАЯ КОЛЛЕГИЯ

Болотов В. А.
Бредихин Г. А.
Васильев Б. М.
Зайдельман Я. Н.
Зубченко А. А.
Киселев Б. Г.
Королев В. А.
Кравцова А. Ю.
Краснов А. Я.
Кузякин А. П.
Курнешова Л. Е.
Лагчик М. П.
Леонов А. Г.
Пахомова Н. Ю.
Савин А. Ю.
Самовольнова Л. Е.
Сапрыкин В. И.
Смекалин Д. О.
Уваров А. Ю.
Угринович Н. Д.
Урнов В. А.
Фурсенко А. И.
Хорошилов В. О.
Христочевский С. А.
Чуриков П. А.
Щенников В. В.

РЕДАКЦИЯ

Заместитель
главного редактора

Кравцова А. Ю.

Ответственный
секретарь

Иванова Т. В.

Редакционная группа:

Васильев Б. М.,
Кириченко И. Б.,
Орлова Т. Н.,
Усенков Д. Ю.

Компьютерная верстка

Кириченко И. Б.

Технический редактор

Луговская Т. В.

Корректор

Антонова В. С.

Экономический отдел

Бородаева З. В.

Отдел по подписки
и распространения

Коптева С. А.
(208-30-78)

Информационное
агентство ИА ИНФО

Васильева Н. А.
208-67-37

НОВЫЕ ПРОЕКТЫ

Тюменцева Е. Ю. MultiVision и MultiVision Pro
в учебном процессе

87

ПРОФЕССИОНАЛЬНЫЙ КОМПЬЮТЕР

Степанова М. И., Сазанюк З. И. Гигиенические
требования к проведению компьютерных занятий
во внеурочное время

97

APPLE ДЛЯ ОБРАЗОВАНИЯ

Конашевич М. Учиться играя

103

ЗАРУБЕЖНЫЙ ОПЫТ

Хирви В. Роль образования в современной Финляндии

105

Кярки С.-Л. Современные тенденции в педагогике
Финляндии

111

Коски Л. Обучение предпринимательской деятельности

113

Виве М. Образовательная инженерия

114

КЛУБ УКНЦ

Демаков А. В., Лебедев М. В. Исполнители
в системе КуМир

117

Фролов М. И. Автоматизированная система
построения расписания «Завуч» для УКНЦ

125

Почту направлять по адресу: 103051, Москва, ул. Садовая-Сухаревская, д. 16, к. 9,
журнал «Информатика и образование».

Телефон: (095) 208-30-78

Факс: (095) 208-67-37

E-Mail: info@tit-bit.msk.su

При перепечатке материалов ссылка на журнал обязательна. Редакция не несет
ответственности за содержание рекламы и используемые в ней товарные знаки.

За содержание листингов программ редакция ответственности не несет.

Подписано в печать с оригинал-макета 27.02.95. Формат 70×100 1/16. Бумага офсетная. Печать офсетная.
Усл. печ. л. 10,40. Усл. кр.-отт. 11,70. Уч.-изд. л. 13,5. Тираж 13 100 экз. Заказ 223

Цена по подписке:

для индивидуальных подписчиков 6 000 руб. (индекс 70423);

для предприятий и организаций 15 000 руб. (индекс 73176).

В розницу цена договорная.

Ордена Трудового Красного Знамени Чеховский полиграфический комбинат
Комитета по печати и информации Российской Федерации. 142300, Чехов Московской обл.

Уважаемые коллеги!



В этом году исполняется 10 лет с момента введения в образовательных учреждениях России курса «Основы информатики и вычислительной техники».

Благодаря огромному энтузиазму сотен и тысяч преподавателей в стране накоплен богатейший опыт преподавания курса в различных возрастных группах, разработаны программно-методические комплексы для различных типов вычислительной техники, созданы вариативные учебники и учебно-методические пособия.

В настоящее время, в условиях формирования единого информационного пространства России, задача подготовки учащихся и студентов к работе с информацией с помощью средств вычислительной техники, формирование их полноценными гражданами, готовыми к жизни в новом информационном обществе, становится все более актуальной и значимой.

Позвольте мне поздравить всех работников системы образования с 10-летним юбилеем курса информатики и пожелать дальнейших успехов в деле подготовки подрастающего поколения к жизни и труду в современном обществе во благо дальнейшего развития и процветания нашей Родины.

Министр образования России

A handwritten signature in black ink, which appears to read "Евгений Викторович Ткаченко". The signature is written in a cursive, flowing style.

Евгений Викторович Ткаченко

10 ЛЕТ

ШКОЛЬНОЙ ИНФОРМАТИКЕ

А. Ю. Уваров,

Научный совет по комплексной проблеме «Кибернетика» РАН

ВСТУПАЯ В ВЕК ИНФОРМАЦИИ

Пришел к нам из роно приказ: послать одного учителя на курсы по информатике и компьютерам. Мы собрались в учительской и бросили жребий. Выпало мне. Я заплакала, заплакала и поехала...

*Из воспоминаний учительницы информатики,
август 1985 г.*

Сегодня, когда все новшества в области вычислительной техники приходят к нам из-за рубежа, многие забыли, что наша страна была пионером **в области обучения школьников работе с ЭВМ**. В 1960 г. учитель московской школы № 444, впоследствии член-корреспондент АПН СССР С. И. Шварцбург выпустил первую группу старшеклассников, получивших квалификацию программиста. С середины 60-х гг. факультативные курсы по вычислительной математике и программированию стали обычным явлением в школах. Во второй половине 60-х стали обсуждаться возможности использования ЭВМ для повышения эффективности системы образования. Примерно в это же время в московской школе № 444 и Киевском высшем инженерно-авиационном училище почти одновременно появились первые «обучающие системы на базе ЭВМ». Основатель отечественной кибернетики, председатель Научного совета по комплексной проблеме «Кибернетика»

академик А. И. Берг был одним из зачинателей этого направления. Неизбежность проникновения компьютеров в образование была убедительно показана и в ходе выполненного в 70-е гг. исследования по прогнозированию развития системы образования. Однако выступление руководителя проекта «Школа 2000 года» академика М. Н. Скаткина на Общем собрании АПН в 1978 г. было встречено полным безразличием. Времена застоя брали свое. Потребовалось еще шесть лет, прежде чем новое поколение политических лидеров страны осознало происходящие в мире перемены. В результате десять лет назад было принято памятное многим постановление «О мерах по обеспечению компьютерной грамотности учащихся средних учебных заведений и широкому внедрению электронно-вычислительной техники в учебный процесс». Министр просвещения, который не соглашался с «компьютеризацией школы», ушел на пенсию, а во всех средних учебных заве-

дениях страны появился новый общеобразовательный учебный предмет — «Основы информатики и вычислительной техники». Сегодня стоит с благодарностью вспомнить академиков Е. П. Велихова, А. П. Ершова, В. А. Мельникова, Б. Н. Наумова и многих других, кто способствовал принятию этого решения, несмотря на сопротивление «педагогического истеблишмента».

Конечно, сегодня лозунги «стратегии ускорения» выглядят довольно нелепо. Мало что из принятых тогда решений удалось реализовать. Монуменстами «ускорения» стоят незавершенные производства, которые должны были выпускать миллионы персональных компьютеров и их комплектующих. Многие учителя сегодня с радостью избавились бы от часто ломающихся УКНЦ, которые глава Минэлектронпрома называл в свое время «лучшим в мире учебным компьютером». Низкокачественные отечественные мониторы продолжают портить глаза школьников. Программа создания отечественных персональных ЭВМ провалилась. Электронная промышленность не выдержала конкуренции с надежной и более совершенной зарубежной техникой. Однако, несмотря на прогнозы пессимистов, которые громко звучали десять лет назад, использование компьютеров в стране стало всеобщим, и они уже являются частью нашей повседневной жизни. Школьники, начавшие изучать информатику десять лет назад, сегодня работают на многочисленных компьютерах в учреждениях, банках, на предприятиях. Подтвердилась известная истина: в условиях глобальных изменений вложения в образование самые выгодные. И сегодня, несмотря на труднейшее финансовое положение, школы продолжают приобретать новейшую вычислительную технику. Начатая десять лет назад «информатизация образования» продолжается.

Программа информатизации образования, подготовленная в Минпросе СССР, была опубликована. Десятилетний юбилей — хороший повод посмотреть, насколько верными оказались представления о реформе школы с использованием средств НИТ.

Можно констатировать, что основные направления и этапы процесса информатизации школы были обозначены доста-

точно верно. Несмотря на то что задания по оснащению школ компьютерами выполнены лишь частично, задача первого этапа (1985—1990) — «обеспечить общеобразовательную подготовку учащихся средних учебных заведений в области информатики, дать части школьников подготовку по профессиям, связанным с вычислительной техникой, ознакомить всех педагогов с возможностями, которые предоставляют компьютеры» — в основном выполнена. Задача второго этапа (1990—1995) — «завершить переход к изучению информатики с обязательным использованием компьютеров, начать широкое использование кабинетов вычислительной техники для проведения лабораторно-практических занятий, «компьютерных уроков», внеклассной работы с учащимися по всем общеобразовательным предметам» — практически решается сегодня во многих школах. Система образования приближается сегодня к следующему, третьему этапу, цель которого — «осуществить подготовленный всем предыдущим развитием переход к изучению общеобразовательных дисциплин по новым программам, предусматривающим естественное использование средств вычислительной техники в процессе их изучения». В ближайшие годы станет ясно, насколько такое кардинальное изменение содержания, методов и организационных форм обучения реально в условиях нашей школы.

Когда программа «Обеспечение всеобщей компьютерной грамотности учащихся и введение ЭВМ в учебный процесс» только разворачивалась, учителя, администраторы, принимающие решение политики видели перед собой прежде всего компьютеры. Некоторым казалось, что компьютеры сделают процессы учения и обучения более эффективными без кардинальных изменений в организационных формах и методах учебной работы. Сегодня всем ясно, что требуется создавать новую организацию учебного процесса, учитывая нашу ситуацию. Наши представления об условиях жизни наших учеников в будущем информационном обществе. Средства НИТ выступают в этих условиях не как самоцель, а лишь как один из инструментов происходящих изменений.

Работу по информатизации школы десять лет назад многие рассматривали как

катализатор будущих глобальных изменений в экономике. Сегодня мы видим, что это ожидание не оправдалось. Централизованно выделенные средства на приобретение и установку в учебных заведениях сотен тысяч наших компьютеров не спасли отечественную электронную промышленность от краха, а лишь ненадолго продлили ее агонию. Однако проявился другой эффект: «обеспечение всеобщей компьютерной грамотности» создало в стране атмосферу наибольшего благоприятствования для повсеместного приобретения и освоения персональных компьютеров. Многие из тех, кто познакомился с информатикой за прошедшие десять лет, сегодня практически работают с компьютерами.

За прошедшие десять лет многое изменилось. Сегодня нет необходимости доказывать, что компьютер — полезный массовый инструмент для повышения эффективности умственного труда. Бесплодные дискуссии о том, какой язык программирования следует изучать, также ушли в прошлое. К сожалению, в некоторых вузах знакомство с информатикой по-прежнему подменяют знанием синтаксиса языка Бейсик. К подготовке учебных пособий по информатике для школы были привлечены лучшие специалисты страны. Неудивительно, что научный уровень этих пособий до сих пор опережает методические разработки, выполняемые сегодня в некоторых вузах.

Информатизация образования — процесс, который является комплексным по своей сути. Для специалистов это очевидно. Нельзя устанавливать в школе компьютеры, не заботясь одновременно и о программном обеспечении, о переподготовке педагогов и внесении соответствующих изменений в учебный план.

Примером комплексного подхода может служить проект «Пилотные школы». Несмотря на известные технические недостатки «пилотных классов», комплексность проекта позволила ему практически повлиять на работу школ. Сегодня многие регионы осуществляют собственные программы информатизации школы. Комплексность этих программ — залог успеха проходящих сегодня реформ.

Опыт прошедшего десятилетия учит: **информатизация образования — это и есть реформа школы.** К сожалению,

далеко не все это действительно понимаем. Основные усилия до сих пор сосредоточены на создании нового поколения учебников и учебно-методических пособий. При этом акцент делается на изменении фактической стороны содержания учебных материалов. Заметно меньше внимания уделяется разворачивающемуся на наших глазах *процессу изменения средств и способов деятельности, перестройке методов и организационных форм обучения.* Решение задачи информатизации образования требует включения в обязательное содержание общего образования большой группы сравнительно новых для практики обязательных составляющих: способности эффективно сотрудничать при выполнении работы в группе, продуктивно планировать свою работу, умения работать с информацией, владения средствами рационального мышления и т. п.

«Осваиваемые способы деятельности», а не «конкретное предметное содержание» или «материал» должны ложиться в основу содержания современных учебных предметов. Это положение не нуждается в обоснованиях. Оно родилось в недрах отечественной педагогической психологии и практически общепризнано. Время требует, чтобы его претворили в жизнь.

Нас ждут коренные преобразования системы образования. Предстоит осваивать новые информационные технологии и одновременно изменять:

- систему учебной работы школьников,
- отношения в педагогическом и учебном коллективах,
- цели и содержание учебной работы,
- способы оценки результатов.

Использование новых информационных технологий — этот «подарок НТР» системе образования — не самоцель, а лишь катализатор необходимых изменений.

В течение следующего десятилетия нам предстоит выстроить модель среды, в рамках которой осуществляется эффективное сотрудничество между участниками учебного процесса, в котором используется весь спектр новейших информационных технологий, позволяющих эффективно осуществлять индивидуальную и коллективную учебную работу учителей и школьников.

А. Г. Гейн, В. Ф. Шолохович,

Екатеринбург

ДЕСЯТЬ ЛЕТ СПУСТЯ

Завершается первое десятилетие существования в школе курса информатики. Он не оказался очередной модной новацией. Курс не только выжил, но и пустил глубокие корни, инициируя проникновение информационных технологий в учебный процесс. Более того, в учительской среде появились люди новой специальности — программисты. Они не были похожи на мастеров производственного обучения, но несли с собой заряд «всамделишной» жизни, были профессионалами в деле «приручения» компьютеров, а не школьников. Тем не менее последние ходили за ними «гуськом». В свою очередь, учителя, отважившиеся на преподавание нового предмета, срочно учили школьный алгоритмический язык, а те счастливицы, которые имели доступ к ЭВМ, — какой-нибудь язык общения с компьютером (в основном, увы, Бейсик или того хлеще — Фокал). Впрочем, выбор языка программирования не представлялся такому учителю чем-то принципиально важным — ведь в тот период почти вся страна была охвачена «безмашинным вариантом» изучения информатики.

В этом «безмашинном море» довольно быстро стали возникать островки машинного варианта. Обычно таким островком была отдельная школа (или несколько школ, прикрепленных к одному УПК, имевшему соответствующий профиль), но встречались и достаточно крупные участки «суши». Одним из них стал, спустя всего лишь два года после введения школьного курса информатики, г. Свердловск (ныне Екатеринбург). Областному координационному научно-методическому совету по компьютеризации под руководством академика Н. Н. Красовского удалось организовать более 20 межшкольных компьютерных центров, закупив для этой цели на средства свердловских предприятий необходимое количество техники. Это дало реальную возможность рабо-

ты с ЭВМ в объеме, предусмотренном программой курса ОИВТ каждому ученику IX и X классов.

Сразу уточним: речь идет об экспериментальной программе, подготовленной советом по компьютеризации и Свердловским пединститутом (ныне — Уральский государственный педагогический университет). В те времена для преподавания по экспериментальной программе требовалось разрешение как минимум Министерства просвещения РСФСР. Такое разрешение было получено, поскольку предлагаемая программа была поддержана учебным пособием «Основы информатики» [4] и соответствующими программными средствами, обеспечивавшими проведение лабораторных работ, дидактическими материалами и методическими рекомендациями для учителей информатики, созданными к началу 1987/88 учебного года в проблемной лаборатории Свердловского пединститута по применению микропроцессорной техники в учебном процессе. Эта программа была рассчитана на 102 учебных часа, половина из которых отводилась на выполнение лабораторных работ в компьютерном классе.

Таким образом, в Свердловске был создан целостный программно-методический комплекс, обеспечивавший учителей и учащихся всем, что необходимо для эффективного освоения курса информатики в тогдашнем его понимании. Состав и структура комплекса отражали сформированную советом по компьютеризации концепцию «компьютерного всеобуча» при наличии необходимой технической базы.

Одна из принципиальных установок этой концепции, не утратившая своего значения и сегодня, состоит в том, что компьютер для учащегося — рабочий инструмент, используемый в различных сферах деятельности. Школьники должны знать, какие задачи обычно решаются

с помощью ЭВМ, каковы этапы решения этих задач, какими средствами снабжают ЭВМ, чтобы облегчить общение человека с машиной. Учащиеся должны владеть навыками работы на ЭВМ и уметь анализировать результаты ее применения. Поэтому основной акцент в разработанном курсе ОИВТ был сделан на освоении учащимися всех этапов решения задач с помощью компьютера: постановки задачи, построения компьютерной модели, разработки, если необходимо, алгоритма, проведении компьютерного эксперимента и анализа полученных результатов.

Эта парадигма курса ОИВТ была реализована в учебнике [4], созданном авторским коллективом в составе: А. Г. Гейн, В. Г. Житомирский, Е. В. Линецкий, М. В. Сапир, В. Ф. Шолохович. Одна из трудных проблем, которую пришлось решать авторам учебника, состояла в том, чтобы новый предмет, выстроенный в такой парадигме, не воспринимался как инородный в системе школьных дисциплин. Обсуждая аналогичную проблему для учебника [6], А. П. Ершов отстаивал «идею перехода в информатику от математики, через понятие алгоритма, вырастающего из уже освоенной алгебраической символики, через разработку алгоритма, предшествующего выходу на машину». Это положение стало определяющим для той серии учебников, где центральное место занимает изучение школьного алгоритмического языка.

Мы же разделяем позицию, в которой главным звеном в изучении школьной информатики как дисциплины, посвященной применению ЭВМ для решения задач, является это построение компьютерных моделей нечетко сформулированных (иногда говорят: «плохо поставленных») задач [3]. С подобными задачами школьники не встречаются ни в одном из остальных предметов. Для учителей здесь тоже «terra incognita». Но у этого подхода есть одна чрезвычайно привлекательная черта — он позволяет рассматривать задачи из разряда жизненных проблем, которые приходится ежедневно решать каждому. Применение ЭВМ для решения именно «жизненных» задач играет боль-

шую мотивационную роль в изучении курса ОИВТ.

Реалии того времени, когда информатика только вводилась в число школьных предметов, не могли не наложить свой отпечаток на созданный нами курс и соответствующий учебник. Весьма скудные ресурсы имевшейся в то время вычислительной техники, отсутствие необходимого учебного программного обеспечения в (меж)школьных компьютерных классах (хотя оно и было разработано в проблемной лаборатории для основных видов школьной компьютерной техники) — все это заставило нас, во-первых, ограничиться рассмотрением лишь математических моделей из всего разнообразия компьютерных моделей, во-вторых, сделать алгоритмизацию главным средством решения задач с помощью ЭВМ, в-третьих, вынести в конец курса знакомство с основными пользовательскими пакетами. Последнее решение, которое сейчас выглядит довольно странно для тех, кто считает пользовательскую часть курса основной, было продиктовано нашим стремлением обеспечить учителя материалом на первые 70 часов преподавания в условиях слабого и «голового» компьютерного «железа».

Приоритет модельной парадигмы перед алгоритмической в нашем учебнике проявляется многообразно. Прежде всего мы разрешаем ученику записывать алгоритмы на естественном (родном для школьника) языке, предъявляя к этой записи минимальные требования — отсутствие двусмысленности в записи действий (например, разрешается использовать лишь те действия, которые допустимы для исполнителя) и возможность однозначно восстановить из записи алгоритма порядок выполнения действий. Образно выражаясь, можно сказать, что ученик, составляя алгоритм, создает модель будущей программы, учитывая лишь два вышеуказанных требования. Методическая целесообразность такого подхода очевидна: он позволяет ученику сосредоточить свое внимание именно на алгоритмизации без опасений допустить синтаксическую ошибку в тексте алгоритма.

Приверженность авторов модельной

парадигме проявляется и в подходе к изучению темы «Устройство ЭВМ». Мы отказались от изложения системы машинных команд реальной ЭВМ. Более того, в нашем учебнике читатель не найдет интригующих рассказов о вентилях и многочисленных регистрах. В рамках жесткой алгоритмической парадигмы без этого, по-видимому, не обойтись. Мы же предлагаем учащимся поработать с моделью ЭВМ, отражающей самое существенное в архитектуре компьютера — принципы фон Неймана.

Эти два примера, как нам кажется, довольно выпукло отражают главную особенность учебника, общий тираж которого в четырех изданиях, выполненных «Просвещением», уже превысил 1,5 млн. экземпляров. Это позволяет нам считать, что он хорошо знаком учительской обществу.

Активная информатизация общества вводит компьютер в повседневную практику многих профессий, и изучение курса информатики только на последних этапах школьного образования объективно не способно решить в полной мере задачи формирования у школьников информационной культуры уже потому, что к этому времени у учащихся оказываются закрепленными такие стереотипы информационной деятельности, которые нередко требуют переформирования. К тому же использование в других предметах знаний, умений и навыков, получаемых в курсе информатики, затруднено ввиду того, что этот курс стоит в заключительной (по времени изучения) обойме школьных дисциплин.

Учителя-энтузиасты, неравнодушные к судьбе своего предмета, почувствовали этот дисбаланс и уже несколько лет предпринимают попытки начать преподавание информатики с более раннего возраста.

Разумеется, названный выше авторский коллектив не был безучастным наблюдателем процесса понижения границы начала применения НИТ и, в частности, преподавания информатики в школе. В 1992 г. была опубликована разработанная нами и утвержденная Министерством образования Российской Федерации

программа курса информатики для VIII—IX классов общеобразовательной школы [7]. Создание этой программы, как и создание подобных программ другими авторскими коллективами, является одним из кирпичиков строящегося «здания» школьной информатики. Комплекс этих программ, с одной стороны, и опыт, накопленный учительской общественностью в преподавании информатики, с другой, без сомнения, сыграли определенную позитивную роль при подготовке нового базисного учебного плана, определившего VII класс как нижнюю границу изучения базового курса информатики в основной школе. Главное направление в этой программе, как и прежде, — обучение решению задач с помощью ЭВМ через построение компьютерных моделей. Однако для внедрения соответствующего курса в школьную практику одной лишь программы, разумеется, недостаточно. Вышедший в 1994 г. учебник информатики для VIII—IX классов средней школы (авторский коллектив: А. Г. Гейн, Е. В. Линецкий, М. В. Сапир, В. Ф. Шолохович), обеспеченный компьютерной поддержкой, позволяет приступить к реализации названной выше установки базисного учебного плана в масштабах всей страны. Чтобы предоставить преподавателям информатики возможность подробнее познакомиться с дидактическими установками и методическими особенностями данного учебника, нами подготавливается серия статей к публикации в журнале «Информатика и образование».

Параллельно с созданием базового курса информатики для VIII—IX классов была начата разработка углубленного профильного курса информатики для специализированных классов. Речь идет не об информатике для физико-математических классов, а именно о курсе, предназначенном для классов углубленного изучения информатики. На наш взгляд, способности к информатике вовсе не идентичны математическим способностям — среди школьников, легко усваивавших созданную нами программу, были учащиеся с весьма средними математическими способностями, и, наоборот, были ученики,

математические способности которых проявлялись достаточно отчетливо, но для которых данная программа оказалась довольно трудной. Главная установка созданного нами курса: информатика — это основа промышленной обработки данных. Это, в частности, определило «производственный» характер данной дисциплины — учащиеся не только осваивают теорию, приобретают умения и отрабатывают навыки на учебных заданиях, но и создают программный продукт, имеющий потребительскую ценность. В основном это программное обеспечение, которое используется в той же школе, где учится обучаемый. Тем самым, подросток видит, как результаты его учения овеществляются в продукте, востребованном обществом. Может ли для школьника существовать более сильное моральное поощрение, более сильная мотивация к обучению?

Программа углубленного курса информатики и его основные дидактические установки описаны в [2]. Ее реализация рассчитана на 4 учебных года — с VIII по XI класс со следующей недельной раскладкой часов: в VIII—IX классах — 4 часа, в X классе — 5 часов, и в XI классе — 6 часов. Для курса созданы программная поддержка и учебник (авторы А. Г. Гейн и А. И. Сенокосов), содержащий материал первых двух лет обучения. В настоящее время этот учебник готовится к изданию в издательстве «Просвещение» и должен увидеть свет в 1995 г.

Перемещение базового курса в основную школу, на наш взгляд, вовсе не означает, что в двух заключительных классах общеобразовательной школы должен царствовать информационный «вакуум», а изучение информационных технологий будет рассредоточено по другим школьным дисциплинам. Наша концепция информатики в X—XI классах общеобразовательной школы предусматривает создание «веера» курсов информатики (и соответствующих учебников), дифференцированно ориентированных на профильное предпрофессиональное образование. Мы выделяем здесь четыре направления: гуманитарное, химико-биологическое,

физико-математическое и программистское. С одной стороны, для всех этих направлений имеется общее ядро, основу которого составляет компьютерное моделирование объектов, процессов и явлений, освоение системного подхода к построению таких моделей, знакомство с информационными принципами управления (управление по принципу «обратной связи», понятие «черного ящика» и т. п.). С другой стороны, проектируемые нами курсы данного «веера» различаются предметными областями, в которых ведется компьютерное моделирование, методами, используемыми для построения моделей, акцентами на различных парадигмах программирования и, наконец (по порядку, но не по значению), стилем изложения материала. Сразу оговоримся, что «веер» этих курсов создается в комплексе с новым учебником информатики для VII—IX классов, который параллельно с уже изданным учебником для VIII—IX классов призван обеспечить изучение информатики в базовом звене. Концептуально эти учебники не различаются, но сниженные границы на год и полутора кратное увеличение числа часов заставили нас пересмотреть многие методические установки. Рукопись этого учебника к настоящему времени создана (авторский коллектив: А. Г. Гейн, А. И. Сенокосов, В. Ф. Шолохович), к нему сделана программная поддержка, и в настоящее время он проходит апробацию в 18 школах Екатеринбурга.

Имеется рукопись учебника информатики для гуманитарного направления в указанном выше «веере» предпрофессиональных курсов (авторский коллектив: С. А. Бешенков, А. Г. Гейн, С. Г. Григорьев). Понимая, что «скоро сказка сказывается, да не скоро дело делается», мы создали вариант этой книги, который может как использоваться в качестве самостоятельного учебника в X—XI классах школ, лицеев, гимназий гуманитарного профиля, так и «пристыковываться» к уже вышедшему в свет учебнику «Информатика VIII—IX».

Процесс снижения границы начала изучения курса информатики сопровож-

дается превращением информатики в фундаментальную дисциплину, занимающую в школьном образовании столь же значительное место, как математика и литература. Курс обогащается идейно за счет осознания, что само обучение в школе — это прежде всего информационный процесс. Внутренние методологические установки данного предмета таковы, что предполагают изучение ряда его аспектов еще в начальной школе. Хорошо известна работа различных исследовательских коллективов над проблемой обучения информатике уже в начальной школе. Мы со своей стороны разрабатываем в настоящее время концепцию «сквозного» модульного курса информатики для I—XI классов. Компьютер в этом курсе в первую очередь выступает как мощный инструмент информационных технологий, а не как объект изучения.

Завершая этот довольно беглый рассказ о нашем восприятии прошедшего десятилетия школьной информатики, мы хотим привести слова А. П. Ершова о том, какой он видит школьную информатику спустя десять лет после ее появле-

ния в школе: «За это время курс информатики перейдет в VII—VIII классы*, оставив последние два года для углубленного предпрофессионального употребления ЭВМ. В недрах научной и экспериментально-педагогической работы созреет так называемый интегрированный курс информатики IV—VIII классов, где освоение работы с компьютером, фундаментальных понятий и умений информатики будет тесно связано со всей школьной практикой, требующей активного применения компьютеров как постоянного орудия интеллектуальной работы. Есть немало сторонников объединения математики и информатики, и эта идея получит свое воплощение и проверке. Регулярное применение компьютеров в начальной школе вряд ли получит полное распространение, но будет полностью подготовлено в результате опережающей совместной работы учителей и специалистов по психологии, дидактике, математике, языку и информатике».

Такой точности прогноза остается только позавидовать.

Литература

1. Ершов А. П. Компьютеризация школы и математическое образование // Программирование. 1990. № 1. С. 5—25.

2. Гейн А. Г., Сенокосов А. И. Программно-методический комплекс для классов с углубленным изучением ОИВТ // Информатика и образование. 1991. № 6; 1992. № 1.

3. Компьютеры, модели, вычислительный эксперимент: Введение в информатику с позиций математического моделирования. М.: Наука, 1988.

4. Гейн А. Г., Житомирский В. Г., Линецкий Е. В. и др. Основы информатики: Учебное пособие для X—XI классов средней школы. Свердловск: УрГУ, 1989.

5. Гейн А. Г., Житомирский В. Г., Линецкий Е. В. и др. Основы информатики и вычислительной техники: Учебник для X—XI классов средней школы. 1—4-е изд. М.: Просвещение, 1991, 1992, 1993, 1994.

6. Ершов А. П., Кушниренко А. Г., Лебедев Г. В. и др. Основы информатики и вычислительной техники: Пробный учебник для средних учебных заведений / Под ред. А. П. Ершова. М.: Просвещение, 1988.

7. Программы средней общеобразовательной школы. Основы информатики и вычислительной техники. М.: Просвещение, 1992.

* В прежней 10-летней схеме нумерации классов. — *Примеч. авторов.*

Фирма АЛТ – НТП «Альтернатива» – НТП «Новая Альтернатива»

ИСТОРИЯ

- 1986 г.** В школах Алтайского края внедряется курс информатики. В Алтайском ИУУ (г. Барнаул) создается кабинет информатики. Его возглавляет А. Н. Гриценко, до этого работавший на кафедре прикладной математики АлтПИ и окончивший в 1985 г. аспирантуру математического факультета Новосибирского университета. Нет ни техники, ни программного обеспечения, ни методики. Во всем крае имеется только три терминальных класса КУВТ-86.
- 1987 г.** Количество терминальных классов в Алтайском крае достигает шести. Попытки найти для них программное обеспечение в Москве и других городах не имеют успеха. Во всех вузах и лучших школах установлены компьютеры «Ямаха» и IBM, а для массовой школьной техники практически ничего нет. Немногие разработки такого качества, что для учебного процесса по ОИВТ непригодны. На одном из совещаний А. Н. Гриценко в ответ на свое резкое заявление о низком качестве одной из таких систем слышит от директора ИУУ: «А у вас есть альтернатива?»
В течение последующих трех месяцев в кабинете информатики Алтайского ИУУ создается первый программно-методический пакет для классов КУВТ-86, получивший название «АЛЬТЕРНАТИВА». Пакет замыслился как комплект программ, документации и методик, который ИУУ должен передать учителю ОИВТ. Никаких задач, кроме обеспечения нормальной работы имеющихся в крае шести терминальных классов, не ставилось.
- 1988 г., февраль** Неожиданно для разработчиков созданный пакет «АЛЬТЕРНАТИВА» приобретает популярность. Поступают многочисленные запросы от школ и учителей других областей. Становится ясна необходимость перевода разработок на постоянную и хозрасчетную основу. При кабинете информатики ИУУ была создана кооперативная фирма АЛТ. Руководитель — А. Н. Гриценко. Программный пакет фирмы быстро совершенствуется и приобретает широкую известность, на него ежедневно поступают заказы из всех регионов и республик СССР. Большинство заказчиков — небогатые сельские школы.
- 1990 г.** В Алтайском крае устанавливаются десятки новых классов типа «Корвет» и УКНЦ. Поступают многочисленные просьбы сделать аналог пакета «АЛЬТЕРНАТИВА» для этих классов. Работа по созданию этих пакетов потребовала полного перехода на профессиональную основу. Создается совместное с Институтом информатики АПН малое предприятие «Альтернатива». Его сотрудниками становятся работники Алтайского ИУУ.
- 1991 г.** Предприятие «Альтернатива» разрабатывает первые версии программно-методических пакетов «АЛЬТЕРНАТИВА-КОРВЕТ» и «АЛЬТЕРНАТИВА-УКНЦ». Начинаются их поставки многочисленным заказчикам. В предприятии начинаются внутренние разногласия. Недовольный диктатом большинства и снижением темпов и качества работ основатель и руководитель предприятия А. Н. Гриценко выходит из него и основывает обновленное предприятие «Новая Альтернатива».
- 1992–1993 гг.** НТП «Новая Альтернатива», начав с нуля, быстро создает новые версии пакетов «АЛЬТЕРНАТИВА» для классов УКНЦ, «Корвет» и КУВТ. Для классов КУВТ-86 создаются пакеты «РОБОТЛАНДИЯ» и «ЛОГО». Устанавливаются агрессивно низкие цены на пакеты, доступные практически всем заказчикам.
- 1994 г.** НТП «Новая Альтернатива» создает новые учебные системы программирования «БЕТТА-ЛОГО» и «АЛТ-ПАСКАЛЬ» для УКНЦ, «ЛОГО» и «РОБОТЛАНДИЯ» для классов БК-11, «ТУРБО-ПАСКАЛЬ» для КУВТ-86. Делаются крупные вложения в создание автоматической системы составления расписаний для школ, ПТУ и техникумов. НТП «Новая Альтернатива» резко расширяет объем разработок и продаж.

**Адрес: 656057, г. Барнаул, а/я 2513,
НТП «Новая Альтернатива»,
директор Гриценко Алексей Николаевич.**

ВОПРОСЫ

ОБЩИЕ

В Министерстве образования Российской Федерации

КОНКУРС НА РАЗРАБОТКУ ФЕДЕРАЛЬНЫХ КОМПОНЕНТОВ ГОСУДАРСТВЕННЫХ ОБРАЗОВАТЕЛЬНЫХ СТАНДАРТОВ

Уважаемые читатели!

По не зависящим от редакции обстоятельствам материалы о конкурсе, представленные ниже, не были опубликованы в журнале «Информатика и образование» ранее. В настоящее время первый этап конкурса завершен (см. стр. 16). Если вы решите участвовать в этом конкурсе, то необходимую информацию можно получить по телефону (095) 923-57-68.

Приложение 1
к приказу Министерства образования
Российской Федерации
от 30.05.94. № 161

ПОЛОЖЕНИЕ

**об условиях проведения конкурса
на разработку федеральных компонентов
государственных образовательных стандартов
начального общего, основного общего, среднего (полного) общего
и начального профессионального образования**

I. Общие положения

1. Конкурс на разработку федеральных компонентов государственных образовательных стандартов (далее – конкурс) проводится Министерством образования Российской Федерации с целью утверждения федеральных компонентов государственных образовательных стандартов (далее – ФК ГОС) начального общего, основного общего, среднего (полного) общего и начального профессионального образования.
2. Конкурс носит открытый характер, предполагающий свободное участие в нем творческих коллективов, образовательных учреждений, а также отдельных авторов.
3. На конкурс принимаются и рассматриваются все представленные материалы по ФК ГОС, отвечающие требованиям к оформлению документов и материалов, присланных на конкурс.

4. Для отбора поступивших на конкурс материалов по ФК ГОС создается конкурсная комиссия (далее – комиссия). Положение о конкурсной комиссии и ее состав утверждаются Министерством образования Российской Федерации.

II. Организация конкурса

5. Конкурс проводится в три этапа.

5.1. Первый этап конкурса проводится с 01 июня 1994 г. по 31 октября 1994 г.

Комиссия работает с 01 ноября 1994 г. по 31 декабря 1994 г.

На этом этапе представляется концепция ФК ГОС начального общего, основного общего, среднего (полного) общего и начального профессионального образования (далее – концепция), которая является заявкой на участие в конкурсе.

По результатам первого этапа отбираются наиболее перспективные концепции для их дальнейшей разработки.

5.2. Второй этап конкурса проводится с 01 января 1995 г. по 31 августа 1995 г.

Комиссия работает с 01 сентября 1995 г. по 30 ноября 1995 г.

На втором этапе конкурса принимаются документы и материалы, оформленные как рукописи ФК ГОС (далее – рукописи), включающие в себя требования к уровню подготовки выпускников и способы квалификации достижений обучающихся, разработанные на основе концепций, победивших на первом этапе.

5.3. Третий этап конкурса проводится с 01 декабря 1995 г. по 30 апреля 1996 г.

Комиссия работает с 03 мая 1996 г. по 30 июня 1996 г.

На заключительном этапе конкурса принимаются рукописи проектов ФК ГОС, допущенные к участию на третьем этапе и доработанные с учетом рекомендаций комиссии.

6. Датой представления рукописи считается дата отправления на штампе почтового отделения, а в случае личной подачи ее автором (авторами) – дата регистрации в журнале приема и выдачи конкурсных материалов.

7. Финансирование работы комиссии производится за счет средств, выделяемых на реализацию Федеральной программы развития образования в России.

III. Подведение итогов конкурса

8. По результатам конкурса комиссия отбирает лучшие рукописи (не более трех) по каждой общеобразовательной программе начального общего, основного общего, среднего (полного) общего образования и по профессиональной программе начального профессионального образования и представляет их Министерству образования Российской Федерации для организации широкого обсуждения в средствах массовой информации.

9. По результатам обсуждения комиссия подводит итоги и выносит решение о распределении по каждой программе 3 (трех) призовых мест.

10. Министерство образования Российской Федерации определяет виды и размеры вознаграждения авторам (автору), занявшим призовые места, а также размеры выплаты грантов авторам работ, допущенных к последующим этапам конкурса.

11. Рукописи проектов ФК ГОС начального общего, основного общего, среднего (полного) общего и начального профессионального образования, занявшие первые места, представляются конкурсной комиссией в Министерство образования Российской Федерации и, при необходимости, направляются на доработку.

12. Доработанные рукописи проектов ФК ГОС направляются на утверждение: основного общего образования – Правительством Российской Федерации, а начального общего, среднего (полного) общего и начального профессионального образования – Министерством образования Российской Федерации.

13. Рукописи, представленные на конкурс, автору (авторам) не возвращаются.

14. Конфликтные ситуации, возникающие в ходе проведения конкурса, рассматриваются конфликтной комиссией, создаваемой Министерством образования Российской Федерации в случае необходимости.

Приложение 3
к приказу Министерства образования
Российской Федерации
от 30.05.94. № 161

ПОРЯДОК прохождения документов и материалов, присланных на конкурс по разработке, утверждению и введению федеральных компонентов государственных образовательных стандартов начального общего, основного общего, среднего (полного) общего и начального профессионального образования

1. Прием документов и материалов, присланных на конкурс по разработке федеральных компонентов государственных образовательных стандартов начального общего, основного общего, среднего (полного) общего и начального профессионального образования осуществляется ответственным секретарем конкурсной комиссии (далее – комиссии) согласно описи, составленной автором (авторами) и предъявляемой им (ими) при сдаче материалов.

2. Все конкурсные материалы, принятые ответственным секретарем, регистрируются в журнале приема и выдачи конкурсных материалов и хранятся в сейфе.

3. Ответственность за хранение конкурсных материалов и обеспечение их конфиденциальности возлагается на ответственного секретаря комиссии.

4. Выдача конкурсных материалов, принятых на хранение, осуществляется в дни работы комиссии только членам комиссии с обязательной регистрацией этого акта в журнале приема и выдачи конкурсных материалов.

5. Контроль за прохождением конкурсных материалов в сроки, предусмотренные этапами конкурса, осуществляет ответственный секретарь комиссии.

6. По окончании каждого этапа конкурса в течение 2 недель каждому участнику данного этапа высылается соответствующая выписка из решения комиссии.

7. Документы и материалы, не прошедшие по конкурсу, подлежат сдаче в архив Министерства образования РФ в течение месяца со дня принятия комиссией решения о завершении очередного этапа конкурса.

Приложение 4
к приказу Министерства образования
Российской Федерации
от 30.05.94. № 161

ТРЕБОВАНИЯ к оформлению документов и материалов, присылаемых на конкурс

1. В конкурсную комиссию (далее – комиссия) на первом этапе должна быть представлена концепция федеральных компонентов государственных образовательных стандартов начального общего, основного общего, среднего (полного) общего и начального профессионального образования (далее – концепция), которая и является заявкой на участие в конкурсе. Концепция – определенный способ трактовки, конструктивный принцип или система взглядов на то или иное явление.

2. Концепция должна быть представлена в конкурсную комиссию не позднее 31 октября 1994 г. Датой представления документов и материалов считается дата отправления на почтовом штемпеле или, в случае личной подачи их автором (авторами), дата регистрации в журнале приема и выдачи конкурсных материалов.

3. В предъявляемой на конкурс концепции должны быть указаны:

3.1. Краткий анализ ситуации и обоснование предлагаемых федеральных компонентов государственных образовательных стандартов начального общего, основного общего, среднего (полного) общего и начального профессионального образования (далее – ФК ГОС).

3.2. Цель (цели) разработки ФК ГОС.

3.3. Основные задачи, которые предполагается решить в процессе введения в действие предлагаемых ФК ГОС.

- 3.4. Принципы формирования содержания учебного предмета (дисциплины).
- 3.5. Принципы формулирования требований к уровню подготовки выпускников.
- 3.6. Краткое описание технологии и средств обучения, используемых для достижения необходимого уровня образования.
- 3.7. Краткое описание системы измерителей (тесты, устный опрос, сочинение и пр.) достижений обучающихся.
4. Все документы и материалы, представляемые на конкурс, должны быть отпечатаны на белой бумаге форматом 210×297 на одной стороне листа с двумя интервалами между строками.
5. К документам и материалам, представляемым на конкурс, должны быть приложены:
- справка об авторе (авторах) с указанием фамилии, имени и отчества, ученой степени, звания, специальности, места работы, занимаемой должности и домашнего адреса;
 - опись;
 - подписанная папка с завязками.
6. Автор (авторы) концепций, решением конкурсной комиссии допущенных к участию во втором этапе конкурса, представляет документы и материалы, содержащие механизмы реализации концепции.
7. Требования к структуре и содержанию документов и материалов, представляемых на второй и третий этапы конкурса, разрабатываются конкурсной комиссией и публикуются в средствах массовой информации до 01.01.95 г.

* * *

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

РАСПОРЯЖЕНИЕ

08.02.95

Москва

№ 09-Р

В соответствии с Положением об условиях проведения конкурса на разработку федеральных компонентов государственных образовательных стандартов (утверждено приказом Минобразования России от 30.05.94 г. № 161) подведены итоги первого этапа конкурса.

На первом этапе членами конкурсной комиссии были рассмотрены общие концепции федеральных компонентов государственных образовательных стандартов (ФК ГОС), представленные творческими коллективами Института общеобразовательной школы РАО (рук. В. С. Леднев), Научно-методического центра стандартизации и диагностики Института общего образования Минобразования России (рук. М. Ф. Королев), МИПКРО (рук. Ю. С. Сахаров), Координационного совета по технологиям индивидуализации обучения (рук. Ю. С. Драль), Красноярского краевого центра развития образования (рук. Г. Н. Блинов), и три концепции по отдельным направлениям функционирования системы образования: концепция стандарта учебно-материальной базы общеобразовательной школы (рук. Т. С. Назарова, Институт средств обучения РАО), концепция стандарта государственной аттестационной службы (рук. С. Д. Некрасов, Государственная аттестационная служба Краснодарского края) и концепция школьного учебника (Д. Д. Зуев, Институт общего образования Минобразования России).

По итогам рассмотрения вышеперечисленных концепций из 90 членов конкурсной комиссии рецензии прислали 62 человека. В них дана экспертная оценка материалов и высказаны предложения по их ранжированию в порядке значимости.

Большинство членов конкурсной комиссии высказали мнение о необходимости доработки представленных концепций ФК ГОС. Высказывались предложения о создании коллектива для выработки единой общей концепции ФК ГОС. Большинство членов конкурсной комиссии считают, что основой для ее разработки должна стать концепция, представленная творческим коллективом ИОШ РАО под рук. В. С. Леднева.

Данная концепция получила наибольшее одобрение (1-е место – 38 чел., 2-е – 9 чел., 3-е – 7 чел.) и заняла первое место.

На втором месте – концепция коллектива под рук. М. Ф. Королева (1-е место – 9 чел., 2-е – 26 чел., 3-е – 5 чел., 4-е – 2 чел., 5-е – 3 чел.).

На третьем месте – концепция, разработанная творческим коллективом под рук. Ю. С. Сахарова (1-е место – 2 чел., 2-е – 14 чел., 3-е – 16 чел., 4-е – 6 чел., 5-е – 6 чел.).

Четвертое место заняла концепция, представленная коллективом под рук. Г. Н. Блинова (1-е место – 1 чел., 2-е – 5 чел., 3-е – 14 чел., 4-е – 9 чел., 5-е – 5 чел.).

Пятое место – концепция коллектива под рук. Ю. С. Драля (2-е место – 1 чел., 3-е – 9 чел., 4-е – 16 чел., 5-е – 5 чел.).

В отношении концепций по отдельным направлениям функционирования системы образования были высказаны мнения о дальнейшей их разработке, но вне рамок конкурса.

На основании решения конкурсной комиссии (протокол заседания конкурсной комиссии от 26.01.95 г. № 3) прошу:

1. Главное управление развития общего среднего образования (М. Р. Леонтьева):

1.1. Создать рабочую группу для разработки проекта общей концепции федеральных компонентов государственных образовательных стандартов и представить ее состав на утверждение в срок до 15 февраля 1995 года.

1.2. Подготовить проект общей концепции федеральных компонентов государственных образовательных стандартов в срок до 15 марта 1995 года.

2. Опубликовать проект общей концепции федеральных компонентов государственных образовательных стандартов для широкого обсуждения педагогической общественностью.

3. Контроль за выполнением распоряжения возложить на заместителя Министра А. Г. Асмолова.

Министр

Е. В. Ткаченко

СОСТАВ

конкурсной комиссии по образовательным стандартам

В состав конкурсной комиссии по образовательным стандартам входят: председатель, заместитель председателя, ответственный секретарь и 90 членов комиссии из 16 регионов Российской Федерации:

- 4 республик в составе России;
- 1 автономного округа;
- 1 края;
- 10 областей;
- Москвы и Санкт-Петербурга.

Из 90 членов конкурсной комиссии:

- 10 академиков (7 академиков РАО, 2 академика РАН, 1 академик Академии наук Татарстана);
- 6 членов-корреспондентов (4 – РАО, 2 – РАН);
- 1 член Президентского Совета;
- 2 вице-президента РАО;
- 6 работников ИПК;
- 14 практических работников (руководители органов управления образованием, учителя);
- 51 – преподаватели вузов, сотрудники НИИ, работники издательств.

Уважаемые читатели!

Напоминаем вам, что с 1 апреля начинается подписка на второе полугодие 1995 г. Сведения, необходимые для подписки на журнал «Информатика и образование»:

	Индекс издания	Цена издания по каталогу в руб.	Периодичность
для индивидуальных подписчиков	70423	24 000 (за три номера)	1 раз в 2 месяца
для предприятий и организаций	73176	75 000 (за три номера)	

ОБРАЩАЕМ ВАШЕ ВНИМАНИЕ НА ТО, ЧТО НА ЖУРНАЛ МОЖНО ПОДПИСАТЬСЯ ПО БЕЗНАЛИЧНОМУ РАСЧЕТУ

Телефон для справок: (095) 208-30-78

Факс: (095) 208-67-37

E-Mail: info@tit-bit.msk.su

И. Ф. Базлов,

*главный специалист Управления развития муниципальной системы образования
Комитета по образованию Санкт-Петербурга*

А. Н. Шляго,

доцент кафедры информатики и вычислительной техники РГПУ им. А. И. Герцена

ЭКЗАМЕНАЦИОННЫЕ МАТЕРИАЛЫ ПО ИНФОРМАТИКЕ (К разработке образовательных стандартов Санкт-Петербургской школы)

Пояснительная записка

Весной 1994 г. учащиеся ряда школ Санкт-Петербурга сдавали выпускной экзамен по информатике независимой городской экзаменационной комиссии. Это мероприятие проводилось впервые, поэтому перед комиссией, созданной Комитетом по образованию Санкт-Петербурга, встал целый ряд проблем, сущность которых становится понятной, если учесть современное состояние информатики как учебной дисциплины. А состояние это следует охарактеризовать как крайне неопределенное: цели, задачи и даже состав и содержание базовых понятий курса трактуются в высшей степени свободно, если не сказать — произвольно. Это приводит к избыточному разнообразию конкретных учебных программ, в большинстве случаев оказывающихся односторонними, далеко не отражающими состояние информатики как науки, изучающей ВСЕ аспекты получения, хранения, преобразования, передачи и использования информации.

Можно было бы привести целый ряд причин, обуславливающих такое положение дел, однако в первую очередь следует указать на главную — сложную дисциплинарную структуру самой информатики как науки, объединяющей целый ряд научных направлений, исследующих разные стороны одного и того же объекта — информации.

Большинство же реально действующих в школах учебных программ решают задачи формирования компьютерной грамотности (в лучшем случае — информационной культуры) и/или обучения программированию. Несостоятельность столь узкой трактовки задач школьного курса информатики можно, по-видимому, считать сегодня уже общепризнанной.

В целом ряде публикаций четко выражена мысль о том, что если оставить за школьной информатикой лишь ее традиционную задачу: формирование компьютерной грамотности, то уже в скором будущем встанет вопрос о целесообразности изучения в школе информатики как самостоятельной дисциплины. Аргумент прост — в условиях предстоящего массового внедрения вычислительной техники в школу умения, составляющие «компьютерную грамотность», приобретают характер общеучебных и будут осваиваться почти в каждом школьном курсе.

Недопустимым считается и ограничение курса информатики в основном задачами обучения алгоритмизации и программированию. Интересно, что аргументация, выдвигаемая в поддержку этого мнения, во многом совпадает с оценкой перспектив и целей преподавания информатики, сформулированной в журнале «Education and Computers» (1985, № 1), т. е. когда мы еще только приступали к массовому введению курса информатики в средних школах. Там пишут: «В большинстве стран компьютеры впервые использовались в школах для обучения программированию. Имеет смысл объяснить детям, как работает ЭВМ, но это не более и не менее важно, чем рассказать о телевизоре, телефоне, реактивном двигателе или атомном реакторе, которые являются частью нашего технического окружения. С другой стороны, если ребята интересуются программированием, то нет причин, чтобы не сделать такой курс факультативным... нет причин, по которым следовало бы обучать именно программированию, а не водопроводному делу, ремонту электроприборов, телевизоров и т. п.». Там же отмечается, что нет оснований полагать, будто приемы профессионального мышления программистов обладают общечеловеческой значимостью, как, например, математическая культура.

Все это, конечно, не значит, что из курса информатики должны быть исключены вопросы алгоритмизации и программирования, однако несомненно, что акценты следует изменить.

Подобное изменение акцентов неизбежно, если встать на путь формирования нового подхода к целям изучения информатики как фундаментальной научной дисциплины. Основы такого подхода достаточно четко сформулированы и изложены в целом ряде публикаций, важнейшими среди которых нам представляются проект Российского стандарта школьного образования по информатике (УГ. 1993. № 36) и «Информатика. Энциклопедический словарь для начинающих» под редакцией академика Д. А. Поспелова, выпущенный издательством «Педагогика-Пресс» в 1994 г.

Все сказанное позволяет утверждать, что мы являемся свидетелями и участниками процесса обретения школьной информатикой нового лица — процесса, который можно характеризовать как смену парадигмы.

В проекте стандарта можно выделить семь

содержательных линий школьного курса информатики: наряду с четырьмя традиционными — алгоритмы, компьютер, программное обеспечение, технология решения задач с использованием компьютера — там сформулированы три новые линии — формальное исчисление, информационные процессы, моделирование.

Расширение курса новыми содержательными линиями предъявляет жесткие требования к организации учебного процесса. Перед педагогическими коллективами школ встает непростая проблема реализации практически нового курса в условиях отсутствия адекватного учебника, неразработанности содержания (новые линии в стандарте только намечены), ограниченного объема часов (учебный план ориентирован на традиционное содержание курса).

Отдавая в связи с этим себе отчет в том, что требования проекта стандарта обращены в первую очередь к разработчикам новых учебных программ и учебных пособий, следует признать, что обозначенные в проекте новые содержательные линии могут быть приняты за основу при создании Петербургского стандарта школьного образования по информатике. Детальная содержательная интерпретация этих линий позволит учесть специфику Петербургской школы, ориентированной на достижение выпускниками повышенного уровня образованности, развитие у них целостного видения мира, формирование умения решать комплексные задачи, понимать глобальные проблемы современности. В частности, такие содержательные линии, как *информационные процессы и моделирование*, при соответствующем методическом исполнении открывают широкие возможности для реализации принципов проблемного обучения, подведения учащихся к осмыслению методологии научного познания, развития у них навыков диалектического мышления. Здесь предоставляется возможность по-новому сформулировать и реализовать цели курса школьной информатики: вместо утилитарной ориентации на развитие у учащихся информационной культуры на первый план выдвинуть задачу освоения ими современной методологии приобретения знаний об окружающем мире и о себе. В такой постановке школьная информатика из некой вспомогательной превращается в дисциплину, способствующую формированию у учащихся целостного мировоззрения, характеризующегося осознанием мира (природы и общества) как единой системы энергоинформационных процессов; осознанием интеллекта в роли решающего фактора глобального исторического процесса; а самого этого процесса как частного в эволюционном процессе Земли.

Все сказанное, на наш взгляд, явилось достаточным основанием для того, чтобы использовать проект Российского стандарта школьного образования по информатике (далее — ПРС) в качестве ориентационного нормативного материала при проведении выборочной аттестации выпускников образовательных учреждений Санкт-Петербурга.

В процессе аттестации комиссией в основном решались две следующие задачи:

1. Исследовать, соответствуют ли требованиям Федерального образовательного стандарта и ориентации Петербургской школы:

- предлагаемые образовательными учреждениями экзаменационные материалы;
- методическая ориентация учителей;
- имеющаяся в образовательных учреждениях техническая база и программное обеспечение.

2. Установить, в какой мере демонстрируемый на экзаменах уровень подготовки учащихся соответствует заявленному образовательному уровню класса.

Для решения первой из перечисленных задач форма, в которой опубликован ПРС, малоприспособна. В связи с этим нам пришлось осуществить некоторую адаптацию этого материала. Дело в том, что проект стандарта опубликован в двух вариантах, во многом дублирующих друг друга. Нами были объединены оба варианта. При этом все дублирования устранены, а также выполнены некоторая перегруппировка материала и разделение содержательных линий. Все это без каких-либо потерь содержания. В таком виде Основные содержательные линии Российского стандарта школьного образования по информатике (см. ниже) вполне могут быть использованы для анализа, необходимого в процессе решения первой из поставленных перед комиссией задач.

Вторая задача потребовала разработки нормативной базы, которая должна обеспечить единообразие и сопоставимость результатов работы комиссии при оценке уровня подготовки выпускников различных классов. Исходным материалом для создания такой базы послужили Примерные вопросы по информатике и вычислительной технике, рекомендованные Министерством образования для использования на выпускных экзаменах в 1993/94 учебном году (Вестник образования. 1993. № 12).

По каждому вопросу были сформулированы требования к знаниям и умениям, которые должен продемонстрировать экзаменуемый для того, чтобы получить высшую оценку в классах, заявленных на повышенный образовательный уровень. А затем произведено ослабление этих требований по трем уровням, соответствующим отличной, хорошей и удовлетворительной оценкам для базового уровня подготовки. Таким образом, по каждому вопросу были выдвинуты четыре уровня требований, что обеспечило возможность раздельной оценки для базового (три нижних уровня) и повышенного (три верхних уровня) вариантов подготовки, и таким образом создана нормативная база, необходимая для работы комиссии (см. ниже Экзаменационные требования к выпускникам за курс средней школы). Пометкой (5п) обозначены требования к знаниям и умениям, которые должен продемонстрировать экзаменуемый, чтобы получить высшую оценку, если класс заявлен на повышенный образовательный уровень. Пометками (3), (4), (5) обозначены требования к знаниям и умениям, которые должны продемонстрировать экзаменуемые для получения удовлетворительной, хорошей и отличной оценок при подготовке по базовому варианту. Требования сформулированы «по нарастанию», т. е. для полу-

чения хорошей оценки необходимо обладать знаниями с пометками (3) и (4), а для отличной — с пометками (3), (4), (5). В классах, заявленных на повышенный образовательный уровень, оценка ведется по схеме:

удовлетворительно — (3)+(4),
хорошо — (3)+(4)+(5),
отлично — (5п).

Необходимо отметить, что эти требования формулировались, исходя главным образом из действующих на настоящий момент программ (Программы средней общеобразовательной школы. Основы информатики и вычислительной техники. М.: Просвещение, 1991), и ни в коей мере не отражают новых содержательных линий проекта Российского стандарта школьного образования по информатике, т. е. оценка велась с ориентацией на традиционные содержательные линии школьного курса информатики, что в настоящий момент, очевидно, является единственно возможным.

Кроме всего прочего, была разработана схема отчета о ходе и результатах экзамена (см. ниже), которая регламентирует деятельность членов комиссии в процессе проведения, предварительной обработки и обобщения результатов экзамена в отдельном классе.

Не останавливаясь подробно на результатах работы комиссии, приведем здесь лишь те рекомендации, которые комиссия сочла необходимым сформулировать по результатам своей работы.

1. Рекомендации руководству РОУНО

Организовать осенью 1994 г. широкое обсуждение проекта Российского стандарта школьного образования по информатике, а также экзаменационных материалов предметной комиссии по информатике с целью учесть мнение педагогической общественности при подготовке к итоговой аттестации в следующем учебном году.

Стимулировать содержательную и методическую разработку новых направлений (формальное исчисление, информационные процессы, моделирование), декларированных проектом Российского стандарта школьного образования по информатике, но не подкрепленных на настоящий момент соответствующей учебной и методической литературой.

2. Рекомендации Университету педагогического мастерства

Предложить методическим службам города усилить контроль за использованием авторских программ, предъявляя повышенные требования к обоснованности таких программ и полноте охвата ими фундаментальных понятий и методов информатики.

Организовать исследование возможности проведения письменных экзаменов по информатике, что позволило бы резко повысить сопоставимость и достоверность оценок качества подготовки выпускников, а также эффективность и широту охвата независимой экспертизой образовательных учреждений города.

Организовать постоянно действующий городской семинар по школьной информатике с целью объединить ведущие педагогические коллективы школ и вузов исследования в этой области, обеспечить возможность регулярной публикации его материалов.

3. Рекомендации администрации школ

Более тщательно и ответственно подходить к решению вопроса о целесообразности использования авторских программ по информатике, учитывать их обоснованность и согласованность с методическими службами города.

Утверждая программы классов, подготовка которых ведется с ориентацией на потребности того или иного вуза, особое внимание обратить на то, чтобы это делалось не в ущерб общеобразовательным целям курса, который в любом случае должен обеспечивать прочное усвоение фундаментальных понятий и методов информатики.

Принять меры к оснащению школы техническими и/или программными средствами, абсолютно необходимыми для глубокого и всестороннего усвоения школьного курса информатики.

Стимулировать профессиональный рост и творческую активность педагогических кадров в области информатики.

4. Рекомендации учителям информатики

Принять к сведению существенно обновленную по сравнению с традиционной концепцией школьного курса информатики, сформулированную в проекте Российского стандарта школьного образования по информатике.

Больше внимания обратить на изучение фундаментальных понятий, принципов и методов информатики, таких, а частности, как

- информация (сигнал как носитель информации; схема передачи сообщений; общность информационных процессов в самоуправляемых системах различной природы; особенности восприятия сообщений);
- алгоритм (свойства алгоритма, исполнители алгоритмов, практические навыки алгоритмизации);
- компьютер (функциональная организация ЭВМ; общие принципы работы компьютера; спектр современной компьютерной техники, ее характеристики и области применения);
- программное обеспечение (основные виды ПО; практические навыки работы, а не просто знакомство, с текстовым и графическим редакторами, электронной таблицей, СУБД, сетевым ПО, операционной системой).

Особое внимание обратить на технологию решения задач с использованием компьютера, поскольку в программах большинства школ крайне слабо отражены вопросы технологии решения задач с использованием компьютера, о чем, в частности, свидетельствует сам характер задач, предлагавшихся на экзамене: это были простейшие задачи, не позволявшие выявить владение принципами модульного программирования и проектирования «сверху вниз»; при этом никаких требований к тестированию полученных программ, к форматированию ввода-вывода, к контролю и верификации вводимых данных не предъявлялось.

Решая любую задачу, акцентировать все звенья технологической цепочки объект — постановка задачи — построение информационной модели — построение алгоритма — компьютерная реализация — получение и анализ результатов.

Как можно раньше и систематически прививать учащимся навыки последовательной детализации, структурной алгоритмизации и модульного программирования.

Добиваться от учащихся при решении любой задачи реализации удобного пользовательского интерфейса, обеспечивающего, помимо прочего, контроль и возможность верификации вводимых данных.

Основные содержательные линии Российского стандарта школьного образования по информатике (адаптированный вариант)

1. Формальное исчисление

Базовый уровень предъявления материала должен обеспечить учащимся возможность:

- 1.1. овладеть понятиями *знак, буква, информация*;
- 1.2. овладеть понятием *информация*;
- 1.3. овладеть понятиями и аппаратом исчисления высказываний: *истина, ложь, высказывания, связи*;
- 1.4. освоить основные кванторы;
- 1.5. понимать соотношение естественного языка и формального исчисления.

В соответствии с уровнем обязательной подготовки учащиеся должны:

- знать содержание понятий *знак, буква, информация*;
- знать основные типы логических связей;
- уметь составлять таблицы истинности для простейших логических формул;
- уметь записывать предложения на естественном языке в виде формул исчисления высказываний.

2. Алгоритмы

Базовый уровень предъявления материала должен обеспечить учащимся возможность:

- 2.1. усвоить (на основе анализа примеров) смысл

понятия *алгоритм*, понять возможность автоматизации деятельности человека при выполнении алгоритмов;

2.2. освоить основные алгоритмические конструкции (цикл с предусловием, цикл с постусловием, цикл с параметром, серия, ветвление, выбор, процедура) и типы данных (числовые и текстовые величины, массив, связанный список, дерево), научиться использовать их для решения задач на составление алгоритмов;

2.3. получить представление о «библиотеке алгоритмов», уметь использовать библиотеку для построения более сложных алгоритмов;

2.4. научиться использовать какой-либо формальный язык (язык программирования или учебный алгоритмический язык) для записи алгоритмов решения простых задач.

В соответствии с уровнем обязательной подготовки учащиеся должны:

- понимать сущность понятия алгоритм, знать его основные свойства, иллюстрировать их на конкретных примерах алгоритмов;
- понимать возможность автоматизации деятельности человека при выполнении алгоритмов;
- знать и уметь распознавать основные алгоритмические конструкции в записи алгоритма на каком-либо формальном языке;
- решать стандартные типы задач на составление алгоритмов с использованием основных алгоритмических конструкций;
- определять возможность применения исполнителя для решения конкретной задачи по системе его команд;
- построить и исполнить на компьютере простой алгоритм для учебного исполнителя (типа «черепаха», «робот» и т. д.).

3. Информационные процессы

Базовый уровень предъявления материала должен обеспечить учащимся возможность:

3.1. получить представление о структуре и назначении основных элементов самоуправляемых систем, функциях обратной связи;

3.2. получить представление об общности информационных процессов, строения и функционирования самоуправляемых систем различной природы;

3.3. получить представление о сигнале как носителе информации, процессе передачи информации, канале связи, его структуре и назначении отдельных элементов;

3.4. познакомиться с двоичной формой представления информации, ее особенностями и преимуществами;

3.5. познакомиться со способом измерения информации, единицами количества информации (бит, байт, килобайт и т. д.);

3.6. получить представление о содержании процесса информатизации общества и роли в нем средств вычислительной техники и связи;

3.7. понять смысл и значение информационной точки зрения на мир.

В соответствии с уровнем обязательной подготовки учащиеся должны:

- уметь приводить примеры передачи, хранения и обработки информации в деятельности человека, живой природе, обществе и технике;
- иметь представление об особенностях самоуправляемых систем, общности информационных принципов строения и функционирования управляющих органов этих систем независимо от их природы;
- иметь представление об обратной связи, о принципах работы замкнутых и разомкнутых систем управления;
- понимать назначение датчиков, исполнительных органов, преобразователей информации в системах управления различной природы;
- иметь представление о двоичной системе счисления;
- иметь представление о мере количества информации, знать основные единицы количества информации.

4. Компьютер

Базовый уровень предъявления материала должен обеспечить учащимся возможность:

4.1. иметь представление о функциональной организации ЭВМ, общих принципах работы основных устройств компьютера и его периферии;

4.2. понять принципы кодирования и представления информации в компьютере;

4.3. понять смысл принципа автоматического исполнения программ на ЭВМ;

4.4. прийти к пониманию того, что компьютер — это универсальное орудие для работы с формально-знаковыми конструкциями;

4.5. ознакомиться со спектром современной компьютерной техники, ее характеристиками и областями применения.

В соответствии с уровнем обязательной подготовки учащиеся должны:

- знать функциональную схему (название и назначение основных устройств) организации компьютера;
- понимать принцип автоматического исполнения программ;
- иметь представление о принципах кодирования информации;
- иметь представление о характеристиках основных типов современных компьютеров;
- знать технику безопасности при работе с персональным компьютером.

5. Программное обеспечение

Базовый уровень предъявления материала должен обеспечить учащимся возможность:

5.1. знать названия и иметь представление о назначении основных видов программного обеспечения ЭВМ: функций базового программного обеспечения, назначения программы-транслятора, применении языков программирования, прикладного программного обеспечения;

5.2. пользоваться текстовым редактором, организовывать хранение текстов во внешней памяти и вывод их на печать в соответствии со стандартным форматом;

5.3. пользоваться простым графическим редактором;

5.4. обращаться с запросами к базе данных, выполнять основные операции над данными;

5.5. осуществлять основные операции над электронными таблицами, выполнять простейшие вычисления над таблицей;

5.6. пользоваться экспертной системой;

5.7. познакомиться с основными этапами развития информационно-вычислительной техники и программного обеспечения ЭВМ.

В соответствии с уровнем обязательной подготовки учащиеся должны:

- иметь представления о программном обеспечении компьютера;
- уметь пользоваться клавиатурой ЭВМ;
- уметь использовать «меню», «запрос о помощи», инструкции для пользователя;
- исполнить в режиме диалога простую прикладную программу (типа: «решение квадратного уравнения», «построение графиков функций» и т. п.);
- набрать на компьютере и откорректировать простой текст;
- построить простейшее изображение с помощью графического редактора;
- уметь обращаться с запросами к базе данных;
- выполнять простейшие вычисления, используя электронную таблицу (типа подсчета общей стоимости покупок в магазине);
- уметь самостоятельно выполнить на компьютере простое задание, используя основные функции инструментальных программных средств, прикладных программ.

6. Технология решения задач с использованием компьютера

Базовый уровень предъявления материала должен обеспечить учащимся возможность:

6.1. понимать назначение и содержание основных звеньев технологической цепочки «объект — постановка задачи — построение информационной модели — построение алгоритма — компьютерная реализация — получение и анализ результатов»;

6.2. уметь правильно подбирать программно-аппаратные средства, необходимые для решения поставленной задачи;

6.3. применять учебные пакеты прикладных программ для решения типовых учебных задач;

6.4. научиться реализовывать учебно-исследовательские проекты.

В соответствии с уровнем обязательной подготовки учащиеся должны:

- знать структуру основной технологической цепочки, содержание и назначение ее звеньев;
- уметь решать типовые задачи с использованием учебных программных средств.

7. Моделирование

Базовый уровень предъявления материала должен обеспечить учащимся возможность:

7.1. получить представление о моделировании как о методе познания и об основных типах моделирования (математическое, физическое и пр.);

7.2. понять смысл отношений «объект — модель — язык»;

7.3. узнать смысл аналогии «язык — средства моделирования»;

7.4. получить представление о методе информационного моделирования как методе современного естествознания, узнать его сущность;

7.5. овладеть навыками структурирования, т. е. уметь строить описания объектов в терминах «элемент — свойство — отношение»;

7.6. научиться использовать графы для представления простейших моделей;

7.7. узнать основные виды моделей (классификационные, динамические и формально-языковые) и сферы их применения;

7.8. понимать связь типов программных средств и видов информационных моделей;

7.9. познакомиться с возможностями изучения «поведения» моделей на компьютере.

В соответствии с уровнем обязательной подготовки учащиеся должны:

- иметь представление о методе моделирования, основных типах моделирования;
- знать сущность метода информационного моделирования;
- уметь строить простейшие информационные модели;
- понимать смысл отношения «объект — модель».

Экзаменационные требования к выпускникам за курс средней школы

1. Компьютер. Магистрально-модульный принцип построения

(5п): Знать назначение, а также общие принципы работы и организации взаимодействия основных устройств компьютера (центральный процессор со схемами системной поддержки, основная память в составе постоянного запоминающего устройства и оперативной памяти, внешние устройства, контроллеры внешних устройств, системная плата с системной шиной и разъемами расширения, блок питания). Знать принципы хранения программы,

двоичного представления информации, автоматического исполнения программы.

(3): Знать назначение основных устройств ПК и структурную схему ПЭВМ.

(4): Знать общие принципы работы основных устройств ПК (процессор, память, устройства ввода/вывода). Знать принципы хранения программы, двоичного представления информации и автоматического исполнения программы.

(5): Знать принципы организации взаимодействия основных устройств ПК.

2. Разрядность процессора. Адресное пространство процессора

(5п): Знать назначение и разрядность программно доступных регистров центрального процессора. Иметь понятие о минимальной адресуемой процессором ячейке основной памяти. Знать схему формирования абсолютного (физического) адреса, размер адресного пространства и конструктивные особенности, его определяющие.

(3): Иметь понятие о минимальной адресуемой процессором ячейке основной памяти; владеть понятиями *адресное пространство процессора* и *разрядность процессора*.

(4): Владеть понятием *размер адресного пространства*; знать схему формирования абсолютного (физического) адреса.

(5): Знать назначение и разрядность программно доступных регистров центрального процессора.

3. Структура памяти. Оперативная память (область программ пользователя и видеопамять). Постоянное запоминающее устройство

(5п): Знать назначение основной памяти; специфику и диапазоны адресов ПЗУ, ППЗУ и ОЗУ; назначение видеопамати; распределение оперативной памяти после загрузки операционной системы. Владеть понятием *виртуальный диск*.

(3): Знать назначение памяти ЭВМ и ее структуру; виды памяти, их назначение.

(4): Знать назначение основной памяти, ПЗУ, ОЗУ, видеопамати; распределение памяти после загрузки системы.

(5): Знать специфику ППЗУ; диапазоны адресов основной памяти, ОЗУ, ПЗУ и видеопамати.

4. Внешняя память (дисководы, диски)

(5п): Знать назначение и принципы управления дисковой памятью (контроллеры, команды обслуживания портов ввода/вывода, дисководы); отличия и специфику использования НГМД и НТМД (НЖМД); устройство дисковода, физическую и логическую систему гибкого диска. Уметь копировать, форматировать и создавать системные диски.

(3): Знать назначение, виды дисковой памяти и правила обращения с дискетами.

(4): Уметь копировать, форматировать и создавать системные диски.

(5): Знать устройство дисковода; физическую и логическую структуру гибкого диска.

5. Матричный принцип печати. Принтер

(5п): Знать назначение и принципы действия принтеров основных типов (матричные, струйные, литерные; лазерные); роль буфера в организации их работы; потребительские качества принтеров (скорость и качество печати, надежность, возможность автоматической подачи бумаги, количество и разнообразие шрифтов, наличие русских букв, совместимость по управлению с наиболее распространенными типами принтеров); преимущества и недостатки основных типов принтеров. Знать основные параметры печати: режим печати (текстовый/графический), шрифт и его модификации (размер букв и способы их выделения), максимальное число символов в строке и

строк на странице, размеры полей, интервал между строками, установка /отмена остановки при отсутствии бумаги.

(3): Знать назначение и основные типы принтеров (матричные, струйные, литерные, лазерные); уметь вывести файл на принтер.

(4): Знать основные параметры печати и потребительские качества принтеров (см. основной текст); понимать принцип действия матричного принтера.

(5): Знать преимущества и недостатки основных типов принтеров; понимать роль буфера в организации работы принтера.

6. Операционная система (назначение, состав, загрузка)

(5п): Иметь представление об основных классах программных систем (операционные системы, системы программирования, прикладные программные системы). Знать назначение и состав операционной системы (ОС). Перечислить встроенные и дисковые компоненты ОС и охарактеризовать их функции в процессе загрузки, а также при управлении процессом обработки данных. Знать основные команды управления ОС.

(3): Знать назначение ОС; перечислить ее встроенные и дисковые компоненты. Уметь подготовить компьютер к работе (загрузить ОС).

(4): Иметь представление об основных классах программных систем (операционные системы, системы программирования, прикладные программные системы). Знать основные команды управления ОС.

(5): Охарактеризовать функции встроенных и дисковых компонент ОС в процессе загрузки.

7. Файл. Файловая система

(5п): Владеть понятиями *файл, каталог, подкаталог, текущий и рабочий каталог, текущий и рабочий дисковод, полное имя файла, групповое имя*. Знать принципы организации файловой системы, включая логическую структуру диска. Уметь использовать основные команды работы с файлами и каталогами (просмотр и создание каталога, смена текущего каталога, смена текущего дисковода, копирование и создание файла, просмотр файла, удаление и переименование файла), а также создавать командные файлы.

(3): Владеть понятиями *файл, каталог, подкаталог и уметь использовать основные команды работы с файлами и каталогами (просмотр каталога, смена текущего дисковода, копирование, просмотр, удаление файлов)*.

(4): Владеть понятиями *текущий и рабочий дисковод, а также текущий и рабочий каталог*. Уметь правильно использовать полное и групповое имя файла (копировать, удалять и переименовывать группу файлов).

(5): Уметь правильно использовать полное и групповое имя файла во всех командах работы с файлами и каталогами; владеть операциями перенаправления и страничного вывода.

8. Машинная графика. Назначение и основные функции

(5п): Знать назначение графических редакторов, основные их типы и отличия друг от друга. Владеть основными возможностями, предоставляемыми графическим редактором: выполнение и редактирование рисунков, сохранение рисунков на магнитном носителе, чтение оглавления рисунков, вывод рисунков на печатающее устройство, передача рисунков по сети.

(3): Знать назначение графического редактора и владеть основными возможностями, им предоставляемыми: выполнение и редактирование рисунков, сохранение рисунков на магнитном носителе, чтение оглавления рисунков.

(4): Уметь вывести рисунок на печатающее устройство.

(5): Уметь передать рисунок по сети.

9. Системы управления базами данных (СУБД). Назначение и основные функции

(5п): Знать назначение СУБД и владеть основными возможностями, ими предоставляемыми: создание БД, наполнение БД, редактирование, поиск, упорядочение информации, печать найденной информации, применение макроязыка. Владеть понятиями *предметная область, модель предметной области, концептуальная схема БД*. Иметь представление о трех основных моделях данных. Для реляционной модели уметь охарактеризовать принципы организации данных и основные операции над данными.

(3): Знать назначение СУБД. Уметь наполнять и редактировать БД, а также осуществлять в ней поиск.

(4): Иметь представление о трех основных моделях данных. Уметь создавать БД.

(5): Для реляционной модели уметь охарактеризовать принципы организации данных и основные операции над ними. Уметь упорядочивать информацию в БД; вывести найденную информацию на печать.

10. Электронные таблицы. Назначение и основные функции

(5п): Знать назначение электронных таблиц и владеть основными возможностями, ими предоставляемыми: заполнение бланков, автоматическое выполнение расчетов, внесение изменений в данные, вывод бланков на печать, запись/чтение бланков с магнитного носителя, передача бланков по сети, получение справок о системе команд, применение макроязыка. Уметь выполнять сложные расчеты с использованием функций и условных выражений.

(3): Знать назначение электронных таблиц. Уметь получать справки о системе команды, заполнять бланки электронной таблицы, автоматически выполнять расчеты.

(4): Осуществлять формирование бланков, запись/чтение бланков с магнитного носителя, выполнять операции с блоками ячеек.

(5): Уметь строить графики функций, заданных в таблице; передавать текст по сети.

11. Текстовый редактор. Назначение и основные функции

(5п): Знать назначение текстовых редакторов и их преимущества по сравнению с пишущей машинкой. Иметь понятие об издательских системах (их возможности и требования к аппаратному обеспечению). Владеть основными возможностями, предоставляемыми текстовыми редакторами: подготовка и редактирование текстов (включая работу с блоками и поиск в тексте), запись/чтение текстов с магнитного носителя, вывод текстов на печать, передача текстов по сети, получение справок о системе команд.

(3): Знать назначение текстового редактора и его преимущества по сравнению с пишущей машинкой. Уметь подготовить и вывести на печать простейший текст; получить справку о системе команд.

(4): Уметь работать с блоками, осуществлять поиск в тексте, производить запись/чтение текстов с магнитного носителя.

(5): Уметь работать с окнами; форматировать абзацы и страницы; осуществлять установку режимов печати; передавать текст по сети.

12. Локальные и телекоммуникационные сети. Компьютерная почта

(5п): Знать назначение, преимущества и общие принципы организации компьютерных сетей; понимать различия между локальными и глобальными компьютерными сетями; владеть понятиями *файлсервер и терминал*; знать аппаратное и программное обеспечение работы в локальных сетях; уметь продемонстрировать работу в локальной сети (включение, регистрация, передача файлов); иметь представление об условиях, необходимых и достаточных для осуществления телекоммуникационного сеанса связи (линии связи, помехи, протокол исправления ошибок); о скорости передачи информации в телекоммуника-

циоинной сети, ее значении; о принципе работы модема, о телекоммуникационных компьютерных сетях в России и в мире (назначение, области использования). Уметь производить действия для осуществления телекоммуникационного сеанса связи; отправить и получить почту по e-mail, продемонстрировать работу в телеконференции.

(3): Знать назначение, преимущества и общие принципы организации компьютерных сетей; понимать различие между локальными и глобальными компьютерными сетями; владеть понятиями *файлсервер* и *терминал*; иметь представление об условиях, необходимых и достаточных для осуществления телекоммуникационного сеанса связи (линии связи, помехи, протокол исправления ошибок); об аппаратном и программном обеспечении работы в локальных сетях.

(4): Знать локальные сети с выделенным и невыделенным файлсервером. Иметь представление о скорости передачи информации в телекоммуникационной сети, ее значении; о принципе работы модема, о телекоммуникационных компьютерных сетях в России и в мире (назначение, области использования).

(5): Продемонстрировать работу в локальной сети (включение, регистрация, передача файлов). Уметь производить действия для осуществления телекоммуникационного сеанса связи; отправить и получить почту по e-mail, продемонстрировать работу в телеконференции.

13¹. Информатизация общества. Развитие вычислительной техники

(5п): Уметь охарактеризовать информацию как ресурс науки и производства; раскрыть сущность процесса информатизации общества; проиллюстрировать темпы роста научно-технической информации; обосновать необходимость совершенствования методов и средств обработки, хранения и передачи информации. Знать основные этапы (поколения) и перспективы развития вычислительной техники (специфику аппаратных и программных средств, архи текстурных решений, а также режимов взаимодействия с пользователем). Приводить примеры применения ЭВМ в научных исследованиях, проектировании, производстве, обслуживании, управлении. Знать основные потребительские характеристики ЭВМ.

(3): Знать основные этапы (поколения) и перспективы развития вычислительной техники. Приводить основные примеры применения ЭВМ, понимать сущность процесса информатизации общества.

(4): Знать перспективы развития вычислительной техники, охарактеризовать информацию как ресурс науки и производства, раскрыть сущность процесса информатизации общества, обосновать необходимость совершенствования методов и средств обработки, хранения и передачи информации. Приводить примеры применения ЭВМ в научных исследованиях, проектировании, производстве, обслуживании, управлении.

(5): Знать перспективы развития вычислительной техники со спецификой аппаратных и программных средств, архитектурных решений, а также режимов взаимодействия с пользователем.

14. Информация. Единицы измерения информации

(5п): Уметь охарактеризовать обыденное представление об информации как о сведениях или знаниях, которые кого-либо интересуют; привести примеры различных видов информации; проиллюстрировать зависимость воспринимаемой человеком информации от его интереса к ней, характера имеющихся у него знаний, а также от способности ее понять.

Обосновать необходимость введения «кибернетического» понятия информации как произвольного текста, т. е. последовательности символов некоторого алфавита. Понимать, что «кибернетическое» определение информации и ее количества необходимо при ее обработке, хранении и передаче в автоматических каналах связи. Иметь представление о сигнале как носителе информации; о процессе передачи информации. Знать схему передачи

сообщений. Знать особенности двоичного представления информации; названия и соотношения основных единиц измерения информации. Понимать, что количество информации, содержащейся в передаваемом сообщении, зависит от способа кодирования.

Уметь определять количество информации в конкретных сообщениях (при заданном способе кодирования); определять объем памяти ЭВМ, необходимой для размещения данных. Знать способы защиты информации при ее передаче от сбоя.

(3): Уметь охарактеризовать обыденное представление об информации, привести примеры различных видов информации.

Иметь представление о сигнале как носителе информации, о процессе передачи информации. Знать схему передачи сообщений, особенности двоичного представления информации; названия основных единиц измерения информации.

(4): Привести примеры различных видов информации; проиллюстрировать зависимость воспринимаемой человеком информации от его интереса к ней, характера имеющихся у него знаний, а также от способности ее понять. Знать соотношения основных единиц измерения информации. Понимать, что количество информации, содержащейся в передаваемом сообщении, зависит от способа кодирования.

(5): Уметь определять количество информации в конкретных сообщениях (при заданном способе кодирования), определять объем памяти ЭВМ, необходимой для размещения данных.

15. Понятие алгоритма. Исполнители алгоритмов

(5п): Иметь представление о свойствах исполнителя (система команд, среда, элементарные действия). Понимать смысл формального исполнения команд. Знать понятие алгоритма как организованной последовательности действий, доступных для некоторого исполнителя; свойства алгоритма (дискретность, детерминированность, результативность, массовость); способы записи алгоритма; понятие программы как алгоритма, записанного на языке, понятном исполнителю, имитируемому на ЭВМ. Понимать, что исполнитель алгоритмов — сочетание «рабочего инструмента» и устройства управления; что каждый исполнитель может использоваться для решения лишь определенного круга задач; что имитация с помощью ЭВМ исполнителя алгоритмов означает имитацию его элементарных (допустимых) действий и устройства управления. Владеть понятием правильности алгоритма. Уметь исполнять готовые алгоритмы, находить в них ошибки, оценивать область применимости, определять примерный набор допустимых действий для решения данного класса задач.

(3): Иметь представление о свойствах исполнителя (система команд, среда, элементарные действия). Понимать смысл формального исполнения команд. Знать понятие алгоритма как организованной последовательности действий, доступных для некоторого исполнителя; свойства алгоритма (дискретность, детерминированность, результативность, массовость); способы записи алгоритма; понятие программы как алгоритма, записанного на языке, понятном исполнителю, имитируемому на ЭВМ.

(4): Понимать, что исполнитель алгоритмов — сочетание «рабочего инструмента» и устройства управления; что каждый исполнитель может использоваться для решения лишь определенного круга задач; что имитация с помощью ЭВМ исполнителя алгоритмов означает имитацию его элементарных (допустимых) действий и устройства управления. Владеть понятием правильности алгоритма, знать типы алгоритмов.

(5): Уметь исполнять готовые алгоритмы, находить в них ошибки, оценивать область применимости, определять примерный набор допустимых действий для решения данного класса задач.

Иметь представление о принципах структурной алгоритмизации.

16. Разветвляющийся алгоритм

(5п): Владеть основными принципами структурной алгоритмизации; знать базовые структуры и правила их применения.

Знать определение разветвляющегося алгоритма и две формы ветвления: полное (имеющее две ветви) и неполное (имеющее одну ветвь). Понимать, что ветвление в алгоритмах появляется, когда исполнителю необходимо сделать выбор одного из нескольких наборов действий в зависимости от некоторого условия; что проверка такого условия должна являться допустимым действием исполнителя; что любой выбор можно свести к одному или нескольким ветвлениям; что при записи ветвлений необходим указатель конца ветвления, отделяющий ветвление от остальной части алгоритма (при отсутствии такого указателя алгоритм становится двусмысленным). Уметь составлять «протоколы» выполнения разветвляющихся алгоритмов; записывать разветвляющиеся алгоритмы, не допуская двусмысленности записи; применять ветвления при решении задач (при переходе от модели к алгоритму) использовать схемы (блок-схемы) или алгоритмический язык для записи разветвляющихся алгоритмов.

(3): Знать определение разветвляющегося алгоритма и две формы ветвления: полное (имеющее две ветви) и неполное (имеющее одну ветвь). Понимать, что ветвление в алгоритмах появляется, когда исполнителю необходимо сделать выбор одного из нескольких наборов действий в зависимости от некоторого условия; что проверка такого условия должна являться допустимым действием исполнителя. Уметь записывать разветвляющиеся алгоритмы, не допуская двусмысленности записи; применять ветвления при решении задач (при переходе от модели к алгоритму); использовать схемы (блок-схемы) или алгоритмический язык для записи разветвляющихся алгоритмов (все для какого-либо конкретного исполнителя).

(4): Уметь составлять «протоколы» выполнения разветвляющихся алгоритмов (для какого-либо конкретного исполнителя).

(5): Демонстрировать перечисленные умения применительно к любому из изучавшихся исполнителей.

17. Циклический алгоритм

(5п): Владеть основными принципами структурной алгоритмизации; знать базовые структуры и правила их применения.

Знать определения циклического алгоритма и особенности организации циклов с предусловием, с постусловием, с параметром. Понимать, что появление циклов в алгоритме обусловлено необходимостью повторять определенный набор действий до тех пор, пока выполняется некоторое условие; что при записи цикла необходим указатель конца цикла, отделяющий его от остальных действий алгоритма. Уметь составлять «протоколы» выполнения циклических алгоритмов; записывать циклические алгоритмы, не допуская двусмысленности записи; применять циклы при решении задач (при переходе от математической модели к алгоритму); использовать схемы (блок-схемы) или алгоритмический язык для описания циклических алгоритмов.

(3): Знать определения циклического алгоритма и особенности организации циклов с предусловием, с постусловием, с параметром. Понимать, что появление циклов в алгоритме обусловлено необходимостью повторять определенный набор действий до тех пор, пока выполняется некоторое условие; что при записи цикла необходим указатель конца цикла, отделяющий его от остальных действий алгоритма. Уметь применять циклы при решении задач (при переходе от математической модели к алгоритму); использовать схемы (блок-схемы) или алгоритмический язык для описания циклических алгоритмов.

(4): Уметь составлять «протоколы» циклических алгоритмов, содержащих простые циклы.

(5): Уметь составлять «протоколы» циклических алгоритмов, содержащих вложенные циклы.

18. Конструирование алгоритмов методом последовательной детализации. Вспомогательный алгоритм

(5п): Владеть приемами модульного проектирования алгоритмов. Понимать смысл и уметь обосновать целесообразность функциональной декомпозиции исходной постановки задачи.

Знать определение вспомогательного алгоритма как произвольного алгоритма, снабженного заголовком, позволяющим вызывать этот алгоритм из других алгоритмов. Понимать, что вспомогательные алгоритмы — этапы в пошаговой детализации алгоритмов; что при решении большинства практических задач можно пользоваться библиотекой вспомогательных алгоритмов (демонстрировать на примерах). Понимать, что в роли вспомогательного алгоритма может выступать любой алгоритм, если его оформить соответствующим образом, снабдив заголовком. Уметь применять метод пошаговой детализации алгоритмов; оформлять и вызывать вспомогательные алгоритмы; составлять протоколы алгоритмов, содержащих вспомогательные алгоритмы.

Владеть принципами и уметь приводить примеры рекурсивного использования вспомогательных алгоритмов.

(3): Знать определение вспомогательного алгоритма как произвольного алгоритма, снабженного заголовком, позволяющим вызывать этот алгоритм из других алгоритмов. Понимать, что вспомогательные алгоритмы — этапы в пошаговой детализации алгоритмов; что при решении большинства практических задач можно пользоваться библиотекой вспомогательных алгоритмов (демонстрировать на примерах). Уметь оформлять и вызывать вспомогательные алгоритмы.

(4): Уметь применять метод пошаговой детализации для простых задач.

(5): Уметь составлять «протоколы» алгоритмов, содержащих вспомогательные алгоритмы.

19. Этапы решения задач на ЭВМ

(5п): Понимать назначение, содержание и взаимосвязь основных звеньев технологической цепочки «объект — постановка задачи — построение информационной модели — построение алгоритма — компьютерная реализация — получение и анализ результатов». Уметь объяснить и проиллюстрировать на примерах важность постановки задачи и зависимость ее от целей исследования. Знать основные виды моделей (классификационные, динамические, формально-языковые) и сферы их применения. Понимать и иллюстрировать на примерах зависимость решения задачи от выбора модели. Владеть понятием правильности алгоритмов и программ. Иметь понятие о принципах подбора тестов для проверки алгоритмов и программ, используя индуктивные рассуждения. Понимать значение вычислительного эксперимента. Уметь анализировать соответствие модели и исходной постановки задачи.

(3): Понимать назначение, содержание и взаимосвязь основных звеньев технологической цепочки «объект — постановка задачи — построение информационной модели — построение алгоритма — компьютерная реализация — получение и анализ результатов».

(4): Уметь объяснить и проиллюстрировать на примерах важность постановки задачи и зависимость ее от целей исследования. Знать основные виды моделей (классификационные, динамические, формально-языковые) и сферы их применения. Понимать и иллюстрировать на примерах зависимость решения задачи от выбора модели. Владеть понятием правильности алгоритмов и программ.

(5): Иметь понятие о принципах подбора тестов для проверки алгоритмов и программ, используя индуктивные рассуждения. Понимать значение вычислительного эксперимента. Уметь анализировать соответствие модели и исходной постановки задачи.

20. Переменная: имя и значение. Идентификатор

(5п): Иметь понятие о величине как об элементарном средстве представления данных в процессе их обработки. Знать атрибуты переменной величины (имя, тип, значение) и уметь охарактеризовать их роль для задания процессов обработки данных. Знать виды величин, фигурирующих в описаниях алгоритмов. Уметь схематично описывать изменения памяти ЭВМ при исполнении команды присваивания.

(3): Иметь понятие о переменной как об участке памяти ЭВМ. Знать атрибуты переменной (имя, тип, значение). Уметь описать процесс задания значения переменной с помощью оператора присваивания.

(4): Знать типы переменных, способы их определения, а также допустимые операции над ними. Уметь охарактеризовать роль атрибутов переменных для задания процессов обработки данных.

(5): Уметь иллюстрировать с помощью модели памяти ЭВМ изменение памяти при исполнении команды присваивания. Знать виды величин, используемых в описании алгоритмов.

21. Типы переменных (целый, вещественный, символьный, логический)

(5п): Владеть понятием типа переменной величины как множества ее допустимых значений и множества применимых к ней операций. Уметь охарактеризовать каждое из этих множеств для переменного целого, вещественного, символьного и логического типов. Знать способы представления в памяти значений переменных перечисленных типов, а также правила описания типов переменных в программах.

Понимать условность понятий «целая переменная» и «вещественная переменная», связанную с ограниченностью разрядной сетки компьютера. Уметь определять погрешности представления конкретных вещественных чисел в памяти ЭВМ.

(3): Владеть понятием типа переменной величины. Знать правила описания типов переменных в программе.

(4): Уметь охарактеризовать множество допустимых значений и множество применимых операций для переменного целого, вещественного, символьного и логического типов. Знать способы представления в памяти значений для переменных перечисленных типов.

(5): Понимать условность понятий «целая переменная» и «вещественная переменная», связанную с ограниченностью разрядной сетки компьютера.

22. Функции обработки символьных (литерных) переменных

(5п): Уметь охарактеризовать тип символьной переменной (множество допустимых значений, множество применимых операций, способ представления в памяти ЭВМ). Знать правила описания переменных символьного типа. Знать функции обработки символьных переменных (формат, типы аргументов, тип значения).

(3): Знать функции обработки символьных переменных (формат, типы аргументов, тип значения).

(4): Знать правила описания переменных символьного типа.

(5): Уметь охарактеризовать тип символьной переменной (множество допустимых значений, множество применимых операций, способ представления в памяти ЭВМ).

23. Оператор присваивания. Арифметические выражения

(5п): Описать формат оператора присваивания, порядок его выполнения и происходящие при этом изменения в оперативной памяти для всевозможных типов переменной и выражения (в том числе, когда в выражении фигурирует переменная, стоящая слева от знака присваивания).

Знать правила записи и порядок вычисления значе-

ний арифметических выражений. Уметь записывать и читать арифметические выражения.

(3): Описать формат оператора присваивания. Уметь записывать и читать простые арифметические выражения.

(4): Описать порядок выполнения оператора присваивания и происходящие при этом изменения в оперативной памяти для всевозможных типов переменной и выражения (в том числе, когда в выражении фигурирует переменная, стоящая слева от знака присваивания).

(5): Уметь записывать и читать арифметические выражения произвольной сложности, свободно переходя от линейной формы записи к обычной и обратно.

24. Условный оператор. Полный и неполный варианты

(5п): Уметь охарактеризовать ветвление как одну из основных конструкций структурного программирования. Описать формат условного оператора и порядок выполнения его в полном и неполном вариантах. Знать правила записи и порядок вычисления значений логических выражений. Выполнять отрицание сложных логических выражений. Уметь записывать разветвляющиеся алгоритмы в виде программ. Владеть приемами отладки разветвляющихся программ; проводить вычислительные эксперименты по разветвляющимся программам.

(3): Описать формат условного оператора и порядок выполнения его в полном и неполном вариантах. Уметь записывать простые разветвляющиеся алгоритмы в виде программ.

(4): Знать правила записи и порядок вычисления значений логических выражений. Уметь записывать сложные разветвляющиеся алгоритмы (вложенные ветвления, композиция условий) в виде программ.

(5): Выполнять отрицание сложных логических выражений; владеть простейшими приемами отладки разветвляющихся программ.

25. Цикл, тело цикла, параметры цикла

(5п): Уметь охарактеризовать цикл как одну из основных конструкций структурного программирования.

Описать форматы всех операторов организации циклов в изучаемом языке программирования и порядок исполнения циклов, организованных с помощью этих операторов. Владеть понятиями *параметр цикла, тело цикла*. Уметь записывать циклические алгоритмы в виде программ; организовывать эквивалентные циклы без использования специальных операторов. Владеть приемами отладки циклических программ; проводить вычислительные эксперименты по циклическим программам.

(3): Описать семантику и синтаксис циклических конструкций. Знать порядок исполнения циклов, организованных с помощью операторов цикла. Владеть понятиями *параметр цикла, тело цикла*. Уметь записывать простые циклические алгоритмы в виде программ.

(4): Иметь четкое представление о характере изменения параметра в процессе выполнения цикла и о возможности преждевременного выхода из цикла. Уметь определить значение параметра после выхода из цикла. Уметь записывать сложные циклические алгоритмы (вложенные циклы) в виде программ. Понимать недопустимость включения в тело цикла операторов, семантически с циклом не связанных.

(5): Уметь организовывать эквивалентные циклы без использования специальных операторов. Владеть простейшими приемами отладки циклических программ.

26. Операторы ввода/вывода информации

(5п): Описать форматы всех операторов ввода (клавиатура, файл) и вывода (экран, файл) в изученном языке программирования и охарактеризовать порядок и специфику их использования для организации ввода/вывода. Уметь использовать операторы ввода/вывода для организации полноэкранный ввода данных (фиксированный кадр, контроль и верификация вводимых данных).

(3): Описать форматы простейших операторов, обеспечивающих ввод данных с клавиатуры и вывод символов

на экран. Продемонстрировать владение этими операторами для организации ввода/вывода значений переменных и массивов различных типов.

(4): Уметь использовать операторы ввода/вывода для организации полноэкранного ввода данных (фиксированный кадр, контроль, верификация вводимых данных).

(5): Описать форматы операторов, обеспечивающих ввод данных с дискового файла и вывод их в дисковый файл. Продемонстрировать владение этими операторами для организации ввода/вывода.

27. Стандартные функции.

Определение функций пользователя

(5п): Уметь объяснить значение функций для организации программирования по методу последовательной детализации. Знать правила определения функций пользователя и способ обращения к ним. Владеть понятиями фактических и формальных параметров; понимать характер связи между ними.

Перечислить стандартные (встроенные) функции в изучаемом языке программирования, описав их форматы, а также типы аргументов и значений.

Уметь определять функции пользователя и использовать их в выражениях.

(3): Знать стандартные функции в изучаемом языке программирования; уметь описать их форматы, типы аргументов, типы значений.

(4): Знать правила определения функций пользователя; уметь использовать их в выражениях.

(5): Владеть понятиями фактических и формальных параметров, локальных и глобальных переменных; понимать характер связи между ними. Уметь на примерах иллюстрировать механизм передачи параметров в программе.

28. Операторы графики

(5п): Дать характеристику имеющихся в изученном языке программирования средств управления экраном (описать форматы соответствующих операторов), а именно: управление режимом работы экрана, управление цветом; изображение точки, линии, прямоугольника, эллипса, дуги, окружности; заполнение области экрана; определение цвета пиксела, координат последнего участвовавшего в обработке пиксела; работа со спрайтами; «черепаховая» графика; язык графического вывода. Уметь использовать перечисленные возможности для построения изображений.

(3): Описать форматы операторов, обеспечивающих изображение на экране точки, линии, прямоугольника, эллипса, дуги, окружности и заполнение области экрана. Продемонстрировать умение пользоваться этими операторами.

(4): Описать форматы операторов, обеспечивающих управление режимом работы экрана, управление цветом, определение цвета пиксела и координат последнего участвовавшего в обработке пиксела. Продемонстрировать умение пользоваться этими операторами.

(5): Продемонстрировать умение работать со спрайтами; создавать движущиеся изображения.

29. Массивы (размерность, типы, заполнение)

(5п): Владеть понятием массива как совокупности однотипных данных. Понимать отличие линейных таблиц от прямоугольных с точки зрения организации их хранения в памяти и доступа в них к отдельным переменным. Понимать различие имени массива и имени переменной, в него входящей. Уметь объяснить понятие типа и размерности применительно к одномерным и двумерным массивам. Знать правила описания массивов в изученном языке программирования. Уметь составлять «протоколы» выполнения алгоритмов, содержащих действия с таблицами; использовать простейшие приемы отладки программ работы с таблицами.

(3): Владеть понятием массива как совокупности однотипных данных. Понимать отличие линейных таблиц

от прямоугольных с точки зрения организации их хранения в памяти и доступа в них к отдельным переменным. Различать индекс и значение элемента массива. Знать правила описания массивов в изучаемом языке программирования. Уметь организовать ввод значений двухмерной таблицы с клавиатуры.

(4): Владеть основными приемами присвоения значений элементам массива (при определении, путем организации циклов, путем ввода с клавиатуры, с использованием специальных операторов языка).

(5): Уметь составлять «протоколы» выполнения алгоритмов, содержащих действия с таблицами; использовать простейшие приемы отладки программ, работающих с таблицами.

30. Подпрограмма, обращение к ней. Возврат из подпрограммы

(5п): Владеть понятием подпрограммы как средства реализации вспомогательных алгоритмов. Понимать значение подпрограмм как средства реализации модульного программирования. Владеть понятием библиотеки стандартных подпрограмм.

Знать виды подпрограмм — процедуры и функции. Владеть правилами описания подпрограмм, обращения к подпрограммам. Знать различия между процедурами и функциями. Владеть понятиями формальных и фактических параметров; локальных и глобальных переменных. Знать способы передачи данных в/из подпрограммы. Понимать процесс исполнения программ, содержащих обращения к подпрограммам.

Уметь определять подпрограммы и использовать их для организации программных комплексов.

(3): Владеть понятием подпрограммы как средства реализации вспомогательных алгоритмов. Уметь определять подпрограммы и использовать их для организации простейших программ.

(4): Знать правила определения подпрограмм и обращения к подпрограммам. Владеть понятиями формальных и фактических параметров; локальных и глобальных переменных. Понимать процесс исполнения программ, содержащих обращения к подпрограммам.

(5): Знать способы передачи данных в/из подпрограммы. Уметь использовать подпрограммы для организации программных комплексов.

31. Задания

(5п): Продемонстрировать владение всеми звеньями технологической цепочки «постановка задачи — построение информационной модели — построение алгоритма — компьютерная реализация — получение и анализ результата». Задание выполнить методом последовательной детализации и в соответствии с принципами структурного программирования. Реализовать пользовательский программный интерфейс. Обеспечить контроль и возможность верификации вводимых данных с выдачей соответствующих сообщений. Обеспечить максимально возможную независимость программ от исходных данных. Самостоятельно осуществить подбор тестов.

(3): Продемонстрировать владение всеми звеньями технологической цепочки «постановка задачи — построение информационной модели — построение алгоритма — компьютерная реализация — получение и анализ результата».

(4): Задание выполнить в соответствии с принципами структурного программирования. Самостоятельно осуществить подбор тестов.

(5): Обеспечить контроль и возможность верификации вводимых данных с выдачей соответствующих сообщений.

Схема отчета о ходе и результатах экзамена

В отчете должны быть обозначены все позиции (если заполнить позицию невозможно, ставится ее номер и прочерк).

К отчету прилагается учебная программа аттестуемого класса.

1. Номер и точное наименование образовательного учреждения (ОУ)

2. Заявленный статус ОУ

3. Аттестуемый класс

4. Профиль класса (физ.-мат., техн., эконом. и т. п.)

5. Заявленный образовательный уровень класса (базовый, повышенный)

6. Характеристика КУВТ, имеющихся в ОУ

Для каждого КУВТ указываются:

6.1. Тип КУВТ

6.2. Количество машин

6.3. Сетевое программное обеспечение (ПО)

6.4. Операционная система

7. Тип модема (если есть)

8. Характеристика ПО, используемого для подготовки по информатике

8.1. Системы программирования (Ассемблер, Бейсик, Си, Паскаль, Фортран, Пролог, Лисп и т. п.)

8.2. Текстовые редакторы

8.3. Графические редакторы

8.4. Электронные таблицы

8.5. СУБД

8.6. Экспертные системы

8.7. Другие педагогические программные средства

9. Характеристика методической ориентации учителя

9.1. Учебники и учебные пособия

9.2. Задачники, сборники тестов

9.3. Методические пособия и рекомендации

9.4. Дополнительная литература

10. Соответствие предложенных экзаменационных материалов Российскому стандарту школьного образования по информатике (далее — РС)

10.1. Перечислить номера позиций РС, которые не отражены в экзаменационных материалах или отражены частично (в последнем случае указать, что отсутствует)

10.2. Указать номера позиций РС, избыточно представленных в экзаменационных материалах (обосновать)

11. Для каждого экзаменуемого:

11.1. Ф.И.О.

11.2. Номер билета

11.3. Для каждого вопроса в билете:

11.3.1. Текст вопроса (задачи)

11.3.2. Оценка ответа на вопрос (решение задачи)

11.3.3. Номер(а) рекомендуемых Министерством образования (МО) вопросов, послуживших основанием для оценки ответа на вопрос (решение задачи)

11.4. Общая оценка ГЭК

11.5. Экзаменационная оценка (за ответ по билету)

11.6. Предварительная оценка

12. Аналитическая справка (дать обоснованные ответы на следующие вопросы):

12.1. Соответствует ли предложенный экзаменационный материал требованиям РС и вопросам МО?

12.2. Соответствует ли методическая ориентация учителя требованиям РС?

12.3. Обеспечивает ли имеющаяся техническая база требования РС?

12.4. Обеспечивает ли имеющееся программное обеспечение (базовое, инструментальное, прикладное) требования РС?

12.5. В какой степени выявленный уровень подготовки учащихся соответствует заявленному образовательному уровню класса (п.5)?

Анализ вести не по экзаменационной оценке (п.11.5), а по оценке ГЭК (п. 11.3; 11.4):

- обосновать расхождение оценок в пунктах 11.4 и 11.5 (возможные причины: отсутствие необходимой техники, отсутствие необходимого ПО, неадекватная методическая ориентация учителя, смена учителя в середине года и пр. выявленные причины);

указать процент учащихся, удовлетворяющих заявленному образовательному уровню класса на 5, на 4, на 3 и не удовлетворяющих этому уровню (все по оценке ГЭК).

12.6. Соответствует ли выявленный уровень организации и результатов подготовки по информатике заявленному статусу (п.2) образовательного учреждения?

13. Дата проведения экзамена

14. Дата сдачи отчета

Подписи членов ГЭК

1. (И. О. Фамилия)

2. (И. О. Фамилия)

3. (И. О. Фамилия)

Приложение: Учебная программа класса

Уважаемые читатели!

Напоминаем вам, что с 1 апреля начинается подписка на второе полугодие 1995 г. Сведения, необходимые для подписки на журнал «Информатика и образование»:

	Индекс издания	Цена издания по каталогу в руб.	Периодичность
для индивидуальных подписчиков	70423	24 000 (за три номера)	1 раз в 2 месяца
для предприятий и организаций	73176	75 000 (за три номера)	

ОБРАЩАЕМ ВАШЕ ВНИМАНИЕ НА ТО, ЧТО НА ЖУРНАЛ
МОЖНО ПОДПИСАТЬСЯ ПО БЕЗНАЛИЧНОМУ РАСЧЕТУ

Телефон для справок: (095) 208-30-78

Факс: (095) 208-67-37

E-Mail: info@tit-bit.msk.su

РЕКОМЕНДАЦИИ

международной научно-практической конференции «Организация единого информационного образовательного пространства средствами телекоммуникаций» (г. Пермь, 6—9 декабря 1994 г.)

Конференция проводилась по плану Министерства образования Российской Федерации Главным управлением образования Пермской области, Институтом информатизации образования Минобразования РФ, Институтом средств обучения РАО, Пермским государственным педагогическим университетом, Пермским институтом повышения квалификации работников образования, журналом «Информатика и образование».

Основная цель конференции — обсуждение проблемы организации информационного обеспечения системы образования на региональном, республиканском, федеральном и международном уровнях. Обсуждались проблемы:

- состояния и перспектив развития телекоммуникаций в системе образования;
- организации единого международного информационного образовательного пространства;
- организации региональных, республиканских, федеральных и международных банков педагогической информации, инновационных данных системы образования;
- компьютерной (информационной) технологии обучения.

В конференции приняли участие 112 человек из 22 регионов России: Москвы и Московской области, г. Омска, Томска, Новосибирска, Самары, Костромы, Калуги, Ярославля, Нижнего Новгорода, всех регионов Урала, Якутии, других территорий. В конференции также участвовал представитель педагогического института Копенгагена (Дания).

Отличительная особенность данной конференции — совместное участие работников региональных органов управления образованием (от начальников главных управлений и ведущих специалистов до научных и научно-педагогических работников), что позволило совместно обсудить как организационные, так и научно-методические проблемы организации единого информационного образовательного пространства.

В ходе конференции было заслушано 12 пленарных и 30 секционных докладов, в которых поднимались практически все аспекты обсуждаемой проблемы. Некоторые доклады касались иных сторон процесса информатизации образования — методических, программно-технических и т. д.

Создание единого информационного образовательного пространства в России, в том числе на уровне регионов, — актуальная задача, начало реализации которой предусмотрено, в частности, Программой информатизации образования в Российской Федерации на 1994—1995 гг. Россия с ее просторами, разбросанностью образовательных и культурных учреждений по всей территории испытывает острый дефицит в информационном обеспечении по всем направлениям образовательной деятельности. Такое обеспечение, поставленное на современный технический и методологический уровень, будет способствовать:

1) широкой интеграции новых педагогических и информационных технологий в учебно-воспитательный процесс разных звеньев образования, типов учебных заведений;

2) переводу административного управления образовательных учреждений на безбумажные сетевые технологии;

3) обмену педагогическим опытом преподавательских кадров различных регионов, анализу и изучению зарубежного опыта;

4) формированию и развитию глобального мышления школьников и педагогов, отражающему социальные, экономические, нравственные потребности глубоко интегрированного человеческого общества.

Участники конференции ознакомились с опытом работы как в центре (институт информатизации Минобразования РФ), так и в ряде регионов: г. Перми, Москве и Московской области, Санкт-Петербурге, Омске, Башкортостане, Екатеринбурге, Калининграде, Томске, Новосибирске, Нижнем Новгороде и других.

Особое внимание конференции привлек опыт создания в Пермском регионе системы, обеспечивающей функционирование единого регионального информационного образовательного пространства, предусматривающей развитие компонентов данной системы на региональном, субрегиональном уровнях и уровне образовательного учреждения. Этот опыт заслуживает одобрения, изучения и заимствования его элементов другими регионами. Отмечается особая ценность создания единого информационного образовательного пространства в регионе в комплексе с другими направлениями информатизации образования: подготовкой и переподготовкой кадров, программно-методическим и научно-методическим обеспечением, техническим обеспечением и обслуживанием.

Однако, наряду с положительными моментами, конференция отмечает и существенные недостатки в организации и осуществлении работы. Таковыми, в частности, являются:

- отсутствие финансирования Федеральной программы информатизации образования крайне замедляет организаторскую работу, проводимую Минобразованием России;
- отсутствие единой республиканской политики в вопросах стратегии и развития системы телекоммуникаций в образовании, отсутствие федеральной образовательной телекоммуникационной сети вынуждает органы управления образованием, учреждения образования осуществлять подключение к коммерческим сетям, плата за услуги которых значительно превышает возможности образовательных учреждений и приводит порой к неоправданному расходованию средств, с одной стороны, а с другой – к потере возможностей информационного обмена между учреждениями в пределах республики или мира в целом;
- осуществляемая институтом информатизации образования деятельность по развитию сети TV-информ, предусматривающая передачу больших массивов информации из центра к периферии, имеет существенный недостаток – отсутствие обратной связи с абонентами сети, а следовательно, не решает всех проблем;
- в регионах стали создаваться свои региональные телекоммуникационные сети, не имеющие возможности стыковки их с мировыми сетями;
- проводимая по заданию Министерства образования научно-исследовательская работа в рамках ВНИКа на базе Рязанского государственного педагогического института по разработке моделей региональных и федеральных телекоммуникационных сетей в образовании ведется крайне медленно, региональные органы управления и научные подразделения регионов не информируются о ее результатах;
- медленно осуществляется перевод на современные компьютерные технологии центральных и региональных научно-педагогических библиотек, что снижает эффективность деятельности создаваемых образовательных телекоммуникационных сетей;
- не решены многие принципиальные вопросы информационного поиска в банках педагогической информации, что затрудняет (или делает невозможным) этот поиск для педагогов-практиков.

На основании изложенного конференция рекомендует

Министерству образования России:

- приложить максимум усилий для получения финансирования Федеральной программы информатизации образования;
- с целью объединения усилий по созданию Российской образовательной телекоммуникационной сети взять на себя координацию соответствующих работ, ведущихся в регионах, добиться участия в них в той или иной форме всех регионов;
- ходатайствовать перед правительством России о выделении единого канала связи на территории страны с льготной системой оплаты для нужд образовательной сети России;
- обратиться к Государственному комитету по высшему образованию РФ с просьбой распространить льготный доступ различного типа образовательных учреждений к

уже действующим сетевым ресурсам и банкам данных высшей школы, оказывать содействие в процессе подготовки работников образования к квалифицированному использованию сетевых информационных ресурсов;

региональным органам управления образованием:

- считать работы по созданию единого регионального информационно-образовательного пространства важным направлением деятельности на ближайший период, предусматривать выделение на это соответствующих ресурсов;
- вести соответствующие работы в тесной интеграции с соседними регионами и с Минобразованием РФ;
- предусмотреть организацию учебно-методических центров на уровне школ – медиатек – с целью развития информационной культуры учащихся, педагогов и администраторов, для организации консультативно-методической поддержки и обмена опытом учителей-практиков;
- создать межрегиональные центры по реализации телекоммуникационных проектов и банков педагогической информации. Эти центры могут взять на себя функции экспертных советов, организацию повышения квалификации всех работников образования, другие функции (по усмотрению соорганизаторов). В частности, в Уральском регионе такой центр мог бы использовать ресурсы лаборатории информатизации образования Пермской области;

Институту средств обучения РАО:

- разработать концепцию дистанционного обучения на основе компьютерных телекоммуникаций;
- организовать комплексные исследования и проектные разработки курсов дистанционного обучения по различным направлениям;
- разработать методы эффективного использования телекоммуникаций в школьной практике (электронной почты, телеконференций, удаленных источников информации и т. д.);

редакции журнала «Информатика и образование»:

- регулярно пропагандировать на страницах журнала необходимость создания единого информационного образовательного пространства, освещать соответствующие научно-методические проблемы;
- опубликовать (выборочно) материалы конференции и данные рекомендации в ближайшем номере.

Участники конференции благодарят Главное управление образования Пермской области, Лабораторию информатизации образования, оргкомитет за высокий уровень организации конференции.

В Министерстве образования Российской Федерации

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ИНФОРМАЦИОННОЕ ПИСЬМО

04.01.95

Москва

№ 5/11

В настоящее время в школах при органах управления образованием наряду с созданием студий замкнутого учебного телевидения, выделением учебного телевизионного канала местного телевидения все активнее поднимается вопрос о расширении комплекса технических средств и средств информации, создании центров, представляющих многокомпонентную информационно-педагогическую среду.

Для образовательных учреждений региона подобного рода медиacentры могут стать основой интеграции баз данных по учебной и учебно-методической литературе, средствам обучения, методической, культурно-эстетической и профессионально-ориентационной работе, а также для разнообразной внеклассной творческой деятельности учащихся и учителей. Центр может оказать помощь родителям и их детям при новых формах получения образования — семейном образовании и экстернате. Медиациентр по своей сути охватывает различные варианты применения средств новых информационных технологий и традиционных средств обучения для обеспечения разнообразных образовательных услуг.

Медиациентры могут создаваться и в школах как самостоятельное структурное подразделение «медиаотека». Как правило, они строятся на базе библиотек, кабинетов технических средств обучения или школьных телевизионных студий.

Установка пульта кабельного телевидения в медиатеке способствует активному включению разнообразных видеопрограмм в классноурочную деятельность. Такого рода студии в школах России начали развиваться еще в 80-е гг. Появление в настоящее время высококачественной видеотехники, цветных видеокамер позволило в значительной степени разнообразить и интенсифицировать учебный процесс.

Так, в школах г. Самары накоплен интересный опыт создания четырехканальных студий учебно-информационного кабельного телевидения, разработанного производственно-технической фирмой КВН и Поволжским институтом информатики, телевидения и радиовещания. Разработчики ставили своей целью перевод и замену всех ранее применявшихся аудиовизуальных средств обучения учебными видеозаписями, предоставление возможности средствами видеостудии (видеоперсональный комплекс) не только накопления банка аудиовизуальной информации, но и создания новых видеопрограмм с использованием старого видеоряда.

Студии предоставляют возможность одновременного транслирования 4 разных учебных программ. Кроме того, имеется возможность приема и трансляции 2—3 программ местного и республиканского телевидения. Созданные студии укомплектованы обширным банком видеoinформации (более 250 видеокассет).

Опыт работы школ г. Самары показывает, что создание школьных медиатек на базе подобных студий с их широчайшими телекоммуникативными возможностями значительно повышает эффективность творческой работы учителя и учащегося.

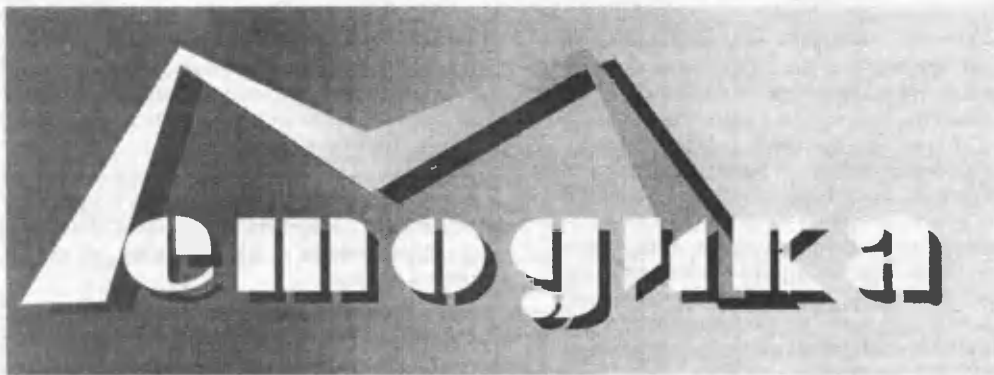
Исходя из опыта работы территорий России (Москва, Санкт-Петербург, г. Тула, Алтайский край, Ульяновская обл. и др.) по организации новых или преобразовании существующих структурных подразделений: в системе образования — в медиациентры, а в школах — в медиатеки. Министерство образования Российской Федерации разработало проект Примерного положения о школьной медиатеке.

В помощь работникам образования издательством «Информатика и образование» выпущено методическое пособие «Медиатека. Как создать в школе медиатеку» (авт. Е. Н. Ястребцева), где рассмотрены вопросы актуальности создания медиатек, их примерного положения, требования к материально-технической базе.

Министерство образования рекомендует руководителям органов управления образованием оказать содействие в создании и развитии медиациентров в своих регионах.

Заместитель Министра

А. Г. Асмолов



И. Г. Семакин,

*кандидат физико-математических наук, Пермский государственный университет,
Лаборатория информатизации образования Пермской области*

БАЗОВЫЙ КУРС ОИВТ: «ПЕРМСКАЯ ВЕРСИЯ» (часть 5)*

Тема 8. Компьютер дает справку

Информационная модель – что это?

Изучая темы «Компьютер пишет» и «Компьютер рисует», ученики познакомились с использованием ЭВМ в качестве инструмента для создания текстов и рисунков. Эти системы заменяют человеку пишущую машинку, ножницы, клей, кисточки, краски, линейки и пр. Иначе говоря, компьютер выполняет роль универсального инструмента для соответствующей работы.

Дальше речь пойдет о применении ЭВМ несколько в другом плане. Хотя определенная инструментальная роль останется, но изучаемые в дальнейшем системы будут иметь качественно новое назначение: они предназначены для создания информационных моделей.

Информационная модель — одно из узловых понятий в информатике. Раскрыть его перед учениками — важная и непростая задача.

Слово «модель» для большинства детей знакомо. Они сами приведут примеры: модель самолета, корабля, автомобиля и пр. Учитель к этому может добавить: модель плотины электростанции, модель застройки города, модель движения транспорта и многое другое. Понятно, что во всех этих примерах речь идет о материальных (физических) моделях реаль-

ных объектов. Такие модели создаются конструкторами, архитекторами, инженерами, исследователями. «Моделью называется некий объект-заменитель, который в определенных условиях может заменять объект-оригинал, воспроизводя интересующие нас свойства и характеристики оригинала, причем имеет существенные преимущества, удобства (наглядность, обозримость, доступность испытаний, легкость оперирования с ними и пр.)» [1]. Иначе говоря, модель — это некоторое упрощенное подобие реального объекта.

Информационная модель — это информация (т. е. знания, сведения) о реальном объекте, процессе, явлении.

Представьте себе, что вам подробно описали внешность человека, которого вы никогда не видели. Затем по описанию вы узнали этого человека. Стало быть, в вашем сознании сложился некоторый образ человека в результате сообщенной информации, т. е. создалась информационная модель. В чем-то она не совпала с реальностью, но если основные признаки совпали, то этого хватило для того, чтобы узнать человека.

Большинство знаний, которые ученики получают на уроках, носят характер информационных моделей. На физике ученики узнают о борновской модели атома, не имея возможности разглядеть реальный атом; знакомятся с описанием Солнечной системы, моле-

* Продолжение. Начало см.: Информатика и образование. 1994. № 5, 6; 1995. № 1.

кулярной структурой вещества, схемой кровеносной системы человека и многим другим.

Следовательно, наши знания о реальном мире — это множество информационных моделей. И часто наши успехи или неудачи зависят от того, насколько эти модели адекватны (т. е. соответствуют) реальности.

Информация, отражающая существенные признаки объекта, процесса или явления и хранящаяся в памяти ЭВМ, представляет собой компьютерную информационную модель. При построении такой модели приходится решать две проблемы:

- какие признаки считать существенными;
- как их организовать в памяти ЭВМ (какой тип и структура информации).

Информация и структуры данных [2,3]

Изучение вопроса о структурах данных можно начать с разговора о структурах в материальном мире. Большинству людей интуитивно понятно, что структура — это что-то упорядоченное, определенным образом организованное. Говорят: «структура молекулы», «структура института», «структура государства».

Задайте ученикам вопрос: чем отличается куча кирпичей от стены, построенной из такого же кирпича?

Очевидно, ответ будет такой: куча кирпича не имеет структуры. Если кирпичи в ней

перемешать, то она все равно останется кучей. Стена же — это структура, построенная из кирпичей. Если ее нарушить, то останутся кирпичи, но не будет стены!

Структура несет новое содержание, новое качество, новую информацию! Из тех же кирпичей можно построить гараж, дом, фонтан и пр. Составляющие — одни и те же, но содержание — разное.

Очень часто целью научного поиска является определение структуры исследуемого объекта. Например, к началу XX в. было известно, что атомы вещества состоят из частиц с положительными и отрицательными электрическими зарядами. Но только после экспериментов Э. Резерфорда в 1911 г. и теоретического открытия Н. Бора стала понятна структура атома (боровская модель атома). В результате нашли объяснение многие экспериментальные факты.

А теперь рассмотрим пример с информацией. Запишите на доске следующее: 1, 3, 5; ТУ-154; Воронеж; Москва; 8-40; АН-24; 4,7; Ижевск; 16-20; ТУ-134 и т. д. Этот список представляет собой набор слов, цифр, мало о чем говорящий. Конечно, можно догадаться, что речь идет о рейсах самолетов, но извлечь какую-то полезную информацию из этой «кучи данных» нельзя.

А теперь покажите ученикам табл. 1.

Т а б л и ц а 1

Аэропорт назначения	Номер рейса	Тип самолета	Дни отправления	Время отправления
Москва	320	ТУ-154	1, 3, 5	16-20
Ижевск	308	АН-24	4, 7	17-35
Воронеж	3107	ТУ-134	1, 3, 5, 7	8-40

Всем все стало ясно! Это расписание вылета самолетов. В таком виде эта информация становится жизненно необходимой для пассажира Аэрофлота.

Расписание — пример информационной модели, представленной в табличном виде. Объектами, отраженными в этой модели, являются рейсы самолетов. В расписании указаны только те данные, которые интересуют пассажиров (вопрос о существенных признаках). Сведения о составе экипажа, техническом состоянии самолета и другие сюда не включены. Эти сведения входят в информацию, которой пользуется, например, начальник аэропорта.

Все данные, входящие в расписание, взаимосвязаны. Если эту связь исказить, то информация станет ложной, что может привести к печальным последствиям.

В информатике совокупность взаимосвя-

занных данных, называется информационной структурой, или структурой данных. В этом примере структура представляет собой прямоугольную таблицу, состоящую из строк и столбцов. Объектами, описанными в этой таблице, являются рейсы самолетов. Свойства (признаки) этих объектов выделяются в отдельные столбцы и озаглавлены («Аэропорт назначения», «Номер рейса», «Тип самолета» и пр.). Их называют также атрибутами объектов. Каждая строка этой таблицы есть совокупность значений атрибутов, относящихся к конкретному объекту. Такую строку называют записью (другое название — кортеж), а столбцы — полями записи.

Прямоугольная таблица — одно из возможных представлений структуры данных. На схеме 1 представлен другой вид структурированной информации. В этой схеме содержится информация о структуре института:

Схема 1



составляющих его факультетах и кафедрах. Такая структура в информатике называется деревом.

Дерево на схеме содержит три типа объектов: институт, факультет, кафедра. Каждый из этих объектов также описывается своими атрибутами, например: институт (название, адрес, ректор); факультет (название, количество студентов, декан); кафедра (название, количество преподавателей, зав. кафедрой).

Объекты в дереве разделены на уровни. На первом уровне находится единственный объект, называемый корнем дерева. На втором — уже три объекта, порожденных корнем (три ветви). В свою очередь, каждый объект второго уровня порождает несколько объектов третьего уровня. Объекты самого нижнего уровня называются листьями.

В форме дерева описываются системы объектов, имеющие иерархическую структуру. Для таких структур характерна подчиненность объектов нижнего уровня объектам верхнего уровня. В дереве соотношения между верхними и нижними объектами имеет характер «один ко многим». Хотя в атрибутах, описывающих кафедру, нет названия факультета. Но, глядя на дерево по линиям связи, мы видим, какая кафедра какому факультету принадлежит.

Аналогично через линии связей в дереве отражена принадлежность факультетов к институту.

Третья разновидность структуры данных называется сетью. Пример сети виден наглядно в схеме 2.

Эта схема иллюстрирует то, на каких факультативах занимаются какие ученики. Здесь тоже есть два уровня взаимосвязанных объектов, но соотношение между ними — «многие ко многим». Такое соотношение характерно для сетевых структур.

Структурная организация данных об объектах позволяет получить дополнительную информацию, помимо той, что непосредственно указана в атрибутах. Например, из табл. 1 можно получить ответы на следующие вопросы (предположим, что таблица содержит сведения о всех рейсах):

- летают ли из данного аэропорта самолеты в Ташкент;
- сколько рейсов обслуживают самолеты ТУ-154;
- есть ли вылеты из аэропорта в интервале от 12 до 13 часов?

Из иерархической структуры на схеме 1 (с учетом атрибутов каждого объекта) можно узнать:

Схема 2



- сколько всего студентов учится в институте;
- на каком факультете наибольшее число преподавателей;
- на каком факультете работает деканом И. И. Иванов;
- есть ли на биологическом и физическом факультетах заведующие кафедрами — однофамильцы.

Пусть в сетевой структуре на схеме 2 данные об учениках и факультативах состоят из следующих атрибутов:

ученик — фамилия, имя, класс;
факультатив — название, преподаватель, кабинет, день занятий, время.

Тогда из этой схемы можно получать следующую информацию (ответы на вопросы):

- сколько учеников занимаются на каждом факультативе;
- сколько факультативов посещает каждый ученик;
- сколько учеников в 18-00 находится в школе на занятиях различных факультативов;
- посещают ли ученики VIII Б класса факультатив преподавателя В. П. Рябова.

Из рассмотренных примеров мы видим, что важно не только собрать вместе нужную информацию, но и удачно структурировать ее. Самым ярким примером удачной информационной структуры в науке является таблица Д. И. Менделеева. Она показывает, как в структуре данных могут отражаться законы природы.

Всякая информационная структура создается для дальнейшего извлечения из нее нужной информации. Поэтому построение структуры данных производится в следующей методической последовательности:

- 1) определяются объекты описания;
- 2) определяются атрибуты этих объектов;
- 3) выбирается тип структуры, отображающий связи между объектами (таблицы, деревья, сети);
- 4) строится конкретный экземпляр информационной структуры.

Реляционные базы данных

Информацию, структурированную определенным образом, можно хранить в журналах, книгах, на плакатах, стендах и пр., т. е. в письменном виде. В докомпьютерный период только так и было. С появлением ЭВМ возникла возможность хранения информации в компьютерной памяти. Очевидно, для этих целей применяется внешняя магнитная память ЭВМ, так как информацию нужно хранить долго и часто большого объема. Преимущества такого способа хранения очевидны: нет расхода бумаги; программным путем можно организовать изменения и дополнения информации; с помощью специальных программ поиска компьютер отыщет нужные сведения гораздо быстрее, чем это можно сделать, перелистывая страницы.

Информационные структуры, содержащие взаимосвязанные данные о реальных объектах и хранящиеся во внешней памяти ЭВМ, называются базами данных (БД).

Прямоугольные таблицы наиболее часто встречающийся на практике тип информационных структур. Можно привести массу примеров: дневник ученика, классный журнал, расписание поездов, телефонная книга, таблица футбольного чемпионата и многое другое.

Базы данных, содержащие информацию в виде прямоугольных таблиц, называются реляционными. Это слово происходит от английского relation — отношение. В этом подходе таблица как совокупность атрибутов объекта называется отношением.

Таблицы хранятся на магнитном диске в виде файлов. Если вся база данных представляет собой одну таблицу, то она образует один файл. Чаще всего реляционная база данных — это множество таблиц, и поэтому на диске — это множество файлов. Различные таблицы связаны между собой через общие атрибуты.

Пусть, например, в базе данных учреждения хранятся две следующие таблицы (табл. 2, 3).

Таблица 2

Ф. И. О.	Адрес	Отдел	Должность	Оклад
Абрамов А. В.	Мира, 30-45	310	программист	400
Савин Р. Н.	Геологов, 5-13	315	инженер	550
Ярцев К. С.	Крылова, 26-46	302	инженер	600
Березников И. П.	Геологов, 13-24	310	бухгалтер	450

Таблица 3

Отдел	Комната	Начальник
302	124	Иванов И. И.
310	250	Петров П. П.
315	222	Козлов К. К.

Эти две таблицы связаны между собой общим полем «Отдел». Через эту связь можно, например, узнать:

- в какой комнате работает А. В. Абрамов;
- сколько инженеров трудится в отделе, руководимом П. П. Петровым;
- работают ли в комнате 222 программисты и пр.

В терминологии реляционных БД табл. 2 и 3 представляют собой экземпляры двух различных отношений, составляющих базу данных. Каждое из этих отношений должно иметь свое имя. Пусть табл. 2 есть отношение с названием СОТРУДНИК, а табл. 3 — отношение ОТДЕЛ. Тогда структура БД, содержащая два типа отношений, опишется так:

СОТРУДНИК (Ф. И. О., адрес, номер отдела, должность, оклад)

ОТДЕЛ (номер отдела, комната, начальник).

Подчеркнутые поля называются ключевыми. Ключ должен удовлетворять требованию однозначной идентификации записи (строчки в таблице). В первом отношении таким полем является Ф. И. О. сотрудника, во втором — номер отдела.

Реляционные базы данных обладают целым рядом преимуществ. Во-первых, это наглядность и понятность для пользователя. Именно с табличной формой структур данных мы чаще всего сталкиваемся на практике. Во-вторых, к реляционной форме можно свести любой тип структуры данных (деревья, сети). Например, в рассмотренном выше примере если добавить в качестве корневого элемента отношение

УЧРЕЖДЕНИЕ (название, адрес, директор),

то будет отражена трехуровневая иерархическая структура:

УЧРЕЖДЕНИЕ—ОТДЕЛ—СОТРУДНИК.

В-третьих, для получения ответов на различные запросы к базе данных существует разработанный математический аппарат, который называется исчислением отношений или реляционной алгеброй. Ответы на запросы получаются путем «разрезания» и «склеивания» таблиц по строкам и столбцам. При этом ясно, что ответы будут иметь также форму таблиц. Например, используя табл. 2, требуется получить список фамилий и должностей всех сотрудников отдела 310. Алгоритм исполнения такого запроса в терминах «вырезать», «склеить» будет следующим:

- вырезать все строки, где отдел = 310;
- склеить полученные строки в таблицу;
- вырезать в полученной таблице столбцы Ф. И. О. и должность;
- склеить полученные столбцы в таблицу.

БД — это, собственно, хранилище информации. Но для чего хранится информация? Очевидно, для того чтобы извлекать из нее по мере необходимости нужные сведения. Иначе говоря — осуществлять поиск нужной информации. Такой поиск в БД производится с помощью специальных программ, обслуживающих запросы пользователя. Кроме того, в распоряжении пользователя имеются программы, позволяющие модифицировать БД (удалить устаревшую информацию, добавить новую информацию, изменить значение отдельных полей и т. п.).

Совокупность базы данных и обслуживающих программ называется и н ф о р м а ц и о н н о - п о и с к о в о й с и с т е м о й (ИПС). ИПС обычно имеет конкретное предметное назначение, например: ИПС книжного фонда публичной библиотеки; ИПС кадровой информации крупного предприятия и т. п.

И еще одно общее понятие: система управления базами данных (СУБД). СУБД — программное обеспечение, с помощью которого строится база данных, на основе которого создаются и функционируют ИПС. СУБД обычно ориентируются на один из типов структур данных: деревья (иерархические СУБД), сети (сетевые СУБД), отношение (реляционные СУБД). На современных персональных компьютерах наибольшее распространение получили реляционные СУБД.

А п п а р а т н ы е с р е д с т в а баз данных. Базы данных бывают самого разного назначения и масштаба: от БД индивидуального пользования на ПК до многопользовательских банков данных, работающих в глобальных сетях. Соответственно, различаются и технические средства, применяемые в этих случаях.

При работе с БД индивидуального пользования задействованы все аппаратные средства ПК. Особая роль здесь принадлежит устройствам внешней памяти. От их информационной емкости, скорости и надежности работы зависят размеры БД и качество создаваемых на их основе ИПС.

На современных ПК используются различные типы дисковых накопителей.

Накопители на гибких магнитных дисках (флоппи-диски). Носителем информации является диск на синтетической основе, покрытый слоем ферромагнетика. Широко используются два стандарта гибких дисков: диаметром 5,25 дюйма и диаметром 3,5 дюйма. Объем информации, помещающейся на таких дисках, — от 360 Кбайт до 2 Мбайт. Гибкие диски — сменные. Поэтому на них можно хранить практически неограниченное количество информации (при большом количест-

ве дисков). Гибкие диски на ПК используются для долговременного хранения данных, накопленных в БД.

Накопители на жестких дисках (винчестерские диски). Это несколько магнитных дисков на алюминиевой основе, надетых на общую ось (пакет дисков). Этот тип дисков используется в профессиональных ПК. В отличие от флоппи-дисков, винчестерские диски — несменные устройства, встроенные в машину. Пользователь доступа к ним не имеет. Емкость и быстрдействие этих дисков по сравнению с гибкими существенно выше, объемы их измеряются десятками, сотнями и даже тысячами мегабайт.

Кассетные пакеты дисков. Это сравнительно новый тип дисковых устройств, объединяющих в себе достоинства винчестерских и гибких дисков: с одной стороны — большой объем и быстрдействие, с другой — сменность.

Оптические (лазерные) диски. Это тип дисков, на котором достигается самая высокая плотность хранения информации (более 10^6 бит/см²). В устройстве памяти на лазерном диске отсутствуют какие-либо движущиеся механические детали, поэтому их надежность существенно выше других.

Оптический диск — сменное устройство. Однако на нем возможна только однократная запись, поэтому на оптических носителях хранятся данные, которые не меняются со временем.

Работа с реляционной СУБД

В отличие от других средств КИТ (текстовые и графические редакторы, электронные таблицы), различные системы управления базами данных не имеют унифицированной среды работы пользователя. Поэтому единообразно описать работу с различными СУБД весьма затруднительно. И все же, безусловно, в работе с различными реляционными базами данных есть сходство.

Можно определять типичные режимы работы с БД. Это:

- 1) режим создания БД;
- 2) режим редактирования БД;
- 3) режим манипулирования БД;
- 4) режим поиска в БД.

Для работы в каждом режиме существует своя система команд. Поэтому всякая работа пользователя с БД строится в форме алгоритма, составленного из этих команд.

Такие алгоритмы могут выполняться в режиме прямого исполнения (отдается команда и сразу выполняется) и в режиме автоматического исполнения, т. е. в программном режиме. Не все СУБД реализуют автоматиче-

ский режим, некоторые могут работать только в режиме прямого исполнения команд.

Одна из наиболее популярных сегодня СУБД называется dBASE (читается: «ди бэйс») [4]. Она создана для персональных ЭВМ IBM PC. Кроме того, версия dBASE имеется на школьном КУВТ «Корвет». С dBASE можно работать как в прямом, так и в программном режиме выполнения команд.

А теперь вернемся к режимам работы с БД. Примеры команд будем приводить из СУБД dBASE. Система приглашает пользователя к диалогу выводом на экран символа «.»(точка). После точки пользователь с клавиатуры набирает команду. Как правило, ввод различных параметров, необходимых для выполнения команды, происходит в диалоге, т. е. СУБД запрашивает параметры, пользователь вводит их значения с клавиатуры.

Итак, режим создания БД. Работа в этом режиме происходит в такой последовательности:

1) отдается команда создать БД (.CREATE);

2) на запрос системы пользователь сообщает имя базы данных (FILE NAME: primer);

3) запрашивается общее количество полей (столбцов) записи. Затем по соответствующему запросу вводятся характеристики полей БД. К характеристикам полей относятся:

- имя поля;
- тип поля;
- ширина (число символов);
- формат числовых данных;

4) после определения структуры записей производится ввод информации в БД. Ввод также происходит в диалоге — значение каждого поля, каждой записи запрашивается отдельно.

Кроме понятия «структура данных» в этой теме ученики впервые узнают о понятии «т и п ы д а н н ы х». Для каждого поля (столбца) таблицы должен быть определен тип. Понятие типа связано с тремя свойствами данных:

- множеством возможных значений;
- формой представления в памяти ЭВМ;
- допустимыми действиями над данными.

Различаются символьные и числовые типы данных. Поле может быть символьной строкой. Способы кодирования символов в памяти ЭВМ уже рассматривались. При описании полей для символьных строк указывается максимальная длина.

Символьная строка — это любая последовательность любых печатаемых символов из алфавита ЭВМ. Над символьными строками можно выполнять следующие действия:

- соединять несколько строк в одну;
- выделять (вырезать) подстроку;

- удалять или вставлять символы;
- сравнивать две строки на совпадение или несовпадение и др.

Числовые данные представляются в памяти ЭВМ в двоичной системе счисления, однако вводятся и выводятся в десятичной форме. Различаются целые и вещественные (десятичные) числа. Обычно для вещественных чисел указывается количество цифр, сохраняемых после запятой (в дробной части). С числовыми данными можно выполнять математические операции, а также операции сравнения на =, <, <, >, <=, >=.

В большинстве СУБД используются поля логического типа. Величины логического типа принимают всего два значения: «да» (истина), «нет» (ложь). Для кодирования такой величины в памяти ЭВМ достаточно одного бита.

Режим редактирования БД. Под редактированием понимается внесение любых изменений в уже созданную базу данных. Типичные операции редактирования:

- добавление новых записей в базу (в конец или вставка внутрь) (APPEND, INSERT);
- удаление записей из базы (DELETE);
- редактирование значений полей записей (EDIT, CHANGE, BROWSE);

Под манипулированием базой данных будем понимать некоторые действия, выполняемые с БД в целом. К этой группе можно отнести следующие команды:

- просмотр всей базы на экране (LIST);
- копирование файла БД в новый файл (COPY);
- сортировка записей БД по значениям некоторого поля (столбца) и создание файла с отсортированными записями (SORT);
- активизация файла БД для обработки всех видов (USE) и др.

База данных становится основой информационно-поисковой системы (ИПС), если СУБД обеспечивает режим поиска. Очевидно, любая СУБД должна поддерживать такой режим. Поиск — это выбор из базы данных записей, удовлетворяющих заданным условиям поиска. Поля, по значениям которых осуществляется поиск, называются ключами поиска.

Условие поиска представляет собой логическое выражение, в котором ключи сравниваются с некоторыми значениями. Результатом вычисления такого выражения является логическая величина «истина» (true), «ложь» (false).

Сравнения на равенство, неравенство, больше, меньше представляют собой простые условия. Вот примеры таких условий:

должность = «инженер»
отдел <> 310
оклад < 500
возраст >= 45.

Слева от знака сравнения пишутся имена полей, справа — значения, с которыми производится сравнение. Подобные выражения отражают условия поиска по одному ключу. Если требуется осуществить поиск по нескольким ключам или на значение одного ключа накладывается несколько условий, то условие поиска будет представлять собой сложное (комбинированное) логическое выражение. Например, нужно выбрать всех людей в возрасте от 35 до 45 лет; или получить сведения о всех инженерах и программистах из отдела 301; или получить сведения о всех сотрудниках, кроме лаборантов.

Для формулировки сложных логических выражений используются логические операции. Их три: отрицание (не — not); логическое умножение (и — and); логическое сложение (или — or). Например, сформулированные выше сложные условия могут быть записаны в форме следующих логических выражений:

возраст > 30 и возраст <= 45
должность = «инженер» или должность = «программист» и
отдел = 301 не (должность = «лаборант»).

Операция отрицания (не) применяется к одному логическому операнду (отношению или логической величине) и изменяет его значение на противоположное. Операции «и», «или» действуют между двумя операндами. Операция логического умножения («и») дает в результате значение «истина», если оба операнда есть «истина». Во всех других случаях в результате получается «ложь». Операция логического сложения («или») дает «истина», если хотя бы один операнд равен «истина».

Табл. 4 называется таблицей истинности. Она показывает результаты всех вариантов выполнения логических операций. В табл. 4 использованы следующие обозначения: И — истина; Л — ложь.

Таблица 4

А	В	не А	А и В	А или В
И	И	Л	И	И
И	Л	Л	Л	И
Л	И	И	Л	И
Л	Л	И	Л	Л

Результатом поиска в БД может быть выделение в таблице записей, удовлетворяющих условию поиска. Это выделение происходит

путем установки указателя на соответствующую запись или изменением цвета этой записи. В dBASE такое действие выполняется по команде LOCATE <условие поиска>. Например:

LOCATE FOR оклад > 500.

Указатель устанавливается на первую запись, удовлетворяющую условию поиска. А затем с помощью специальной команды (в dBASE — CONTINUE) перемещается к следующей аналогичной записи. Так можно продолжать до конца.

Результатом поиска может быть получение отдельной таблицы, в которой собраны все строки, удовлетворяющие условию. Например, в dBASE в результате выполнения следующей команды:

LIST FOR оклад > 500

выведутся на экран все записи о сотрудниках, имеющих оклад больше 500. В таблицу с результатами поиска можно включать не все поля, а лишь некоторые указанные. Например, по команде

LIST ФИО, должность, FOR оклад > 500

будет выдана таблица с фамилиями и должностями сотрудников, имеющих оклад больше 500.

Повторим еще раз, что можно делать с базой данных:

- создавать БД;
- дополнять, редактировать, манипулировать;
- осуществлять поиск в БД;
- формировать новые таблицы из выбираемых записей по заданным условиям.

Некоторые СУБД позволяют проводить определенные виды обработки значений полей записей. Например, вычисление сумм или средних значений для числовых полей.

Немного о языках

В данной теме достаточно отчетливо должно прозвучать еще одно понятие: язык общения с ЭВМ.

Снова вернемся к опорной точке всего курса — к теме «Человек и информация». Там было дано следующее определение: «Язык — это знаковая система представления информации, это средство обмена информацией».

Общение человека с ЭВМ происходит, безусловно, в знаковой форме и, значит, на определенном языке. Собственный язык машины — это язык двоичных кодов. Однако человеку пользоваться такой формой языка неудобно. Поэтому с помощью средств программного обеспечения повышается уровень

языка общения с ЭВМ; уровень языка тем выше, чем он ближе к человеческому.

Каждое средство компьютерных информационных технологий имеет свой язык для взаимодействия с человеком.

В рассмотренных нами текстовых и графических редакторах также используются определенные языки общения. Поскольку они, как правило, реализованы через меню команд, то такой способ общения можно назвать языком меню. Язык меню — это совокупность слов-команд, предоставляемых пользователю на выбор. Чаще всего этот выбор производится либо установкой указателя на соответствующее слово, либо нажатием определенной клавиши (функциональной или буквенной).

При работе с СУБД или готовой ИПС также приходится иметь дело с языком общения с компьютером. Но такое общение усложняется тем, что оно выходит за рамки языка меню. Пользователь сам, по определенным правилам описывает и вводит исходные данные, формирует команды обработки данных. Чтобы это делать правильно, надо знать правила языка. Правила языка сообщаются пользователю в форме описания языка.

В данной и в последующих темах курса учителю придется давать ученикам описание различных языков работы с разными видами программного обеспечения.

Всякий язык определяется тремя составляющими: алфавитом, синтаксисом и семантикой.

Алфавит (иногда говорят — словарь) — это набор всех символов, а также служебных слов (стандартных символьных последовательностей), используемых в языке.

Синтаксис определяет правила построения предложений языка из символов алфавита. Иначе говоря, синтаксис — это правописание на данном языке.

Семантика — это система правил истолкований предложений данного языка. Иначе говоря, семантика — это смысловое содержание языковых конструкций.

Предложения или конструкции на любом языке описывают либо данные (обрабатываемую информацию), либо действия над данными, т. е. команды из СКИ соответствующего исполнителя.

В настоящем разделе курса мы главным образом говорили о смысловом (семантическом) содержании данных и команд при работе с БД. В этом смысле разные СУБД имеют много общего. Здесь и далее мы будем стремиться показать прежде всего семантическую общность языков различных версий однотипного программного обеспечения.

Разрабатывая урок, учитель ориентирует-

ся на конкретное программное обеспечение своего компьютерного класса. Поэтому наряду с семантикой языковых конструкций необходимо четко определять их синтаксис. Необходимо определить алфавит языка, за рамки которого ни в коем случае нельзя выходить. Для этого можно пользоваться строгими приемами метаязыков (БНФ, синтаксические диаграммы), но вполне можно обойтись и не столь формальными средствами.

Организация практических занятий

по данной теме происходит с использованием конкретной СУБД или готовой ИПС на имеющемся КУВТ. Возможности программных средств определяют содержание занятий. В любом случае рекомендуется организовать практику в таком порядке:

- работа с готовой БД; отработка приемов манипулирования и поиска в БД;
- проектирование БД по конкретной теме с целью создания справочника;
- создание и заполнение БД на компьютере;
- отработка алгоритмов поиска для получения справок по типичным запросам.

Следовательно, в рамках темы «Компьютер дает справку» решаются два типа информационных задач [5]:

1) получение информации из имеющейся БД;

2) создание новой БД и поддержка ее в определенном состоянии.

Решение задач 1-го типа происходит в таком порядке:

- формулируется запрос

(кто из инженеров предприятия получает оклад больше 500 рублей?);

- определяются исходные данные и результаты в терминах полей БД

(Дано: оклад > 500, должность — «инженер»; Найти: Ф. И. О., отдел);

- строится план (алгоритм) поиска (если допускаются логические операции, то поиск осуществляется по условию: оклад > 500 и должность — «инженер»; если логических операций нет, то сначала ищутся все инженеры, а затем среди них те, у кого оклад больше 500. Из полученных записей выбираются только поля Ф.И.О., отдел и выводятся на экран);
- составляется соответствующая команда или последовательность команд, реализующих данный алгоритм на языке СУБД;
- устанавливается режим поиска, вводится и выполняется команда, получают результаты;

- результаты интерпретируются.

Методическая последовательность решения задач второго типа:

- анализируются информационные потребности пользователя к будущей БД, т. е. продумываются все возможные типы запросов;
- на основе результатов предыдущего этапа строится структура БД; определяются имена полей, типы и форматы исходных данных;
- структура БД описывается средствами описания данных СУБД;
- описание структуры вводится в ЭВМ и происходит машинное проектирование БД;
- производится заполнение БД информацией.

СУБД, рекомендуемые для использования на уроках. Наиболее развитые профессиональные СУБД имеются на IBM PC. Существует множество реляционных СУБД для этого класса компьютеров. Одна из наиболее распространенных — СУБД dBASE. Она имеет множество версий. Последняя версия — dBASE-IV. В отличие от предыдущих dBASE-IV, в частности, имеет удобную диалоговую оболочку с системой меню. Русифицированные аналоги dBASE — СУБД «Карат», «Ребус».

На КУВТ «Корвет» реализована СУБД dBASE-2. Систему поставляют НТП «Альтернатива». Подробное описание прилагается в текстовых файлах.

На КУВТ УКНЦ имеется реляционная СУБД (РБД). К ней также прилагается документация в текстовых файлах. К сожалению, данная СУБД не работает в многопользовательском режиме в сети. Но она достаточно хорошо работает в монопольном режиме.

Для КУВТ УКНЦ существует имитатор реляционной СУБД (ИРБД), разработанный преподавателем кафедры информатики ПГПУ Г. П. Ереминой. ИРБД работает с оперативной памятью в режимах создания, манипулирования и поиска. Поэтому с помощью данной системы можно создавать только маленькие базы данных (до 50 строк). Но зато не возникает проблем с коллективной работой в классе. Сформированная учеником база данных может быть сохранена на магнитном диске центральной машины КУВТ.

Для КУВТ-86 имеется учебная база данных BASE, на которой реализована информационно-поисковая система «Страны мира». Работа с системой BASE подробно описана в [6]. Там же приводится поурочный план занятий.

Планирование уроков

Урок 1

Теоретическая часть. Информация и структуры данных. Три типа информационных структур: деревья, сети, таблицы.

Практическая часть (без ЭВМ). Ответы на вопросы и выполнение заданий.

1. Какой тип информационной структуры представляет расписание уроков в классе на неделю?

2. В форме какой структуры может быть представлена информация о трех типах объектов: класс, учитель, ученик? Постройте небольшой фрагмент такой структуры на примере вашей школы.

3. Какой тип информационной структуры получается; если объединить сведения о следующих объектах: страна, республика, область, район, населенный пункт?

4. В качестве домашнего задания предложить ученикам самостоятельно придумать пример трех типов информационных структур, взятых из различных областей жизни. Пусть, кроме того, ученики сформулируют запросы, на которые можно получить ответы из построенных структур.

Урок 2

Теоретическая часть. Реляционные базы данных. Понятия: поле (столбец), запись (строка), отношение (таблица). Связь между различными отношениями через общие поля. Операции «вырезать» и «склеить».

Практическая часть (без ЭВМ). Спроектировать реляционную базу данных, содержащую сведения о книгах в школьной библиотеке. Придумать несколько вариантов запросов к ней. Описать алгоритмы обслуживания запросов через операции «вырезать», «склеить». Дать задание на дом: спроектировать реляционную базу данных по индивидуальной теме для создания справочника.

Урок 3

Теоретическая часть. Первое знакомство с СУБД, имеющейся на школьном КУВТ. Инициализация работы СУБД. Открытие готовой БД для работы с ней. Способы (команды) просмотра БД на экране. Сортировка записей. Поиск по простым запросам.

Практическая часть. Инициализировать работу СУБД на ученических машинах. Вызвать заранее заготовленный экземпляр базы данных. Просмотреть БД на экране. Выполнить сортировку записей по какому-нибудь ключу. Реализовать поиск по нескольким простым запросам.

Урок 4

Теоретическая часть. Логические выражения, логические операции. Таблица истин-

ности. Правила построения условий поиска в форме сложных логических выражений. Получение ответа на запрос в форме отдельной таблицы.

Практическая часть. На той же готовой БД организовать поиск по сложным логическим условиям. Результаты поиска получить в форме отдельной таблицы (если это возможно).

Урок 5

Теоретическая часть. Типы данных (символьные строки, числа, логические величины). Режим создания БД. Команды создания и редактирования БД.

Практическая часть. Работа в режиме создания БД. Описать структуру и заполнить данными БД по индивидуальной теме (подготовка справочника).

Урок 6

Теоретическая часть. Команды редактирования и манипулирования базой: удаление записей; вставка в конец, вставка в середину новых записей; редактирование полей; изменение структуры БД.

Практическая часть. Практические задания на всевозможные операции манипулирования и редактирования БД.

Урок 7

Зачетный. Завершение работы над справочником. Демонстрация учениками своих справочников (БД). Демонстрация обслуживания запросов к справочнику.

Дополнительные уроки по этой теме могут быть посвящены знакомству с программным режимом работы с БД. Даются команды программирования, способы создания и исполнения программ.

Контрольные вопросы

1. Что называется информационной структурой данных?
2. Какие существуют виды информационных структур?
3. Что такое реляционная база данных?
4. Что такое отношение?
5. Почему реляционные БД являются наиболее распространенным типом баз данных?
6. Какие две основные операции используются для извлечения информации из реляционных таблиц?
7. Что такое ИПС?
8. Что такое СУБД?
9. Назовите основные режимы работы СУБД?
10. В чем назначение режима создания БД?
11. Какие характеристики данных определяются в режиме создания БД?
12. Что такое тип данных? С какими свойствами данных связано это понятие?

13. Какие действия можно выполнять с символьными данными, с числовыми данными?

14. Что такое «логическая величина»?

15. Что такое «простое и сложное логическое выражение»?

16. Какие существуют логические операции и как они выполняются?

17. Какие действия выполняются в режиме манипулирования данными?

18. Каким образом (по каким командам) происходит поиск в БД?

19. В какой форме отражаются результаты поиска?

20. В чем состоят основные этапы работы с целью создания БД?

21. В чем состоят основные этапы работы с целью получения информации по запросу к БД?

Т а б л и ц а 5

Функциональные возможности СУБД	1	2	3	4	5
1. Создание БД					
описание структуры (количество полей, имена, типы и форматы полей)		+	+	+	+
заполнение	+	+	+	+	+
формирование БД на диске			+	+	+
формирование БД в оперативной памяти с сохранением на диске	+	+			
2. Редактирование БД					
корректировка полей	+	+	+	+	+
добавление записей	+	+	+	+	+
удаление записей	+	+	+	+	+
изменение структуры	-	-	-	+	+
3. Манипулирование БД					
сортировка по одному ключу	-	+	+	+	+
сортировка по многим ключам	-	-	+	+	+
просмотр базы на экране	+	+	+	+	+
создание копии БД	-	-	+	+	+
математическая обработка числовых полей	-	-	+	+	+
4. Поиск в БД					
поиск по одному ключу	+	+	+	+	+
поиск по многим ключам	+	+	+	+	+
выделение активных записей в БД	-	-	+	+	+
получение активных записей в отдельной таблице	+	+	-	+	+
определение формата выходного документа с результатами поиска	-	-	+	+	+
печать выходного документа	-	-	+	+	+
5. Программирование работы с БД					
6. Слияние нескольких БД в одну					
	-	-	+	+	+

Обозначения:

1	— BASE	КУВТ-86
2	— ИРБА	УКНЦ
3	— РБА	УКНЦ
4	— dBASE	«Корвет»
5	— dBASE	IBM PC

Литература

1. *Церегулов Ф. И., Тарасенко Ф. П.* Введение в системный анализ. М.: Высшая школа, 1989.
2. *Тиори Т., Фрай Дж.* Проектирование структур баз данных: В 2 кн. М.: Мир, Кн. 1. 1985.
3. *Бургин М., Степенко Г.* Информационный поиск и компьютерная грамотность//Информатика и образование. 1990. № 1.
4. *Смирнов Н. Н.* Программные средства

персональных ЭВМ. Л.: Машиностроение, 1990.

5. *Фрейман В.* Методические подходы к обучению школьников работе с базами данных//Информатика и образование. 1991. № 3.

6. *Гриценко А.* Информационно-поисковая система для КУВТ-86//Информатика и образование. 1989. № 6.

В. Н. Исаков, В. В. Исакова

АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ: МЕТОДИЧЕСКИЕ АСПЕКТЫ

В настоящее время трудно однозначно ответить на вопрос: каким должен быть школьный курс информатики? Предлагаются многочисленные варианты курса для образовательных программ разных уровней, ведется поиск его инвариантного содержательного и методического ядра. Имея в виду прежде всего общеобразовательный курс ОИВТ для массовой школы, мы попытаемся обсудить два дидактических положения, которые, на наш взгляд, должны учитываться при разработке курса и организации его преподавания. Исходя из собственного опыта работы в вузе и школе, попутно обратим внимание на некоторые проблемы методического характера и возможные пути их решения.

Одна из важнейших целей курса, на наш взгляд, формирование и развитие навыков алгоритмического мышления.

В первые годы обучения основам информатики эта цель была фактически единственной. Со временем, когда в школах появились компьютерные классы, к различным КУВТ были адаптированы серьезные операционные системы и системы программирования, редакторы, электронные таблицы, СУБД и т. п. Спектр целей расширился. Тем не менее, работа с алгоритмами занимает центральное место и в других действующих курсах, в новых базовых и экспериментальных учебных программах. С методикой обучения основам алгоритмизации можно познакомиться в книге [1], а также в книге [2], в которой сказано, возможно, с излишней категоричностью: «Различные школьные предметы используют и развивают разные аспекты мышления. ... Для нас важно то, что существует специфический стиль мышления, который принято называть алгоритмическим. Развитие этого стиля мышления и есть основная цель курса информатики. ... Мы исходим здесь из того, что современная школа должна ... закладывать общую базу мышления и культуры, формировать всесторонне развитого человека, создавать условия для дальнейшего развития в любом направлении. Поэтому, если мы выделяем в человеческом мышлении какой-то специфический стиль, то его развитие представляет самостоятельную ценность...»

Привить учащимся навыки алгоритмического мышления и развивать их — трудная

задача. Трудность ее заключается в специфичности образа мыслительной деятельности, необходимой для составления алгоритма. Если в курсах математики, физики, лингвистики и т. п. решение задач в конечном счете сводится к исполнению готовых алгоритмов, то в курсе информатики учащиеся должны сами составлять такие алгоритмы. А это уже творческая работа, сродни научной деятельности, и не каждому под силу быстро освоить ее! Поэтому и важно начинать изучение информатики с младших классов, приучая детей к алгоритмическому мышлению.

В своей преподавательской практике мы стараемся донести до студентов специфичность алгоритмического стиля мышления: ведь учителю следует с пониманием относиться к тем учащимся, которые не могут самостоятельно написать, казалось бы, совсем простые алгоритмы. Если на начальной стадии изучения основ алгоритмизации после разбора, например, задачи на составление алгоритма вычисления суммы нескольких чисел ученик не справляется с алгоритмом вычисления их произведения, то это не удивительно: переход к нему не прост. С таким заданием самостоятельно могут справиться ученики, имеющие врожденное алгоритмическое мышление. Вот вам и тест на проверку способностей ученика!

Как формировать навыки алгоритмического мышления? Предлагаются разные рецепты. Один из них — постоянная умственная работа [2]. Ученики должны решить большое количество упражнений, причем эти упражнения необходимо располагать по возрастающей сложности. При этом средства описания алгоритмов должны появляться по мере необходимости. Аналогичное мнение высказано в книге [3]: «Занятия надо строить таким образом, чтобы решение одной задачи указывало путь решения другой, более сложной задачи, чтобы учащиеся пусть медленно, шаг за шагом, но самостоятельно покоряли вершины Алгоритма».

Такой проблемный подход, предполагающий к тому же «отзадный» метод предъявления теоретического материала, представляется очень полезным при изучении алгоритмических и информационных структур, средств их описания.

Существует еще один путь рационального обучения учащихся технике алгоритмизации: систематическое и целенаправленное применение *идей структурного подхода* (в частности, метода последовательной детализации). Он полезен и в плане воспитания учащихся, так как развивает умение планировать свои действия при решении сложных задач, способности к общению и коллективной деятельности. Следовало бы его применять с самого начала, уже при знакомстве с линейными алгоритмами, тем более что первый шаг проектирования алгоритма — разбивка задачи на подзадачи — обычно порождает линейную структуру.

К сожалению, предложить такой путь учителям пока невозможно: отсутствует соответствующая система упражнений. Разработать такую систему — интересная и важная задача для исследователя-методиста.

Эффективная форма работы с алгоритмами — их *исполнение*. Наибольший методический эффект, с точки зрения пояснения смысла алгоритма, проверки его правильности и, вообще, обучения навыкам алгоритмизации, может быть достигнут тогда, когда в процессе исполнения алгоритма, во-первых, прослеживаются все действия исполнителя и, во-вторых, наглядно и точно «протоколируются» результаты этих действий. В книге [4] подчеркнута, что исполнение должно быть не только темой в учебнике, но и инструментом в руках учителя. Там же приведена одна из схем ручного исполнения алгоритмов. Кстати, в работе [1, гл. 2] приведен обзор различных способов протоколирования при ручном исполнении.

Аналогичные мысли высказаны в статье [5], описывающей методику компьютерного моделирования схематично описанных алгоритмов.

Естественным исполнителем компьютерных программ является сам компьютер. Но если после выполнения программы на экране дисплея появляется только результат (как это чаще всего и бывает), то польза от такого исполнения с точки зрения понимания смысла работы отдельных операторов и динамики выполнения программы в целом близка к нулю. Большой эффект может быть достигнут при выполнении программ в режиме трассировки. Если такой режим в системе программирования не предусмотрен, необходимо организовывать хотя бы выходы промежуточных результатов с подробными комментариями.

Мы считаем, что и при работе с компьютером необходимо практиковать ручное исполнение программ. Опыт показывает, что

при сочетании ручного и компьютерного исполнений обучаемые быстрее осваивают язык программирования и овладевают программистскими навыками. Об этом же говорится в книге [6], где предложен способ диаграммной иллюстрации ручного исполнения Бейсик-программ.

В этом плане представляют интерес специальные программы, позволяющие организовывать исполнение Бейсик- или Паскаль-программ так, как это производится, например, в системе «Е-практикум». Полезной была бы также разработка систем упражнений с ориентацией на ручное исполнение алгоритмов и на сочетание ручного и компьютерного исполнений программ.

Система задач курса информатики должна иметь практическую направленность.

Это положение относится прежде всего к тем разделам, которые посвящены алгоритмизации и программированию.

Мы уже говорили о том, насколько трудно формировать алгоритмическое мышление у учащихся. К предложенным рецептам следует добавить еще один: повысить интерес учащихся к составлению алгоритмов, подняв *уровень мотивированности задач*. Для этого следует по возможности отказаться от формальных, отвлеченных формулировок. Отвлеченные задачи воспринимают обычно только те люди, которые имеют склонность к формализованному мышлению. Они хороши для развития техники алгоритмизации и программирования у тех, кто воспринял деятельность по составлению алгоритмов как естественную потребность. Наш опыт показывает: для большинства учащихся такие задачи не несут никакой мотивационной нагрузки.

О теме «Массивы данных» в книге [7] говорится: «Эта тема представлена задачами поиска суммы элементов массива, количества элементов больше нуля, максимального (минимального) элемента массива и т. п. Трудно спорить с тем, что они не вызывают у ученика ничего, кроме скуки». И затем предлагается в этой теме решать задачи на обработку данных типа ведомостей, журналов, каталогов и т. п.

Вряд ли следует строить общеобразовательный курс информатики, на наш взгляд, и на задачах из области математики. Их мотивационный уровень для большинства учеников в обычном школьном классе также низок. Мы не отрицаем, что учащимся надо уметь составлять алгоритмы решения различных математических задач, но в то же время считаем, что такие алгоритмы должны быть преимущественно средством решения других задач, а не самоцелью.

Нельзя забывать, что информатика —

межпредметная наука. Более того, спектр решаемых ею задач выходит за пределы изучаемых в школе наук. Выпускник школы должен иметь представление о самых разных областях человеческой деятельности, где применяются ее методы. И не только о тех, где применяются текстовые процессоры, электронные таблицы и другие прикладные системы.

Если молодой человек понимает, каким образом работает автоматизированная система продажи билетов на самолеты и поезда, если он в принципе может понимать алгоритм работы игрового автомата или автопилота, знает, как можно автоматизировать обработку результатов соревнований, процедуру получения информации в библиотеке и т. п., это значит: он изучал информатику не зря. Если же он не только понимает алгоритмическую суть различных реальных процессов и методов решения задач, но и знает, как при помощи компьютера смоделировать эти процессы и решить эти задачи, то, значит, информатика дала ему средства исследования окружающего мира.

Но дать эти средства информатика не сможет, если не будет сформировано алгоритмическое мышление учащихся на системе упражнений практического, прикладного характера.

Опыт показывает, что упражнения с практическим смыслом заинтересовывают учащихся больше всего. Учащиеся начинают говорить на языке той предметной области, на котором сформулирована задача. Они с интересом разбирают пути ее решения, предлагают другие возможные формулировки этой же задачи или ее аналогов. На основе этих заданий значительно проще реализовать проблемный и опережающий методы обучения, так как они естественным образом порождают более сложные свои продолжения. Они становятся сквозными, т. е. используемыми на разных этапах обучения.

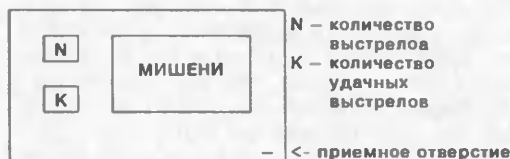
Не вызывает сомнения и то, что метод последовательной детализации лучше всего применять при решении прикладных задач. При их решении возникает также потребность в математическом моделировании.

Рассматривая задачи с реальным смыслом, учитель получает возможность попутно знакомить учащихся с разными видами профессиональной деятельности, т. е. вести профориентационную работу.

Приведем пример практической задачи и покажем, как на ее основе можно реализовывать те формы и методы обучения, о которых мы вели речь.

Задача 1

Обращаем внимание учащихся на обилие игровых автоматов и предлагаем окантоваться на месте их создателей. Выбираем один из самых простых — автомат «Меткий стрелок». Рисуем модель его внешнего вида:



Возникает вопрос: каким должен быть алгоритм работы этого автомата?

Выбор стратегии разработки алгоритма зависит от того, хочет или нет учитель применить здесь метод последовательной детализации. Если да, то необходимо дать полное описание работы автомата. Если нет, то можно выбрать путь «от простого к сложному» и сформулировать только основной принцип его работы:

Опустив в приемное отверстие жетон, игрок получает возможность стрелять по мишеням. При удачном попадании в мишень ему начисляется одно очко, а при промахе — нет.

Тем самым впоследствии учащиеся получают возможность «изобрести» такой автомат, какой им хочется: с любым количеством выстрелов на один жетон, с призовой игрой или без нее, давать приз только за стрельбу без промаха или за определенное количество набранных очков, за то, уложился стрелок в некоторый временной норматив или нет. И т. д.

Рассмотрим второй путь.

Обсуждаем смысл задачи и составляем словесное или схематичное описание алгоритма при одном выстреле. Предполагается, что жетон автоматом принят и стрельба разрешена. Вариант алгоритма:

1. Принять выстрел.
2. Если стрелок попал в мишень, то добавить очко, в обратном случае очко не добавлять.

На этом примере можно объяснить необходимость структуры «неполная развилка», так как фраза «не добавлять очко» на самом деле не предполагает выполнения какого-либо действия и может быть опущена.

Либо в процессе написания алгоритма, либо в конце его следует поговорить о разных способах физической реализации таких действий автомата, как фиксация выстрела, проверка «попал — не попал», высвечивание результатов стрельбы в окошечках. Надо также выяснить математический смысл попадания,

например, в круглую мишень. Последнее может пригодиться при переходе к алгоритму с числовыми величинами, разработке компьютерной модели этого автомата или при решении других аналогичных задач.

Далее выясняется, что на самом деле за один жетон игрок может сделать, например, 20 выстрелов. Тогда возникает идея повторить 20 раз приведенные выше действия и организовать суммирование набранных очков.

Вырисовывается такой вариант алгоритма:

1. Принять выстрел.
2. Если стрелок попал в мишень, то добавить ему очко к уже набранным.
3. Если количество выстрелов меньше 20, то перейти к команде 1, иначе закончить стрельбу.

Таким образом выявляется необходимость в структуре «цикла». При этом по желанию учителя можно рассмотреть и ввести разные ее модификации.

Автомат имеет дело с двумя числовыми величинами — количеством произведенных выстрелов (N) и количеством очков, набранных при попаданиях (K). Значения N и K высвечиваются в окошечках автомата после каждого выстрела. Тогда фразу «принять выстрел» можно заменить командой присваивания $N:=N+1$, а предписание «добавить очко» — командой $K:=K+1$.

Иногда трудно убедить некоторых учеников, что перед выполнением цикла, где происходит суммирование, необходимо обнулить ту переменную, которая станет потом суммой. Здесь это сделать легко, так как учащиеся понимают, что при включении автомата до первого выстрела в соответствующих окошечках высвечиваются нули. Значит, необходим блок подготовки цикла: присваивания $N:=0$ и $K:=0$.

Здесь же можно обсудить с учащимися другой вариант конструкции автомата: когда при включении его в окошке (N) высвечивается число 20. Тогда и сам алгоритм, и блок подготовки цикла должны измениться. Как именно — это и должны узнать учащиеся.

Затем ученики или учитель «догадываются», что автомат может предложить призовую игру при стрельбе без промаха. Естественное усложнение алгоритма и повод для обсуждения: при помощи какой алгоритмической структуры это можно реализовать, достаточно уже имеющихся или необходимо вводить новую? Так возникает новое задание.

Итак, мы видим, что попытка исследовать принцип управления работой автомата «Меткий стрелок» дает интересную возможность составлять алгоритмы различной структуры с

величинами и без них. Поэтому эта задача может быть сделана сквозной.

При желании и возможности можно создать компьютерную модель этого автомата с помощью элементов графики. «Меткий стрелок» может стать тренажером для ориентировки на координатной плоскости: в качестве мишеней могут служить круги, появляющиеся на некоторое время на экране. После их исчезновения игроку нужно набрать координаты точки, которая, по его мнению, попадет в этот круг. А дальше все может работать так, как описано выше.

Много образцов задач прикладного характера можно привести к теме «Массивы (табличные величины)». О необходимости придания конкретной смысловой или ситуативной нагрузки элементам массивов и задачам их обработки уже говорилось выше.

Следующие примеры интересны тем, что они, с одной стороны, порождают большое количество заданий различной сложности, с другой — побуждают учеников включиться в коллективную разработку достаточно сложных программных продуктов (естественным образом возникает потребность в применении идей структурного подхода).

Задача 2

Дадим учащимся задание:

разработать алгоритмы (программы) решения задач, выполняемых автоматизированной железнодорожной билетной кассой.

Предположим, что касса продает билеты в один вагон и в каждом из билетов указан номер места в вагоне. В какой-то момент времени надо:

- выяснить, сколько билетов продано;
- узнать, можно ли купить билет на конкретно указанное место;
- вывести на экран дисплея информацию о том, какие места свободны;
- определить количество свободных нижних мест;
- узнать, можно ли приобрести три билета в одно купе;
- организовать продажу одного билета на свободное место;
- осуществить операцию возврата билета

и т. п.

Эту же задачу можно вспомнить, когда будут изучаться двумерные массивы. Тогда можно производить операции с билетами на целый поезд.

Аналогичное поле деятельности представляет ситуация, приведенная в задаче 3.

Задача 3

Бригада из N человек изготавливает детали. Известны цена одной детали и величина оплаты за каждую произведенную деталь. После рабочего дня с пульта ЭВМ вводятся данные о выработке каждого рабочего. Необходимо организовать учет производимой продукции и начисление заработной платы.

Можно предложить задания практически на все виды обработки одномерных и двумерных массивов (суммирование элементов, сортировка, поиск, отыскание максимальных и минимальных элементов и т. д.).

Применяя задачи конкретного содержания из разных областей жизнедеятельности

Литература

1. Антипов И. Н. Основы информатики и вычислительной техники: Методическое пособие для преподавателей техникумов. М.: Высшая школа, 1991.

2. Зайдельман Я. Н., Лебедев Г. В., Самовольнова Л. Е. Три кита школьной информатики//Информатика и образование. 1993. № 3. С. 19—23.

3. Сочнев С. Путь от простого к сложному, или Покорение алгоритма//Информатика и образование. 1989. № 4. С. 33—38.

4. Исаков В. Н. Исполнение алгорит-

человека, учитель действительно получает возможность добиться многих целей: привить интерес к своему предмету, показать его связь с жизнью, разнообразить методы учебно-воспитательной работы, дать ученикам прочные знания, умения и навыки по применению методов информатики в повседневной жизни.

Многие учителя давно уже применяют такие задачи в своей практике. Есть они в учебных пособиях, публикуются и на страницах ИНФО. Но, к сожалению, это только наборы задач, их следовало бы привести в методически продуманную, дидактически обоснованную систему.

мов//Информатика и образование. 1987. № 2. С. 30—43.

5. Руденко В. Имитационное моделирование учебных алгоритмов//Информатика и образование. 1990. № 5. С. 42—45.

6. Поливанный И. Исполнение Бейсик-программ//Информатика и образование. 1989. № 3. С. 44—52.

7. Парсаганов О. Роль задач и упражнений в курсе информатики//Информатика и образование. 1992. № 2. С. 25—29.

Фонд новых технологий в образовании «Байтик» (г. Троицк Московской обл.) при участии журнала «Информатика и образование»

с 29 июня по 2 июля 1995 г.

проводит традиционную

Шестую Международную конференцию «Использование новых технологий в образовании».

В конференции примут участие американские представители организации Computer Using Educators (CUE) и специалисты других стран.

Конференция посвящена вопросам практического использования новых технологий в образовании и будет проводиться по пяти секциям:

- компьютер для детей дошкольного и младшего школьного возраста;
- использование компьютера для преподавания школьных дисциплин;
- новые технологии в учебном процессе;
- использование компьютерных приложений;
- компьютер для учителя и школьной администрации.

Во время конференции будет проводиться коммерческая выставка-продажа программных и технических средств. Принимаются заявки на участие в выставке.

По всем вопросам организации конференции обращаться в оргкомитет.

Телефон/факс (095) 334-03-67.

С. Е. Пронина

ЛОГО В ШКОЛЬНОМ КУРСЕ ИНФОРМАТИКИ

В статье раскрывается вклад Лого в методику обучения информатике, его роль в формировании мышления школьников. Рассматривается возможность использования языка для изучения основ как процедурного, так и функционального программирования, а также для изучения машинной графики. Излагается краткая история создания языка, приводятся аргументы в пользу введения Лого в школьный курс информатики, отмечается также, что Лого является не столько зарубежным аналогом школьного алгоритмического языка, сколько его альтернативой.

За рубежом лучшим стартовым языком обучения детей основам программирования и машинной графики признан язык Лого. Ему посвящены десятки книг, издаются специальные журналы. В Болгарии, Японии, США и Франции уроки информатики с использованием языка Лого начинаются с V класса и уже обсуждается вопрос о введении его изучения с III класса.

К сожалению, в России Лого пока мало известен. Практически отсутствуют методические пособия и учебники, лишена программной совместимости на международном уровне русифицированная версия языка.

В российской школе используется для обучения основам алгоритмизации и программирования школьный учебный алгоритмический язык, разработанный А. П. Ершовым и усовершенствованный А. Г. Кушниренко. Лого является не столько аналогом этого языка, сколько его альтернативой.

Выбор начального языка программирования имеет принципиальное значение, поскольку от этого во многом зависит процесс овладения компьютерной грамотностью и дальнейшая деятельность по применению компьютеров в решении практических задач.

Выбор языка программирования для начинающих должен определяться психологическими требованиями и методическими рекомендациями, позволяющими воспитывать у обучаемых навыки хорошего стиля программирования. В общем виде можно сформулировать следующие требования к начальному языку программирования:

- логическая и синтаксическая простота;
- лингвистическая естественность;
- наглядность основных действий;
- исключение второстепенных понятий, заслоняющих основные принципы программирования;
- привлекательность в обучении;
- простота в общении с компьютером, дружественный интерфейс, возможность написания программ без предварительного длительного изучения языка.

В соответствии с этими рекомендациями была предпринята попытка разработать спе-

циальный язык обучения начинающих основам программирования. В результате появился язык, получивший международное признание и массовую популярность.

Он был разработан в США Сеймуром Пейпертом в конце 60-х гг. и получил название Лого, которое происходит от греческого «логос» — «слово, мысль, смысл, идея».

Основные характеристики языка:

- синтаксис, близкий к естественному языку;
- приспособленность к интерактивному режиму работы;
- ориентация на формирование самых общих представлений о программировании;
- необязательность глубоких знаний архитектуры ЭВМ;
- система графических примитивов.

Даже без графических возможностей Лого пригоден для обучения основам программирования. Для того чтобы этот язык был пригоден для обучения учащихся младших классов, в нем была реализована идея исполнителя — Пайперт предложил так называемую *черепашку*. Название «черепашка» связано с работой нейрофизиолога Грея Уолтера, проводившего эксперименты над маленькими электронными роботами-черепашками. За всю историю Лого существовало несколько видов черепашки: реальные физические объекты типа напольной механической игрушки-робота; графопостроители, чертящие рисунок на бумаге; абстрактные объекты, обитающие на экране дисплея. Сегодня черепашка — это управляемое с помощью компьютера кибернетическое животное, существующее в языковой среде Лого, с овладением которым связывается усвоение основных понятий программирования. При движении черепашка может оставлять след, и эта ее способность используется для вычерчивания фигур на экране.

Неоценим вклад Лого в методику обучения. Среда обучения Лого является примером новых технологий обучения, направленных на освоение средств, при помощи которых учащиеся могут самостоятельно добывать зна-

ния. Пайперт, создавая Лого, основывался на учении швейцарского психолога Жана Пиаже о стихийном, непринужденном обучении ребенка при взаимодействии со средой. Среда Лого позволяет так построить процесс обучения, что *овладение языком программирования становится таким же естественным процессом, как и изучение родного языка.*

Слова «компьютерное обучение» обычно означают, что компьютер учит («программирует») ребенка. Занятия в среде Лого переводят традиционную ситуацию компьютерного обучения. В среде Лого ребенок сам учит и программирует компьютер и, делая это, овладевает основами предмета.

Лого реализует новые подходы к обучению, нацеленные не на заучивание правил, а на *формирование процесса мышления.* В ситуации традиционного обучения наблюдать за мыслительной деятельностью ребенка просто невозможно. Среда Лого является тем окном, которое позволяет заглянуть в мыслительные процессы ребенка. У учителя появляется возможность проанализировать накопленные в памяти компьютера данные о работе ребенка и получить представление о том, как ребенок думает, как он понимает задачу, расчленяет ее и т. п.

В среде Лого ребенок сам управляет процессом обучения. Как и в реальной жизни, он сам ставит себе задачу и сам находит пути ее решения. Вместо привычного ожидания, чтобы все за него сделали другие, ребенок попадает в ситуацию, управлять которой может только он сам. Из постоянно ждущего помощи от других он превращается в человека, самостоятельно ищущего и находящего решения. На собственном опыте ребенок учится делать выводы и обобщения.

В Лого изначально заложены *принципы конструктивного обучения.* Согласно этим принципам в процессе создания реального продукта (компьютерной программы) значительно повышается эффективность обучения. Это возможно только потому, что Лого — полноценный язык программирования, допускающий возможность создания настоящих, графически оформленных, работоспособных программ.

Важную роль в новых направлениях обучения играет отношение к ошибкам и их устранение. Отладка программы (поиск и исправление в ней ошибок) обычно является трудоемким процессом. Язык Лого значительно упрощает его. Решение разбивается на множество мелких шагов, каждый из которых может быть проверен отдельно. Если допущена ошибка, то черепашка просто не выполнит задания. Так как ученики могут самостоятельно найти ошибку и установить ее источник,

они перестают бояться ошибок. Школьники чувствуют себя исследователями, не боящимися творить. Они обретают уверенность в своих силах и могут выразить свое знание предмета в форме, отличной от вербальной.

Лого привлекает и тем, что программирование из скучной дисциплины превращается в интересную игру, в процессе которой происходит быстрое и прочное усвоение основных понятий и навыков программирования.

В настоящее время Лого — одно из немногих достижений в области методики преподавания и обучения. За годы своего существования Лого произвел настоящую революцию в обучении.

Известно, что школьный курс должен отражать основные направления развития информатики. К ним относятся процедурное и функциональное программирование. *Обучение процедурному и функциональному программированию при плавном переходе от одного к другому, без введения нового языка в изучение, возможно только при использовании языка Лого.* На первых этапах обучения Лого — классический процедурный язык, позволяющий легко усваивать принципы структурного программирования. При дальнейшем изучении языка (рекурсии, обработки списков и др.) в Лого становится видна его функциональная сторона.

Универсальный, с богатыми возможностями язык Лого позволяет решать вычислительные и невычислительные задачи, включающие в себя все задания, рассматриваемые в сложившемся курсе информатики средней школы.

Графические возможности Лого позволяют использовать его для обучения машинной графике, тем более что все современные языки программирования включают в себя элементы Лого-графики. Но популярность Лого-графики имеет и свои недостатки. Существует опасная тенденция отождествлять возможности Лого только с его графическими возможностями. Не стоит ограничивать изучение языка овладением исполнителем черепашкой и Лого-графикой. Это отвлекает от других, не менее важных преимуществ языка и превращает его только в интересную игрушку.

Лого объединяет в себе черты многих языков программирования. *Изучение Лого как начального языка значительно облегчает дальнейшее изучение профессиональных языков программирования.*

Так же как и язык Бейсик, Лого является интерпретатором, обеспечивающим диалоговый характер общения с пользователем, превращающим компьютер в мощный калькулятор в режиме непосредственных вычислений.

Структурное построение программы роднит Лого и Паскаль. Лого, как и Форт, — не только структурные, но и саморасширяющиеся языки, допускающие введение новых слов и расширение словарного запаса языка. Мощный аппарат обработки объектов языка Лого аналогичен языку искусственного интеллекта Лисп.

В заключение приведем аргументы в пользу введения Лого в школьный курс информатики:

- Лого удовлетворяет требованиям начального языка программирования;
- философия Лого — истинное достижение в методике обучения;
- Лого дает широкий взгляд на программи-

рование (на его процедурный и функциональный аспекты);

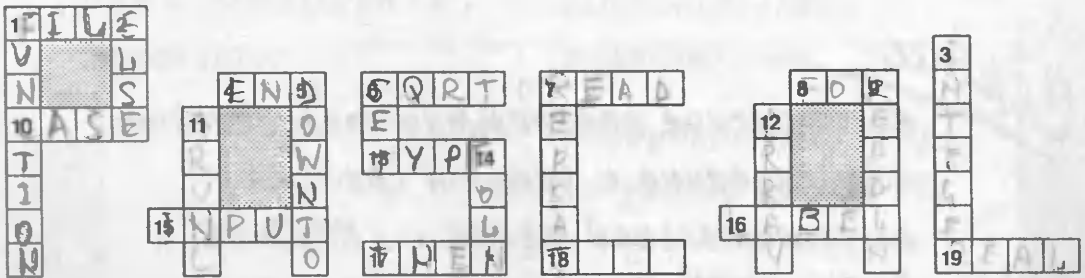
- обучение Лого является пропедевтикой изучения профессиональных языков программирования;
- Лого пригоден для изучения основ машинной графики.

Кроме того, обучение в среде Лого вызывает у учащихся повышенный интерес к предмету, развивает математическую интуицию и геометрические представления, является своеобразным математическим тренажером, формирует алгоритмический, структурный, логический и комбинаторный типы мышления.

* * *

КРОССВОРД

Составитель — А. Домнич (пос. Каменка Воронежской обл.)



По горизонтали:

1. Именованная совокупность данных на диске.
4. Правая операторная скобка.
6. Стандартная функция языка Паскаль.
7. Процедура для работы с текстовым файлом.
8. Часть заголовка цикла.
10. Оператор варианта.
13. Команда операционной системы.
15. В Бейсике — команда ввода.
16. Безобразие с точки зрения структурного программирования.
17. Часть оператора ветвления.
18. Тип файла.
19. Тип данных.

По вертикали:

1. Служебное слово.
2. Часть оператора ветвления.
3. Тип данных.
5. В Бейсике — отрицательный шаг.
6. В Фокале — оператор присваивания.
7. Заголовок цикла.
9. Процедура чтения для текстового файла.
11. Стандартная процедура.
12. Служебное слово для описания массива.
14. Признак конца строки.

Ответы на кроссворд см. на стр. 124.



ЭКСИ

АОЗТ Фирма "ЭКСИ"

Классы

Проектирование конфигурации и поставка "под ключ" компьютерных классов и комплексов на базе IBM-совместимой техники, Apple

Мебель

Изготовление современной мебели для учебных заведений и офисов (доставка осуществляется по Москве и Московской области)

Сервис

Комплексное программно-техническое обслуживание и ремонт средств вычислительной техники (КУВТ-86, УКНЦ, УКНЦ-01, ДВК, БК, Партнер, Корвет, Агат, IBM-совместимая техника, Apple)

107005, г. Москва, Волховский пер., 11
АОЗТ Фирма "ЭКСИ" (бывш. "Колледж")
Тел./факс: (095) 265-62-65
Тел.: (095) 267-70-58

Программы

Обеспечение системными, инструментальными, программно-педагогическими и административными средствами для IBM-совместимой и техники типа УКНЦ, УКНЦ-01, КУВТ-86, ДВК. Каталог программ высылается бесплатно

Учебные пособия

Карасева Т.В. Сборник задач и упражнений по основам информатики и вычислительной техники.

Поддубная Л.М. и др. 10 уроков на IBM PC.

Каймин В.А. Основы компьютерной технологии.



ЭКСИ

АОЗТ Фирма "ЭКСИ"

А. Г. Юдина

ИНФОРМАТИКА НА ЛОГО ДЛЯ СТАРШЕКЛАССНИКОВ*

15. Глава для учителя

Исполнитель «Весы»

Предыдущие главы были предназначены для учащихся и могут служить учебным пособием (если вырезать готовые программы, помещенные в конце каждой главы). Данная, заключительная, часть адресована учителю.

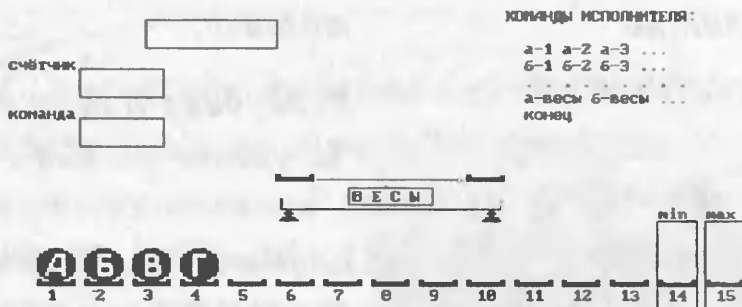
Вероятно, большинство преподавателей информатики уже успели по достоинству оценить «Роботландию». Программы этого замечательного комплекса можно использовать разнообразно и даже в старших классах.

А почему бы не пополнить самостоятельно коллекцию исполнителей «Роботландии»? Легко ли словесно, «на пальцах», объяснять алгоритмы поиска максимума и минимума, сортировки различными способами? Вот если бы можно было предложить ребятам практические работы по взвешиванию камушков или яблок, дело пошло бы наверняка лучше.

В этой главе рассказывается, как создать исполнителя «Весы», оформленного почти как в «Роботландии».

Вообще, Лого прекрасно вписывается в идеологию «сделай сам». Для ученика: сделай сам модель, скажем, движения — и проводи с ней эксперименты, открывая зависимости, устанавливая закономерности. Для учителя: можно самостоятельно создавать подсобные средства — «демонстрашки», «обучалки», тесты, а затем фрагменты из них использовать в качестве заданий ученикам. Исполнитель «Весы» (кроме использования по прямому назначению) позволит при разборе с учащимися отдельных частей программы продемонстрировать элементы системного подхода и дать примеры иного использования списков, чем в предыдущих главах.

Итак, на экране — двухчашечные весы и ряд пронумерованных подставок, на которых могут лежать шары, обозначенные буквами. Размеры шаров одинаковые, а вес — разный, случайный. Перекладывая шары на свободные подставки и взвешивая их, можно находить минимум, максимум и производить сортировку. На экране также есть поля команд, счетчика команд и реакции программы на ошибки. Число шаров можно устанавливать перед началом работы.



Система команд исполнителя:

1) команды переноса шара на подставку

(например: а-8

г-5

б-10);

* Окончание. Начало см.: Информатика и образование. 1994. № 6; 1995. № 1.

2) команды переноса шара на весы

(например: а—весы
б—весы);

3) команда «конец», после которой над каждым шаром распечатывается его вес (для проверки правильности проделанной работы).

Реакция программы:

«не понимаю» — при синтаксических ошибках, т. е. если набранная команда не входит в состав команд исполнителя;

«не могу» — в случае попытки переложить шар на весы, когда обе чашки уже заняты, или на занятую подставку; а также при попытке отправить на весы шар, который уже лежит там.

Шары и подставки

Для решения поставленной задачи опишем систему объектов — шаров и подставок (в их число входят правая и левая чашки весов, они отличаются от других подставок подвижностью).

Каждый шар можно описать следующим набором признаков:

- 1) подставка, на которой он находится;
- 2) вес;
- 3) номер формы

(в листе форм необходимо заготовить формы шаров с буквами — по максимально возможному числу шаров).

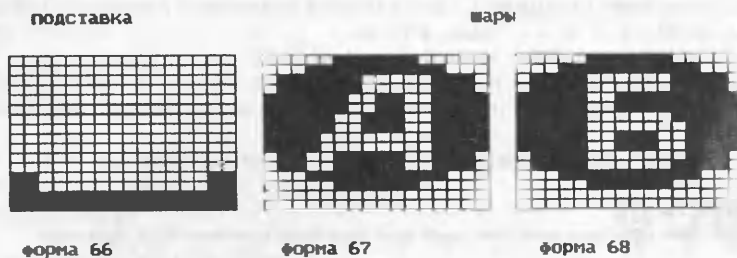
Каждую подставку можно описать списком из двух элементов:

- 1) координаты;
- 2) имя шара, находящегося на ней, или 0 — если подставка свободна.

Таким образом, шары и подставки связаны друг с другом.

Пусть у нас будет 15 неподвижных подставок. Имена переменных для них: 1, 2, 3, ..., 15; номер формы подставки — 66. Максимальное число шаров — 13, имена переменных — а, б, ..., н. Номера форм шаров — с 67 по 79. Центры форм шаров и подставок должны совпадать.

Пример изображения объектов в листе форм:



Первая процедура образует 15 переменных для подставок, рисует их на экране (вернее, отпечатывает) и проставляет под ними номера:

```
to подставки
setc 2 setsh 66
make "i 1
pu setpos [-280 -127]
repeat 15 [name list pos 0 :i
pd stamp pu
sety y - 40 setx x - 5
pd label :i pu
sety y + 40 setx x + 45
make "i :i + 1]
end
```

Первая строка в цикле — заведение очередной переменной. Так как имя переменной явно не задано — оно само является текущим значением переменной *i* — использована команда NAME (парная и обратная команде присваивания MAKE). Вторая строка отпечатывает

подставку. Строки 3—5 — отпечатывание номера под подставкой и продвижение на 40 шагов вправо (на место следующей подставки).

После выполнения процедуры в памяти образуются переменные:

имя	значение
1	[[-280 -127] 0]
2	[[-240 -127] 0]
3	[[-200 -127] 0]
...	...
15	[[280 -127] 0]

Следующая процедура заводит переменные для шаров (называет их начиная с «а», далее — по алфавиту); значение каждой — список из имени подставки, веса и номера формы. Шары раскладываются (отпечатываются) на подставки по порядку, начиная с первой. Во вторые элементы занятых подставок записываются имена шаров.

```
to шары :n
  pu setc 14
  make "i 1
  repeat :n [make "Ш char 159 + :i
    make "П :i
    name (se bl thing :П :Ш) :П
    make "форма 66 + :i
    make "вес 5 + random 20
    name (list :П :вес :форма) :Ш
    setpos first thing :П
    setsh :форма
    pd stamp pu
    make "i :i + 1 ]
ht
end
```

Первая строчка цикла образует имя очередного шара (оно является значением переменной :Ш). Вторая и третья строчки связывают шар со следующей пустой подставкой — в переменной для этой подставки второй элемент (0) заменяется на имя шара. Здесь впервые встречается операция THING — она выдает значение переменной и используется в тех случаях, когда имя переменной не задано явно (например, само является значением некоторой переменной). Если имя задано явно, то thing "Т и :Т — одно и то же.

Строки 4—6 — формирование значения переменной для очередного шара.

Строки 7—9 — отпечатывание изображения шара на занятой им подставке.

После выполнения процедуры (например, ШАРЫ 3) вид экрана и состояние переменных следующие:



имя	значение
1	[[-280 -127] а]
2	[[-240 -127] б]
3	[[-200 -127] в]
4	[[-160 -127] 0]
5	[[-120 -127] 0]
...	...
15	[[280 -127] 0]
<hr/>	
а	[1 3 67]
б	[2 21 68]
в	[3 14 69]

Вторые элементы в списках для шаров — случайные веса.

Теперь необходимо описать перенос шаров с подставки на подставку. Для этого нужны следующие действия:

1) черепашка принимает форму указанного шара;

2) черепашка стирает шар на старом месте (перекрашивается в цвет фона и ставит свой отпечаток);

3) черепашка перемещается на указанную подставку и отпечатывает шар на новом месте.

При этом должны измениться значения всех трех переменных: подставки, на которой шар находился (второй элемент списка становится нулевым); шара (меняется первый элемент списка — вместо старой подставки должна быть указана новая); подставки, на которую шар переносится (второй элемент списка принимает значение имени шара).

При попытке перенести шар на занятую подставку должно появиться сообщение «не могу».

```
to на_подставку :Ш :П
```

```
ifelse занята :П
```

```
  [ ответ [не могу]]
```

```
  [ перенос :Ш :П ]
```

```
end
```

```
to занята :П
```

```
or not (last thing :П) = 0
```

```
end
```

```
to перенос :Ш :П
```

```
старое_место
```

```
pu setpos first thing :П
```

```
новое_место
```

```
end
```

```
to старое_место
```

```
ht pu setsh last thing :Ш
```

```
setpos first thing first thing :Ш
```

```
setc 0 st pd stamp setc 14
```

```
name ( se bl thing first thing :Ш 0) first thing :Ш
```

```
end
```

```
to новое_место
```

```
pd stamp ht pu
```

```
name (se :П bf thing :Ш ) :Ш
```

```
name (se bl thing :П :Ш) :П
```

```
end
```

```
to ответ :текст
```

```
top cd tab tab settc 12
```

```
insert :текст
```

```
make "q readchar
```

```
select sol cut settc 1
```

```
end
```

Процедура НА_ПОДСТАВКУ использует два входных данных: имя шара и имя подставки. Например, в результате выполнения команды

```
НА_ПОДСТАВКУ "б 4
```

шар «б» будет перемещен на подставку 4.

Функция ЗАНЯТА — логическая, т. е. ее результатом является значение "ИСТИНА или "ЛОЖЬ.

Разберем процедуру СТАРОЕ_МЕСТО.

Первая строчка — установка формы шара, указанного значением переменной :Ш (last thing :Ш).

Вторая строчка — установка черепашки на позицию подставки (она записана в переменной для этого шара).

В нашем примере в начале выполнения команды

```
НА_ПОДСТАВКУ "б 4
```

шар «б» с весом 21 находится на второй подставке (ее координаты [-240 -127]). Тогда значение переменной Ш (:Ш) — б.

Значение переменной б (thing :Ш) — [2 21 68]. Здесь имя подставки — первый элемент списка — first thing :Ш (переменная 2).

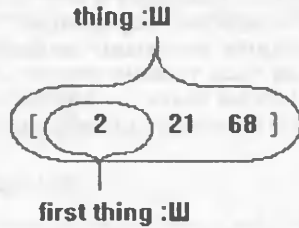
Значение переменной 2 (thing first thing :Ш) — [-240 -127] б].

А координаты — первый элемент этого списка — first thing first thing :Ш.

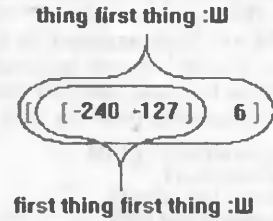
Значение
переменной Ш:



Значение
переменной б:



Значение
переменной 2:



Третья строка процедуры СТАРОЕ_МЕСТО стирает шар на старом месте.

Четвертая строка заменяет второй элемент в списке для подставки, на которой шар находился, на ноль (так как она теперь освободилась).

Процедура НОВОЕ_МЕСТО отпечатывает шар на новом месте, меняет значение первого элемента в переменной для шара и второго элемента в переменной для новой подставки. Картинка на экране и состояние таблицы переменных в нашем примере станут следующими (подчеркиванием выделены изменившиеся значения):

	А		В	Б											
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
имя	значение														
1	[[-280 -127] а]														
2	[[-240 -127] 0]														
3	[[-200 -127] в]														
4	[[-160 -127] б]														
5	[[-120 -127] 0]														
...	...														
15	[[280 -127] 0]														
а	[1 3 67]														
б	[4 21 68]														
в	[3 14 69]														

Процедура ОТВЕТ включает в себя задержку до нажатия любой клавиши (чтобы можно было сообразить, в чем ошибка) и после этого стирает сообщение.

Процедура ПЕРЕНОС осуществляет «мгновенное» перемещение шаров. Можно «украсить» программу, вставив вместо строчки

```
pu setpos first thing :Ш
```

процедуру движения шара по прямоугольной траектории (подъем — полет → спуск) или даже по параболической с замедлением и ускорением.

Шары уже можно перемещать с подставки на подставку; следующая задача — моделирование весов и процесса взвешивания.

Весы

Если до сих пор с переносом шаров справлялась одна черепашка (ее номер по умолчанию — 0), то для взвешивания удобно использовать все четыре черепашки — две будут чашами весов (пусть левая чаша — черепашка 3, а правая — черепашка 2), а оставшиеся — лежащими на чашах шарами (слева — черепашка 1, справа — черепашка 0).

Процедура ЧАШИ устанавливает вторую и третью черепашки на исходные позиции и заводит две переменные для чаш весов.

```

to чаша
ask 2 3 [setsh 66 setc 15 st pu]
ask 2 [ setx 80 sety 0]
make "лев. ч [[-80 0] 0]
make "пр. ч [[ 80 0] 0]
end

```

В нашем примере с тремя шарами и уже переложённым шаром «б» вид экрана и таблица переменных станут следующими:



имя	значение
1	[[-280 -127] а]
2	[[-240 -127] 0]
3	[[-200 -127] в]
4	[[-160 -127] б]
5	[[-120 -127] 0]
...	...
15	[[280 -127] 0]
<u>лев. ч</u>	[[-80 0] 0]
<u>пр. ч</u>	[[80 0] 0]
а	[1 3 67]
б	[4 21 68]
в	[3 14 69]

В процессе взвешивания система из двух чаш весов и грузов движется до тех пор, пока не займет одно из трех положений:

- 1) равновесия;
- 2) левая внизу, правая вверху;
- 3) левая вверху, правая внизу.

Движение черепашки до некоторого предела описывает рекурсивная процедура ДВИЖ:

```

to движ :предел
if y = :предел [stop]
ifelse y < :предел [fd 1] [bk 1]
движ :предел
end

```

Такое движение должна осуществлять система из 2, 3, 4 черепашек (в зависимости от наличия грузов), причем если третья и первая едут вниз, то вторая и нулевая — вверх (и наоборот). За точку отсчета примем положение третьей черепашки.

```

to движение :предел
ask 3 [make "P y]
if :P = :предел [замена_поз_чаш stop]
ifelse :P < :предел
[ask левые [fd 1] ask правые [bk 1]]
[ask левые [bk 1] ask правые [fd 1]]
движение :предел
end

```

```

to левые
ifelse занята "лев. ч [ор [1 3]] [ор [3]]
end

```

```

to правые
ifelse занята "пр. ч [ор [0 2]] [ор [2]]
end

```

```

to замена_поз_чаш
name list se -80 :предел last :лев. ч "лев. ч
name list se 80 -1 * :предел last :пр. ч "пр. ч
end

```

Процедура ЗАМЕНА_ПОЗ_ЧАШ переписывает после взвешивания координаты чаш.

Команда ASK передает указанному списку черепашек список инструкций.

Функции ЛЕВЫЕ и ПРАВЫЕ формируют список черепашек (если чаша весов занята, должны двигаться две черепашки, в противном случае — одна).

При взвешивании левая чаша весов двигается до верхнего положения, до положения равновесия или до нижнего положения, что зависит от соотношения грузов (ординаты положений приняты равными 20, 0 и -20):

```

to взвешивание
ifelse (груз "лев. ч) = груз "пр. ч
  [движение 0]
  [ifelse (груз "лев. ч) > груз "пр. ч
    [движение -20]
    [движение 20]]
end

```

```

to груз :чаша
ifelse занята :чаша
  [op item 2 thing last thing :чаша]
  [op 0]
end

```

Функция ГРУЗ выдает 0, если соответствующая чаша пустая, или вес шара.

Теперь можно описать процедуру переноса шара на весы. Она будет похожа на процедуру переноса на неподвижную подставку.

Во-первых, необходимо выявить случай невозможности положить шар на весы — если заняты обе чаши или производится попытка поместить на весы шар, который уже там лежит (если первый элемент списка для шара — "лев. ч или "пр. ч, т. е. не число).

Во-вторых, проанализировать, на какую чашу поместить шар, и активизировать соответствующую черепашку (если на левую — черепашку номер 1, если на правую — черепашку номер 0). Пусть в случае, если свободны обе чаши, груз помещается на левую.

```

to на_весы :Ш
ifelse or (and занята "лев. ч
  занята "пр. ч)
  (or (first thing :Ш) = "лев. ч
    (first thing :Ш) = "пр. ч)
  [ответ [не могу]]
  [ifelse занята "лев. ч
    [ask 0 [перенос :Ш "пр. ч]]
    [ask 1 [перенос :Ш "лев. ч]]]
end

```

Процедура ПЕРЕНОС уже описана, но в нее придется внести некоторые изменения, чтобы учесть случай переноса не только с подставки на подставку, но и с подставки на весы и с весов на подставку. Также придется подкорректировать старшую процедуру НА_ПОДСТАВКУ, чтобы она правильно работала в случае, когда переносимый шар лежит на весах. Окончательные варианты для процедур НА_ПОДСТАВКУ, ПЕРЕНОС и НОВОЕ_МЕСТО:

```

to на_подставку :Ш :П
ifelse занята :П
  [ответ [не могу]]
  [ifelse (first thing :Ш) = "лев. ч
    [ask 1 [перенос :Ш :П]]
    [ask 0 [перенос :Ш :П]]]
end

```

```

to перенос :Ш :П
старое_место
pu setpos first thing :П
новое_место
взвешивание
end
    
```

```

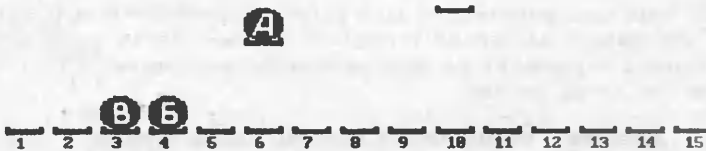
to новое_место
setc 14
if number? :П [pd stamp ht pu]
name (se :П bf thing :Ш) :Ш
name (se bf thing :П :Ш) :П
end
    
```

Внесены три изменения:

- 1) в процедуре НА_ПОДСТАВКУ перенос поручается черепашке номер 1, если надо переносить шар с левой чаши весов;
- 2) в процедуре ПЕРЕНОС добавлено взвешивание (в самом редко встречающемся случае — переносе с одной неподвижной подставки на другую — это действие лишнее, но и не мешает);
- 3) в процедуре НОВОЕ_МЕСТО отпечатывание черепашки на новом месте стало условным, так как если шар переносится на весы, отпечатывать его не нужно — он может начать двигаться.

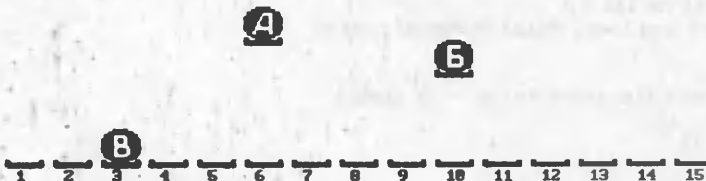
Теперь все процедуры увязаны друг с другом. Результаты выполнения еще двух команд в нашем примере:

НА_ВЕСЫ "a



имя	значение
1	[[-280 -127] 0]
2	[[-240 -127] 0]
3	[[-200 -127] a]
4	[[-160 -127] b]
5	[[-120 -127] 0]
...	...
15	[[280 -127] 0]
лев. ч	[[-80 -20] a]
пр. ч	[[80 20] 0]
a	[лев. ч 3 67]
b	[4 21 68]
a	[3 14 69]

НА_ВЕСЫ "b



имя	значение
1	[[-280 -127] 0]
2	[[-240 -127] 0]
3	[[-200 -127] в]
4	[[-160 -127] 0]
5	[[-120 -127] 0]
...	...
15	[[280 -127] 0]
лев. ч	[[-80 20] а]
пр. ч	[[80 -20] б]
а	[лев. ч 3 67]
б	[пр. ч 2 68]
в	[3 14 69]

Основные два вида команд исполнителя — перекладывание шаров на подставку и на весы — готовы. Осталась последняя команда «конец», которая распечатывает над каждым шаром его вес.

```
to конец
tell 3
pd stamp pu
ht setc 12 make "i 1
repeat :n
  [pu
  setpos first thing first thing char 159 + :i
  sety y + 40
  pd label item 2 thing char 159 + :i
  make "i :i + 1]
end
```

Первая строчка поручает выполнение этой работы черепашке номер 3. Вторая строка отпечатывает эту черепашку на экране (чтобы не пропала левая чаша весов). В цикле вторая строка помещает черепашку на позицию очередного шара, а третья и четвертая строки отпечатывают над ним его вес.

Анализ и выполнение команд исполнителя

Теперь наступил черед главной процедуры, которая должна принимать, анализировать и запускать команды на исполнение. Эта процедура должна в первую очередь выявлять команды, не входящие в состав команд исполнителя, что проще всего сделать, если заранее подготовить список всех команд (пусть он хранится в переменной СПИСОК_КОМАНД). Например, для трех шаров такой список состоит из 49 команд:

```
конец
а-весы  а-1  а-2  а-3  а-4  а-5  а-6  а-7  а-8  а-9
         а-10 а-11 а-12 а-13 а-14 а-15
б-весы  б-1  б-2  б-3  б-4  б-5  б-6  б-7  б-8  б-9
         б-10 б-11 б-12 б-13 б-14 б-15
в-весы  в-1  в-2  в-3  в-4  в-5  в-6  в-7  в-8  в-9
         в-10 в-11 в-12 в-13 в-14 в-15
```

Функция формирования списка всех возможных команд:

```
to команды :n
make "список [конец]
make "i 1
repeat :n
  [make "буква char 159 + :i
  make "список lput (word :буква "- "весы) :список
  make "j 1
  repeat 15
    [make "список lput (word :буква "- :j) :список
    make "j :j + 1]
  make "i :i + 1]
ор :список
end
```


После синтаксической проверки команды анализатор должен запустить для ее выполнения соответствующие процедуры, обновить на экране значение счетчика команд, стереть выполненную команду, ввести новую и начать цикл сначала:

```

to анализатор :команда
ifelse not member? :команда :список_команд
  [ответ [не понимаю] ]
  [if :команда = "конец [конец stop]
   ifelse (bf bf :команда) = "весы
    [на весы first :команда]
    [на подставку first :команда bf bf :команда]
   make "счетчик :счетчик + 1
   стереть_стр 5 insert :счетчик ]
стереть_стр 8
анализатор first readlist
end

to стереть_стр :i
top repeat :i - 1 [cd]
sol select eol cut tab
end

```

Заключительный этап работы — сборка всех процедур в единое целое:

```

to весы
rg ct
подставки ht
pr [Укажите число шаров (от 2 до 13):]
make "n first readlist
ct шары :n
make "список_команд команды :n
чаши
make "счетчик 1
repeat 4 [pr []] tab pr :счетчик
repeat 2 [pr []] tab
анализатор first readlist
end

```

Декорации — изображение весов, рамочки вокруг полей команды, счетчика ходов и реакций программы, подсказку о системе команд исполнителя и т. д. — вы можете сделать на свой вкус (программно или используя графредактор и первоначальную загрузку картинки).

Возможны варианты дальнейшего усовершенствования программы (они могут послужить заданиями для ваших учеников):

1. Написать процедуру перемещения шаров (не мгновенного, возможны различные виды движения); входные данные — координаты исходной и конечной позиций.
2. Написать процедуру для задания числа шаров, как в «Роботландии» (с помощью стрелок и перебора возможных значений, сменяющихся в экранном окне). Эта процедура будет похожа на МЕНЮ.
3. Осуществить запоминание списка введенных команд и начальных значений (числа шаров и их весов), чтобы можно было «прокрутить» ход решения еще раз.
4. Реализовать «откатку» (возможность «обратного» выполнения команд) и «накатку».



Кто же нас не знает?

ПРОГРАММНО-МЕТОДИЧЕСКИЕ СИСТЕМЫ ПО ИНФОРМАТИКЕ

КуМир-Гипертекст для IBM PC - Базовый комплект,
Расширенный (инструментальный) комплект,
Планиметрия, Функции и Графики, Информатика-Десять

Информатика для младшего возраста
для IBM PC - Конструктор сказок, Веселые картинки,
Классификаторы, ТОРТ, Раскрашка, Magic Land
Буквоед, Считалка, Бюро находок

КуМир для Корвета, Ямахи, УКНЦ
ФортранМир для УКНЦ

Нов!
КуМир-Гипертекст для Apple Macintosh

ФортранМир,
ПаскальМир для IBM PC

Нов! *МикроМир* для Windows
ТЕКСТОВЫЙ РЕДАКТОР+ЭЛЕКТРОННЫЕ ТАБЛИЦЫ

Заявка на приобретение программных средств

Мой адрес: _____

Я хочу приобрести: _____

Заявки направляйте по адресу: 103051, Москва,
Садовая-Сухаревская, д.16, к.9 "Информатика и образование"
Для получения полного каталога вложите конверт с маркой.

ЗАДАНИЯ ДЛЯ ВСТУПИТЕЛЬНЫХ ЭКЗАМЕНОВ ПО ИНФОРМАТИКЕ В НГУ

В предлагаемой вашему вниманию статье приведены задачи, предложенные абитуриентам на конкурсных испытаниях по информатике для групп «Системы информатики» и «Экономическая информатика» ВКИ НГУ в 1994 г. Варианты 1 и 2 — для групп абитуриентов первого потока, варианты 3 и 4 — второго потока. В статье указаны требования к знаниям по информатике, даны образцы решения задач, а также программа собеседования по экономике, литература для самостоятельной подготовки.

Для абитуриентов обоих потоков вступительный экзамен проходит в два этапа: письменная работа и собеседование. Цель собеседования — проверка способности абитуриентов к алгоритмическому мышлению. В процессе собеседования оценивается: степень подготовленности к обучению в колледже, точность изложения, оригинальность решения, настойчивость в поиске решения и опыт практической работы по программированию.

Требования к абитуриенту

Абитуриентам на собеседование по информатике предлагаются математически довольно простые задачи. Требуется только написать решение в виде алгоритма (последовательности действий).

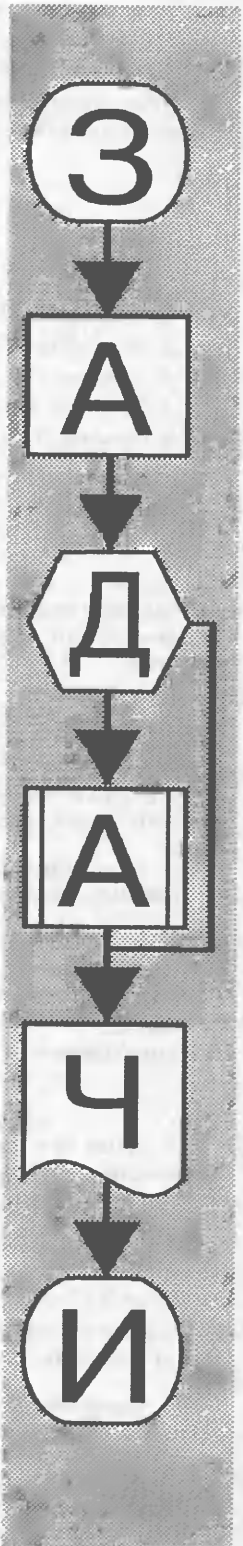
Подготовиться к собеседованию можно в школе на уроках информатики, на подготовительных курсах ВКИ НГУ или самостоятельно, изучив соответствующую литературу. Если вы хотите более основательно подготовиться к вступительным экзаменам, можно поступить в заочную, воскресную, вечернюю школу по информатике ВКИ НГУ или участвовать в работе летней школы юных программистов.

В письменной работе абитуриенту предлагались четыре задачи для групп «Системы информатики» и три задачи для групп «Экономическая информатика», расположенные в порядке нарастания сложности. Необходимо было описать алгоритм решения этих задач.

Абитуриенты, не знакомые с языками программирования, записывали алгоритмы в виде последовательности действий с комментариями, как показано в примерах решения задач.

Абитуриенты, имеющие навыки программирования, в качестве решения писали только программу, реализующую алгоритм, на любом известном им языке программирования (Бейсик, Паскаль, Фортран, Си).

Беседа с членами комиссии начиналась с проверки письменной работы и обсуждения ошибок. В процессе беседы абитуриент мог самостоятельно найти и исправить свои ошибки, что положительно влияло на общую оценку. По результатам первой части беседы выставлялась оценка за письменную работу.



Для выяснения подготовленности абитуриента к учебе предлагались дополнительные задачи разной степени сложности. Во время второй части беседы качеством своих ответов абитуриент мог повысить общую оценку за собеседование.

Абитуриенты групп «Экономическая информатика» проходили дополнительное собеседование по экономике.

Литература для подготовки к экзаменам

1. Светозарова Г. И., Мельников А. А., Козловский А. В. Практикум программирования на языке Бейсик. М.: Наука, 1988.
2. Персональный компьютер в играх и задачах. М.: Наука, 1988.
3. Косневски Ч. Занимательная математика и персональный компьютер. М.: Мир, 1987.
4. Абрамов С. А., Зима Е. В. Начала программирования на языке Паскаль. М.: Наука, 1987.
5. Этюды о персональных компьютерах: Пер. с венгерского. М.: Знание, 1988.
6. Абрамов В. Г., Трифонов Н. П., Трифонова Н. Г. Введение в языке Паскаль. М.: Наука, 1988.

Задания для групп «Системы информатики»

Вариант 1

1. Даны три натуральных числа A , B , C . Составьте алгоритм (или напишите программу), определяющий, образуют ли они арифметическую прогрессию и выдающую на печать ее разность.

2. Чтобы получить одну молекулу серной кислоты H_2SO_4 , нужно 2 атома водорода H , 1 атом серы S и 4 атома кислорода O . В химическом реакторе имеется A атомов водорода, B атомов кислорода и C атомов серы.

Составьте алгоритм (или напишите программу), вычисляющий, сколько молекул серной кислоты может получиться из такого количества атомов.

3. Даны натуральное число n и вещественное число X . Составьте алгоритм (или напишите программу), вычисляющий $A = X^{n^2} / 3^n$, не используя операцию возведения в степень.

4. Два прямоугольника, расположенные в первом квадранте, со сторонами, параллельными осям координат, заданы координатами своих левого верхнего и правого нижнего углов. Для первого прямоугольника это точки (x_1, y_1) и $(x_2, 0)$, для второго — (x_3, y_3) , $(x_4, 0)$.

Составьте алгоритм (или напишите программу), определяющий, пересекаются ли данные прямоугольники, и вычисляющий общую площадь, если они пересекаются.

Вариант 2

1. Даны три натуральных числа A , B , C . Составьте алгоритм (или напишите программу), определяющий, образуют ли они геометрическую прогрессию, и выдающий на печать ее знаменатель, если это прогрессия.

2. Для образования одной молекулы щелочи $Ca(OH)_2$ нужно 1 атом кальция Ca , 2 атома кислорода O и 2 атома водорода H . В химическом реакторе имеется A атомов кальция, B атомов водорода и C атомов кислорода.

Составить алгоритм (или написать программу), вычисляющий, сколько молекул щелочи может получиться из такого количества атомов.

3. Даны натуральное число n и вещественное число X . Составить алгоритм (или написать программу), вычисляющий $A = X^{n^2} / 2^n$, не используя операцию возведения в степень.

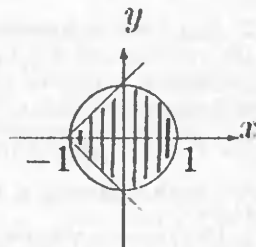
4. Два прямоугольника, расположенные в первом квадранте, со сторонами, параллельными осям координат, заданы координатами своих левого верхнего и правого нижнего углов. Для первого прямоугольника это точки $(0, y_1)$ и (x_2, y_2) , для второго — $(0, y_3)$, (x_4, y_4) .

Составьте алгоритм (или напишите программу), определяющий, пересекаются ли данные прямоугольники, и вычисляющий общую площадь.

Вариант 3

1. Составьте алгоритм (или напишите программу), который вычисляет сумму трех последних цифр произвольного шестизначного числа N .

2. Пусть D — заштрихованная часть плоскости, как указано на рисунке. Определить, принадлежит ли заданная точка с координатами (X_0, Y_0) области D .



3. Даны натуральное число n , действительное число X . Составьте алгоритм (или напишите программу), который вычисляет:

$$\text{tg}X + \text{tg}^2X + \dots + \text{tg}^n X$$

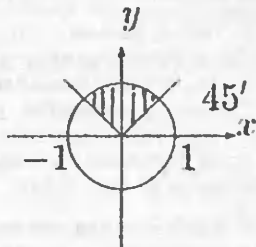
4. В N -этажном общежитии по 20 комнат на каждом этаже. Комнаты расположены по одну сторону коридора и пронумерованы на каждом этаже слева направо. Лестница находится слева.

Составьте алгоритм (или напишите программу), который по заданным натуральным номерам K и M дает оптимальный способ перехода из K -й комнаты в M -ю.

Вариант 4

1. Составьте алгоритм (или напишите программу), который вычисляет сумму первых трех цифр произвольного шестизначного числа N .

2. Пусть D — заштрихованная часть плоскости, как показано на рисунке. Определить, принадлежит ли заданная точка с координатами (X_0, Y_0) области D .



3. Даны натуральное число n , действительное число X . Составьте алгоритм (или напишите программу), который вычисляет:

$$\text{tg}X + \text{tg}X^2 + \dots + \text{tg}X^n$$

4. Составьте алгоритм или напишите программу для робота, управляющего лифтом. Условия: в небоскребе N этажей и всего один подъезд; на каждом этаже по 3 квартиры; лифт может останавливаться только на нечетных этажах.

Человек садится в лифт и набирает на клавиатуре номер нужной ему квартиры — M (клавиатура из цифр от 0 до 9). Лифт сам «вычисляет», как доставить пассажира к нужной квартире.

Задания для групп «Экономическая информатика»

Вариант 1

1. Дано действительное число A . Составьте алгоритм (или напишите программу), вычисляющий $f(A)$, если

$$f(x) = \begin{cases} 1 - \sin x & \text{при } -1 < x < 1 \\ 0 & \text{в остальных случаях} \end{cases}$$

2. Даны два треугольника ABC и DEF с координатами вершин: $(Ax, Ay), (Bx, By), (Cx, Cy), (Dx, Dy); (Ex, Ey), (Fx, Fy)$.

Составьте алгоритм (или напишите программу), определяющий, какой из треугольников имеет больший периметр.

3. У школьников было S руб. Бутылка с соком стоит P руб., пустая бутылка стоит B руб. Составьте алгоритм (или напишите программу), вычисляющий, сколько бутылок сока могут

выпить школьники, если они сдают пустые бутылки и на вырученные деньги покупают полные.

Вариант 2

1. Даны три действительных числа A, B, C . Составить алгоритм (или написать программу), выбирающий из заданных чисел те, которые больше 1, но меньше 3.

2. Даны два треугольника ABC и DEF с координатами вершин:

$(Ax, Ay), (Bx, By), (Cx, Cy); (Dx, Dy), (Ex, Ey), (Fx, Fy)$.

Составить алгоритм (или написать программу), определяющий, какой из треугольников имеет большую площадь. Использовать формулу Герона:

$$S = \sqrt{P(P-a)(P-b)(P-c)},$$

где P — полупериметр, a, b, c — длины сторон треугольника.

3. Покупатель должен заплатить в кассу S руб. У него имеются купюры достоинством 1, 5, 10, 50, 100, 1000 и 10000 руб.

Составьте алгоритм (или напишите программу), вычисляющий, сколько купюр разного достоинства отдаст покупатель, если он начинает платить с самых крупных купюр.

Вариант 3

1. У вас есть доллары. Вы хотите их обменять на рубли. Есть информация о стоимости купли-продажи долларов в банках города. В городе N банков.

Составьте алгоритм (или напишите программу), определяющий, какой банк выбрать, чтобы выгодно обменять доллары на рубли.

2. Перед входом в столовую огромная очередь ребят. Контролер проверяет у них чистоту рук. Если руки грязные, то отправляет мыть. Мытье рук занимает N секунд. Проверка чистоты рук — M секунд. Предполагается, что у каждого второго ученика грязные руки.

Составьте алгоритм (или напишите программу), определяющий, какое минимальное количество умывальников надо поставить перед входом в столовую, чтобы не было очереди. Как надо расположить умывальники, чтобы их количество не зависело от времени, за которое необходимо до них дойти.

3. Ежемесячная стипендия студента составляет A руб., а расходы на проживание превышают стипендию и составляют B руб. в месяц. Рост цен ежемесячно увеличивает расходы на 3%.

Составьте алгоритм (или напишите программу) расчета необходимой суммы денег, которую надо одновременно попросить у родителей, чтобы можно было прожить учебный год (10 месяцев), используя только эти деньги и стипендию.

Вариант 4

1. У вас есть рубли. Вы хотите их обменять на доллары. Есть информация о стоимости купли-продажи долларов в банках города. В городе N банков.

Составьте алгоритм (или напишите программу), определяющий, какой банк выбрать, чтобы выгодно обменять рубли на доллары.

2. Вы подъехали к автозаправочной станции и не пользуетесь льготами служебной машины. На автозаправочной станции имеется две бензоколонки. К каждой из них стоит очередь автомобилей. Заправка одного автомобиля занимает 3 минуты. В первой очереди — N автомобилей, во второй — M . Через каждые три автомобиля в первой очереди заправляется не стоявшая в очереди служебная машина.

Составьте алгоритм (или напишите программу) выбора бензоколонки, в которой очередь на заправку подойдет быстрее. Определите время, за которое подойдет ваша очередь.

3. У студента имеются накопления S руб. Ежемесячная стипендия составляет A руб., а расходы на проживание превышают стипендию и составляют B руб. в месяц. Рост цен ежемесячно увеличивает расходы на 3%.

Составьте алгоритм (или напишите программу) расчета времени (в месяцах), которое может прожить студент, используя только накопления и стипендию.

Подумайте, изменится ли алгоритм, если инфляция не постоянна.

Примеры оформления решений

Задача 1

Даны три натуральных числа A, B, C . Составьте алгоритм (или напишите программу), определяющий, образуют ли они арифметическую прогрессию, и выдающий на печать ее разность.

Предположим, что числа A, B, C стоят в арифметической прогрессии рядом.

Алгоритм

- 1) переменные $A, B, C, R1, R2$: тип — целый;
- 2) ввести A, B, C ; $B - A$
- 3) вычислить $R1 = A - B$;
- 4) вычислить $R2 = B - C$; $C - B$
- 5) если $R1 = R2$, то
6 = выдать на печать $R1$;
- иначе:
6 = выдать на печать: заданные числа не являются прогрессией.

Самостоятельно рассмотрите вариант, когда только A, B или B, C являются соседними членами прогрессии. Более сложная задача: когда A, B, C — произвольные натуральные числа. В этом случае задача может иметь несколько решений.

Задача 2

Чтобы получить одну молекулу серной кислоты H_2SO_4 , нужно 2 атома водорода H , 1 атом серы S и 4 атома кислорода O . В химическом реакторе имеется A атомов водорода, B атомов кислорода и C атомов серы.

Составьте алгоритм (или напишите программу), вычисляющий, сколько молекул серной кислоты может получиться из такого количества атомов.

Алгоритм

- 1) переменные $A, B, C, M1, M2, K$: тип — целый;
- 2) ввести A, B, C ;
- 3) вычислить целую часть от деления A на 2:
6 = $M1 = \text{целая часть } (A/2)$;
- 4) вычислить целую часть от деления B на 4:
6 = $M2 = \text{целая часть } (B/4)$;
- 5) найти минимальное из трех чисел $C, M1, M2$:
6 = $K = \min(C, M1, M2)$;
- 6) выдать на печать: в реакторе образуется K молекул кислоты.

Алгоритм нахождения минимального из тех заданных чисел напишите самостоятельно. Обобщите алгоритм для нахождения минимального из N заданных чисел.

Задача 3

Даны натуральное число n и вещественное число X . Составьте алгоритм (или напишите программу), вычисляющий $A = X^{n^3} / 3^n$, не используя операцию возведения в степень.

Алгоритм

- 1) переменная $n, n3$: тип — целый;
- 2) переменная X, A : тип — вещественный;
- 3) задать $A1 = 1$;
- 4) задать $A2 = 1$;
- 5) вычислить $n^3 = n \times n \times n$;
- 6) в цикле от 1 до n
6 = вычислить $A1 = A1 \times 3$
конец цикла;
- 7) в цикле от 1 до 3
6 = вычислить $A2 = A2 \times X$
конец цикла;

$$A = 1$$

$$A = (A \cdot X \cdot X \cdot X) / 3$$

8) вычислить $A=A2/A1$;

9) выдать на печать A .

Примечание. Обратите внимание на порядок вычисления при возведении в степень $A2=X^n^3$. Сначала n возводится в степень 3, а потом X в степень n^3 .

Задача 4

Два прямоугольника, расположенные в первом квадранте, со сторонами, параллельными осям координат, заданы координатами своих левого верхнего и правого нижнего углов. Для первого прямоугольника это точки $(x1, y1)$ и $(x2, 0)$, для второго — $(x3, y3)$, $(x4, 0)$.

Составьте алгоритм (или напишите программу), определяющий, пересекаются ли данные прямоугольники, и вычисляющий общую площадь, если они пересекаются.

Алгоритм

Нарисуйте первый квадрант декартовой системы координат и пересекающиеся прямоугольники. Обозначьте левую верхнюю точку пересечения (a, b) , правую нижнюю точку (лежащую на оси X) — (c, d) .

1) переменные $x1, y1, x2, x3, y3, x4, a, b, c, d, S$: тип — вещественный;

2) вычислить максимальное из чисел $x1$ и $x3$. Это координата по оси X левой стороны общего прямоугольника:

$b = a = \max(x1, x3)$;

3) вычислить минимальное из чисел $x2$ и $x4$, это координата по оси X правой стороны общего прямоугольника:

$b = c = \min(x2, x4)$;

4) Если $a > c$, то:

$b =$ выдать на печать: прямоугольники не пересекаются;

$b =$ выдать на печать: решение задачи прекращается;

$b =$ стоп;

иначе:

$b =$ присвоить $d=0$ (по условию задачи);

$b =$ вычислить минимальное из чисел $y1$ и $y3$. Это высота общего прямоугольника.

$b = d = \min(y1, y3)$;

$b =$ вычислить общую площадь $S=(c-a) \times b$;

5) выдать на печать S .

Задача 5

Дано действительное число A . Составьте алгоритм (или напишите программу), вычисляющий $f(A)$, если

$$f(x) = \begin{cases} 1 - \sin x & \text{при } -1 < x < 1 \\ 0 & \text{в остальных случаях} \end{cases}$$

Алгоритм

1) переменная A, FA : тип — вещественный;

2) ввести A ;

3) если $A > -1$ и $A < 1$ то

$b = FA = 1 - \sin A$;

иначе:

$b = FA = 0$;

4) вывести на печать FA .

Задача 6

Даны два треугольника ABC и DEF с координатами вершин:

$(Ax, Ay), (Bx, By), (Cx, Cy), (Dx, Dy), (Ex, Ey), (Fx, Fy)$.

Составьте алгоритм (или напишите программу), определяющий, какой из треугольников имеет больший периметр.

Алгоритм

Чтобы решить эту задачу, нужно знать, как вычислить длину отрезка, если заданы декартовы координаты его концов:

1) переменные $A_x, A_y, B_x, B_y, C_x, C_y, D_x, D_y, E_x, E_y, F_x, F_y, AB, BC, AC, DE, EF, DF, P_1, P_2$: тип — вещественный;

2) ввести $A_x, A_y, B_x, B_y, C_x, C_y, D_x, D_y, E_x, E_y, F_x, F_y$;

3) вычислить длины сторон треугольников:

$$AB = \sqrt{(B_x - A_x)^2 + (B_y - A_y)^2},$$

$$BC = \sqrt{(C_x - B_x)^2 + (C_y - B_y)^2},$$

$$AC = \sqrt{(C_x - A_x)^2 + (C_y - A_y)^2},$$

$$DE = \sqrt{(E_x - D_x)^2 + (E_y - D_y)^2},$$

$$EF = \sqrt{(F_x - E_x)^2 + (F_y - E_y)^2},$$

$$DF = \sqrt{(F_x - D_x)^2 + (F_y - D_y)^2},$$

4) вычислить периметры треугольников:

$$P_1 = AB + BC + AC;$$

$$P_2 = DE + EF + DF;$$

5) если $P_1 > P_2$, то

6 = выдать на печать: треугольник ABC имеет больший периметр;
иначе:

6 = выдать на печать: треугольник DEF имеет больший периметр.

Задача 7

У школьников было S руб. Бутылка с соком стоит P руб., пустая бутылка стоит B руб. Составьте алгоритм (или напишите программу), вычисляющий, сколько бутылок сока могут выпить школьники, если они сдают пустые бутылки и на вырученные деньги покупают полные.

Алгоритм

1) переменные S, P, B, S_1, S_2 : тип — вещественный;

2) переменная k_1, k_2 : тип — целый;

3) ввести S, P, B ;

4) присвоить $k_2 = 0$; это счетчик количества выпитых бутылок сока;

5) пока $S < -P$ делать 6—10;

6 = 6) вычислить, сколько бутылок сока можно купить на S руб.:

$$k_1 = \text{целая часть}(S/P);$$

6 = 7) изменить значение счетчика:

$$k_2 = k_2 + k_1;$$

6 = 8) вычислить, сколько денег осталось после покупки сока:

$$S_1 = S - k_1 \times P;$$

6 = 9) вычислить, сколько денег получено после сдачи пустых бутылок:

$$S_2 = k_1 \times B;$$

6 = 10) вычислить новую сумму денег:

$$S = S_1 + S_2;$$

конец цикла;

11) выдать на печать k_2 .

Программа собеседования по экономике

Экономические науки. Задачи, которые они решают.

Потребности человека: понятия, виды, границы. Экономические ресурсы общества, проблемы их ограниченности. Типы экономических систем, различающихся организацией использования ресурсов: централизованное, рыночное.

Рынок, элементы рынка: фирмы, домохозяйство, рынок товаров, рынок ресурсов. Товар,

его свойства. Спрос на товар. Предложение товара. Доходы фирмы, издержки и прибыль. Конкуренция продавцов, конкуренция покупателей. Цена товара.

Заработная плата. Реальная заработная плата. Безработица и ее формы: нормальная, структурная, застойная, скрытая.

Деньги и их функции: мера стоимости, средство обращения, средство накопления. Денежное обращение, его регулирование. Инфляция: понятие, причины, типы (инфляция спроса, инфляция издержек). Гиперинфляция. Валюта. Валютный курс.

Экономика страны, ее структура.

Формы собственности: государственная, муниципальная, частная, собственность общественных организаций. Основные организационно-правовые формы предприятий: государственная, муниципальная, индивидуальная, акционерные общества закрытого типа (товарищество с ограниченной ответственностью), акционерные общества открытого типа. Акции: простые, привилегированные, именные. Дивиденды.

Другие виды ценных бумаг: облигации, векселя, сертификаты и др. Доходность по ним.

Показатели экономического развития государства. Валовой национальный продукт. Национальный доход. Национальное богатство. Производительность труда. Инвестиции: валовые, чистые. Уровень жизни населения.

Мировое хозяйство. Международное разделение труда. Мировая торговля. Политика государства в мировой торговле. Торговые барьеры. Общее представление о платёжном балансе страны.

Экономическая интеграция стран, существующие группировки.

Литература

1. Липсин И. Экономика без тайн. М.: Дело, 1993.
2. Кемпбел Р. Макконнелл, Стезлунд Л. Брю. Экономикс. М.: Республика, 1992.
3. Закон о собственности в РСФСР.
4. Закон Российской Федерации (о предприятиях и предпринимательской деятельности).



РЕСПУБЛИКАНСКИЙ ЦЕНТР ИНТЕРАКТИВНЫХ СРЕДСТВ ОБУЧЕНИЯ

Телефоны: (095) 155 8737, 155 8747

Факс: (095) 155 8727

Сборник видеофильмов

"Основы информатики и вычислительной техники"



- Основы теории информации
- Первоначальные сведения об информации
- Устройство ЭВМ
- Первоначальные сведения об ЭВМ
- История развития вычислительной техники
- Компьютерные средства в учебном процессе

Сборник поставляется на 90 минутной кассете VHS

Цена сборника, включая НДС

\$10

а также более 70 других наименований видеофильмов по различным предметам.

Оплата в рублях по текущему курсу ММВБ.

По желанию заказчика кассета может быть выслана почтой

Стоимость пересылки без возмещения убытков при пропаже бандероли
с возмещением стоимости

3000 рублей
15000 рублей

РЕКВИЗИТЫ ЦЕНТРА

Адрес: 125315, г. Москва, ул. Часовая, 21-б

Банковские реквизиты для Москвы: р/с 1609325 в Ленинградском отделении МББ, МФО 201694.

Для других регионов: кор. счет 48616100 в РКЦ ГУ ЦБ РФ, МФО 201791

В. М. Нечаев

ОДНА ЗАДАЧА — ТРИ ЯЗЫКА

Предлагаемая статья — вторая из серии статей, объединенных общим названием и предназначенных для демонстрации стиля программирования, предполагающего разработку структурированных программ. Первую статью этой серии см.: Информатика и образование. 1995. № 1.

Задача «Точки на плоскости»

На координатной плоскости в произвольном порядке располагается некоторое количество точек. Требуется определить две точки, ближайšie друг к другу.

Простая программа на Бейсике

```

10 CLS
20 INPUT "Количество точек —>"; n
30 DIM x(n), y(n)
40 FOR i=1 TO n
50 x(i)=INT(RND(1)*100)
60 y(i)=INT(RND(1)*100)
70 PRINT "x("; i; ")="; x(i), "y("; i; ")="; y(i)
80 NEXT i
90 l=1 : h=2 : b=SQR((x(1)-x(2))^2+(y(1)-y(2))^2)
100 FOR i=1 TO n-1
110 FOR j=i+1 TO n
120 r=SQR((x(i)-x(j))^2+(y(i)-y(j))^2)
130 IF r<b THEN b=r : l=i : h=j
140 NEXT j
150 NEXT i
160 PRINT "Ближайшие друг к другу точки: "; l; " и "; h
170 PRINT "Расстояние между ними: "; INT(b*100)/100

```

Тоже Бейсик, но структурно

```

10 REM Программа points .....
20 REM Из множества точек на плоскости находит две ближайšie
30 REM Переменные: n — количество точек
40 REM i, j — номера текущей пары; l, h — номера базисной пары
50 REM r, b — текущее и базисное расстояния
60 REM x(n), y(n) — массивы координат

100 CLS : REM Основная программа .....
110 GOSUB 200 : REM Вводим координаты точек
120 GOSUB 300 : REM Даем базису начальное значение
130 FOR i=1 TO n-1 : REM Начинаем перебирать точки
140 FOR j=i+1 TO n : REM по парам
150 GOSUB 400 : REM Находим расстояние для текущей пары
160 GOSUB 500 : REM Обновляем, если требуется, базис
170 NEXT j : NEXT i : REM Конец перебора
180 GOSUB 600 : REM Выводим ответ
190 END : REM Конец программы .....

200 REM Подпрограмма ввода координат
210 INPUT "Количество точек —>"; n
220 FOR i=1 TO n
230 x(i)=INT(RND(1)*100)
240 y(i)=INT(RND(1)*100)
250 PRINT "x("; i; ")="; x(i), "y("; i; ")="; y(i)
260 NEXT i
290 RETURN : REM Конец подпрограммы .....

```

```

300 REM Подпрограмма начального значения базиса
310 l=1 : h=2 : i=1 : j=2
320 GOSUB 400 : REM Находим базисное расстояние
330 b=r
390 RETURN : REM Конец подпрограммы .....

400 REM Подпрограмма нахождения расстояния для пары точек
410 r= SQR((x(i)-x(j))^2+(y(i)-y(j))^2)
490 RETURN : REM Конец подпрограммы .....

500 REM Подпрограмма обновления базиса
510 IF r<b THEN l=i : h=j : b=r
590 RETURN : REM Конец подпрограммы .....

600 REM Подпрограмма вывода ответа
610 PRINT "Ближайшие друг к другу точки: "; l; " и "; h
620 PRINT "Расстояние между ними: "; INT(b*100)/100
690 RETURN : REM Конец подпрограммы .....

```

Программа на QBasic

```

' Программа points. Среди точек на плоскости находит две ближайшие
' Объявляем имена процедур и функций .....
DECLARE SUB coord ( ) ' Процедура ввода координат
DECLARE FUNCTION distance (i, j) ' Функция расстояния
DECLARE SUB init ( ) ' Процедура начального значения базиса
DECLARE SUB basis (r, b) ' Процедура обновления базиса
DECLARE SUB result (l, h, b) ' Процедура вывода ответа

' Объявляем имена переменных .....
DIM SHARED n, i, j, l, h, r, b
' n — количество точек
' i, j — номера текущей пары
' l, h — номера базисной пары
' r, b — текущее и базисное расстояние между точками
' Массивы координат точек

DIM SHARED x(1 TO 255), y(1 TO 255)

' Начинаем программные действия .....
CLS
CALL coord ' Вводим координаты точек
CALL init ' Даем базису начальное значение
FOR i=1 TO n-1 ' Начинаем перебирать точки
FOR j=i+1 TO n ' по парам
r=distance(i, j) ' Находим расстояние для текущей пары
CALL basis(r, b) ' Обновляем, если потребуется, базис
NEXT j
NEXT i ' Конец перебора
CALL result(l, h, b) ' Выводим ответ
END ' Конец программы .....

' Описываем работу процедур и функций

SUB coord ' Процедура ввода координат
INPUT "Количество точек —>", n
RANDOMIZE TIMER
FOR i=1 TO n
x(i)=INT(RND*100)
y(i)=INT(RND*100)
PRINT "x("; i; ")="; x(i), "y("; i; ")="; y(i)
NEXT i
END SUB ' Конец процедуры .....

FUNCTION distance (i, j) ' Функция расстояния между парой точек
distance=SQR((x(i)-x(j))^2+(y(i)-y(j))^2)
END FUNCTION ' Конец функции .....

SUB init ' Процедура начального значения базиса
l=1 : h=2 : b=distance(l, 2)
END SUB ' Конец процедуры .....

SUB basis (r, b) ' Процедура обновления базиса

```

```

IF r<b THEN
  b-r : l=i : h=j
END IF
END SUB
SUB result (l, h, b)
  PRINT "Ближайшие друг к другу точки: "; l; " и "; h
  PRINT "Расстояние между ними: "; INT(b*100)/100
END SUB

```

' Если текущее расстояние меньше базисного,
' то впредь считать базисным его

' Конец процедуры

' Процедура вывода ответа

' Конец процедуры

Программа на TurboPascal

```

PROGRAM points;
USES CRT;
{ Объявляем имена переменных ..... }
VAR n: BYTE;
    i, j, l, h: BYTE;
    r, b: REAL;
    x, y: ARRAY [BYTE] OF BYTE;
{ Объявляем процедуры и функции и описываем их работу ..... }

PROCEDURE coord;
BEGIN
  WRITE('Количество точек —> '); READLN(n);
  RANDOMIZE;
  FOR i:=1 TO n DO
    BEGIN
      x[i]:=TRUNC(RANDOM(100));
      y[i]:=TRUNC(RANDOM(100));
      WRITELN('x(, i, ')', x[i]:2, 'y(, i, ')', y[i]:2);
    END;
END;

FUNCTION distance(i, j: BYTE): REAL;
BEGIN
  distance:=SQRT(SQR(x[i]-x[j])+SQR(y[i]-y[j]));
END;

PROCEDURE init;
BEGIN
  l:=1; h:=2; b:=distance(l, 2);
END;

PROCEDURE basis(r: REAL; var b: REAL);
BEGIN
  IF r<b THEN
    BEGIN
      b:=r; l:=i; h:=j;
    END;
END;

PROCEDURE result(l, h: BYTE; b: REAL);
BEGIN
  WRITELN('Ближайшие друг к другу точки: ', l, ' и ', h);
  WRITELN('Расстояние между ними: ', b:4:2);
END;

{ Начинаем программные действия }

BEGIN
  ClrScr;
  coord;
  init;
  FOR i:=1 TO n-1 DO
    FOR j:=i+1 TO n DO
      BEGIN
        r:=distance(i, j);
        basis(r, b);
      END;
    result(l, h, b);
  END.

```

{ Среди точек на плоскости находит две ближайшие }

{ Количество точек; не более 255 }

{ Номера текущей (i, j) и базисной (l, h) пар точек }

{ Текущее и базисное расстояния между точками }

{ Массивы координат }

{ Процедура ввода координат }

{ Начало цикла }

{ Конец цикла }

{ Конец процедуры

{ Функция расстояния между точками }

{ Конец функции

{ Процедура начального значения базиса }

{ Конец процедуры

{ Процедура обновления базиса }

{ Если текущее расстояние меньше базисного, }

{ то впредь считать базисным его }

{ Конец оператора условия }

{ Конец процедуры

{ Процедура вывода ответа }

{ Конец процедуры

{ Вводим координаты точек }

{ Даем базису начальное значение }

{ Начинаем перебирать точки }

{ по парам }

{ Находим расстояние для текущей пары }

{ Обновляем, если потребуется, базис }

{ Конец перебора }

{ Выводим ответ }

{ Конец программы }

Алгоритм решения задачи

Прежде всего, пронумеровав все точки, возьмем первую и определим расстояние от нее до второй. Примем этот результат за основу для дальнейших сравнений.

Затем продолжим перебор всех точек, определяя расстояния от первой до третьей, до четвертой и т. д. до последней. Каждый новый результат будем сравнивать с тем, который принят за основу. Если он окажется меньше, то основным (базисным) будем считать уже его.

Примечание: под базисом будем понимать не только расстояние между какими-либо точками, но и их номера.

Таким образом, после окончания перебора мы будем знать, какая точка ближе всего расположена к первой и какое расстояние их разделяет.

Теперь повторим этот же процесс по отношению ко второй точке, т. е. найдем расстояние от нее до третьей, четвертой и т. д., опять-таки сравнивая каждый результат с базисным. То же самое сделаем и для всех остальных точек; в итоге в качестве основы будем иметь расстояние между парой ближайших соседей. Их номера и являются ответом на поставленную задачу.

Итак, мы имеем два вложенных друг в друга цикла.

БАЗИС (Определяем начальное значение базиса)

НАЧАЛО ВНЕШНЕГО ЦИКЛА (Берем очередную точку с номером i)

НАЧАЛО ВНУТРЕННЕГО ЦИКЛА (Берем очередную точку с номером $j > i$)

РАССТОЯНИЕ (Определяем расстояние между точками с номерами i и j)

СРАВНЕНИЕ (Сравниваем полученный результат с базисным)

ОБНОВЛЕНИЕ (Изменяем, если нужно, значение базиса)

КОНЕЦ ВНУТРЕННЕГО ЦИКЛА (по точкам с номером j)

КОНЕЦ ВНЕШНЕГО ЦИКЛА (по точкам с номером i)

Как видно, главным действием внутри циклов является определение расстояния между двумя точками по их заданным координатам. Поэтому целесообразно выделить эту операцию в отдельную подпрограмму. Оформить ее удобно в виде ФУНКЦИИ, которая в качестве аргументов имела бы номера каких-либо двух точек и выдавала бы расстояние между ними. Дадим ей имя **distance**, а механизм ее работы опишем позже, в соответствующем месте.

Два других действия — **СРАВНЕНИЕ** и **ОБНОВЛЕНИЕ** — тоже выделим в отдельную подпрограмму. Ее удобнее оформить в виде ПРОЦЕДУРЫ. Дадим ей имя **basis** и опять-таки опишем механизм ее работы. В качестве параметров ей будут передаваться только что полученное (текущее) расстояние между точками очередной пары и базисное расстояние. Произведя над ними операцию сравнения, процедура вернет в основную программу обновленное, если потребуется, значение базиса.

Задание базису начального значения также оформим в виде ПРОЦЕДУРЫ. Для этого дадим ей имя **init** и опишем действия, входящие в нее. Эта процедура должна возвратить в основную программу начальное значение базиса.

Кроме того, в программе должен еще присутствовать блок ввода исходных данных, а именно координат точек. Этот блок независим от остальной программы, и поэтому его также следует выделить в отдельную подпрограмму. Оформить ее удобно в виде ПРОЦЕДУРЫ с именем **coord**. Параметров, передаваемых ей из основной программы, не будет, однако сама она должна возвращать значения координат.

Ну и, наконец, вывод полученного результата тоже представим в виде ПРОЦЕДУРЫ с именем **result**. В качестве параметров ей будет передаваться окончательное значение базиса.

ПРОГРАММА **points**

ВВОД КОЛИЧЕСТВА ТОЧЕК **coord** (Процедура **coord** — ввод координат точек)

БАЗИС **init** (Процедура **init** — начальное значение базиса $l=1$, $h=2$, $b=distance(1, 2)$)

НАЧАЛО ВНЕШНЕГО ЦИКЛА (Берем очередную точку с номером i)

НАЧАЛО ВНУТРЕННЕГО ЦИКЛА (Берем очередную точку с номером $j > i$)

РАССТОЯНИЕ $r=distance(i, j)$ (Находим расстояние между точками с номерами i и j)

СРАВНЕНИЕ **basis**(r, b) (Процедура **basis** — обновление базиса)

КОНЕЦ ВНУТРЕННЕГО ЦИКЛА (по точкам с номером j)

КОНЕЦ ВНЕШНЕГО ЦИКЛА (по точкам с номером i)

ВЫВОД РЕЗУЛЬТАТА **result**(l, h, b) (Процедура **result** — вывод конечного значения базиса)

КОНЕЦ ОСНОВНОЙ ПРОГРАММЫ

ПОДПРОГРАММА **coord** (Описание процедуры **coord**)

ВВОД КОЛИЧЕСТВА ТОЧЕК

НАЧАЛО ЦИКЛА ПО ТОЧКАМ
ВВОД КООРДИНАТ ОЧЕРЕДНОЙ ТОЧКИ
КОНЕЦ ЦИКЛА

ПОДПРОГРАММА distance(i,j) (Описание функции distance)
 $\text{distance}(i,j) = \text{SQR}\{[x(i) - x(j)]^2 + [y(i) - y(j)]^2\}$

ПОДПРОГРАММА init (Описание процедуры init)
 $l=1, h=2, b=\text{distance}(1, 2)$

ПОДПРОГРАММА basis(r, b) (Описание процедуры basis)
СРАВНЕНИЕ (текущего расстояния с базисным)
ЕСЛИ r TO
ОБНОВЛЕНИЕ (Новые значения для основы)
 $b=r, l=i, h=j$

ПОДПРОГРАММА result(l, h, b) (Описание процедуры result)
ВЫВОД (конечного значения базиса)

Описываемые функции и процедуры могут, естественно, содержать в себе ссылки на стандартные функции и процедуры, уже встроенные в транслятор, такие, как, например, SQR — функция, вычисляющая квадратный корень из какого-либо числа.

Разбитая на отдельные процедуры программа называется СТРУКТУРИРОВАННОЙ. Она имеет перед неструктурированной два существенных преимущества.

Во-первых, ее легко можно трансформировать применительно к изменяющимся условиям задачи. Например, если потребуется находить не самые близкие, а, наоборот, самые удаленные друг от друга точки, то изменить надо будет лишь одну подпрограмму basis, не затрагивая всю остальную программу. Либо если условия задачи изменятся так, что точки располагаются не на плоскости, а в пространстве, то трансформации будут подлежать только лишь подпрограммы distance и coord; все остальное останется в прежнем виде.

Во-вторых, структурированная программа легче просматривается на глаз. Видна логика в последовательности действий основной программы, ее не загромождают технические подробности вспомогательных операций, поскольку каждая из них описывается в отдельной подпрограмме. Если бы программа была не структурирована, то основные и вспомогательные действия были бы перемешаны и логику их последовательности было бы трудно проследить, не пользуясь карандашом и бумагой. Значение этого фактора особенно возрастает для больших по объему программ.

Комментарии к тексту программы на QBasic

```
DECLARE SUB coord ( )
DECLARE FUNCTION distance(i, j)
DECLARE SUB init( )
DECLARE SUB basis(r, b)
DECLARE SUB result(l, h, b)
```

В первых пяти строчках мы объявляем имена четырех процедур и одной функции, а также параметры, с помощью которых будет осуществляться передача значений между ними и основной программой.

Так, процедура coord не имеет передаваемых ей параметров, в ней просто будет выполняться ввод координат точек.

Функция distance имеет два аргумента, в качестве которых выступают номера пары точек. По координатам точек функция будет вычислять расстояние, их разделяющее.

Процедура init не имеет передаваемых ей параметров, в ней выполняется присваивание базису начального значения.

Процедура basis будет получать от основной программы текущее расстояние r между очередными двумя точками и базисное расстояние b, принятое за основу для сравнения. Произведя с ними некоторые действия, процедура должна вернуть в основную программу обновленное, если потребуется, значение базиса.

Процедура `result` имеет в качестве параметров, передаваемых ей из основной программы, конечные значения трех переменных, представляющих собою базис.

Пока что указанные процедуры и функция только лишь объявлены. Их работа (т. е. те действия, которые они выполняют) еще должна быть описана. В QBasic эти описания помещаются вслед за текстом основной программы.

```
DIM SHARED n, i, j, l, h, r, b
DIM SHARED x(1 TO 255), y(1 TO 255)
```

В этих двух строках объявляются переменные, используемые в программе. Ими будут шесть простых числовых величин n, \dots, b и два одномерных числовых массива x и y . Для массивов в скобках указаны значения, которые могут принимать их индексы.

Существенным здесь является не только служебное слово `DIM`, собственно и предназначенное для объявления переменных, но и следующее за ним слово `SHARED`. Оно придает этим переменным статус «разделяемых» (буквальный перевод с английского). Это означает, что доступ к ним разрешен всем процедурам и функциям программы. Например, если в процедуре `coord` фигурирует переменная n , то начальное значение переменной в процедуре будет равно тому, которое переменная имеет в основной программе в момент вызова процедуры. Если бы после `DIM` не стояло слово `SHARED`, то внутри процедуры значение n считалось бы поначалу равным нулю.

```
CALL coord
CALL init
```

Вызов (буквальный перевод служебного слова `CALL`) на исполнение сначала процедуры `coord`, а затем процедуры `init`.

```
FOR i=1 TO n-1
  FOR j=i+1 TO n
    (ТЕЛО ЦИКЛА)
  NEXT j,
NEXT i
```

Два вложенных друг в друга цикла: внешний со счетчиком i и внутренний со счетчиком j , в тело которого входят два действия. Первое:

```
r=distance(i, j)
```

Переменной r присваивается значение, вырабатываемое функцией `distance`, в качестве аргументов которой выступают значения переменных i и j . Иначе говоря, r — это расстояние между точками с номерами i и j (текущее расстояние).

Второе действие:

```
CALL basis(r, b)
```

Вызывается на исполнение процедура `basis`. При этом ей передаются из основной программы два параметра, а именно только что вычисленное с помощью функции `distance` значение текущего расстояния r и базисное расстояние b . Сравнив их, процедура `basis` должна вернуть в основную программу обновленное, если потребуется, значение базиса.

```
CALL result(l, h, b)
```

Вызывается на исполнение процедура `result`. При этом ей передается из основной программы конечное значение базиса.

```
END
```

На этом основная программа заканчивается, и далее следует описание объявленных процедур и функций. Каждое такое описание начинается со служебного слова `SUB` или `FUNCTION` с указанием имени описываемой процедуры или функции, а заканчивается служебными словами `END SUB` или `END FUNCTION`. Внутри же этих границ перечисляются подлежащие выполнению действия.

```
SUB coord
```

Описание процедуры `coord`.

```
INPUT "Количество точек —>", n
```


Ввод с клавиатуры некоторого числового значения, которое запоминается в переменной n . Чтобы ввод не был «слепым», ему предшествует текстовое сообщение, помещенное в кавычках.

RANDOMIZE TIMER

Включение встроенного генератора случайных чисел из интервала от 0 до 1 (RANDOMIZE), причем генератор включается таким образом, чтобы его работа началась с некоторого стартового положения, которое определялось бы значением системного счетчика времени (TIMER) компьютера в момент обработки транслятором этой строки. Поскольку этот момент может быть каким угодно, случайность вырабатываемых генератором чисел существенно возрастает.

```
FOR i=1 TO n
  (ТЕЛО ЦИКЛА)
NEXT i
```

Оператор цикла со счетчиком i . Тело цикла содержит три действия. Первые два почти одинаковы:

```
x(i)=INT(RND*100)
y(i)=INT(RND*100)
```

Очередным элементам массивов x и y присваиваются значения встроенной функции RND, округленные с помощью встроенной же функции INT. Функция RND осуществляет обращение к уже включенному выше (оператором RANDOMIZE) генератору случайных чисел.

```
PRINT "x(" ; i ; ")=" ; x(i), "y(" ; i ; ")=" ; y(i)
```

Вывод полученных координат для просмотра.

```
END SUB
```

Конец описания первой процедуры.

```
FUNCTION distance(i, j)
```

Описание функции distance.

```
distance=SQR((x(i)-x(j))^2+(y(i)-y(j))^2)
```

Функции distance присваивается значение, вычисленное с помощью встроенной функции SQR, которая находит квадратный корень из выражения, стоящего в скобках.

```
END FUNCTION
```

Конец описания функции.

```
SUB init
```

Описание процедуры init.

```
l=1: h=2: b=distance(1, 2)
```

В качестве начального значения базиса берутся первая и вторая точки и расстояние между ними.

```
END SUB
```

Конец описания второй процедуры.

```
SUB basis(r, b)
```

Описание процедуры basis.

```
IF r<b THEN
  b-r: l-i: h-j
END IF
```

Блок условного оператора с одной только ветвью THEN. Конец блока отмечен служебными словами END IF. Они могли бы быть опущены, если бы внутри блока было только одно

действие. Но их там три (хотя они и записаны в одну строчку); это простые присваивания, перечисленные через двоеточие. Значит, конец блока должен быть помечен обязательно.

END SUB

Конец описания третьей процедуры.

SUB result(l, h, b)

Описание процедуры result.

PRINT "Ближайшие друг к другу точки: "; l; " и "; h

PRINT "Расстояние между ними: "; INT(b*100)/100

Вывод на экран результатов работы программы — номеров l и h ближайших друг к другу точек и расстояния b между ними. При этом расстояние округляется до двух цифр после запятой с помощью встроенной функции INT.

END SUB

Конец описания четвертой и последней процедуры.

Комментарии к тексту программы на TurboPascal

PROGRAM points;

Объявляется имя программы. Эта строка не является обязательной. Обычно в этой же строке (или в нескольких последующих) программист также указывает в качестве комментария краткое содержание программы.

USES CRT;

Объявляется имя используемого в программе модуля CRT, в котором содержатся стандартные функции и процедуры для работы с монитором и клавиатурой.

VAR n: BYTE;

i, j, l, h: BYTE;

r, b: REAL;

x, y: ARRAY [BYTE] OF BYTE;

Объявляются имена фигурирующих в программе переменных с указанием их типов, причем имя от типа отделяется двоеточием. Тип переменной обязательно (в отличие от Бейсика) должен быть указан.

Переменные n, i, j, l, h — это целые числа от 0 до 255, на что указывает их тип BYTE. По разным строчкам они разнесены по принципу смысловой родственности. Такое разделение необязательно, все переменные одного типа могли быть записаны в одну строку через запятые, однако это уменьшило бы наглядность.

Примечание. В отличие от QBasic в Pascal блок объявления переменных обязательно должен присутствовать в тексте.

Переменные r и b — это вещественные числа (REAL), поскольку расстояние между точками может оказаться не целым. Более того, даже независимо от того, каким оно будет, при его нахождении вычисляется квадратный корень, что в Pascal автоматически предполагает вещественность результата.

Наконец, переменные x и y — это два массива (ARRAY) целых чисел (OF BYTE). То же служебное слово BYTE, стоящее в квадратных скобках, указывает на то, что индекс элементов массива пробегает целочисленные значения от 0 до 255.

Все перечисленные переменные считаются глобальными, т. е. доступными из любой подпрограммы.

На этом описание переменных заканчивается, и мы переходим к процедурам и функциям, конструируемым пользователем. Описание каждой из них начинается служебным словом PROCEDURE или FUNCTION с указанием соответствующего имени. Далее следует перечисление действий, составляющих эту процедуру или функцию. Весь список действий окружен так называемыми операторными скобками BEGIN ... END.

Общий вид блока описания процедуры или функции таков:

PROCEDURE имя (параметры); *или* **FUNCTION** имя (параметры): тип значения;

BEGIN

действие первое;

действие второе;

и т. д.

действие последнее;

END;

Примечание: среди действий внутри функции обязательно должно быть хотя бы одно присваивание ей какого-либо значения.

PROCEDURE coord;

BEGIN

Начинаем описание первой из процедур.

WRITE('Количество точек —>'); **READLN**(n);

Ввод числового значения для переменной n с предварительным выводом поясняющего текстового комментария. После вывода курсор остается в той же строке, а после ввода — переходит на следующую строку экрана.

RANDOMIZE;

Запуск в работу встроенного в транслятор программного генератора случайных чисел.

FOR i:=1 **TO** n **DO**

BEGIN

(ТЕЛО ЦИКЛА)

END;

Блок оператора цикла со счетчиком i. Поскольку тело цикла состоит из нескольких действий, оно заключено в операторные скобки **BEGIN ... END**. Горизонтальное смещение этих строчек сделано для увеличения наглядности. Внутри операторных скобок — три последовательных действия. Первые два из них:

x[i]:=TRUNC(RANDOM(100));

y[i]:=TRUNC(RANDOM(100));

— это операции присваивания очередным элементам координатных массивов их значений с использованием двух встроенных функций. Функция **RANDOM** вырабатывает (с помощью запущенного в работу генератора) случайное вещественное число в интервале от нуля до того значения, которое указано в скобках в качестве ее аргумента. Функция **TRUNC** преобразует это вещественное число в целое, отсекая (буквальный перевод слова *trunc*) его дробную часть.

Третье из действий внутри цикла:

WRITELN('x(', i, ')=', x[i]:2, 'y(', i, ')=', y[i]:2);

— вывод на экран полученных координат точек. Среди элементов вывода, перечисленных через запятую, встречаются как текстовые сообщения (они взяты в апострофы), так и значения переменных. Поскольку это действие последнее в цикле, точка с запятой после него не обязательна.

END;

Конец описания первой процедуры.

FUNCTION distance(i, j: **BYTE**): **REAL**;

BEGIN

Начинаем описание функции. Приведено имя функции и в скобках два ее параметра с указанием их типа. Кроме того, после скобок через двоеточие указан тип того результата, который будет вырабатываться функцией.

distance:=SQRT(SQR(x[i]-x[j])+SQR(y[i]-y[j]));

Среди действий, составляющих описание, обязательно должно быть присваивание функции того значения, которое она будет иметь. В данном примере это значение вычисляется с помощью двух встроенных функций **SQRT** и **SQR**: вторая дает квадрат разности координат, а первая — берет квадратный корень от суммы квадратов.

END;

Конец описания функции.

PROCEDURE init
BEGIN

Начинаем описание второй процедуры. Параметров ей никаких не передается.

l:=1; h:=2; b:=distance(1, 2);

В этой процедуре — три присваивания переменным базиса их начальных значений. При этом в третьем действии используется описанная выше функция пользователя `distance` с параметрами, указанными в явном виде. По окончании последнего действия точка с запятой необязательна, хотя и может стоять.

END;

Конец описания процедуры.

PROCEDURE basis(r: REAL; VAR b: REAL);
BEGIN

Начинаем описание третьей процедуры. Приведено ее имя и в скобках — два параметра, передаваемых процедуре (с указанием их типа). Вторым параметром особого вида — это переменная величина, о чем свидетельствует служебное слово `VAR` перед именем `b`. Различие между параметрами состоит в том, что при вызове процедуры значения `r` и `b` передаются в нее из основной программы, но обратно в основную программу возвращается только значение `b`. Если в процедуре `r` и `b` изменят свои значения, то изменения `r` никак не проявятся в основной программе. Переменная `b`, наоборот, будет иметь в основной программе новое значение, принятое ею в процедуре.

IF r<b THEN
BEGIN
b:=r; l:=i; h:=j;
END;

Оператор условия с одной только ветвью `THEN`. Внутри этой ветви не одно, а несколько (три) разделенных точкой с запятой действий, поэтому они взяты в операторные скобки `BEGIN ... END`. Эти три действия — простые присваивания переменным новых значений. Выполняться они будут, только если условие `r` окажется истинным.

END;

Конец описания процедуры.

PROCEDURE result(l, h: BYTE; b: REAL);
BEGIN

Начинаем описание последней, четвертой процедуры. Приведено имя процедуры и в скобках — три ее параметра с указанием их типа.

WRITELN('Ближайшие друг к другу точки: ', l, ' и ', h);
WRITELN('Расстояние между ними: ', b:4:2);

Вывод результатов работы программы. В скобках после служебного слова `WRITELN` перечислены через запятую элементы вывода, причем текстовые элементы помещены в апострофы.

Для расстояния `b` через двоеточие явно указан формат его вывода: всего четыре цифры, из них две — после запятой.

END;

Конец описания процедуры.

В отличие от `QBasic`, где сначала пишется текст программы, а затем описываются процедуры, в `TurboPascal`, наоборот, сначала описываются процедуры, и только потом идет текст

основной программы. Он начинается со служебного слова BEGIN и заканчивается служебным словом END с обязательной точкой после него.

BEGIN

Начало основной программы.

ClrScr;

Очистка экрана — стандартная процедура из модуля CRT.

coord;

Вызываем на исполнение процедуру coord, осуществляющую ввод координат точек.

FOR i:=1 TO n-1 DO

Оператор цикла со счетчиком i . Это внешний (из двух вложенных) цикл. Его тело составляет один-единственный, правда, не простой, а составной оператор внутреннего цикла. Вот он:

FOR j:=i+1 TO n DO

Это тоже оператор цикла, но уже со счетчиком j . То обстоятельство, что он не простой, а составной, отражено в наличии операторных скобок BEGIN и END. Внутри этих скобок перечислены несколько действий, составляющих тело внутреннего цикла. Если бы такое действие было одно, то скобок можно было бы не ставить. Всего перечислено два действия. Первое действие:

BEGIN

r:=distance(i, j);

— нахождение расстояния между точками i и j с помощью функции distance.

Второе действие:

basis(r, b);

— исполнение процедуры basis, в которой базисное расстояние b сравнивается с только что вычисленным текущим расстоянием r . Точку с запятой после него можно не ставить, поскольку этот оператор является последним в теле внутреннего цикла.

END;

На этом внутренний, а вместе с ним и внешний цикл заканчиваются. Для внешнего цикла операторных скобок не потребовалось, так как в его теле — всего одно действие — внутренний цикл.

Далее идет следующий оператор основной программы:

result(l, h, b);

Вызывается на исполнение процедура вывода ответа на поставленный в задаче вопрос.

END.

Конец программы. В этом месте точка после служебного слова END обязательна.

Примечание. Для того чтобы после выполнения программы результат ее работы остался на экране (до нажатия любой клавиши), часто используют один стандартный прием. Непосредственно перед последним служебным словом END. (с точкой) добавляют еще одну строку:

REPEAT UNTIL KeyPressed;

Это пустой, т. е. не содержащий никакого действия, цикл REPEAT UNTIL с постусловием KeyPressed, которое является стандартной функцией из модуля CRT. Эта функция имеет логическое значение FALSE до тех пор, пока не нажата никакая клавиша. Тем самым условие выхода из цикла не выполняется, и он без конца повторяется. При этом на экране ничего не меняется, поскольку цикл пустой.

Задания для самостоятельного решения

1. Дополнить рассмотренную программу процедурой вывода изображения точек на графический экран.
2. Изменить процедуру ввода координат точек так, чтобы их значения не выработывались случайным образом, а вводились с клавиатуры.
3. Изменить процедуру ввода координат точек так, чтобы их значения брались из файла на диске.
4. Найти среди точек наиболее удаленные друг от друга.
5. Найти среднее значение расстояния между точками.
6. Найти крайние (левую, правую, нижнюю и верхнюю) точки.
7. Найти точку, занимающую центральное положение среди всего множества.
8. Предположим, что надо обойти все до одной точки. Из какой точки надо выйти и какую выбрать оптимальную последовательность обхода, так, чтобы длина всего пути оказалась наименьшей? Как эта длина зависит от числа точек?

Исследовательские задания по теме «Точки на плоскости»

1. Определить, как зависит минимальное расстояние между точками от их числа, при одном и том же диапазоне значений координат.
2. Определить, как зависит максимальное расстояние между точками от их числа, при одном и том же диапазоне значений координат.
3. Определить, как зависит среднее расстояние между точками от их числа, при одном и том же диапазоне значений координат.
4. Определить, как зависит минимальное расстояние между точками от диапазона их координат, при одном и том же числе точек.
5. Определить, как зависит максимальное расстояние между точками от диапазона их координат, при одном и том же числе точек.
6. Определить, как зависит среднее расстояние между точками от диапазона их координат, при одном и том же числе точек.
7. Определить, как зависит длина оптимального обхода всех точек от их числа.

Компьютер для дома по почте!

Вышлем по Вашему адресу по самым низким ценам:

**ZX-SPECTRUM 48 (128),
SCORPION 256,
PROFI 512,**

**IBM-совместимые от PC-XT («Поиск») до IBM PC 486 в любой конфигурации,
любые комплектующие, аксессуары.**

Новейшие программные средства для

**БК-001(01), БК-11(М),
УКНЦ (МС-0511),
ZX-SPECTRUM,**

IBM-совместимых компьютеров.

Наш адрес:

189510, Санкт-Петербург, Ломоносов, в/я 649, «КИ-ЧПМ».

Для ответа: подписанный конверт с марками.

Н. В. Софронова

РЕШЕНИЕ ЗАДАЧ НА МАССИВЫ С ПОМОЩЬЮ ЭЛЕКТРОННЫХ ТАБЛИЦ

На наш взгляд, сегодня существует разрыв между теми задачами, которые предлагаются учащимся при изучении основ программирования, и теми, которые школьники решают при знакомстве с пакетами прикладных программ. Мы предлагаем с помощью соответствующего подбора задач выявить и довести до понимания учащихся связь между структурной организацией массивов и данными электронных таблиц (ЭТ), показав тем самым, как знание основ программирования используется для работы с готовым программным обеспечением.

Школьные задания по ЭТ ориентированы прежде всего на решение статистических задач, обычных в практике пользователей электронных таблиц. Мы предлагаем расширить диапазон задач, которые можно решать на уроках информатики в школе с помощью ЭТ. Наиболее близкими по смыслу и содержанию являются задачи на массивы. Ниже приведено несколько примеров, в которых задачи на программирование решаются с помощью электронных таблиц. Решения даны для системы ФРЕЙММОНТАЖ (ЭТЛАС-Ashton-Tate, IBM), но задачи могут быть решены и в другой системе электронных таблиц.

Задача 1.

Найти наименьший элемент в числовой таблице.

Решение. Пусть числа расположены в ячейках A1:T5. Тогда формула, которую можно поместить в произвольную свободную ячейку, имеет вид: @min(A1:T5).

Задача 2.

Найти наибольший элемент в числовой таблице.

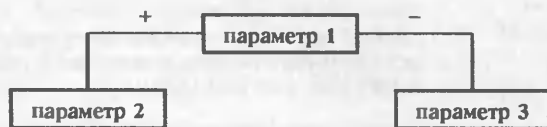
Решение. Пусть числа расположены в ячейках A1:T5. Тогда формула в произвольной свободной ячейке имеет вид: @max(A1:T5).

Задача 3.

В одномерной числовой таблице все положительные элементы заменить нулями.

Решение. Количество элементов в таблице не имеет значения. Допустим, числа расположены в ячейках A1:T1. Тогда в ячейку A2 записываем формулу @if(A1>0, 0, A1), далее копируем (нажатием клавиши F8) по второй строке.

Функция @if(параметр 1, параметр 2, параметр 3) аналогична условному оператору структурных языков программирования. Ее можно представить в виде:



Если условие, заданное параметром 1, выполняется (+), то выполняется действие, описанное параметром 2, иначе (-) — параметром 3.

Для нашего примера: если значение ячейки A1 > 0, то в ячейку записывается 0. Если A1 <= 0, то значение не изменяется.

Задача 4.

Заменить в одномерной числовой таблице отрицательные элементы их квадратами.

Решение аналогично предыдущей задаче. Формула, которую надо записать в ячейку A2, имеет вид: @if(A1 < 0, A1*A1, A1).

Задача 5.

Заменить в одномерной числовой таблице элементы, большие числа M, на число A. Рассмотреть случай M=2, A=-7.

Решение аналогично задаче 3. Формула имеет вид: @if(A1 > 2, -7, A1).

Задача 6.

Проверить, является ли одномерная числовая таблица упорядоченной по возрастанию.

Решение. Вариантов решения этой задачи несколько. Допустим, мы будем выводить слово «да», если $A1 \leq B1$, и «нет» в противном случае. Тогда формула для ячейки B2 будет иметь вид: $@if(A1 \leq B1, \text{«да»}, \text{«нет»})$. Затем копируем эту формулу по второй строке. Результат будет представлен в виде последовательности слов «да», «нет». Если слово «нет» встречается хотя бы однажды, то одномерная числовая таблица не будет упорядоченной по возрастанию. Подобное решение позволяет показать использование текстовых констант в формуле.

Задача 7.

Проверить, имеется ли в одномерной числовой таблице хотя бы одна пара чисел, совпадающих по величине.

Решение задачи аналогично задаче 6. Формула для ячейки B2 будет иметь вид: $@if(A1=B1, \text{«да»}, \text{«нет»})$. Если хотя бы один раз встретится слово «да», то ответ на вопрос задачи будет положительным.

Задача 8.

Определить в одномерной числовой таблице число соседств из двух положительных чисел.

Решение.

Примечание. Задачи 8—10 требуют предварительного присвоения ячейке фиксированного значения. Если этого не сделать, то значение искомого числа будет меняться при копировании каждой формулы и мы не сможем получить верный результат.

Пусть искомое число будет накапливаться в ячейке A2. Тогда в ячейку A2 запишем формулу: $A2:=1$, в ячейку B2 и остальные (через копирование) запишем формулы вида: $@if(@and(A1>0, B1>0), \$A\$2:=$A\$2+1)$.

В этой формуле условие описывается функцией $@and()$. Оператор присвоения записан через фиксированную ссылку, как обычно в языках программирования. Команда для случая, когда условие функции $@if()$ не выполняется, отсутствует, т. е. в этом случае не выполняется никакого действия.

Задача 9.

Определить в одномерной числовой таблице число соседств из двух чисел одного знака.

Решение задачи аналогично задаче 8. В ячейку A2 запишем формулу $A2:=1$. Если два числа имеют одинаковые знаки, то их произведение положительно. Поэтому формула в ячейке B2 будет иметь вид: $@if(A1*B1>0, \$A\$2:=$A\$2+1)$.

Задача 10.

Найти произведение ненулевых элементов одномерной числовой таблицы.

Решение задачи аналогично задаче 8, но в ячейке A2 будет накапливаться произведение элементов.

В ячейку A2 запишем формулу $A2:=A1$. Если в ячейке A1 значение равно нулю, то начните решение с первой ненулевой ячейки. Пусть в нашем случае первая ненулевая ячейка — это A1. Тогда в ячейку B2 надо записать формулу: $@if(B1<>0, \$A\$2:=$A\$2*B1, 0)$.

Как видно, задачи на массивы с использованием электронных таблиц имеют принципиально иное решение, чем при составлении программ. Но использование таких задач при изучении ЭТ позволит учащимся увидеть единый подход в организации данных в ЭВМ, понять общие принципы решения задач на компьютере:

- обращение к данным осуществляется через имена переменных или ячеек, в которых они расположены,
- команда для исполнения дается на языке, «понятном» компьютеру, т. е. можно использовать только слова языка программирования (для ЭТ это тоже язык программирования: FRED для системы ФРЕЙММОНТАЖ).

НОВЫЕ ПРОЕКТЫ



Е. Ю. Тюменцева

MULTIVISION И MULTIVISION PRO В УЧЕБНОМ ПРОЦЕССЕ

В настоящее время все больше людей, в том числе и работающих в сфере образования, в своей повседневной деятельности сталкиваются с необходимостью использования современных средств передачи информации. Компьютерная грамотность становится неотъемлемой частью общечеловеческой культуры. Потребности общества, формирующего социальный заказ системе образования, ставят задачи широкого внедрения современных информационных технологий в процесс обучения. Над решением этих проблем вот уже 10 лет в России работают различные организации, в том числе и Научный Центр программных средств обучения. За это время его сотрудниками и учителями базовых учебных заведений, накоплен определенный практический опыт по созданию и внедрению в реальный учебный процесс образовательных проектов, связанных с использованием программного обеспечения, видео- и компьютерной техники.

Этим опытом, своими взглядами на информатизацию образования я и хочу сегодня поделиться с читателями журнала, с точки зрения преподавателя далеко не «компьютерного» предмета.

Насыщение учебных заведений современными техническими средствами стало сейчас явлением повсеместным. Но, несмотря на обилие техники, обучающих и тестовых программ, на наличие компьютерных классов в большинстве московских школ, эта техника в основном используется для проведения уроков информатики и вычислительной техники, хотя их применение уже давно выходит за рамки курса ОИВТ. Очевидно, что в каждом общеобразовательном предмете реально существуют

такие задачи, решение которых с помощью информационных средств сделает уроки более зрелищными и интересными и даст образовательный результат. Однако использование компьютера и программ при изучении различных предметов, в том числе и биологии, — явление пока крайне редкое.

Как гласит «теория», компьютеризация обучения должна не только способствовать развитию личности и творческих возможностей учащихся, но и кардинально изменять всю технологию обучения. Но практика показывает, что попытки «лобового» внедрения техники современного уровня в ход урока зачастую не дают того образовательного эффекта, на который учитель вправе рассчитывать.

Такое положение дел объясняется множеством причин: отсутствием обоснования и методических разработок, трудностями организации предметных уроков в компьютерных классах, неподготовленностью учителей общеобразовательных дисциплин; отсутствием «универсальных» (отвечающих требованиям любого учителя) обучающих программ... Этот ряд наверняка мог бы продолжиться любой из вас применительно к конкретно взятой школе (учебному заведению), предмету. Но и при наличии положительного решения вышеперечисленных проблем перед «рядовым» учителем-предметником остается главная задача органичного включения компьютерной техники в канву урока, сохранения при этом за собой традиционных и выработку новых психолого-педагогических и методических позиций.

Именно о возможных путях решения этой проблемы и пойдет речь в предлагаемой ва-

шему вниманию статье. Сразу хочу оговориться, что она печатается в рамках анонсированной в 1994 г. информационной кампании о методических возможностях применения программного пакета MultiVision и комплекса MultiVision Pro в учебном процессе. Сегодняшний рассказ посвящен использованию компьютерной техники в преподавании биологии.

Как показало мое знакомство с компьютерными программами по биологии, большинство из них предполагает либо фрагментарную демонстрацию учебного материала одного-двух уроков, либо вынуждает учителя работать с колоссальным объемом программ, созданных в виде энциклопедий. При этом практически все они написаны в виде баз данных или тренажеров и предназначены для индивидуальной работы учащегося со «своим» персональным компьютером. Понятно, что учителю-предметнику, не искушенному в компьютерной технике, в данном случае отводится роль простого консультанта или, что еще хуже, пассивного наблюдателя, даже при наличии сетевых коммуникаций.

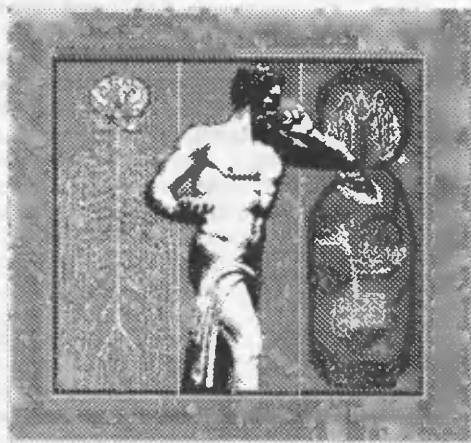
В отличие от этого идеология разработки обучающих программ в среде MultiVision предполагает иной подход. Он основывается на создании программ, пакетов программ (охватывающих учебный курс), ориентированных на учителей и учащихся, на основные требования к знаниям и умениям учащихся, заложенные в действующие учебные программы. При этом учитель уходит от роли послушного исполнителя чужой (хотя в боль-

шинстве случаев правильной) программистской или педагогической мысли, становится АВТОРОМ собственной компьютерной продукции (от поурочных до более глобальных разработок) и методики ее применения, НЕ ИЗУЧАЯ НИ ЕДИНОЙ СТРОКИ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ.

Предусмотренные русскоязычной средой MultiVision возможности доступного, сжатого и наглядного изложения материала значительно облегчают его усвоение, технология подачи информации в программах позволяет легко выделять и отбирать наиболее существенное, устанавливать внутренние взаимосвязи изучаемых процессов, акцентирует внимание учащихся на важнейших понятиях курса и их значении. Тексты, схемы и рисунки дополняются мультфильмами, которые наглядно демонстрируют последовательность протекания различных процессов, дают возможность выделить их основные стадии, получить стоп-кадры, конкретизирующие представления детей о происходящем на каждом этапе. Работа с программами, реализованными в среде MultiVision, также позволяет экономить время учителя на уроке для изложения дополнительного учебного материала.

Одним из примеров подобных разработок является пакет демонстрационных обучающих программ «Организм человека», подготовленный в 1993 г. автором этой статьи, тогда еще учителем школы № 209 Северного округа, учениками XI класса и специалистами НЦПСО.

ОРГАНИЗМ ЧЕЛОВЕКА



Версия 1.1

Пакет состоит из 7 отдельных программ, охватывающих все основные темы курса «Биология человека» (анатомия), и предназначен как для деятельности преподавателя, так для работы ученика или группы учащихся.

При разработке программ пакета мы ориентировались на их соответствие следующим требованиям:

- способствовать повышению познавательного интереса;
- отвечать практическим потребностям учителя и учащихся;
- обладать универсальностью в использовании;
- опираться на базовую учебную программу;
- содержать краткую, научную информацию по конкретным темам или всему курсу, дополняющую или расширяющую материал учебника;
- быть взаимосвязанным с другими дидактическими материалами.

Предоставленная мне возможность систематического использования компьютерного оборудования и собственного программного обеспечения показала не только реальность органичного включения их в ход проводимых уроков, но и универсальность с точки зрения применения на самых разных занятиях, как по форме, так и по их содержанию, в том числе:

- на уроках объяснения нового материала (вводных уроках и лекциях);
- на семинарах;
- на повторительно-обобщающих занятиях;
- на уроках контроля знаний и др.

Использование компьютерных технологий в практике трехлетней работы сочеталось с традиционными формами (работой с книгой, проведением практических и лабораторных занятий) и применением других дидактических средств (приборов, препаратов, муляжей, раздаточных материалов, таблиц, схем и т. д.). Как мне кажется, разные формы проведения учебных занятий по биологии с включением техники успешно дополняются различными методами организации работы учащихся: фронтальной, групповой, индивидуальной. Применение обучающих программ не ограничивается только уроком, они используются и при внеклассной работе, факультативных, кружковых занятиях, работе с отстающими.

Учебные занятия

Одно из основных предназначений демонстрационных обучающих программ — иллюстрирование учебного материала на занятиях, поэтому я использую их на вводных

лекциях, определяя перспективные задачи темы, а также при дальнейшем изложении материала по данному уроку.

На вводных лекциях логично демонстрировать программу целиком, организуя фронтальную работу. В ходе обзорного изложения можно поставить перед учащимися следующие задачи:

1. Подготовить план изучения новой темы.
2. Выделить основные понятия темы, которую предстоит изучать и др.

Так, например, при проведении вводного занятия на тему «Строение нервной системы и ее значение» с помощью компьютерной программы «Нервная система» я не только предварительно знакомя учащихся с общими принципами работы нервной системы человека, рассказываю о механизме ее работы и значении для жизнедеятельности, но и обращаю внимание на признаки сходства и различия в структурах нервных систем человека и животных, а также ставлю задачу: выделить признаки, установить их сходство и различия и сделать выводы об общности происхождения животного мира.

При дальнейшем изложении нового материала мне представляется возможным фрагментарно использовать программы в соответствии с содержанием конкретного урока. Так, знакомя учащихся с темой «Газообмен в легких и тканях» с помощью экрана «Газообмен» программы «Органы дыхания», организую повторение состава газов атмосферы, рассказываю о содержании газов во вдыхаемом и выдыхаемом воздухе, используя при этом диаграммы обучающей программы. При объяснении процесса газообмена я опираюсь на знание структурно-функциональных особенностей кровеносных сосудов и легочных альвеол. Удачные на мой взгляд схемы, текст экрана способствуют раскрытию сущности процесса обмена газов в организме и причин, его обеспечивающих. В ходе изложения содержания урока акцентируется внимание на понятиях диффузии и парциального давления. Эти термины выделены в тексте, и программа дает возможность предложить вниманию учеников дополнительную информацию, содержащую формулировку данных понятий. С помощью цифровых данных, текстов, схем программы, опираясь на межпредметные связи с химией и физикой, объясняю механизм насыщения легких кровью и клеток кислородом и удаления из них углекислого газа.

Работа с информационными технологиями обучения открывает неисчерпаемые возможности для педагога. Но, к сожалению, в большинстве случаев учителя, имеющие

MultiVision Pro -

Value-Priced

Компьютер

- IntelliView; Model DPS-3
486 SLC2 50
4 Mb RAM
1.44 FDD
256 HDD
8,4" colore Active-Matrix
LCD
Math co-processor
- 14" SVGA monitor 0.28DPI
- KeyBoard

Проекционное оборудование

- Overhead Projector
Medium 5000
- SuperLightScreen 115x150
- Stand for Screen 115x150
- Remote Pointer

Мебель

- Рабочий стол учителя
Medium PT-7

Программное обеспечение

- Windows 3.11
- DOS 6.22
- CAIRC MultiVision 4.5
(дискеты)

\$ 16900

OPTIONS

- Печатающее устройство
HP DJ 320
- Digital Option Card
- CD ROM внешний,
удвоенной скорости
- Сканирующее устройство
HP SJ II cx
- Видеомагнитофон Panasonic
- Видеокамера Panasonic

Value-Priced Desktop

Компьютер

- IBM PC/AT 486
"MultiVision Pro"
Intel 486 DX2 66
8 Mb RAM
256 Kb CASHE
450 HDD
1,2; 1,44 FDD
- 15" SVGA Colore Monitor
Tatung 0.28 DPI
- 101 Focus KeyTrack
KeyBoard
- MultiMedia Kit
Diamand 5000

Проекционное оборудование

- Overhead Projector
Medium 5000
- SuperLightScreen 115x150
- Stand for Screen 115x150
- Remote Point
- Online Colore Works SX
8,4" Dual-scan STN
панель

Мебель

- Рабочий стол учителя
Medium PT-7

Печатающее устройство

- Printer HewlettPackard
DeskJet 320

Программное обеспечение

- Windows 3.11
- DOS 6.22
- CAIRC MultiVision 4.5

\$ 16999

Basic

Компьютер

- IBM PC/AT 486
"MultiVision Pro"
Intel 486 DX2 66
8 Mb RAM
256 Kb CASHE
450 HDD
1,2; 1,44 FDD
- 15" SVGA Colore Monitor
Tatung 0.28 DPI
- 101 Focus KeyTrack
KeyBoard
- MultiMedia Kit
Diamand 5000

Проекционное оборудование

- Overhead Projector
Medium 10K
- Proxima Ovation 820
- Proxima Cyclis 2030
- LazerPointer A90
- SuperLightScreen 115x150
- Stand for Screen 115x150
- Remote Pointer

Мебель

- Рабочий стол учителя
Medium PT-7

Video оборудование

- Видеомагнитофон
Panasonic VHS
- Видеокамера Panasonic
VHS

Печатающее устройство

- Printer HewlettPackard
DeskJet 320

Сканирующее устройство

- Scanner HewlettPackard
ScanJet II cx

Программное обеспечение

- CAIRC MultiVision 4.5
- Windows 3.11
- DOS 6.22

\$ 23999

По вопросам приобретения и
за дополнительной информацией
обращайтесь:



Научный Центр Программных Средств Обучения

Адрес: 109004, г. Москва, ул. Б. Коммунистическая, 9-а
ТЕЛЕФОН. (095) 214 77 84, 214 46 49. Факс (095) 271 04 28

Portable

Компьютер

- MultiMedia Notebook
486 DX4 100
16 Mb RAM
450 HDD
1.44 FDD
Fax Modem (встроенный)
CD ROM (встроенный)
Активная матрица
640x480
- 101 Focus KeyTrack
KeyBoard

**Проекционное
оборудование**

- Proxima DeskTop
Projector 2800
- Proxima Cyclops 2060
- LazerPointer A90
- Tripod Screen Professional
150x150

**Видео
оборудование**

- Видеокамера Panasonic
VHS

**Печатающее
устройство**

- Printer HewlettPackard
DeskJet 320

**Программное
обеспечение**

- CAIRC MultiVison 4.5
- Windows 3.11
- DOS 6.22

\$ 34999

Внимание!!!
Обязательства по
послегарантийному
обслуживанию
принимают на себя наши
деловые партнеры
А/О "НОВАЯ
ЭКСПОЗИЦИОННАЯ
ФИРМА"
Москва, ВВЦ,
п-н "ФИЗИКА"
тел. (095) 181 96 31
факс (095) 181 23 75

Video Work Station

Компьютер

- IBM PC "MultiVision Pro"
Pentium 90
32 Mb RAM
256 Kb CASHE
1 Gb - IDE4 HDD
1.2 1.44 FDD
IDE PCI-controller (4Mb)
Видеокарта JazzCard:
Jacarta;
Projector;
Port of Entry
- MultiMedia Kit
Diamand 5000
- 15" SVGA Colore Monitor
Tatung 0.28 DPI
- Big Tower
- Microsoft KeyBoard
- Microsoft Mouse

**Видео
оборудование**

- Видеокамера Panasonic
S-VHS
- Видеомагнитофон
Panasonic S-VHS
- Телевизор Panasonic
GAO S-VHS

**Печатающее
устройство**

- Printer HewlettPackard
DeskJet 1200 C

Мебель

- Рабочий стол учителя
Medium PT-7
- Видеоподставка
Medium VT-2

**Проекционное
оборудование**

- Overhead Projector
Medium 10K
- Proxima Ovation 842
- Proxima Cyclops 2050
- LazerPointer A90
- SuperLight Screen 130x180
- Stand for Screen 130x180
- Remote Pointer

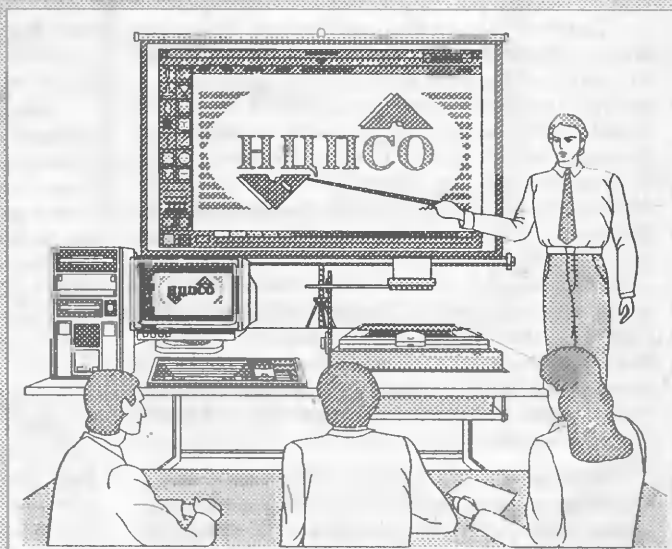
**Сканирующее
устройство**

- Scanner HewlettPackard
ScanJet II c

**Программное
обеспечение**

- Windows 3.11
- MS DOS 6.22
- MS Office 4.3
- Adobe Premier
- AIST AutoAnimation 1.0
- AIST MediaMaster 1.2
- CAIRC MultiVision 4.5

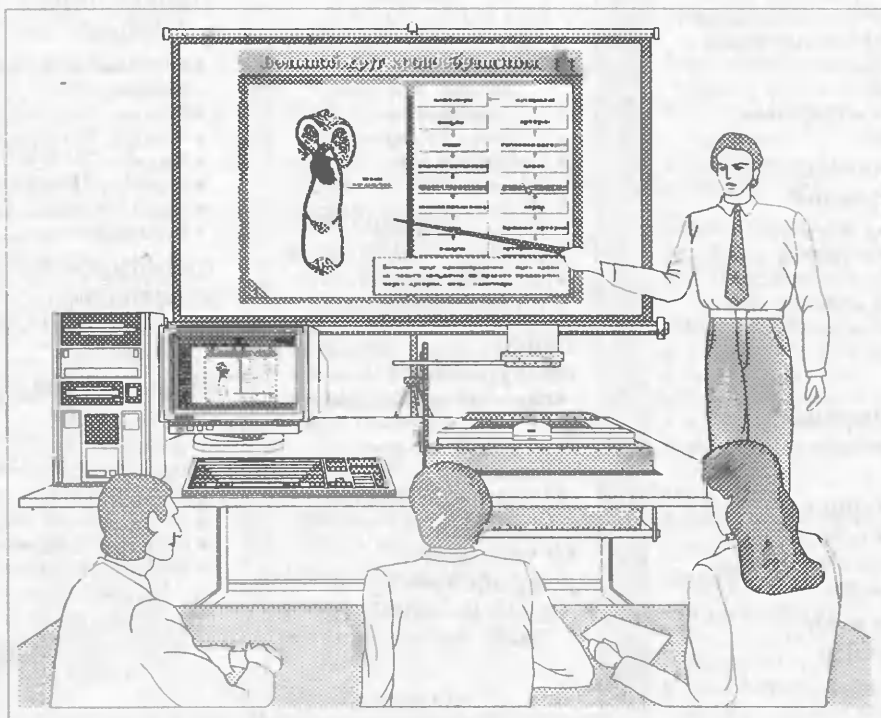
\$ 45999



практику работы с компьютером, при объяснении нового материала вынуждены либо довольствоваться монитором одной из машин, организуя работу класса фронтально, либо (повторюсь) организовывать индивидуальную работу каждого ученика с персональной машиной. В первом случае весь класс вынужден использовать небольшое изображение на дисплее одной из машин, во вто-

ром — нарушаются психологический контакт и взаимодействие учителя и ученика. Эффективность подобного объяснения заведомо снижается.

Проблему использования компьютерной поддержки курса, не отвлекая внимание учащихся от деятельности учителя, удастся решить с помощью программно-аппаратного комплекса MultiVision Pro.



Система видео-компьютерного проекционного оборудования и экрана, выполняющего роль дополнительной доски, оставляет в центре внимания личность учителя, который использует компьютер и демонстрационные программы как одно из действенных средств обучения. Занятия, проводимые с применением комплекса, позволяющего вывести изображение с персональной машины на большой экран, сохранив при этом всю палитру красок программы, становятся более зрелищными и интересными. Такого рода иллюстрация, воздействуя на эмоционально-личностное восприятие учеников, в большей степени привлекает их внимание к рассказу учителя, способствуя в конечном итоге прочному усвоению знаний.

Исключительно важно, на мой взгляд, что подобная проекционная аппаратура позволяет сохранить центральное место во время за-

нятий за преподавателем, совершенно не изменяя его традиционных педагогических действий и привычных атрибутов деятельности.

Использование новых информационных средств обучения позволяет мне решить традиционно трудные предметные вопросы с большим образовательным эффектом, повысить рейтинг своего предмета. Компьютерная поддержка курса биологии в школе значительно сокращает время, отведенное на изучение обязательного учебного материала, дает возможность существенно углубить и расширить круг рассматриваемых проблем и вопросов.

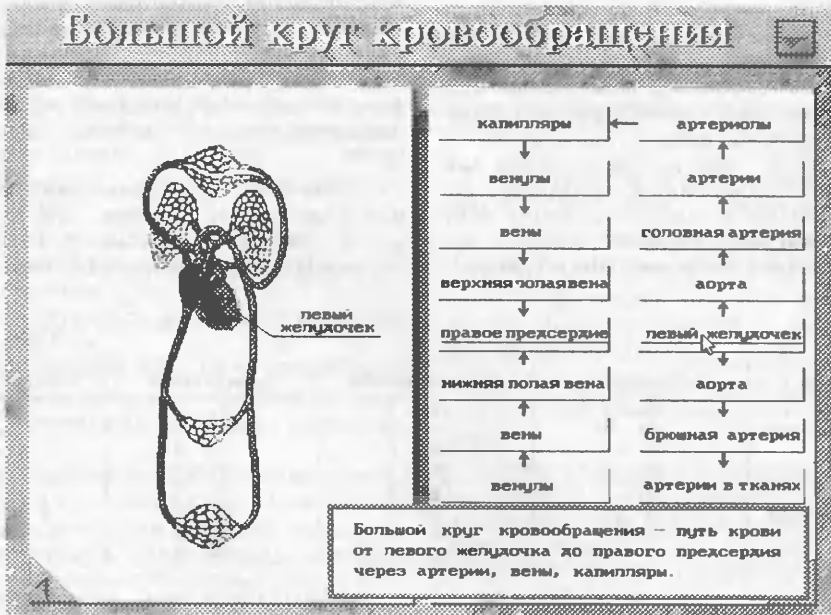
С помощью MultiVision и MultiVision Pro у меня впервые появился реальный шанс разнообразить диапазон своих технологических возможностей как преподавателя. Например, за счет мультипликации в программах и большого экрана я могу наглядно демон-

стрировать всему классу не только статические образы, но и динамику физиологических процессов, происходящих в живых организмах (циркуляцию крови, движения ресничного эпителия, дыхательные движения, мышечные сокращения и т. д.). Подобного эффекта не добиться даже средствами видеотехники. Таким образом значительно расширяются и углубляются возможности представления учебной информации. Очень важным, как мне кажется, является еще и то, что учитель (либо ученик при выполнении самостоятельной работы) может по своему усмотрению не только инициировать и остановить мультфильм, но и получить стоп-кадр. Это позволяет существенно конкретизировать представления учащихся о происходящих явлениях на разных стадиях, досконально разобратся в их сущности.

Не менее значительным, чем техническая поддержка лекционной деятельности учителя, является возможность применения видеокomпьютерных технологий на уроках (семинарах, беседах, практических занятиях), различных по формам организации работы (фронтальной, групповой, индивидуальной).

Примером такой работы может быть урок

на тему «Круги кровообращения». Проводя его, знакомя учащихся со взглядами древних ученых на циркуляцию крови в организме человека, выводя их высказывания с рабочей учительской машины на демонстрационный экран. Таким образом, обсуждаемый текст остается перед глазами учеников практически в течение всего урока. Это дает мне возможность поставить ряд проблемных заданий и организовать работу в группах по их решению, используя сравнение материала обучающей программы «Кровь и кровообращение» и содержания выведенного на экран текста. Каждая группа работает со своим компьютером и фрагментом программы. В результате работы учащиеся, выступая в роли исследователей, выявляют ошибочность взглядов древних ученых с точки зрения современной науки, отмечают правильность в их высказываниях и формулируют свои выводы о закономерностях циркуляции крови в организме. Защита выдвинутых ими тезисов сопровождается демонстрацией на большом экране мультфильма и схемы движения крови по малому, большому кругам кровообращения, по обоим кругам одновременно.



После общего обсуждения при помощи самостоятельно изготовленных бумажных средств такая же схема монтируется на обычной классной доске. Подобная форма проведения работы сохраняет за учителем присутствие ему педагогические функции: выделение вопросов для изучения, организацию работы

групп, оказание консультативной помощи при выполнении заданий, планирование обсуждения, подведение его итогов, оценку работы групп и отдельных учащихся.

Систематическое проведение занятий с использованием групповой формы их организации способствует развитию индивиду-

ально-личностных особенностей учащихся и навыков работы в коллективе. У ребят развиваются внимание, умение выбрать наиболее значимые материалы, подобрать аргументацию, отстаивать свое и выслушать мнения товарищей, составить краткий доклад, сообщение, подобрать более удачный иллюстративный материал. Самостоятельно найденное, аргументированное, проиллюстрированное подобранными материалами и изложенное перед всем классом решение способствует более прочному усвоению знаний.

Применение компьютерных технологий позволяет разнообразить также и формы проведения повторительно-обобщающих занятий.

При фронтальной проверке знаний у учащихся есть возможность для наглядного сравнения своих представлений с научным содержанием предмета. Реализовать эту возможность позволяет одна из особенностей построения программ пакета (наличие скрытой информации, содержащей точные формулировки, сведения о строении органов, значения некоторых величин), а также использование проекционной аппаратуры комплекса.

При необходимости на уроке организуются и индивидуальная работа учащихся с персональным компьютером. В качестве заданий им предлагается за определенное время найти ответы на конкретно поставленные вопросы, подобрать фактический материал для обоснования своих выводов и иллюстрации к сообщениям, ответам и докладам.

Несомненной поддержкой в работе для меня является применение в сочетании с демонстрационными и контролирующими компьютерных программ. Работая с классом, педагог, как правило, не в состоянии одновре-

менно проверить правильность выполнения задания каждым учащимся. Это ведет к накоплению у последних ошибочных представлений, приводящих к появлению существенных пробелов в знаниях, устранить которые впоследствии нелегко.

Используя для контроля тестовую оболочку программы «Знания» (НЦПСО, 1994 г.), я имею возможность подобрать наиболее существенные вопросы для контроля, варьировать порядок их предъявления, получать сведения не только о количестве верных и неверных ответов, но и информацию о том, какие именно вопросы вызвали затруднения у конкретного ученика, каковы типичные ошибки, допущенные в классе, а также изучить тенденцию развития интереса к изучаемым темам.

Своевременный контроль безусловно расширяет возможности коррекции знаний учащихся, а на основе анализа его результатов позволяет подобрать наиболее соответствующие методические приемы, пересмотреть методику преподавания различных тем с целью создания устойчивого интереса к предмету. Взаимодействие учащихся с компьютером в данном случае гарантирует им получение реальной оценки, независимой от субъективного мнения учителя.

В то же время анализ результатов тестирования наряду с оценкой знаний учеников дает учителю определенную возможность оценить и саму эффективность использования информационных технологий в учебном процессе.

Например, при проведении тестирования учащихся IX классов УВК № 1601 по курсу «Биология человека» в 1994 г. были получены следующие результаты:

Т а б л и ц а 1

Класс/группа	Введение	Скелет и мышцы	Пищеварение	Нервная система
IX «Г» 27 учеников	«2» - 2	«2» - 3	«2» - 3	«2» - 2
	«3» - 15	«3» - 12	«3» - 16	«3» - 18
	«4» - 8	«4» - 7	«4» - 8	«4» - 6
	«5» - 2	«5» - 5	«5» - 2	«5» - 1
IX «М» 25 учеников	«2» - 1	«2» - 0	«2» - 0	«2» - 0
	«3» - 15	«3» - 11	«3» - 10	«3» - 9
	«4» - 5	«4» - 9	«4» - 9	«4» - 10
	«5» - 4	«5» - 5	«5» - 6	«5» - 6

Классы для тестирования подбирались с примерно одинаковым интеллектуальным потенциалом, но с целью обеспечения возможности анализа результативности использования информационных технологий были поставлены в различные условия. В гуманитарном классе, ученики которого обладают прекрасно развитым воображением, биология преподавалась традиционными методами, а в

математическом классе, куда менее «склонном» к предметам данного цикла — в сочетании с информационными технологиями (Организм человека, Знания, комплекс MultiVision Pro). Результаты, как мы видим, показывают относительную стабильность качества знаний в IX «Г» и определенную тенденцию повышения уровня знаний в течении года в IX «М», что делает положительные ре-

зультаты использования информационных технологий обучения достаточно очевидны.

Сжатость, четкая отобранность материала компьютерных программ оказывают учащимся реальную поддержку в повторении основного содержания курса и в ходе подготовки к выпускному (вступительному) экзамену. Для оказания дополнительной помощи я провожу консультации и обзорные занятия, на которых демонстрирую программу целиком на большом экране, актуализируя в памяти учащихся конкретные знания, восстанавливаю уже знакомые зрительные образы. Экономия времени при этом дает возможность в дальнейшем остановиться на подробном рассмотрении вопросов, традиционно вызывающих затруднения на экзамене.

Внеурочная деятельность

На мой взгляд использование компьютерных технологий обучения оправдано не только на уроках, но и во внеурочной деятельности: при проведении факультативов, кружков, при работе с отстающими. При этом основной формой организации работы учащихся является самостоятельная работа ученика с персональной машиной по заданию учителя. Планируя работу подобного рода, я учитываю доступность, простоту, удобство интерфейса программ, индивидуальные особенности ученика, подбираю для него соответствующие задания, например:

- познакомиться с содержанием программы (экрана);
- выучить основные определения;
- проследить за осуществлением физиологических процессов;
- найти структурно-функциональные закономерности;
- сформулировать выводы о происхождении тех или иных особенностей, об условиях протекания реакций, процессов и т. д.

По результатам работы ученики оформляют отчеты в письменном виде, либо готовят устные сообщения. Это приучает их формулировать краткие и четкие выводы, анализировать закономерности осуществления процессов. Функции учителя при этом: постановка задачи, консультирование в ходе выполнения работ и оценка результатов.

Развитие интереса учащихся к биологии осуществляется во внеурочной деятельности также и за счет использования их естественного увлечения компьютерной техникой. Так, в течение ряда лет на кружке по изучению пакетов TEACNCAD и MultiVision учениками

VIII—IX классов были разработаны собственные сценарии программ по биологии и после их защиты реализованы на уроках. В перспективном плане такая деятельность способствует укреплению сотрудничества учащихся с учителем.

Сочетание с традиционными методами преподавания

В зависимости от дидактических целей, поставленных на уроке, я обладаю широкими возможностями включения в его ход разнообразных технических средств, используя их изолированно или сочетая в различных вариантах друг с другом, а также с традиционными средствами обучения (работа с книгой, лабораторное оборудование, модели, муляжи, натуральные объекты и т. д.).

В настоящее время использование компьютера в сочетании с проведением лабораторных, самостоятельных работ, опросом, работой с учебником не только не вызывает психологического дискомфорта у меня и моих учеников, но наоборот делает эти занятия более интересными и продуктивными.

Так, например, при подготовке к лабораторной работе на уроке «Ткани, органы, система органов, функциональная системы» на основе содержания экрана «Ткани» учащимся предлагаются следующие задания:

1. Повторить определения тканей.
2. Познакомиться с основными группами тканей человеческого организма.
3. Выявить общие признаки строения тканей, их расположения и функций.
4. Установить наличие взаимосвязей между строением тканей и выполняемыми ими функциями.
5. Оформить результаты задания по пункту 3 в виде письменных теоретических выводов.

Затем проводится лабораторная работа по выявлению структурно-функциональных особенностей тканей различных типов, в ходе которой учащиеся должны найти практическое подтверждение своим теоретическим выводам.

Сочетание материала программы «Органы дыхания» (экран «Дыхательные движения») с текстом учебника на уроке «Дыхательные движения» предоставляет учащимся возможность выявить закономерности дыхания, проследить за реально происходящим физиологическим процессом и самостоятельно заполнить таблицу:

Т а б л и ц а 2

Изменения, происходящие при вдохе и выдохе

Органы, принимающие участие в дыхательных движениях	Вдох	Выдох
1. Межреберные мышцы		
2. Диафрагма		
3. Ребра		
4. Объем грудной клетки		
5. Давление в грудной клетке		

Использование разнообразных источников знаний в своей работе дает возможность составить наиболее полную характеристику изучаемых процессов.

* * *

Завершить сегодняшний рассказ хотелось бы следующим. Применение современных информационных технологий обучения во многом изменяет традиционные взгляды на методику преподавания биологии, структуру и организацию учебной деятельности. С использованием компьютерной поддержки курса я работаю третий год. Зачастую приходится проводить двоянные уроки, уроки-беседы, уроки-консультации, уроки-моделирования. По мере работы с комплексом MultiVision Pro убеждаешься в необходимости повседневного использования подобного оборудования, постоянно открываешь для себя новые и новые технические возможности решения стандартных методических проблем. Например, всем биологам известны затруднения при необходимости применения в демонстрационном плане таких дидактических средств как микропрепараты. Недостаток данных средств вынуждает учителя потратить колоссальное количество времени на рассмотрение одного препарата учащимися всего класса. Ситуацию удается в корне изменить, если вывести на экран крупномасштабное изображение непосредственно из-под микроскопа. И так повсеместно!

Сочетание проекционного оборудования, аудио и видео-компьютерной техники на уроке также предоставляет неисчерпаемые возможности для творчества учителя, позволяя ему не только разнообразить материал по

способам воздействия на ученика, но и совершенствовать собственное педагогическое мастерство.

Комплексное влияние различных информационных средств обучения позволяет создать целостные образы, а чередование их друг с другом — возможность избежать чрезмерного утомления. Подобное сочетание средств позволяет соблюсти также и санитарно-гигиенические нормы работы с компьютерной техникой (время общения ученика с компьютером в ходе урока занимает в общей сложности не более 10—15 минут).

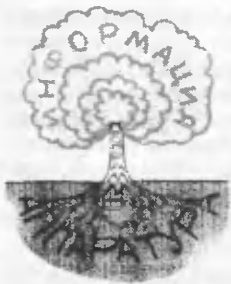
В то же время управление техникой высокого уровня сложности создает для педагога и ряд проблем. Необходимость предъявления информации из разных источников, использование их на уроках требует переосмысления целей и задач, которые ставит учитель, изменения структуры и содержания урока, овладения приемами работы с многофункциональной техникой.

В связи с этими проблемами в идеологии построения комплекса MultiVision Pro и его программного насыщения — MultiVision 4.5, Media Master 1.2 сегодня открывается еще одно перспективное направление.

Представляется возможным в создание видео-компьютерных программ, разработка сценариев таких программ для отдельных уроков, целых тем и курсов. Используя в качестве результирующего технического средства привычный многим и достаточно распространенный источник — видеокассету, видеоманитофон и телевизор или проекционное оборудование, возможно удастся решить следующие проблемы:

- вовлечь в процесс информатизации образования гораздо большее количество учителей;
- во многом избежать тех трудностей, которые возникают у педагогов сегодня, при этом сочетая все преимущества аудио-, видео- и компьютерной техники в едином источнике.

Необходимость таких разработок очевидна, база для их технической реализации практически сформирована (MultiVision Pro Video Work Station), но это тема для отдельного рассказа.



ПРОФЕССИОНАЛЬНЫЙ КОМПЬЮТЕР

М. И. Степанова, З. И. Сазанюк,

НИИ гигиены и профилактики заболеваний детей, подростков и молодежи

ГОСКОМСанЭпидНадзора России

ГИГИЕНИЧЕСКИЕ ТРЕБОВАНИЯ К ПРОВЕДЕНИЮ КОМПЬЮТЕРНЫХ ЗАНЯТИЙ ВО ВНЕУРОЧНОЕ ВРЕМЯ

Одна из проблем компьютеризации школьного образования — гигиеническая регламентация занятий с использованием ВДТ (видеодисплейным терминалом). Необходимость такой регламентации заключается в профилактике негативного воздействия, которое оказывает компьютер на детский организм. В нашем институте проводятся различные исследования, которые дают возможность комплексно изучить состояние организма детей в процессе компьютерной деятельности и выработать соответствующие рекомендации. Нами разработан специальный компьютерный тест для диагностики функционального состояния организма школьника в процессе занятий за ВДТ. Расскажем об этом более детально.

Преимущество такого исследования заключается в том, что оно, с одной стороны, является элементом самих занятий школьника с компьютером, а с другой — учитывает физиологические показатели, отражающие состояние организма.

Компьютерный тест построен на учете психофизиологических характеристик переработки информации. Он представляет собой задание на опознание и запоминание серии из двенадцати предъявляемых на дисплее случайных четырехзначных цифр (для детей младшего школьного возраста — трехзначных) и воспроизведение их с помощью клавиатуры компьютера. Время экспозиции цифрового сигнала минимизируется с помощью специального алгоритма. При ошибочном воспроизведении сигнала или отсутствии от-

вета испытуемого в течение 3 секунд время экспозиции последующего сигнала увеличивается, при правильном ответе — уменьшается на одну и ту же величину. Выполнение теста занимает 1,5—2 минуты.

Для характеристики функционального состояния используются следующие показатели: качество выполнения теста, пороговое время распознавания и запоминания цифровых сигналов с экрана дисплея и латентный период зрительно-моторной реакции — время до начала воспроизведения цифр на клавиатуре. Все эти показатели с помощью специальной формулы интегрируются в единый показатель.

Алгоритм компьютерного теста содержит 4 блока:

- 1) определение индивидуальных исходных показателей функционального состояния;
- 2) определение текущих значений показателей;
- 3) сравнение текущих показателей с исходными и диагностика состояния школьника;
- 4) управление работой (продолжительностью) за компьютером с помощью специальных команд.

Ухудшение функционального состояния школьника в результате занятий за ВДТ выражается в снижении качества выполнения теста, в увеличении времени распознавания и запоминания сигналов, удлинении времени реагирования на сигнал.

Компьютерное тестирование исходного

состояния проводится до занятий. По его результатам вычисляется индивидуальный интегральный показатель, который характеризует дорабочее, исходное состояние школьника. Интегральный показатель принимается равным 100% и вводится в память компьютера.

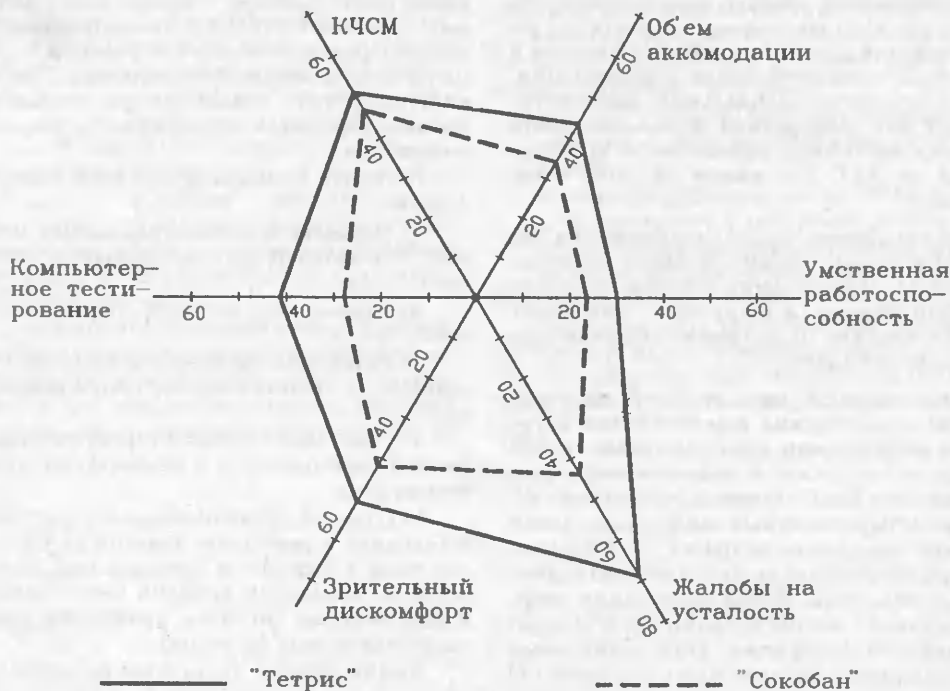
Последующее тестирование проводится в процессе работы школьника за компьютером. Сравнение результатов тестирования с исходными показателями ученика позволяет судить об изменениях в его функциональном состоянии. Установлено, что снижение величины интегрального показателя на 20% и более достоверно свидетельствует об ухудшении функциональных возможностей и появлении утомления. При отсутствии изменений или улучшении функционального состояния интегральный коэффициент равен 100%. В зависимости от значения интегрального показателя после каждого тестирования на дисплее появляется рекомендация о возможности продолжить занятия или прекратить их и отдохнуть.

Для сравнения информативности компьютерного тестирования и традиционных методов диагностики утомления школьников использовался корреляционный анализ. Его результаты выявили, что между показателями работоспособности (корректирующая проба), а также величиной критической частоты слия-

ния световых мельканий (КЧСМ) прослеживается достоверная согласованность. Из этого следует, что предлагаемый компьютерный тест отражает главным образом те психофизиологические изменения, которые происходят в функциональном состоянии центральной нервной системы (ЦНС) под влиянием компьютерной деятельности школьников.

Проиллюстрируем этот вывод результатами наших исследований. Изучение влияния различных по характеру кружковых компьютерных занятий детей младшего и среднего школьного возраста выявило, что наиболее утомительные — занятия, которые целиком посвящены компьютерным играм. Предлагаемый на этих занятиях набор компьютерных игр весьма разнообразен. Однако в этом многообразии как отечественных, так и зарубежных игровых программ, как правило, можно выделить два типа. Первый — это обучающие и развивающие игры, направленные на тренировку логического мышления, расширяющие кругозор школьника. Второй — главным образом игры, тренирующие скорость реагирования на быстро изменяющуюся ситуацию на экране дисплея, игры с навязанным ритмом работы. С помощью традиционных методов и компьютерного теста была проведена оценка воздействия компьютерных игр разного типа на работоспособность и функциональное состояние. В качестве игры первого

Число учащихся (в %) с неблагоприятными значениями показателей функционального состояния организма после игровых занятий с компьютером.



типа была взята игра «Сокобан», второго типа — игра «Тетрис». В исследованиях участвовали школьники 7—13 лет и подростки 16—18 лет (всего 42 человека). Исследование проводилось во второй половине дня, после учебных занятий. Перерыв между компьютерными занятиями и окончанием уроков в школе составлял 1 час и более. Наряду с компьютерным тестированием до и после занятий за ВДТ проводили регистрацию физиологических показателей, отражающих функциональное состояние организма школьников: умственной работоспособности, величины КЧСМ, объема аккомодации зрительного анализатора, учитывались также жалобы на общее и зрительное утомление. Продолжительность игровых занятий независимо от типа игры составляла для детей 7—10 лет — 30 минут, для детей старше 10 лет — 45 минут. Все испытуемые с интересом относились к этим играм.

Как эти игры влияли на функциональное состояние школьников? Наибольшее количество неблагоприятных сдвигов по всем регистрируемым показателям отмечалось после игры второго типа: ухудшились результаты компьютерного тестирования и почти в 2 раза чаще регистрировались неблагоприятные сдвиги работоспособности, отражающие выраженное утомление. При этом испытуемые значительно чаще жаловались на усталость и предъявляли астенопические жалобы (покраснение век, боли в области глаз, ощущение мельканий перед глазами и др.).

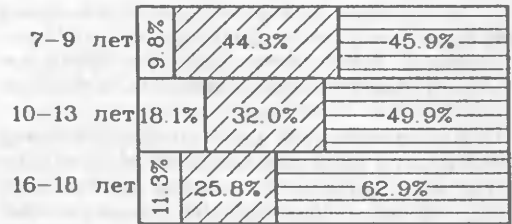
Исследования показали: чем младше дети, тем большее число неблагоприятных реакций наблюдается со стороны исследованных нами систем организма (после обоих типов игр). Так, после игры «Сокобан» ухудшение показателей функционального состояния организма отмечалось в 44,3% случаев у детей 7—9 лет, в 32,0% случаев — у детей 10—13 лет и в 25,9% случаев — у подростков 16—18 лет. Аналогичная динамика изменений показателей отмечалась и после игры «Тетрис». Однако число ухудшений показателей функционального состояния в этом случае было больше: в 58,9, 37,4 и 29,6% случаев. Таким образом, степень и частота утомления более выражена у школьников 7—9 лет, несмотря на то что продолжительность компьютерной игры у них была меньше.

Вывод: сравнение влияния разного типа игр выявило большую утомительность игры с навязанным ритмом работы, рассчитанную на скорость реагирования игрока на быстро сменяемую игровую ситуацию.

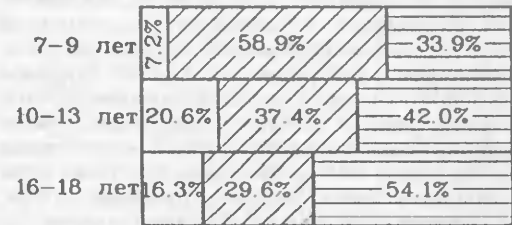
Применение компьютерного теста позволит педагогу регламентировать продолжительность занятий за компьютером каждого ребенка в зависимости от его возможностей, условий, в которых проводятся эти занятия.




Изменение показателей функционального состояния организма школьников разного возраста в результате игровых компьютерных занятий.

I тип игры
("Сокобан")



II тип игры
("Тетрис")



-  — улучшение
-  — ухудшение
-  — без изменений

Большое достоинство компьютерного тестирования — возможность проведения экспресс-диагностики функциональных возможностей детей в школах и других детских учебных заведениях педагогами без участия врачей.

Эта методика может быть использована для диагностики утомления взрослых пользователей ЭВМ.

Результаты компьютерного тестирования в основном совпадают с результатами, полученными при использовании других методов диагностики утомления школьников.

Разработанный нами компьютерный тест был использован при комплексном изучении функционального состояния организма детей в процессе внеучебных компьютерных занятий. Исследовались центральная нервная, сердечно-сосудистая системы, зрительные функции, работоспособность и субъективные ощущения учащихся, а также проводились хронометражные наблюдения на кружковых занятиях с ВДТ. Под наблюдением находилось

205 школьников I—VII классов, посещающих компьютерный кружок. Все учащиеся были практически здоровы.

Занятия в компьютерных кружках проводились во второй половине дня, после учебных занятий, 1—2 раза в неделю. Продолжительность их колебалась от 45 минут до 1,5 часа. Работа за компьютером проходила в свободном ритме. За дисплеем, как правило, занимался один ученик. Кабинеты вычислительной техники оборудованы ПЭВМ «Правец».

Анкетный опрос детей показал, что они проявляют к занятиям большой интерес. Однако, как показали наблюдения за поведением детей, такая увлеченность нередко маскирует утомление, вызванное работой с ВДТ. Дети, особенно младшего школьного возраста, нередко еще не в состоянии адекватно оценить свое самочувствие. Между тем с помощью объективных методов у них регистрируется утомление: снижается работоспособность, ухудшается работа аккомодационной мышцы глаза. Результаты хронометражных наблюдений показали, что, так же как и на учебных занятиях с использованием ВДТ, непосредственная работа с экраном дисплея занимает большую часть времени и составляет почти 73,0% всей их продолжительности. Этот вид деятельности — доминирующий и определяющий влияние на функциональное состояние организма детей.

Чем старше дети, тем более устойчивым становится уровень функционирования систем организма, обеспечивающих их компьютерную деятельность. Наиболее выраженные неблагоприятные сдвиги в работоспособности, функциональном состоянии ЦНС, сердечно-сосудистой систем и зрительного анализатора выявлены у младших школьников, и в первую очередь у первоклассников. Так, у них после занятий за ВДТ работоспособность снижалась в 5 раз по сравнению с ее уровнем до занятий. Наряду с этим уменьшалась скорость реагирования на световой раздражитель, снижалась величина КЧСМ, отмечалось значительное снижение объема аккомодации и увеличение коэффициента утомления цилиарной мышцы глаза, что особенно опасно, поскольку именно в этом возрасте происходит формирование рефракции глаза.

У учащихся 8 лет (II класс) неблагоприятные изменения работоспособности отмечались в меньшей степени, по сравнению с первоклассниками. Работоспособность второклассников после кружковых компьютерных занятий снижалась в 1,5 раза, по сравнению с ее показателем до занятий.

Лишь у трети школьников III класса отмечались неблагоприятные сдвиги работоспособности и незначительное снижение функциональных возможностей ЦНС.

У детей среднего школьного возраста — более устойчивое состояние ЦНС и работоспособность, которые под влиянием занятий за ВДТ практически не изменялись. Небольшие изменения у школьников V—VII классов регистрировались со стороны зрительного анализатора: отмечалось существенное снижение объема аккомодации, по сравнению с исходной величиной в обеих возрастных группах. Это вполне естественно, поскольку в первую очередь при работе с ВДТ страдают функции зрительной системы. Однако у пятиклассников эти неблагоприятные изменения более выражены. Так, коэффициент утомления цилиарной мышцы глаза после компьютерных занятий у них в 3 раза выше, чем у старших школьников.

Исследование сердечно-сосудистой системы до и после занятий с ВДТ не выявило специфических изменений. Реакция сердечно-сосудистой системы на компьютерную деятельность аналогична той, которую мы наблюдаем во время традиционных учебных занятий: снижение систолического, повышение или понижение диастолического давления, снижение частоты сердечных сокращений. Ухудшение гемодинамических показателей регистрировалось у учащихся начальной школы: снижение систолического давления наблюдалось в 51% случаев, а у учащихся среднего школьного возраста — в 40% случаев. У младших школьников выше число случаев увеличения систолического давления, чем у более старших детей, что свидетельствует не только о значительном напряжении сердечно-сосудистой системы во время компьютерных занятий, но и об увеличении статического компонента в режиме дня детей.

Сравнительный анализ результатов исследований показал, что степень утомления школьников в значительной степени определяется характером и содержанием компьютерных занятий в кружке. В зависимости от вида компьютерной деятельности все занятия были разделены на 3 типа: программирование, смешанного типа, игровые.

На занятиях программирования ребята учились составлять простейшие программы, вводили их в компьютер, осваивали возможности изобразительной деятельности, конструирования, сочинения музыки с помощью компьютера и т. п.

Игровые занятия целиком посвящались компьютерным играм. Игры, как правило, динамичные, рассчитанные на быстроту реакции, эмоционально напряженные, проводятся в темпе, который задает компьютер. Занятия смешанного типа сочетали программирование с компьютерными играми, которые обычно проводились в конце занятия (последние 20—30 минут).

Наиболее утомительными для детей как 7—10, так и 11—13 лет оказались занятия компьютерными играми. Хронометражные наблюдения показали, что на таких занятиях свыше 88% времени занимает непосредственная работа с экраном дисплея, на других занятиях эта величина не превышала 66—67%.

Помимо высокой плотности занятий большая утомляемость во время игр связана с тем, что они, как правило, проводятся в заданном ритме, требуют высокой концентрации внимания и быстрой реакции, что предъявляет повышенные требования к состоянию ЦНС. Так, комплексный показатель работоспособности школьников после игровых занятий снижается почти в 3 раза. В 48% случаев удлинялся латентный период зрительно-моторной реакции, у 57% детей существенно снижалась величина КЧСМ. Аналогичные, но менее выраженные сдвиги в функциональном состоянии отмечались у школьников 11—13 лет.

Игровая деятельность, особенно на начальном этапе обучения навыкам общения с компьютером, занимает наибольший удельный вес.

Имеющиеся в настоящее время в распоряжении преподавателей игровые программы носят преимущественно развлекательный характер. Чаще всего это остросюжетные, эмоциональные, нередко милитаристского характера игры. Если взрослые видят в этих играх возможность эмоциональной разрядки, то дети ищут в них источник напряжения, риска, самоиспытания. Практика показывает, что они очень привлекательны для детей и определенная категория школьников стремится сесть за компьютер лишь для занятий играми такого типа. Психологи видят в этом ряд негативных моментов, считая развлекательные компьютерные игры малополезными и даже вредными. Они снижают волевые качества детей, приучают к легкому времяпрепровождению и таят в себе опасность наркотизирующего эффекта.

Промежуточное положение по утомитель-

ности между компьютерными играми и занятиями смешанного типа занимают занятия программированием. После занятий программированием выраженное утомление отмечалось в 42% случаев, у 48% детей снижалась величина КЧСМ, увеличение латентного периода зрительно-моторной реакции регистрировалось в 34,9% случаев.

Наименее утомительными для школьников I—VII классов оказались компьютерные занятия смешанного типа. Частая смена деятельности и меньшая занятость на них непосредственной работой с экраном дисплея обеспечивают более устойчивый уровень работоспособности и функционального состояния. Занятия такого типа в большей степени соответствуют возрастным особенностям детей.

Изучение влияния компьютерных занятий разного типа позволило установить оптимальную и допустимую их продолжительность для детей разного возраста. Так, для учащихся 7—10 лет оптимальная продолжительность компьютерных игр составляет 30 минут, допустимая для игр и занятий смешанного типа — 60 минут. Для школьников 11—14 лет оптимальная продолжительность компьютерных игр составляет 30 минут, а допустимая — 60 минут, для занятий смешанного типа соответственно — 60 и 90 минут. При соблюдении рекомендуемой нами продолжительности работы с ВДТ отмечается устойчивый уровень не только функционального состояния ЦНС и работоспособности, но и зрительного анализатора.

Для снижения общего и зрительного утомления школьников, работающих с ВДТ, важно регламентировать не только продолжительность занятий, но и режим работы. Наиболее благоприятная динамика функционального состояния организма школьников отмечается в том случае, когда перерыв между уроками и началом занятий в кружке — не менее часа. Эффективное средство профилактики утомления — организация небольшого перерыва (10 минут) в середине занятий для подвижного отдыха и проведения специальных упражнений.

Контактный телефон: (095) 917-10-60.

Литература

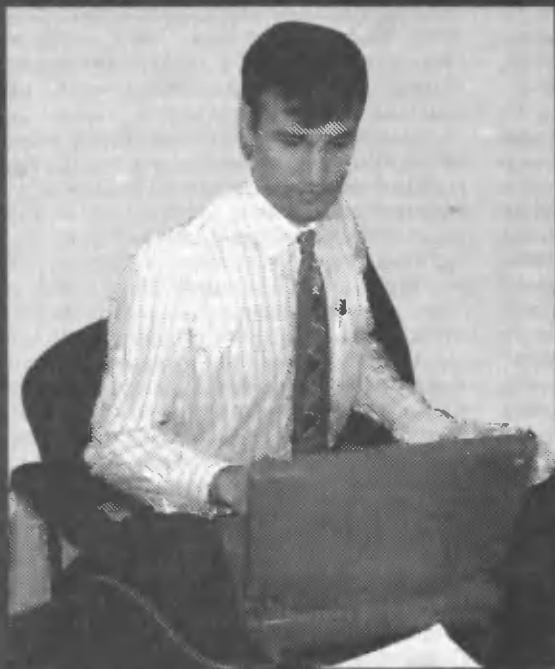
1. Розенблюм Ю. З., Ланцбург М. Е., Боковинов А. М. Зрительное утомление и умственная работоспособность при работе с дисплеем// Гигиенические проблемы компьютеризации общеобразовательной школы. М., 1988.

2. Шмелев А. Детская болезнь компьютерного всебуча// Информатика и образование. 1987. № 1.

3. Болтянский В. Г., Рубцов В. Р. Вопросы компьютеризации школьного обучения// Вопросы психологии. 1985. № 6.

4. Маргулис Е. Д. Компьютерная игра в учебном процессе// Советская педагогика. 1989. № 4.

5. Учиться отдыхать// Информатика и образование. 1987. № 5.



Компьютеры Macintosh широко и успешно применяются в сфере образования во всем мире. Однако в России они появились совсем недавно. Связано это с существовавшим запретом на поставки в СССР новейших технологий. Сейчас эти запреты сняты, и компьютеры Macintosh могут, наконец, прийти в любую школу.

Для компании Apple применение производимых ею компьютеров в образовании является одним из самых приоритетных направлений. Первой среди производителей компьютеров компания создала специальное подразделение, осуществляющее разработку стратегий использования компьютеров в образовании. Результатом работы этого отдела было создание современных моделей Macintosh, в которых были воплощены многочисленные предложения и пожелания учителей школ, преподавателей университетов, а также учеников и студентов.

Компания Apple предоставляет ряд специальных скидок для образования. Существуют

также специальные программы поддержки разработчиков учебного программного обеспечения и апробации новых путей использования этой техники и программ в учебном процессе (например, проект Apple Classrooms of Tomorrow — Эпловские классы завтрашнего дня).

Фирма RUI IMC Apple Computer, официальный представитель компании Apple в России, имеет серьезные намерения в продвижении компьютеров Macintosh на российский рынок. Мы будем поддерживать разработчиков программного обеспечения и предоставлять специальные скидки при продаже техники в сферу образования.

Конечно, сейчас компьютеров Macintosh в России не так много, но с каждым днем их становится все больше, на них стали ориентироваться целые регионы, а тот, кто начал работать с ними, практически никогда им не изменит.

Также мы рады сообщить вам, что Министерство образования РФ и фирма RUI IMC Apple Computer договорились о координации действий и проведении совместных мероприятий по созданию моделей использования современных информационных технологий фирмы Apple в образовании. Для их реализации разработана программа "Технологии Apple в образовании".

Сообщество пользователей компьютеров Macintosh ждет вас, мы приглашаем вас к сотрудничеству.

Генеральный директор RUI IMC Apple Computer

Дэвид Краускопф



APPLE

ДЛЯ ОБРАЗОВАНИЯ

М. Конашевич,

отдел образования RUI IMC Apple Computer

УЧИТЬСЯ ИГРАЯ

Можно долго спорить о перспективности использования компьютера Macintosh в российском образовании. Специалистами высказываются самые различные мнения — от полного неприятия до вытеснения в недалекой перспективе всех других платформ. Сегодняшняя реальность такова: компьютеры Macintosh уже стоят в наших учебных заведениях. Мы представляем нашим читателям новую рубрику «Apple для образования».

Уже более 10 лет работают компьютеры корпорации Apple Computer во всем мире в различных сферах образования — от детского сада до университета. 60% школ и 45% высших учебных заведений США используют компьютеры Macintosh. Популярна эта техника и в других странах. Причина успеха в том, что Apple Computer — единственная в мире компания, имеющая специальное отделение, разрабатывающее новые информационные технологии для образования. Совершенствуя свои компьютеры вместе со школьниками, студентами и преподавателями, фирма накопила уникальный опыт, позволивший создать лучшие образовательные программы и методики.

Среди многих преимуществ компьютеров Apple Macintosh важным является то, что они имеют наиболее дружелюбную среду, интуитивно понятную и взрослому, и ребенку. Пользователи этой техники отмечают простоту в обучении, совместимость с другими компьютерами,

высокую производительность, очень удобный графический интерфейс.

Принцип всего образовательного процесса на Macintosh — «play and learn» (учиться играя) — позволяет использовать компьютер как инструмент для обучения различным предметным дисциплинам. Благодаря операционной системе 7.1 (кстати, русскоязычной) Macintosh совместим практически с любыми операционными системами. Он не только работает с дискетами MS DOS и OS/2, но легко подключается к любой компьютерной сети, а также может исполнять программы, написанные для DOS, Windows или UNIX.

Все модели Macintosh, используемые в образовании, снабжены микрофонами и динамиками. Программное обеспечение позволяет записывать речь или музыку на жесткий диск и воспроизводить их. Это идеальный компьютер для реализации всего спектра современных технологий мультимедиа. Кроме того, каждый компьютер оснащен сетевыми средствами AppleTalk. Это позволяет легко объеди-

нить машины в локальную сеть и подключиться к телекоммуникационной сети AppleLink, соединяющей пользователей Macintosh во всем мире.

Для системы образования очень важна еще одна особенность Macintosh — высочайшее качество комплектующих и сборки. Ведь эти компьютеры производятся только в двух точках планеты, где созданы все необходимые для этого условия. Эта техника практически не ломается.

Все мониторы Macintosh обеспечивают очень высокую четкость и стабильность изображения, чистоту цвета и типографское качество букв, а экран компьютера, снабженный специальной внутренней защитной сеткой, безвреден для здоровья и удовлетворяет самому строгому на сегодняшний день шведскому стандарту.

Корпорация Apple Computer вышла на российский рынок совсем недавно. Фирма RUI IMC Apple Computer, официальный представитель фирмы Apple в России, и Министерство образования Российской Федерации приняли программу совместных действий «Технологии Apple в образовании», которая предусматривает создание нескольких базовых моделей использования компьютеров Macintosh в сфере российского образования.

Первой на путь оснащения школ компьютерами Macintosh встала республика Карелия. Еще в 1993 г. здесь была принята и начала осуществляться программа «Информатизация школьного образования на платформе Macintosh». В настоящее время завершены первый и второй этапы программы — оснащение вычислительной техникой Macintosh и базовым программным обеспечением школ г. Петрозаводска и районных школ Карелии, определяемых в качестве базовых для своего района. Третий этап начался в первом квартале 1995 года. (Подробно об опыте использования компьютера Macintosh в Карелии вы прочтете в одном из следующих номеров.) Он предусматривает оснащение компьютерами Macintosh школ Карелии, готовых принять участие в реализации Программы, а также дооснащение наиболее передовых базовых школ дополнительными техническими средствами.

Для поддержки Карельского проекта был создан Образовательный Apple Центр на базе Центра информационных технологий Петрозаводского государственного университета. Работает также Образовательный Apple Центр в Санкт-Петербурге.

Началось оснащение школ компьютерами Macintosh и в Москве. За период с августа по декабрь 1994 г. компьютерами Macintosh были оснащены порядка 50 московских школ, в том числе 21 школа-новостройка по проекту Московского департамента образования «Создание модели школьного информационного пространства».

На сегодняшний день уже более 250 дошкольных, средних и высших учебных заведений России используют компьютеры Macintosh. Кроме Москвы и Петрозаводска дети уже сидят за экранами этих компьютеров в Санкт-Петербурге, Тольятти, Самаре, Ижевске, Волгограде, Екатеринбурге, Томске, Сургуте, Новокузнецке, Владивостоке и других городах. В России уже создана сеть авторизованных дилеров, осуществляющих поставку, техническое обслуживание оборудования, обучение работе с ним и весь комплекс поддержки пользователей.

Фирма RUI IMC Apple Computer уделяет большое внимание стратегии работы на образовательном рынке России. Разработана система специальных скидок для образовательных учреждений. Отдел образования фирмы проводит учебу дилеров, координирует разработку русскоязычных программных продуктов для преподавания информатики и других предметов, занимается организацией семинаров, презентаций и встреч для ознакомления с возможностями компьютера Macintosh в сфере образования.

Следуя доброй традиции, фирма продолжает выступать спонсором ежегодного конкурса «Учитель года России», выражая таким образом свою признательность и уважение российским учителям.

С приходом Macintosh в наше образование у школ появилась возможность выбора. Фирма RUI IMC Apple Computer постарается сделать все, чтобы Macintosh стал массовым явлением в российской школе.

ЗАРУБЕЖНЫЙ ОПЫТ

Вилхо Хирви,

генеральный директор Главного управления просвещения Финляндии



РОЛЬ ОБРАЗОВАНИЯ В СОВРЕМЕННОЙ ФИНЛЯНДИИ

На пути к глобальному обществу взаимодействия

В настоящее время происходят глубокие перемены общественного и культурного характера как в международном, так и национальном плане. Ускорителями этого процесса служат всеобщая интернационализация, распад блоков, неустойчивость рыночного хозяйства и экономический кризис. На мировом рынке больше торгуют информацией и капиталом, чем товарами и услугами. Речь идет о новой индустриальной революции и росте производительности труда. Товары можно изготовить из все меньшего количества сырья, энергии и трудовых затрат, благодаря новым технологиям и автоматизации.

За последние годы финская сфера труда была быстрыми темпами реформирована, чтобы фирмы смогли успешно работать в объединяющейся Европе. Сейчас происходит переход от постиндустриального общества ко все более глобальному, сервисному, мозаикоподобному, инфоинтенсивному, интерактивному, где будет ра-

сти значение сотрудничества между людьми и странами, взаимной зависимости, новой информации и постоянного образования.

Информация может стать четвертым производственным фактором и важнейшим рычагом власти и богатства. В промышленности, например, новая техника уже сегодня стала важнейшим видом капитала.

Инфотехнологии, такие, как гипер- и мультимедиа, расширяют методологические возможности образования. Возникают новые медиа-школы и глобальные классы. Например, совсем недавно знающие язык ученики одной из школ Финляндии провели без особых трудностей сеанс космической связи с астронавтами на «Шаттле» с помощью компьютера и спутника связи.

Новые задачи образования

В настоящее время задача образования состоит в том, чтобы установить тесную связь с производством.

Делается акцент на более тесном сотрудничестве между народным хозяйством и учебными заведениями Финляндии в рамках различных программ развития, учебной практики на предприятиях, а также организации обучения на местах и в регионах, в так называемых центрах обучения. Региональное сотрудничество между различными отраслевыми профтехучилищами позволит обогатить формы модульного обучения.

Обучение новым технологиям и предпринимательству должно в первую очередь содействовать подъему в секторе малых и средних фирм, где появятся новые рабочие места. Роль непромышленного производства и услуг будет расти. В США, например, уже почти 80% всей рабочей силы заняты в сфере услуг.

Учащиеся — будущие специалисты, должны проявить умения во внедрении современных систем качества, инновационной разработке изделий, комплексного маркетинга, улучшения качества услуг, предприимчивости и универсального владения компьютер-

ной технологией. Ученик должен скорее быть мыслящим субъектом своего труда, а не объектом обучения.

Квалификационные требования меняются быстро

Реформируя образование, важно предугадать, какие произойдут изменения в характере труда и какие профессии будут иметь спрос. Образование должно обязательно опираться на новейшие исследования в области труда и профессий. Квалификационные требования меняются быстро, срок жизни профессий сокращается, и человеку придется менять профессию несколько раз в течение жизни.

В обучении пора больше обращать внимания на качества, которые необходимы для предоставления сложных услуг в таких сферах, как администрирование, управление, маркетинг или продажа.

Будут расти требования к квалификации. Сегодня такими значимыми квалификационными факторами считаются профессиональная или отраслевая квалификация, способность учиться, готовность к общению, выживаемость при переменах, инфоинтенсивные навыки. Труд в совокупности становится все более творческим: нет конструкторов и исполнителей.

Специалисты считают, что к 2017 г. СПОСОБНОСТЬ УЧИТЬСЯ будет самым главным качеством.

Высокая квалификация и широкая профессиональная образованность благодаря образованию

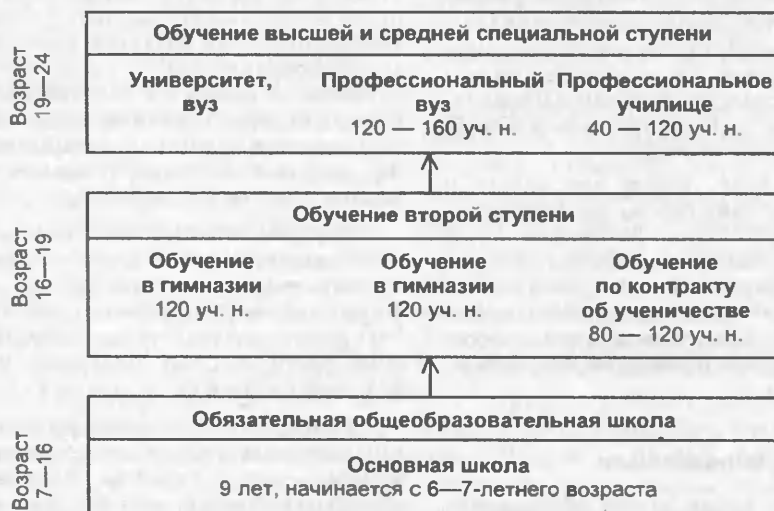
Высокая квалификация и широкая профессиональная образованность — это те основные качества, которые позволяют учащимся после поступления на работу справиться со своими задачами. Профессиональная образованность — это основательное и обширное профессиональное умение и общее образование.

В интернационализирующемся мире, в объединяющейся Европе профессиональная квалификация означает многосторонние знания об общественном строе разных стран, знание их культуры, экономики, торговли, производства и технологий. Образовательные системы должны приспосабливаться к этим требованиям, чтобы сделать возможным перемещение учащихся и рабочей силы.

Основные черты реформы образования

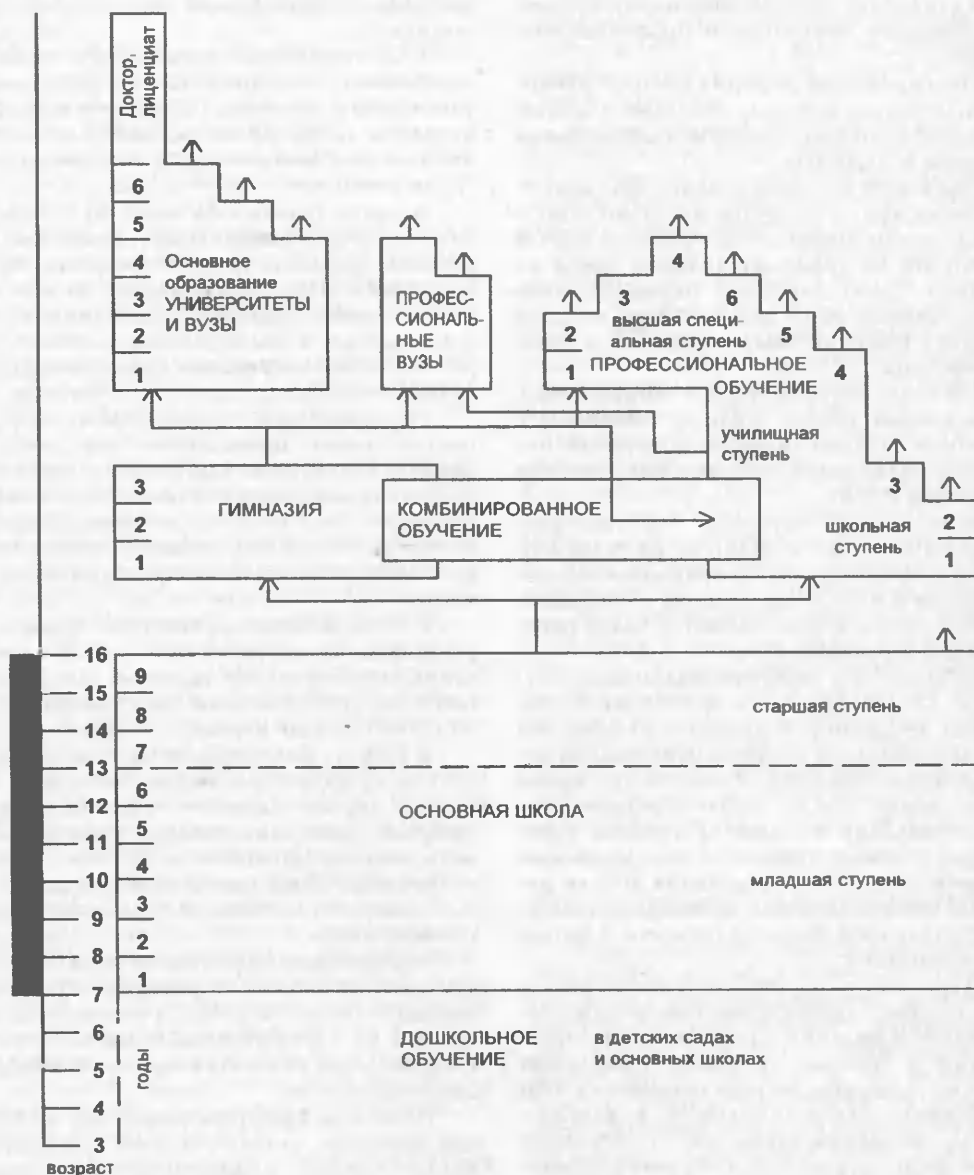
В Финляндии, как и во всех индустриально развитых странах, в недавнем прошлом была проведена реформа системы образования.

Начиная с 70-х гг. для всех возрастных групп введена единая основная школа. В дальнейшем естественным развитием стала реформа средней ступени, которая направле-



(Под учебной неделей подразумевается в среднем 40-часовая работа учащегося)

Схема образования в Финляндии



Структура образования Финляндии

В основной школе учатся в возрасте 7–16 лет. Основная школа подразделяется на шестилетнюю младшую ступень и трехлетнюю старшую ступень. После основной школы учащийся может перейти на трехлетнюю учебу в гимназию, в которой ведется общеобразовательное обучение, или на 2–5-летнюю профессиональную основную учебу. На профессиональное обучение можно поступать и после гимназии, в этом случае экзамен на специальность можно сдать за 1–4 года.

Лица, закончившие гимназию или получившие диплом о профессиональном образовании училищной или высшей специальной ступени, могут поступать на учебу в университет или профессиональный вуз. Профессиональные вузы являются экспериментальными, при помощи их ищутся способы поднятия уровня профессионального обучения. В профвуз могут поступать также и те, кто сдал экзамен на специальность школьной ступени соответствующей или близкой по характеру сферы. Получение основного высшего образования в университете длится в среднем 4–6 лет, а учащиеся профессиональных вузов заканчивают учебу за 3–4 года.

на на изменение соотношения между общеобразовательной гимназией и профтехобразованием.

Была проведена реформа высшей школы.

Нынешнюю реформу последнего десятилетия можно назвать реформой образования молодежи и взрослых.

Еще в 80-е гг. планирование объемов образования велось на основе довольно детальных прогнозов потребности в рабочей силе, и отчасти это по-прежнему является одной из исходных точек. Детальное государственное регулирование стало более общим, и право принятия решений было передано на местный уровень.

Все более остро ощущают потребность в образованных кадрах отрасли, связанные с развитием научных знаний и технологий, интернационализацией, изменением структур экономики и т. п.

В Финляндии обучается 200 тыс. учащихся в 500 профтехучилищах. До недавнего времени училища были сравнительно небольшими и имели определенную специализацию. Сегодня образуются более крупные и универсальные.

Государство и коммуны, как правило, отвечают за содержание и финансирование учебных заведений. В прошлом руководство и финансирование системы образования велись централизованно. В настоящее время школы имеют все большую независимость при составлении учебной программы, организации учебного процесса и использовании ресурсов. В системе образования 90-х гг. акцент делается на широкие возможности выбора для учащегося, большей гибкости и интернационализации.

Владельцами профтехучилищ являются коммуны, государство и частные фонды. Находящиеся в ведении промышленных предприятий и торговли училища выполняют функцию в первую очередь повышения квалификации. Муниципальные и частные учебные заведения также имеют государственное финансирование, и обучение организовано в них по тем же принципам, что и в государственных. Обучение, как правило, бесплатное.

Задача реформы средней ступени в 80-е гг. заключалась в развитии после девятилетнего образования некоей единой школы. Идея состояла в том, чтобы сделать профтехобразование конкурентоспособным гимназическому и обеспечить таким образом наличие образованной рабочей силы.

Структура профтехобразования была унифицирована: образованы широкие отраслевые и функциональные профили (26 основных профилей) по 250 специальностям,

которые соответствовали потребностям экономики.

В настоящее время около 90% из числа окончивших основную школу продолжают учебу либо в гимназии (55%), либо в профтехучилище (35%). Диплом вуза или профтехучилища получают почти 80% всех возрастных групп учащихся.

В рамки профтехобучения на базе девятилетки входят обязательные общеобразовательные предметы: родной язык, иностранные языки, математика, физика, химия, вычислительная техника, обществоведение, физкультура и гигиена, художественное и экологическое воспитание. Они должны обеспечить возможности продолжения учебы.

Обучение всем профессиям на базе основной школы продолжается как минимум два года или больше. Профессиональная подготовка на базе основной школы по половине специальностей занимает три года. Обучение на базе гимназии (если общеобразовательных предметов нет) продолжается в среднем на год меньше.

В целях повышения престижа профтехобразования была создана возможность поступления из него в вуз. На практике, однако, доля таких выпускников среди поступающих в вузы остается очень низкой (3—4%).

В 80-е гг. была создана система учебных планов профтехобразования, которая в основных чертах сохранена и по сей день. В профтехобразовании средней ступени внедрены общегосударственные основы учебных планов по профессиям, на базе которых учебные заведения составляли свои собственные учебные планы.

Реформа средней ступени была началом динамичной работы по развитию профтехучилищ, и около 80 новых училищ было открыто в 80-е гг. Преобладающая часть их — это небольшие училища социальной сферы и здравоохранения.

На нужды профтехобразования и обучения взрослых направляется 30% государственных средств, выделенных бюджетом на образование.

Стратегия развития

Реформа средней ступени не привела в полной мере к ожидаемым результатам.

Популярность гимназии по-прежнему оставалась выше популярности профтехучилища. Тогда начался поиск новых форм обучения. Было решено провести эксперимент с «молодежной ступенью» и профессиональными вузами. Стояла задача — повысить престиж профессионального образования, выра-

ботать новые требования к различным квалификациям.

Эксперимент заключается в поиске сочетания новых форм общего и профессионального образования с учетом индивидуальных интересов учащихся, а именно:

- гимназисты могут обогащать свою учебу профессионально-техническими знаниями;
- учащиеся профтехучилищ могут получить более многостороннюю квалификацию, выбирая курсы в других училищах или изучая общеобразовательные предметы в гимназии;
- приоритетным является новый диплом, который выдается после комбинированного обучения на основе индивидуального подхода к обучению;
- экзамены на профессиональную квалификацию и в гимназии можно сдать параллельно;
- в диплом можно включить занятия, полученные в культурно-просветительской системе, вузах, на производстве, учеба может происходить в рамках проектов.

Занятия можно выбирать. Их доля может составлять 30—40% всего объема учебной программы. Сроки учебы зависят от содержания занятий, но экзамены на аттестат или диплом необходимо сдать за 2—4 года.

Образовательный эксперимент с «молодежной ступенью» начался в 1992 г. на 15 экспериментальных площадках (140 учебных заведений, 13 500 учащихся).

Временные высшие профтехвузы были образованы в таких учебных заведениях, где велась профтехподготовка на базе профтехучилища или высшей школы. Это многоотраслевые учебные заведения были организованы по европейскому образцу, хотя по сравнению с ними они более мелкие.

Формы обучения похожи на работу в высшем учебном заведении. Вместо школярства идет более самостоятельная учеба в разнообразных формах.

Экспериментальные учебные заведения составляют учебные планы самостоятельно. Появилось много новых и оригинальных учебных программ, в которых идет поиск интеграции обучения нового типа.

Разработаны курсы, ориентированные на международный рынок труда, где обучение ведется на иностранном языке.

Профтехвузы должны наряду с университетами стать учебными заведениями высшего образования, которые имеют административную независимость и самостоятельность от остальной системы высшей школы и профтехобразования. Диплом профтехвуза считается дипломом высшей школы.

Наряду с вышеописанными экспериментами Главное управление просвещения приступило к обширной работе по обновлению учебных планов основной школы, гимназии и профтехучилищ.

Акцент делается на повышение уровня обучения, обновление содержания обучения и расширение возможностей для индивидуального выбора. Внедрение новых учебных планов произойдет в 1994—1996 гг.

Учебные планы профтехобразования стыкуются с учебными планами основной школы и гимназии. В системе обучения взрослых вместо учебных планов будут использоваться квалификационные уровни, но профессиональные задачи обучения молодежи и взрослых остаются единими.

В современном профтехобучении больше возможностей для педагога: стремление научить учащихся учиться, активизировать их инициативу и предпринимательство, способность к решению проблем, связанных с социальным воздействием и сотрудничеством, подготовить к интернациональным контактам. Воспитать ответственность за окружающую среду — ведущая задача для всех форм обучения. В современном профтехобучении больше возможностей и для выбора учащихся. Научить учащихся активизировать инициативу, способность к самостоятельному решению проблем, сотрудничеству, подготовить их к интернациональным контактам — ведущая задача всех форм обучения 90-х гг.

Сближение школы и производства

Взаимодействие образования и производства осуществляется в Финляндии в рамках различных проектов развития, прохождения учебной производственной практики на фирмах, а также при организации обучения на местах или в регионах в так называемых центрах учения или умения. Региональное сотрудничество между учебными заведениями различных профилей может служить обогащению форм модульного обучения.

Примером сотрудничества между производством, в первую очередь сектором мелких и средних фирм, и училищами служит проект под названием «Программа развития дизайна». В рамках названной программы дальше всего продвинулся проект «СИЛВА» по созданию экспортной мебели. В сотрудничестве между промышленностью и ремесленными и художественно-прикладными училищами были разработаны модели мебели, которые с успехом показывались на международных мебельных ярмарках.

Акцент в стратегии образования направ-

лен на то, как ведется обучение и меньше — что преподается.

Подведем итоги

Говоря о направлениях реформирования системы образования в Финляндии, следует говорить прежде всего об изменении соотношений между общим образованием, профтехобразованием и высшим.

Что это за изменения?

- Задачи обучения. Задачи общего и профтехобразования сближаются. Широкое профессионально-техническое образование включает в себя, например, коммуникативную готовность, знание иностранных языков, способность решать проблемы, инновационность, социальную готовность, экологическую сознательность, а также общие гражданские и жизненные позиции. Задачи основного профтехобразования уточняются с тем, чтобы желание приобрести профессию не вело к слишком узкой специализации, а ориентировало человека на выбор различных возможностей. Новая задача образования на всех уровнях состоит в том, чтобы поддерживать инициативность и занятость (например, предпринимательская подготовка, фирменные «инкубаторы», молодежные мастерские, направление на работу, программы развития, образующие новые рабочие места).
- Уровень образования. Внутренняя селективность образовательной системы осуществляется в довольно традиционной форме, хотя внешние ее проявления меняются. Гимназия по-прежнему предлагает более широкие возможности для продолжения учебы, чем профтехобразование. Все чаще образование не определяет карьеру, уровень зарплаты и общественное положение, как это было раньше. Даже хорошее образование не гарантирует выхода на рынок труда, хотя образование во все большей мере является жизненно важным качеством на интернационализирующихся рынках труда. С точки зрения реализации прав человека требования к полноценному образованию становятся еще более высокими.
- Структура учебы и способы работы унифицируются. Тенденция к выбору внутренней гибкости и индивидуальным программам обучения присутствует на всех уровнях. Учебные программы профтехвузов и высшей школы по своей структуре все более сближаются. Однако, в принципе, между учебными заведениями существует четкое разделение труда. Академические вузы занимаются подготовкой научных кадров и научно-исследовательской работой, в то время как профтехвузы ведут прикладное обучение и прикладные исследования. Тем не менее во многих сферах их деятельности прослеживается параллельная работа и конкуренция.
- Административно-организационные структуры. С одной стороны, налицо стремление к административной самостоятельности, как это наблюдается в профтехвузах, а с другой, желание сохранить нынешнее административное положение, как в гимназии. Одновременно усиливается функциональное сотрудничество на всех уровнях, в результате чего появляются центры учения и открытая учебная среда. В будущем хорошие школы будут работать в сетевых структурах, которые смогут предложить многосторонние возможности организовать непрерывное обучение. Обучение привязано к различным проектам развития, которые призваны содействовать местному развитию и хозяйственной деятельности.
- Стыковка курсов в горизонтальном и вертикальном планах реализуется постепенно. Система образования предлагает более гибкие возможности для учебы на продолжении всей жизни.
- Количественные соотношения обучения меняются. В утвержденном Государственным советом плане развития прогнозируется, что гимназия и впредь будет расширяться в таком объеме, чтобы около 60% от всей возрастной группы получило общее среднее образование. Профтехвузы будут обеспечивать образование (вместе с академическими вузами) 60—65% от состава возрастной группы. Это приблизительно соответствует нынешнему количеству набора в профтехучилища и на высшую ступень. Это означает, что количество людей с профобразованием значительно возрастет по сравнению с 80-ми гг.
- На всех уровнях происходит интернационализация. Уже сегодня профессионально-технические учебные заведения и вузы Финляндии участвуют в программах Европейского союза и практикуют активный двухсторонний обмен. Проводимая в северных странах политика в сфере образования соответствует современным образовательно-политическим позициям, которые были высказаны в последнее время в Европейском союзе.

Сиркка-Лийса Кярки,

советник просвещения Главного управления просвещения Финляндии

СОВРЕМЕННЫЕ ТЕНДЕНЦИИ В ПЕДАГОГИКЕ ФИНЛЯНДИИ

Задача обновляющейся педагогики состоит в том, чтобы создать условия для роста и развития молодого человека. Задача не из простых. Как повысить престиж образования, как заинтересовать молодежь в приобретении знаний и умений, необходимых обществу? Вопрос не только в том, каково должно быть содержание образования, но и в том, к каким предметам и каким формам самостоятельной работы у учащегося будет пробужден интерес.

Педагогика, нацеленная на стимулирование самостоятельной учебы, строится на представлении об общественно активном человеке.

Согласно определенным целям учащийся сам решает, что, когда, каким образом и в каком порядке он будет изучать. Увеличение объема самостоятельных занятий делает учебу более приятной, но и более ответственной.

Переход к таким формам работы, в которых акцент делается на самостоятельное приобретение знаний, означает изменение культуры учебного заведения и роли преподавателя в учебном процессе. Преподаватель корректирует, направляет работу учащегося.

В Финляндии сотрудничество между производством и школой углубляется путем обновления содержания и форм учебных курсов, реализации более тесной связи преподавателей с производством, увеличения объемов обучения на предприятии.

Центры образования — это школы будущего. В них найдут место новые формы и методы обучения. В них реализуется идея профессионально-технического центра обучения.

Уже сейчас подготавливается база для создания таких центров. Работа ведется в следующих направлениях:

- Большинство учебных программ составлено на индивидуальной основе.
- Новые учебные планы, которые состоят из модулей, предлагают учащимся разнообразные возможности для прохождения

учебы по индивидуальным планам. В них могут войти занятия по профессиям в разных учебных заведениях.

- Учеба может протекать и в рамках проекта, приближенного к условиям работы и реальной жизни. Такой проект развивает предпринимательские качества и становится часто частью «проекта всей жизни» учащегося.
- Режимы работы и учебы должны быть гибкими. Мастерские, читальные залы и компьютерные классы предоставляются в распоряжение учащихся и в вечерние часы. В некоторых странах Европы есть примеры, когда профтехучилища работают с семи утра до десяти часов вечера.
- Открытая учебная среда определяет новые требования к планировке учебных помещений. Требуются различные комнаты для библиотеки, уголки для самостоятельной работы и залы.

Центры образования поддерживают связи с другими учебными заведениями данного региона. Цель сотрудничества — предлагать самые разные образовательные услуги и учебные программы.

В сотрудничестве участвуют профтехучилища, гимназии, высшие учебные заведения и научно-исследовательские центры, фирмы, учреждения, муниципальные хозяйственные структуры, молодежные организации и т. п. Многие профтехучилища имеют партнеров в других странах. Участие в международных программах и экспериментальных проектах постоянно расширяется.

Сотрудничество не самоцель. Оно оправдано лишь в тех случаях, когда приводит к эффективному использованию ресурсов и помогает объединять различные курсы, служит стимулом к дальнейшему совершенствованию содержания и методов обучения.

Центр обучения поддерживает активные контакты с производством.

Специальные учебные программы наце-

лены на создание новых форм предпринимательства и поиск новой стратегии производства и деятельности фирмы. В качестве примера можно привести программу развития дизайнера, которая охватывает проект по разработке и маркетингу экспортной мебели, дизайн изделий из дерева и камня, промышленный дизайн и конструирование с помощью ЭВМ, проект по развитию стекольной промышленности, текстильной и швейной промышленности.

В целях развития сельских профессий и хозяйственной деятельности на селе начат проект, нацеленный на поддержку структурных изменений на селе и оценку того, каким образом гарантировать сельскому предпринимателю необходимые, но высококачественные услуги по обучению, консультированию и исследованиям.

В проекте участвуют профтехучилища, отраслевые консультационные организации, научно-исследовательские центры, официальные власти и центральные организации.

Обучение предпринимательской деятельности заключается в присоединении к фирменным сетям. Обучение предпринимательству — это, например, открытие учащимися магазина, работающего в стенах учебного заведения.

Центр обучения применяет развитую технологию обучения. Электронные средства инфо- и телекоммуникации используются как для управления, так и для обучения. Поиск информации за стенами учебного заведения ведется по каналам инфосетей в библиотеках и базах данных, которые могут быть и международными.

Компьютер находит широкое применение при изучении многих предметов, и компьютерное обучение применяется в большом объеме. Хорошо продуманные формы компьютерного обучения углубляют практические навыки учащихся. Тренажерные и математические модели позволяют наглядно представить трудноусваиваемые вопросы и ситуации, которые нельзя смоделировать в лабораториях. Использование развитых форм мультимедиа и гипермедиа, в которых сочетаются звук и изображение, и неограниченные возможности компьютера, расширяют рамки изучения искусства, техники и пр.

Наряду с учебниками и компьютерным обучением используется продукция средств массовой информации и аудиовизуальные пособия. Преподаватель не всегда должен присутствовать в классе. Иногда он ведет занятия по аудиовизуальной связи, конференц-связи или кабельному телевидению из другого учебного заведения. С помощью спутников можно передавать учебные программы с другой стороны земного шара. Благодаря новой технике можно учиться у себя дома или на рабочем месте.

Библиотечные и информационные услуги должны быть доступны в центре обучения. В результате проведенного эксперимента в профтехвузе разработаны интересные планы дальнейшего развития библиотечных и информационных услуг.

Развитая технология обучения тем не менее хороший слуга, но не хозяин. Техника никогда не заменит человеческое общение ученика и преподавателя. Новая технология обучения, вероятно, раскроет такие горизонты, о которых мы сегодня и не подозреваем.

Степень развитости технологии обучения можно оценивать по следующим критериям:

- наличие универсальных средств и развивающих пособий;
- компьютерное обучение, использование аудиовизуальных и мультимедиа-технологий;
- открытые и многосторонние библиотечные и информационные услуги;
- электронная телекоммуникационная связь и инфосети;
- учебная контактная сеть на базе новой техники.

Центр обучения должен служить примером нового учебного заведения и в новых формах управления. Это прежде всего воспитание высококвалифицированных кадров и общественно активных членов общества.

Центры работают на коммерческой основе. Принятие решений происходит на уровне учебного заведения и его хозяйственных органов. Системы оценки и качества работы гарантируют хороший результат и имидж у контактных групп и пользователей услугами.

Мы верим в те ценности, которые заложены в идее центров.

Лээна Коски,

Управление просвещения Финляндии

ОБУЧЕНИЕ ПРЕДПРИНИМАТЕЛЬСКОЙ ДЕЯТЕЛЬНОСТИ

Традиционно обучение предпринимательской деятельности молодежи в Финляндии велось в коммерческих училищах и частично в техучилищах. В последнее время характер предпринимательской деятельности меняется. Новые рабочие места все чаще появляются в мелких фирмах, а не в крупных или в системе местного или государственного управления. В Финляндии сегодня идет интенсификация деятельности мелких фирм, что ставит определенные проблемы перед системой образования. От работников все чаще требуется умение работать в составе различных групп и программ, понимать значение предпринимательской деятельности. Требования к работнику все возрастают.

Обучение предпринимательской деятельности охватывает изучение как внешнего, так и внутреннего предпринимательства. Внешнее предпринимательство — это работа в качестве предпринимателя. Внутреннее предпринимательство — это предпринимательское отношение работника к труду при работе по найму. Внутреннее предпринимательство должно стать частью всего образования, начиная от основной школы и кончая обучением взрослых. В основной школе дети могут изучать экономические вопросы в объеме коммерческого предмета. В гимназии преподаются основы предпринимательской деятельности более глубоко. В Финляндии уже работают гимназии с предпринимательским уклоном. В профтехучилищах при обучении коммерции и управлению обязательными являются такие предметы, как конторская работа, обслуживание клиентов, предпринимательская деятельность. По всем профессиям учащимся предлагаются все более широкие возможности выбора занятий. Сами учебные заведения могут определить свой профиль в зависимости от потребностей местной экономики.

Важное место занимает обучение предпринимательству и в системе обучения взрослых. Тот, кто уже работает предпринимателем или хочет заняться предпринимательством, может сдать экзамен на квалификацию предпринимателя. Курс состоит из десяти разделов: бизнес-идея, основание фирмы, формы фирмы, контрактное право, маркетинг, работа по сбыту, обслуживание клиентуры, составление сметы, расчет расходов и доходов. Для работающих на фирмах предлагается множество краткосрочных курсов повыше-

ния квалификации. В Финляндии есть курсы для безработных, где много внимания уделяется предпринимательской подготовке.

Базовая квалификация предпринимателя

ПЛАНИРОВАНИЕ ФИРМЫ

Бизнес-идея

ПРЕДПРИНИМАТЕЛЬСТВО И РОСТ ПРЕДПРИНИМАТЕЛЯ

Фирменная сеть/сотрудничество

ФИРМА И ОБЩЕСТВО

Основные факторы народного хозяйства

Основание и основные виды фирм

Контрактное право

МАРКЕТИНГ

Основы и планирование маркетинга

Личная работа по сбыту и с клиентурой

ЭКОНОМИКА ФИРМЫ

Основы бухгалтерии, баланс и основные показатели

Налоги фирмы, налог с добавленной стоимости

Смета, расчет доходов, ценообразование

Высшая квалификация предпринимателя

ПЛАНИРОВАНИЕ ФИРМЫ

Основы планирования фирмы

План хозяйственной деятельности

Планирование производства

Материально-техническое снабжение/логистика

Внешние связи

ПРЕДПРИНИМАТЕЛЬСТВО И РОСТ ПРЕДПРИНИМАТЕЛЯ

Взаимодействие

Умение управлять

План повышения квалификации

Интернационализация

ФИРМА И ОБЩЕСТВО

Конкурентное и рыночное право

МАРКЕТИНГ

Разработка продукции

Исследование рынка

Маркетинговая коммуникация

Международный маркетинг

ЭКОНОМИКА ФИРМЫ

Управление заработной платой

Планирование инвестиций и финансирование инвестиций

Контроль над факторами риска

ЭВМ в управлении экономикой

ЗАДАЧИ ПО ОТРАСЛЯМ

Отраслевое законодательство и т. п. (определяется по отрасли)

М. Виве,

профессор университета г. Ля Ман,

главный редактор журнала

«Sciences et techniques éducatives»

ОБРАЗОВАТЕЛЬНАЯ ИНЖЕНЕРИЯ

В 1994 г. во Франции родился новый журнал «Sciences et techniques éducatives». Его научным содержанием и изданием руководят программный и редакционный комитеты. Журнал призван освещать проблемы в области образовательной инженерии. Ниже публикуется с сокращениями статья главного редактора М. Виве, в которой он рассказывает об основных целях этого издания. Редакция журнала «Информатика и образование» готовит проект соглашения об обмене материалами между журналами. Мы просим заинтересованных читателей присылать в редакцию свои предложения.

Что следует понимать под понятием «образовательная инженерия» («Sciences et techniques éducatives»)?

Это понятие отражает, на наш взгляд, весь круг проблем, касающихся концепций, реализации (включая разработку инструментов реализации), конструирования и внедрения технических устройств, предназначенных для облегчения и улучшения учебного процесса. В журнале будут представлены информационные и коммуникационные технологии (ИКТ), которые играют важную роль в конструировании компьютерных сред обучения. При реализации технологий учитываются не только их содержание и применение, но и экономические и гуманитарные факторы, изучаются условия и среда, в которых эти технологии используются. Например, можно отметить большие различия между использованием таких технологий школьниками при их начальной подготовке и взрослыми при профессиональной переподготовке специалистов на предприятиях. Значительная разница существует также в ситуациях, когда обращаются к малоквалифицированным слушателям с целью их общей подготовки или к специалистам высокой квалификации, которые нуждаются в дополнительной подготовке на современной технике.

Образовательная инженерия и качество образования взаимосвязаны. Качество образования многогранно: это и качество в определении потребностей, качество продукции (программной или аудиовизуальной), качест-



во условий разработки этой продукции. В свою очередь, качество продукции связано с содержанием, формой, общей эргономикой, с качеством взаимодействий, а также с качеством пользовательской документации, описанием эксплуатационных требований и условий, благоприятствующих их реализации в процессе обучения. С методологических позиций весьма важно добиться высокой степени интеграции ИКТ в полную систему непрерывного обучения и удовлетворить различные критерии качества.

Образовательная инженерия формирует необъятное поле исследований на перекрестке многих наук. Термин «наука» «Sciences et techniques éducatives» относит к различным научным дисциплинам, способствующим ее развитию: информатика поддерживает прочные отношения с искусственным интеллектом, когнитивными науками, лингвистикой, эргономикой, психологией, социологией, дидактикой.

Образовательная инженерия остается пока недостаточно распространенной во франкоязычных странах, и в частности во Франции. Поэтому результаты, полученные в исследовательских лабораториях, находят слишком узкое применение, поскольку публикуются, как правило, на английском языке.

Журнал «Sciences et techniques éducatives» станет организатором обменов, центром разработки и передачи знаний и умений, звеном, объединяющим деятельность всех тех, кто занимается проблемами образовательной

инженерии. В частности, здесь найдут свое отражение европейские проекты, развертывающиеся в этом направлении, — ESPRIT, DELTA, COMETT и др.

Журнал «Sciences et techniques éducatives» ориентирован на разные категории читателей, сотрудничество с которыми представляется нам желательным. Это:

- педагоги всех направлений и уровней подготовки, включая администраторов системы образования, использующих так называемые новые информационные технологии;
- дидакты различных дисциплин, компетенция и исследование которых нацелены на создание и использование продуктов образовательной инженерии;
- специалисты в разных технических дисциплинах: информатики разных специальностей, таких, как человеко-машинный интерфейс, графические приложения, искусственный интеллект, вычислительная техника, а также специалисты в средствах коммуникации и периферийной технике, включая носителей изображения и звука, электронные модели, управляемые микророботы и другие устройства;
- специалисты когнитивных наук;
- эргономисты, специалисты по коммуникациям, например графисты, которые, умея передавать сообщения в виде рисунка, могут формировать композицию экрана;
- социологи в сфере образования, изучающие проблемы социального включения новых технологий, исследующие условия социализации знаний, получаемых в средах, которые реализуют эти технологии;
- лингвисты, которые могут участвовать в обсуждении условий, порождающих изменения, привносимые и зачастую навязываемые в лингвистическом плане информационными и коммуникационными технологиями в способы выражений и структуры высказываний, лингвисты, которые изучают глубокие связи между языком и знаниями;
- преподавателей, ведущие переподготовку педагогов, роль которых состоит в передаче технических, экономических, педагогических, социальных знаний, необходимых для интеграции новых технологий в учебные среды на хорошо освоенной научной основе.

Многообразие подходов

Журнал «Sciences et techniques éducatives» рассматривает новые подходы к современной технике: применение информационных носителей мультимедиа и телекоммуникации. Рассматриваются пользовательские подходы в педагогическом аспекте с точки зрения отхода от традиционных схем.

Международные контакты в этой области разнообразны и активны. Общепризнаны экономические, социальные, гуманитарные аспекты реализации ИКТ для обучения людей, проекты индивидуализации обучения и дистанционного обучения, которые позволяют надеяться на то, что удастся частично избежать противоречий, возникающих при традиционных организационных схемах учебного процесса.

- Комиссия Европейского экономического сообщества выступила с инициативой таких проектов, как DELTA и COMETT;
- OCDE (Организация сотрудничества и экономического развития) с помощью CERI (Центр исследований и инноваций в образовании) организует встречи и публикует аналитические отчеты о состоянии компьютеризации образования;
- ЮНЕСКО исследует потенциальные возможности, в частности, в развивающихся странах;
- НАТО организовало с 1988 по 1993 гг. в рамках специальной программы «Advanced Educational Technology» («Передовые образовательные технологии») регулярные встречи, посвященные этим вопросам. В издательстве «Шпрингер Ферлаг» издана серия отчетов НАТО;
- ИФИП (Международная федерация по обработке информации) создала технический комитет (ТСЗ) по этой теме. Эта федерация организует тематические конференции, в числе которых недавняя конференция «Teleteaching-93» в норвежском городе Торндехейм;
- ААСЕ (Ассоциация за внедрение компьютеров в образование), которая проводит конференцию ED-Media (средства мультимедиа в образовании) и конференции «Искусственный интеллект и образование» (AI&ED).

В таком контексте международных исследований внутри AFCET (Французская ассоциация кибернетики, экономики и техники) была создана рабочая группа «Информационные технологии в образовании и переподготовке» (GTIF).

Роль AFCET и рабочей группы «Информационные технологии в образовании и переподготовке» (GTIF)

Внутри AFCET рабочая группа GTIF прикреплена к техническому комитету «Искусственный интеллект и распознавание образов». Она взяла на себя обязанность обсуждения, выработку предложений и объединительные функции по терминологии в области новых информационных технологий, формам выражений и коммуникаций, которых не хватает во франкоязычной общине.

Главные задачи этой рабочей группы таковы:

- объединить исследователей, которые занимаются использованием информационных технологий (информатика и/или аудиовизуальные средства) и коммуникаций в области образования и переподготовки;
- содействовать общению исследователей, обмену и распространению информации;
- установить необходимые связи между исследователями и пользователями с целью определения стоимости разработок;
- разработать структуру диалогов, позволяющую проектам европейского значения участвовать во франкоговорящем мире.

Объединительная роль рабочей группы была главенствующей с момента ее создания в 1988 г. Это позволило объединить не только инициативы в среде самой Ассоциации AFCET, но и предложения внешних источников инициатив из INRP (Национальный институт педагогических исследований) и ряда других организаций. Таким образом, AFCET образовала междисциплинарный форум, в который смогли войти различные участники, имеющие дело с использованием ИКТ в не ограничиваемых школой задачах подготовки кадров.

Среди тем, наиболее часто называемых в проблематике рабочей группы, выделяются:

- приложения искусственного интеллекта к образованию и подготовке кадров (в частности, все то, что сегодня обозначают как интерактивные компьютеризированные обучающие среды (EIAO));
- педагогические приложения телематики (дистанционной обработки информации) и процессы дистанционного обучения;
- разработка методического обеспечения компьютеризованного учебного процесса и создание педагогических сред;
- средства мультимедиа и аудиовизуальная техника;
- педагогическая робототехника и создание

микро-миров, наполненных специфическим содержанием;

- темы когнитивные, эргономические, дидактические и педагогические;
- экономические и технические аспекты развитого инструментария;
- социокультурные следствия информационных технологий.

Цель журнала — объединить различные подходы к «открытым» проблемам. Назовем в качестве примера понятие, относящееся к кооперативной, коллективной работе, — *collecticiel* [«коллектисель», по созвучию с другими французскими терминами из прикладной информатики: *logiciel* — программное обеспечение, *matériel* — техническое обеспечение, *didacticiel* — методическое обеспечение] (англосаксонцы сказали бы «groupware») [ср. *software*, *hardware*, *firmware*], понятие инженерии знаний.

Журнал «Sciences et techniques éducatives» намерен сгруппировать публикации, описывающие исследовательские разработки с конкретным содержанием. Мы планируем широко представить работы молодых исследователей. Будут введены рубрики новостей.

Журнал рассчитывает на широкое взаимодействие с заинтересованной научной общественностью и профессионалами. Ставка велика, если учесть, что сегодня франкоговорящие исследователи слишком часто вынуждены публиковать свои работы на английском языке — языке, который не воспринимается значительной частью организаторов в системе подготовки кадров (на предприятиях или в образовательных учреждениях, таких, как университетские институты усовершенствования преподавателей во Франции). Это означает, что англосаксонцы имеют доступ к нашей практике и результатам исследований, тогда как во франкоговорящем пространстве эти результаты остаются недоступными.

Цель журнала — способствовать развитию перспективных инноваций и совершенствованию образовательных технологий и подготовке кадров.

Поэтому читательский корпус, на который ориентируется журнал, включает широкие круги профессионалов: в области программного обеспечения — программистов; педагогических методов использования компьютеров — методистов.

Журнал «Sciences et techniques éducatives» выпускается издательством AFSET/Hermes. Периодичность — четыре номера в год. Каждый номер состоит примерно из 96 страниц.

Мы благодарим читателей, которые будут читать журнал, давать советы авторам и обеспечивать желаемое качество.

А. В. Демаков, М. В. Лебедев

ИСПОЛНИТЕЛИ В СИСТЕМЕ КУМИР

Лаборатория вычислительных методов механико-математического факультета МГУ традиционно занимается разработкой программного обеспечения для поддержки школьного курса «Основы информатики и вычислительной техники». В частности, здесь разработана система программирования КуМир, которая полностью поддерживает программирование на школьном алгоритмическом языке. В этой статье обсуждается реализация на УКНЦ важной части системы — исполнителей — и описывается процесс, позволяющий пользователю, обладающему некоторым опытом программирования на макроассемблере, создавать собственные исполнители. Поставляемый набор исполнителей достаточно широк (в него входят исполнители Робот, Чертежник, ГраТекс, Экран) и продолжает расширяться (готов исполнитель Редактор, позволяющий использовать команды, с помощью которых можно быстро создать текстовый редактор). Исполнители представляют собой мощный инструмент, существенно расширяющий возможности базового языка, поэтому важно предоставить пользователю самому создавать исполнители по собственному вкусу.

Вместе с КуМиром поставляется система, поддерживающая его работу в локальной сети. После ее загрузки ученикам становятся доступны устройства с логическими именами А и В на машине преподавателя — обычно это логические диски (могут быть и дисководы MZ0 и MZ1). Кроме того, на каждой ученической машине имеется устройство VM — электронный диск (в качестве его используются два плана экранной памяти), на который можно прочитать необходимые файлы с машины учителя. Когда все необходимые файлы имеются на VM, можно запустить КуМир. В ответ на приглашение «Укажите файл .E» можно ввести имена исполнителей (которые будут использоваться), имена обстановок для них, а также имя E-программы. Имя обстановки должно идти сразу после имени соответствующего исполнителя. Если одновременная работа с указанными исполнителями возможна, то кроме программы на экране будут отображаться «Миры» исполнителей, а в программе могут использоваться специальные команды управления исполнителем. В процессе урока ученики не общаются с машиной учителя, сохраняя свои файлы на электронном диске. Это большое достоинство, учитывая невысокую надежность и скорость сетевого оборудования. В конце урока содержимое электронного диска может быть сохранено на машине учителя, но обычно в этом нет необходимости.

Обстановка, или мир, исполнителя — это место, где этот исполнитель работает. Например, мир Робота — это клетчатое поле, между некоторыми клетками которого могут быть стенки. Робот целиком умещается в одной клетке. Мир исполнителя Редактор — это текст, в котором перемещается курсор.

Мир исполнителя может занимать четверть, половину или весь экран. В последнем случае Мир видно только в процессе выполнения программы. У Робота и Чертежника Миры занимают четверть экрана, а у Редактора — половину (хотя реальная длина текста может достигать размеров экрана — просто текст не весь виден на экране).

Исполнитель имеет систему предписаний, т. е. команды, которые умеет выполнять. У Робота это команды перемещения «влево», «вправо», «вверх», «вниз», он умеет закрашивать клетку, на которой стоит, по команде «закрасить», у него также можно спросить, характеристики клетки, на которой он стоит, — вещественные функции «радиация» и «температура» возвращают значения этих величин, а логические функции «сверху стена», «сверху свободно» и т. п. возвращают значения «да», когда соблюдаются соответствующие условия. Курсор редактора умеет перемещаться по тексту («влево», «вправо», «вверх», «вниз»), а также изменять этот текст («вставить пробел», «удалить символ», «вставить строку», «удалить строку», «уста-

новить текущий символ»). Есть также функции, возвращающие значение символа, на котором стоит курсор, проверяющие, находится ли курсор в начале или конце строки (или текста).

Исполнители для системы КуМир на УКНЦ ввиду своих небольших размеров и особенностей экономии памяти написаны на ассемблере с использованием пакета макроопределений MACRO-85. Этот пакет включает в себя макроопределения, дающие возможность использовать конструкции условного ветвления, организации циклов и вызова процедур, аналогично языкам высокого уровня, что значительно облегчает программирование на ассемблере и дает возможность пользователям, имеющим лишь небольшой опыт программирования на ассемблере, самим создавать исполнители.

Исполнитель может использовать функции, реализованные внутри КуМира, для своих целей, например для вывода информации на экран. Чтобы сослаться на эти функции, надо знать их адреса. Можно, конечно, указать эти адреса непосредственно, узнав их из MAP-файла, но гораздо легче и, что важно, надежнее поручить эту работу программе LINK, собрав КуМир вместе с исполнителем.

Другая проблема — как получить исполнитель в виде отдельного файла — решается с использованием команд операционной системы. Прежде всего надо собрать КуМир без исполнителя и записать его размер (можно получить командой операционной системы DIR KUMIR.SAV). Затем надо удалить файл (DEL KUMIR.SAV). Далее необходимо вновь собрать КуМир, но уже с исполнителем, причем при сборке следует указать, что кусок КуМира с кодом исполнителя должен находиться в третьем оверлейном фрагменте (первые два использует сам КуМир), и указать файл с кодом исполнителя последним в списке файлов для сборки (проще всего собирать КуМир, используя командный файл, чтобы не набирать каждый раз на клавиатуре довольно длинную команду для его сборки. Вдобавок это предохранит от неизбежных ошибок при вводе). Код исполнителя будет находиться в конце файла KUMIR.SAV и начинаться с начала блока. Используем особенность операционной системы — файлы хранятся в последовательно расположенных блоках на диске, причем команда DEL лишь изменяет запись о файле в каталоге, но содержимое его продолжает оставаться на диске. Пусть размер файла KUMIR.SAV без исполнителя m блоков, а с исполнителем — $m+n$ блоков. Командой DIR/BL KUMIR.SAV можно узнать номер блока, начиная с которого файл лежит на диске. Пусть это k . Последовательность команд

```
DEL KUMIR.SAV
CRE KUMIR.SAV/START:k./ALL:m.
CRE <ИМЯ_ИСПОЛНИТЕЛЯ>.ISP/START:<k+m>./ALL:<n-m>.
```

удалит файл и создаст на его месте файлы, содержащие КуМир без исполнителя и отдельно исполнитель. Константы надо предварительно вычислить, точка после них указывает на десятичную систему счисления.

Особое внимание надо обратить на два ограничения на код исполнителя, накладываемые из-за того, что при запуске КуМира с различными наборами исполнителей код исполнителя может попасть в различные места в памяти. Во-первых, все обращения к внутренним функциям КуМира исполнитель должен производить в абсолютном режиме адресации:

```
JSR      FUNKUM,      PC
```

должно быть заменено на

```
JSR      @#FUNKUM,   PC
```

Кстати, вызов процедуры без параметров с использованием MACRO-85 выглядит соответственно как

```
CALL     FUNKUM
CALL     @#FUNKUM
```

Удобно в начале каждой подпрограммы, применяющей регистры R2—R4, использовать вызов

```
JSR      R5,@#CSV$
```

сохраняющий в стеке регистры R2—R4 и переход

```
JMP@#CRET$
```

восстанавливающий их содержимое из стека и осуществляющий возврат из подпрограммы (при этом все, что положили в стек после вызова CSV\$, будет уничтожено). Регистр R5

использовать нельзя, он указывает на место в стеке, где сохранено старое содержимое R5, сверху от него лежит адрес возврата, снизу — сохраненные значения регистров.

Во-вторых, код исполнителя должен быть перемещаемым. То есть к своим функциям исполнитель должен обращаться в относительном режиме и использовать численное значение своих меток, вычисляя смещение до них от текущего значения PC. Например,

```
MOV     ADDR,    R0
```

должно быть заменено на

```
MOV     ADDR--,  R0
ADD     PC,      R0
```

Для удобства можно использовать макроопределение

```
.MACRO  MOVR     ADDR,    REG
        MOV     ADDR--,  REG
        ADD     PC,      REG
.ENDM
```

Файл с исполнителем состоит из двух частей. Первая часть — это заголовок стандартного образца, включающий описание некоторых параметров и описание системы команд исполнителя. Вторая часть содержит процедуры, реализующие систему команд исполнителя. Исполнители должны удовлетворять требованиям КуМира, знать о причине этих требований вовсе не обязательно, легче делать по аналогии на основании представленных примеров.

Формат заголовка исполнителя

Сначала лежат три слова, содержащие размер заголовка исполнителя, размер кода исполнителя и количество памяти, требуемой исполнителю. Затем лежит информация, соответствующая определениям Си (адреса должны быть указаны относительно метки START, которая указывает на начало заголовка исполнителя):

```
char _version[4];           номер версии
char name[32];             имя исполнителя
int request;              запрос на окружение
byte far *modes;
int win_color;
int status;
proc Hello;               адрес процедуры приветствия исполнителя
proc LoadMIR;             адреса процедур записи и чтения
proc SaveMIR;             обстановки исполнителя
char MirName[40];         имя обстановки по умолчанию
struct tTable far *texp;
struct cTable far *cexp;
struct fTable far *fexp;  адрес таблицы экспортируемых функций
struct vTable far *vexp;
char **CnfData;
int key[8];
```

Неописанные поля не используются.

ВНИМАНИЕ. Вся таблица функций имеет формат массива структур. Поскольку размер таблицы переменный, в конце (последним элементом массива структур) должен быть помещен элемент '{0}', служащий признаком конца. Его отсутствие может привести к фатальным последствиям!

Таблица функций:

```
typedef struct fTable
{
    char far *name;
    byte type;
    byte far *Parm;
    proc body;
    byte expMod, HelpId;
};
```

byte type — тип функции (тип результата операции). Для вспомогательных алгоритмов, не возвращающих значения, надо указывать 0. Имеются встроенные простые типы:

BOOL	— лог	— логический,
INT	— цел	— целый,
REAL	— вещ	— вещественный,
LIT	— лит	— литерный,
SYM	— сим	— символьный;

byte far *Parm — строка типов параметров (количество параметров определяется по длине строки, указанной в первом байте). Кроме использования простых типов (как в поле типа функции) можно добавлять любые из следующих модификаторов:

ARG	— arg	— аргумент,
RES	— рез	— результат;

использование параметров табличных типов:

DIM1	— таб[*]	— линейная таблица,
DIM2	— таб[*,*]	— прямоугольная таблица,
DIM3	— таб[*,*,*]	— трехмерная таблица.

Пример заголовка исполнителя — исполнитель Робот:

.PSECT C\$A

```
.WORD DSCEND-DSCBEG ; содержащие размер заголовка исполнителя
.WORD CODEND-CODBEG ; размер кода исполнителя
.WORD MINMEM ; количество памяти, требуемой исполнителю
```

\$\$START:
DSCBEG:

```
.ASCII /4001/ ; Номер версии
```

```
..1: .LC <robot> ; Имя исполнителя
```

```
..2--..1 .BLKB 32. - ..2 ; дополнение до 32 байт
```

```
.WORD QUATRSCREEN ; Размер окна вывода
```

```
.WORD 0
```

```
.WORD 0
```

```
.WORD 0
```

```
.WORD R.HELLO-$$START ; смещения процедуры приветствия исполн.
```

```
.WORD R.LOAD-$$START ; процедуры загрузки
```

```
.WORD R.SAVE-$$START ; и сохранения исполнителя
```

EVR\$: .ASCIZ /NONAME.MIR/

```
.REPT 29. ; имя обстановки по умолчанию, терминированное 0
```

```
.BYTE 000 ; дополнение до 40 байт
```

```
.ENDR
```

```
.WORD 0
```

```
.WORD 0
```

```
.WORD FE-$$START
```

```
.WORD 0
```

```
.WORD 0
```

KEY\$: .WORD 0

```
.WORD 0
```

```
.WORD 0
```

```
.WORD 0
```

```
.WORD 0
```

```
.WORD 0
```

```
.WORD 0
```

```
.WORD 0
```

EVR = . - EVR\$

TBLEND:

FE:

```
.WORD $RAD-$START ; адрес имени функции
.BYTE REAL ; тип возвращаемого значения
.EVEN
.WORD 0: .NULL-$START ; количество аргументов
.WORD R.RAD-$START ; адрес тела функции
.BYTE 0
.BYTE 0
.EVEN
... ..
.WORD ... 0 ; терминатор таблицы функций
```

Строки типов параметров (исполнитель Чертежник):

```
.NULL: .BYTE 0
        .EVEN
.R2: .BYTE 2, ARG+REAL, ARG+REAL
      .EVEN
.I1: .BYTE 1, ARG+INT
     .EVEN
```

Строка с именем функции (исполнитель Робот):

```
.MACRO .LC TXT ; Макроопределение для
.IRPC S,<TXT> ; сообщений на русском языке
  .IF LT "S"@
    .BYTE "S"
  .IFF
  .IF NE "S&0040"
    .BYTE "S!0200&`C040"
  .IFF
  .IF EQ "S'"
    .BYTE "S"
  .IFF
  .BYTE "S!40"
  .ENDC
  .ENDC
  .ENDC
  .ENDR
  .ENDM
$RAD: .LCZ <radiaciq>
      .EVEN
```

Фрагмент исполнителя Робот:

```
PROC R.HELLO ; это просто сохранение некоторых значений
MOV PC, R0 ; не вдаваясь в подробности можно просто
ADD #TBLEND-, R0 ; вставлять в свой исполнитель аналогичную
; подпрограмму.

PUSH R0

MOV #HOOK+START, R1
ADD #OLDSTR-CODBEG, (SP)
MOV (R1), @(SP)

MOV R0, (SP)
ADD #R.START-CODBEG, (SP)
MOV (SP), (R1)

MOV #HOOK+BYE, R1
MOV R0, (SP)
ADD #OLDBYE-CODBEG, (SP)
MOV (R1), @(SP)

MOV R0, (SP)
ADD #R.BYE-CODBEG, (SP)
MOV (SP), (R1)

TST (SP)+
ENDPRO
```

DSCEND:
 CODBEG:

CLASS C\$Y, OPT-YES

;
 ; ispolzuyuemye ispolniteli:

;
 .INCLUDE /GLOBL.MAC/

CSIZX - 16. ; razmer kletki labirinta
 CSIZY - 12. ; razmer kletki labirinta
 LXSIZ - <<XXX+1>/2/CSIZX>-1
 LYSIZ - <<YYY+1>/2/CSIZY>-1

MINMEM- <LXSIZ*LYSIZ*SIZ> + 6 + 4 ; razmer trebuemoj pamqti

PROC R.START
 JSR R5, @#CSV\$
 CALL \$SET
 CALL CLEAR
 CALL DRAW.L ; narisovat' labirint
 CALL @#EVRESET
 CALL @OLDSTR
 JMP @#CRET\$

PROC R.BYE
 JSR R5, @#CSV\$

 CMPB 4(R5), #1
 IF EQ
 CALL R.SAVE
 ENDIF

MOV #HOOK+START, R1
 MOV OLDSTR, (R1)

MOV #HOOK+BYE, R1
 MOV OLDBYE, (R1)

CALL @OLDBYE
 JMP @#CRET\$

PROC R.LOAD
 JSR R5, @#CSV\$
 MOV #FX.REA, -(SP)
 MOV PC, R0
 ADD #CODBEG-EVR-, R0
 MOV R0, -(SP)
 CALL @#FXOPEN
 CMP (SP)+, (SP)+
 IF <R0 EQ #-1>
 CLR SCR1
 CLR SCRY1 ;
 MOV PC, R0 ;
 ADD #LAB-, R0 ;
 MOV R0, LABPT1 ;
 MOV SP, QCLRL ; o~istitc=da

ELSE
 MOV R0, CHAN
 MOV #MINMEM-4, -(SP) ; razmer
 MOV #FX.BIN, -(SP) ;
 MOV PC, R1 ;
 ADD #SCR1-, R1 ;
 MOV R1, -(SP) ; kuda
 MOV R0, -(SP) ; kanal
 CALL @#FXREAD

```

MOV PC, R1 ;
ADD #LAB-., R1 ;
ADD R1, LABPT1 ;

CALL @#FXCLOSE
CMP (SP)+, (SP)+
CMP (SP)+, (SP)+
    
```

```

ENDIF
JMP @#CRET$
    
```

```

PROC R.SAVE
JSR R5, @#CSV$
MOV #FX.WRI,-(SP)
MOV PC, R0
ADD #CODBEG-EVR-., R0
MOV R0, -(SP)
CALL @#FXOPEN
CMP (SP)+, (SP)+
IF <R0 EQ #-1>
MOV PC, R0
ADD #FAIL-., R0
MOV R0, -(SP)
CALL @#OTKAZ
    
```

```

ELSE
MOV R0, CHAN
    
```

```

PUSH LABPT1
    
```

```

MOV PC, R1 ;
ADD #LAB-., R1 ;
SUB R1, LABPT1
    
```

```

MOV #MINMEM-4,-(SP) ; razmer
MOV #FX.BIN,-(SP) ;
MOV PC, R1 ;
ADD #SCRX1-., R1 ;
MOV R1, -(SP) ; otkuda
MOV R0, -(SP) ; kanal
CALL @#FXWRITE
CALL @#FXCLOSE
CMP (SP)+, (SP)+
CMP (SP)+, (SP)+
    
```

```

POP LABPT1
    
```

```

ENDIF
JMP @#CRET$
    
```

```

PROC R.RAD ; radiaciq
JSR R5, @#CSV$
MOV LABPTR, R0
MOV BRAD(R0), R1
CLR R0
    
```

```

.1:
;$$JTOR:: R0,R1 - REAL*4
CALL @#$$JTOR
MOV 4(R5), R2
MOV R0, (R2)+
MOV R1, (R2)+
    
```

```

JMP @#CRET$
CODEND:
    
```

```

OLDSTR = CODEND
OLDBYE = OLDSTR + 2
    
```

```

.END
    
```

Вывод на экран исполнитель осуществляет при помощи макроопределений, описанных в файле **GT.MAC**. Эти макроопределения поддерживают вывод на экран текста и графических примитивов:

GTSVM GM	— инициализировать графический режим
GTTERM	— закончить работу с графикой
GTSCOLOR COL	— установить цвет рисования
GTSBCKC COLOR	— установить цвет фона
GTAMOVE X,Y	— сместиться в точку с абсолютными координатами
GTRMOVE X,Y	— сместиться в точку с относительными координатами
GTSWINDOW L, T, R, B	— установить окно вывода
GTADOT X,Y	— поставить точку с абсолютными координатами
GTRDOT X, Y	— поставить точку с относительными координатами
GTPIXEL	— поставить точку в текущую позицию
GTALINE X,Y	— прочертить линию от текущей позиции до точки X,Y
GTRLINE X,Y	— нарисовать вектор X,Y от текущей позиции
GTTEXT S, L	— вывести L символов по адресу S

Имеются и более сложные макросы для рисования прямоугольников, эллипсов, установок стилей рисования и т. п.

В функции **START** исполнителя перед первым вызовом макроопределений ввода/вывода должен быть вызов функции

CALL @#EGSET

а в каждой функции, использующей ввод/вывод, в конце должен быть вызов

CALL @#EVRESET

Для сохранения и загрузки обстановки исполнитель может пользоваться процедурами **FXOPEN** — открыть файл.

При вызове в стеке должна лежать константа, определяющая режим доступа:

FX.REA — на чтение
FX.WRI — на запись

и указатель на имя файла (вид смотреть в примере).

Функция возвращает значение в **R0**. -1 — ошибка, иначе номер канала — байт, по которому производятся все дальнейшие обращения к файлу.

FXREAD — прочитать из файла
FXWRITE — записать в файл

При вызове в стеке должен лежать размер, адрес в памяти — начало массива и канал, по которому происходит обмен.

FXCLOSE — закрыть файл

При вызове в стеке должен лежать номер канала.

Имеются также аналогичные процедуры для ввода/вывода текстовых строк.

* * *

ОТВЕТЫ НА КРОССВОРД, ОПУБЛИКОВАННЫЙ НА СТР. 51

По горизонтали: 1. File. 4. End. 6. Sqrt. 7. Read. 8. For. 10. Case. 13. Type. 15. Input. 16. Label. 17. Then. 18. Text. 19. Real. По вертикали: 1. Function. 2. Else. 3. Integer. 5. Downto. 6. Set. 7. Repeat. 9. Readln. 11. Round. 12. Array. 14. Eoln.

М. И. Фролов,

доктор технических наук

АВТОМАТИЗИРОВАННАЯ СИСТЕМА ПОСТРОЕНИЯ РАСПИСАНИЯ «ЗАВУЧ» ДЛЯ УКНЦ

На сегодняшний день проблема оперативного составления оптимального учебного расписания является одной из наиболее актуальных в организации учебного процесса. Именно поэтому фирмой «Колледж» была разработана автоматизированная система построения расписания «Завуч» версии 2.01 для компьютера УКНЦ (МС0511), имеющего широкое распространение в школах и профессионально-технических училищах.

Система «Завуч» проста в эксплуатации и основана на решении сложных комбинаторных задач, являющихся основой для построения учебного расписания. Эта система позволяет учесть следующие особенности организации учебного процесса:

- включение учебной группы, преподавателя, аудитории в расписание не более одного раза во избежание накладок;
- разбивка учебных групп на подгруппы для проведения занятий по одному и тому же предмету с каждой из подгрупп в отдельной аудитории с отдельным преподавателем;
- отсутствие «окон» в расписании учебных групп;
- существование методических дней преподавателя;
- двухсменная система занятий;
- пятидневная неделя учебных групп или наличие одного свободного дня в неделю для занятий в учебно-производственном комбинате (УПК);
- занятия, которые должны проводиться в строго определенное время (директивное расписание).

Доступность автоматизированной системы «Завуч» определяется удобным для пользователя меню. Главное меню содержит следующие пункты: «Справочники», «Расписание», «Результаты», «Прочее» и «Выход». Все пункты главного меню содержат подменю с самостоятельными наборами пунктов. Выбор пункта главного меню или подменю осуществляется с помощью курсорной рамки (подсвеченного фрагмента меню), которая перемещается с помощью клавиш управления курсором. В подменю «Справочники» есть такие пункты как «Предметы», «Группы», «Преподаватели», «Аудитории», «Учебный план» и «Директивное расписание». Каждая запись в справочнике имеет поля данных, которые, как правило, заполняются пользователем. Исключение составляет поле, в котором приводится информация о нагрузке (в часах) групп или преподавателей. Это поле заполняется системой «Завуч» автоматически в режиме вычисления нагрузки.

Справочник «Предметы» содержит единственную графу «Предметы», в которой указываются названия предметов.

В справочнике «Группы» содержатся следующие графы:

- «Название» — название группы или класса;
- «Д» — номер свободного дня или дня УПК;
- «С» — номер смены;
- «Нг» — планируемая нагрузка группы в часах, заполняемая автоматически по данным из справочника «Учебный план».

Каждая запись справочника «Преподаватели» содержит графы:

- «Фамилия» — фамилия преподавателя;
- «Д» — номер свободного (методического) дня;
- «Нг» — планируемая нагрузка преподавателя в часах.

В справочнике «Аудитории» имеются такие графы, как:

- «Название» — название аудитории;
- «Вместимость» — вместимость аудитории по количеству учебных групп.

Корректировка записей в вышеперечисленных справочниках осуществляется просто и удобно с помощью курсорной рамки, установленной на нужную запись.

Справочник «Учебный план» включает в себя сведения об указанных предметах и занятиях учебных групп на неделю. Записи этого справочника объединены по группам, т. е. на экране одновременно находятся и доступны для редактирования записи только одной учебной группы.

Каждая запись справочника «Учебный план» содержит следующие графы:

«Группа»	— название учебной группы из справочника «Группа»;
«Предмет»	— название предмета из справочника «Предметы»;
«Преп. 1»	— фамилия первого преподавателя из справочника «Преподаватели»;
«Преп. 2»	— фамилия второго преподавателя из того же справочника;
«Аудит-я»	— название аудитории из справочника «Аудитории»;
«Ч»	— количество учебных часов в неделю для данной группы по данному предмету.

В случае деления учебной группы на две подгруппы (заполнены графы «Преп. 1» и «Преп. 2») для проведения одноименных занятий создается одна условная аудитория с вместимостью в две группы, которая заносится в справочник «Аудитории», а ее номер указывается в аналогичной графе справочника «Учебный план».

В справочнике «Дир. расписание» задается фиксированное время проведения занятий по указанному предмету для данной группы и содержатся следующие графы:

«Сводное поле»	— ссылка, соответствующая таким графам пункта «Учебный план», как: «Предмет», «Преп. 1», «Преп. 2» и «Аудит-я»;
«Д»	— номер дня недели проведения занятия;
«Ч»	— номер учебного часа проведения занятия в выбранном дне.

Подменю «Расписание» включает в себя пункты: «Проверка корректности», «Построение расписания», «Планируемая нагрузка». При выборе пункта «Проверка корректности» система «Завуч» производит проверку исходных данных и может обнаружить ошибки следующих типов:

«Не задан»	— не задано значение поля (графы);
«Нет в справ.»	— ссылка на пустое поле соответствующего справочника;
«Перегрузка»	— в справочнике «Дир. расписание» указано больше часов для занятия, чем это требовалось по справочнику «Учебный план»;
«Накладка»	— в справочнике «Дир. расписание» сделана попытка установить группу, преподавателя или аудиторию в то время, когда они уже заняты;
«Нет в справ. Уч. план»	— запись в справочнике «Дир. расписание» не соответствует ни одной записи справочника «Учебный план».

При выборе пункта «Построение расписания» система «Завуч» приступает собственно к построению расписания с выводом на экран дисплея числа нераспределенных занятий и числа «окон» в расписании учебных групп. Кроме того, выдается запрос на сохранение нового варианта расписания.

С помощью пункта «Планируемая нагрузка» производится начисление нагрузки в часах для учебных групп и преподавателей. Начисление осуществляется суммированием количества часов из справочника «Учебный план» для каждого преподавателя и каждой группы.

Подменю «Результаты» содержит следующие пункты: «Расписание», «Нераспределенные занятия» и «Вывод результатов». Выбор пункта «Результаты» позволяет пользователю просмотреть на экране дисплея расписание конкретного преподавателя, группы, аудитории на неделю. При этом на экране одновременно помещается расписание на один день недели.

Пункт «Нераспределенные занятия» позволяет получить на экране дисплея список нераспределенных занятий, которые можно устранить:

- доводкой расписания вручную (наиболее радикальный и наименее удобный подход);
- построением нового варианта расписания при другой последовательности дней недели (см. ниже подменю «Прочее»). Этот подход эффективен при небольшом (от —5 до 7) числе нераспределенных занятий;
- анализом полученных вариантов расписания и внесением соответствующих изменений в справочник «Учебный план» и повторным построением расписания. Например, если обнаруживается, что в нераспределенных занятиях часто фигурирует одна и та же аудитория, то в справочнике «Учебный план» имеет смысл указать другую, менее загруженную аудиторию. Можно также разделить запись справочника «Учебный план» на две записи, указав в них одни и те же предметы и одних и тех же преподавателей, но разные аудитории, и соответственно перераспределить нагрузку.

Пункт «Вывод расписаний» позволяет управлять выводом на принтер или в текстовые файлы всех вариантов расписания без отображения их на экране дисплея.

Подменю «Прочее» содержит пункты «Границы смен» и «Порядок дней».

Пункт «Границы смен» дает возможность установить номер последнего часа (урока) первой смены и номер первого часа (урока) второй смены. В случае если номер последнего часа первой смены больше номера первого часа второй смены (пересечение смен), то на время, общее для

обеих смен, занятия расставляются равноправно, как для первой, так и для второй смены. В противном случае занятия первой и второй смен не пересекаются.

Пункт «Порядок дней» позволяет изменить последовательность номеров дней недели, по которой производится построение расписания. Изменение порядка дней недели дает возможность увеличить число вариантов расписания, из которых выбирается наиболее подходящий.

Содержимое любого справочника или расписания может быть выведено на принтер или на дискету в виде текстового файла в процессе его просмотра или корректировки.

Файлы, поставляемые на дискете вместе с автоматизированной системой построения расписания «Завуч», можно условно разделить на четыре типа: системные файлы, поддерживающие работу операционной системы RT11SJ;

служебные файлы, обслуживающие работу дополнительного программного обеспечения:

- RULON.SAV — обеспечение быстрого скролинга экрана;
- KLAV.SAV — перекодировка клавиш;
- EDIK.SAV — текстовый редактор;
- EDIK.DOC — описание текстового редактора;
- TTY.SAV — страничный просмотр текстовых файлов;
- RESORC.SAV — ревизия системных ресурсов;
- LP.DOC — описание драйвера принтера;

файлы, составляющие собственно систему «Завуч»:

- INF.TXT — подсказка о формах вызова системы;
- HELLO.TXT — заставка;
- RZAV.COM — командный файл, управляющий запуском системы;
- ZAV.EXE — собственно программа «Завуч»;

файлы, которые могут быть созданы системой «Завуч» в процессе ее работы:

- CFG.CFG — файл сохранения настройки;
- GRUP.DAT, PRED.DAT, PREP.DAT, AUDT.DAT, UPL.DAT, DRR.DAT — файлы справочников, соответственно, «Группы», «Предметы», «Преподаватели», «Аудитории», «Учебный план» и «Директивное расписание»;
- RSPG.RSP, RSPP.RSP, RSPA.RSP, NZN.RSP — файлы расписаний, соответственно, «Группы», «Преподаватели», «Аудитории» и «Нераспределенные занятия»;
- GRUP.LST, PRED.LST, PREP.LST, AUDT.LST, UPL.LST, DRR.LST — файлы списков справочников, соответственно, «Группы», «Предметы», «Преподаватели», «Аудитории», «Учебный план» и «Директивное расписание»;
- NZN.LST — файл списка «Нераспределенные занятия»;
- GRxx.LST — файлы списков расписаний групп;
- PRxx.LST — файлы списков расписаний преподавателей;
- AUxx.LST — файлы списков расписаний аудиторий (здесь xx — номер записи из соответствующего справочника).

В заключение следует отметить, что сотрудники фирмы «Колледж» готовы учесть все пожелания, замечания и предложения по усовершенствованию автоматизированной системы построения расписания «Завуч», которые можно направить по адресу:

107005, Москва, Волховский пер., 11.

Телефон: (095) 265-62-65.

Здесь же можно ознакомиться и с условиями приобретения системы «Завуч».



РЕСПУБЛИКАНСКИЙ ЦЕНТР ИНТЕРАКТИВНЫХ СРЕДСТВ ОБУЧЕНИЯ

Телефоны: (095) 155 8737, 155 8747

Факс: (095) 155 8727

КУРСЫ ИНОСТРАННЫХ ЯЗЫКОВ НА CD-ROM

Новый курс английского языка разработан в одном из ведущих американских научных центров - Сиракузском университете. Эффективная оригинальная методика изучения иностранного языка в увлекательной игровой форме включает режим распознавания вводимой через микрофон речи. Курс дополнен подробным руководством на русском языке.

Для детей и взрослых

Новинка!



Компьютер поймет Вас, если произношение будет достаточно корректным!

Курс содержит свыше 1000 слов, фраз и диалогов. Возможны варианты заказа аналогичных курсов французского и испанского языков.

Для ввода устной речи в комплект поставки включен микрофон!

плюс еще 900 наименований CD-ROM!



*Добро пожаловать
в РЦИСО!*

Наш адрес: 125315, г. Москва,
ул. Часовая, 216

Проезд: станция метро "Сокол",
выход к улице Усиевича,
далее по схеме

ЛинТех

**Хватит мечтать - давайте действовать!
Превратите КУВТ УКНЦ, "Корвет" и БК в IBM-PC**

Принципиально новые системы "NET-R111 & DOS-LINE" и "NET-CP/M & DOS-LINE" позволят Вам превратить КУВТ УКНЦ, "Корвет" и БК в классы IBM-PC. На каждом рабочем месте Вы будете работать, как на IBM-PC, под управлением MS-DOS, использовать Norton Commander, Лексикон, Turbo Basic и другие популярные программы для IBM-PC. При этом полностью сохраняется возможность использования всего существующего программного обеспечения для этих КУВТ.

Для модернизации КУВТ достаточно приобрести нашу систему и установить в КУВТ IBM-совместимый головной компьютер.

Локальные сети "NET-R111 & DOS-LINE" и "NET-CP/M & DOS-LINE" объединяют с помощью высокоскоростных сетевых адаптеров в единое целое головной компьютер IBM-PC и ученические машины. Скорость работы повышается в 30-100 раз, на каждом компьютере ученика обеспечивается полноценная работа без сбоев и зависаний благодаря отказу от использования стандартного оборудования и дисководов.

Цена систем ниже цены одного IBM-совместимого компьютера. В настоящий момент ими оснащено более 300 компьютерных классов на территории России, Белоруссии, Украины и Казахстана.

Все системы просты в установке и использовании, не требуют перемонтажа существующих линий связи, весь процесс модернизации стандартного класса занимает 2-3 часа. Гарантия - 3 года со дня приобретения.



**Министерство образования РФ рекомендует
использовать системы
"NET-R111 & DOS-LINE" и "NET-CP/M & DOS-LINE"
для модернизации КУВТ "Корвет" и УКНЦ.**

**ЛинТех
Телефон/факс: (095) 273-50-14
E-mail: shop@lintech.msk.su
119501, Москва, а/я 942**

MultiVision

"Замечательное средство для создания презентаций
с использованием красочных анимаций, текстов и музыки"

"PC WORLD"



Универсальная среда для
гибкой генерации и сборки учебных
курсов, обучающих программ и
компьютерных шоу.