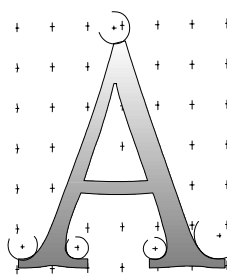


ИНФОРМАТИК

Еженедельная газета Издательского дома «ПЕРВОЕ СЕНТЯБРЯ»
ПОДПИСКА: (095) 249-47-58



Некоронованный король

В 2002 году исполняется 225 лет со дня рождения крупнейшего немецкого математика, астронома и физика, члена Лондонского королевского общества, Парижской и Петербургской академий наук Карла Фридриха Гаусса (1777—1855)

“Гаусс напоминает мне образ высочайшей вершины баварского горного хребта, какой она предстает перед глазами наблюдателя, смотрящего с севера. В этой горной цепи в направлении с востока на запад отдельные вершины поднимаются все выше и выше, достигая предельной высоты в могучем, высящемся в центре великане; круто обрываясь, этот горный исполин сменяется низменностью новой формации, в которую на много десятков километров далеко проникают его отроги, и стекающие с него потоки несут влагу и жизнь”.

*Феликс Клейн,
видный математик,
исследователь научного
творчества Гаусса*

Работы Карла Фридриха Гаусса оказали большое влияние на все дальнейшее развитие высшей алгебры, теории чисел, дифференциальной геометрии, классической теории электричества и магнетизма, геодезии, целых отраслей теоретической астрономии [1–4].

Первым крупным сочинением Гаусса являются “Арифметические исследования” (1801 г.), где рассматриваются вопросы теории чисел и высшей алгебры, постановка и решение которых во многом предопределили развитие данных дисциплин. В конце книги изложена те-

ория уравнений деления круга, т.е. уравнений $x^n = 1$. Помимо общих методов решения этих уравнений, Гаусс установил связь между ними и построением правильных многоугольников. Решив уравнение $x^{17} - 1 = 0$, Гаусс построил правильный семнадцатиугольник при помощи циркуля и линейки. [В его газетной заметке, появившейся в июне 1796 года, сообщалось [1]: “Всякому начинающему геометру известно, что можно геометрически (т.е. циркулем и линейкой) строить разные правильные многоугольники, а именно: треугольник, пятиугольник, пятнадцатиугольник и те, которые получаются из каждого из них путем последовательного удвоения числа его сторон. Это было известно во времена Евклида, и, как кажется, с тех пор было распространено убеждение, что дальше область элементарной геометрии не распространяется: по крайней мере я не знаю удачной попытки распространить ее в эту сторону.

Тем более кажется мне заслуживающим внимания открытие, что, кроме таких правильных многоугольников, может быть геометрически построено множество

других, например, семнадцатиугольник”.] Гаусс считал это решение очень важным и завещал выгравировать правильный семнадцатиугольник, вписанный в круг, на своем надгробном памятнике, что и было сделано.

От этих теоретических изысканий Гаусс скоро перешел к исследова-

Окончание
читайте
на с. 32



Читайте в номере

Страницы повышения квалификации 3–7

И.Н. Фалина. Современные педагогические технологии и частные методики обучения информатике

“Информация”, “алгоритм”, “команда”, “модель”, “операционная система”, “компьютер”, “гипертекст”, “электронная почта”, “база данных”... Сколькими понятиями мы оперируем в школьном курсе информатики?

Приглашаем на лекцию, тема которой — “Анализ (количественный и качественный) определений, вводимых в школьном курсе информатики для средних классов”.

Уроки..... 8–15

А.А. Дуванов. Азы информатики. Материалы Роботландского университета

“Информацию можно хранить, передавать и обрабатывать”.

Продолжение второй книги. (“Бумажная” версия первой книги — “Знакомство с компьютером” — представлена в № 1, 2/2002.)

На стенд в кабинете информатики 16–17

Привод CD-ROM

Знаете ли вы, что это весьма сложное устройство состоит всего из двух основных (правда, тоже сложных) узлов: шпинделя с держателем диска, под которым располагается электродвигатель, и системы оптической головки?

Информация 18–20

В.А. Козлова. Программно-методические комплексы для преподавания информатики в 1–6-х классах

Знакомы ли вам развивающие программы фирмы “Никита”, программы ассоциации “Компьютер и Детство”, мультимедийные развивающие игры фирмы NMG? Приводятся краткие характеристики этих и некоторых других программных продуктов.

Проекты 21–27

Н.Н. Паньгина. Универсальная модель движения

Тема “Моделирование” является одной из самых интересных в школьном курсе информатики. А начинать здесь, как обычно, надо с простого...

Начните с простого 28–31

Л.О. Сергеев. Простая гостевая книга для школьного сайта

“Perl представляет собой универсальный язык программирования высокого уровня”.

Рассматривается задача, являющаяся типичной для языка Perl, причем ее решение можно непосредственно использовать на практике. (Кстати, интерпретаторы этого языка есть почти для всех современных операционных систем.)

Читайте в следующих номерах

В следующем номере мы познакомим наших читателей с планом тематических выпусков серии “Жаркое лето-2002”. В этом году серия будет рекордной — никогда еще летние номера не были такими разнообразными и содержательными. Встречайте!

В № 9 мы начнем публикацию “примерных ответов” на “примерные билеты”. Сами примерные билеты для итоговой аттестации выпускников 9-х классов общеобразовательных учреждений в 2001/2002 учебном году были подготовлены Министерством образования. Мы сочли, что сами билеты, конечно, хорошо, но билеты с ответами — много лучше. Таким образом, начиная с № 9 мы будем публиковать по два билета с ответами на них в каждом номере. Эти публикации завершатся в середине мая.

В весенних номерах продолжатся публикации трех лекционных курсов на “Страницах повышения квалификации”, а уже в марте мы представим вам лекционные курсы, которые будут “прочитаны” на страницах “Информатики” в следующем учебном году.

День учителя информатики в Москве

15 апреля,

Московский городской дом учителя
(улица Пушечная, дом 4, строение 2,
станция метро “Кузнецкий мост”).



Дорогие коллеги! Пожалуйста, постарайтесь найти возможность прийти 15 апреля в Дом учителя. Мы готовим замечательную программу, будет очень интересно.

Современные педагогические технологии и частные методики обучения информатике

Лекции читает И.Н. Фалина

В подготовке материала для данной лекции принимала активное участие студентка МГУ им. М.В. Ломоносова Екатерина Коновальчук, обучающаяся одновременно на факультете Вычислительной математики и кибернетики и факультете Педагогического образования.

Лекция 8. Анализ (количественный и качественный) определений, вводимых в школьном курсе информатики для средних классов

Процесс познания материального мира начинается с чувственного восприятия человеком окружающих предметов и явлений. Отражая в нашем сознании реальную действительность, чувственное восприятие служит источником всех наших знаний. Однако чувственное восприятие есть лишь начальная, простейшая ступень познания; на этой ступени познания человек не может получить знания об общих, внутренних свойствах вещей. От непосредственного познания действительности с помощью органов чувств человек переходит к высшей ступени познания — образованию понятий.

Понятие есть основная и высшая форма нашего мышления. *Понятие* абстрагируется от индивидуальных черт и признаков отдельных представлений и восприятий и является, таким образом, результатом *обобщения* восприятий и представлений очень большого количества однородных явлений и предметов. Например, число, система счисления, компьютер, монитор, гипертекст и т.д.

Формирование понятий (или усвоение понятий) относится к умению выяснять свойства, присущие некоторому классу объектов или идей.

Понятия образуются путем таких логических приемов, как анализ и синтез, абстракция и обобщение. Вообще же надо сказать, что вопрос об образовании понятий чрезвычайно сложен, для его решения формальной логики недостаточно.

Формирование понятий — одна из самых важных когнитивных функций человека. В большинстве наук в период их становления формирование понятий играет решающую роль в организации данных. Расположение элементов в химии, классификация видов памяти в психологии, классификация систем счисления в матема-

тике и информатике — все это примеры формирования понятий, способствующие лучшему пониманию предмета.

Формирование у учащихся системы понятий по конкретному предмету — одна из важнейших задач учителя-предметника.

? Сколькими понятиями мы оперируем в школьном курсе информатики, вернее, каков тот круг основных понятий, без усвоения которого, как мы считаем, учащийся не может нормально изучить курс школьной информатики?

Наиболее простым ответом на поставленный вопрос является анализ основных школьных учебников по информатике. Мы выбрали наиболее распространенные учебники, рекомендованные Управлением общего и среднего образования Министерства общего и профессионального образования РФ:

1. *Семакин И., Залогова А., Русаков С., Шестакова Л.* “Информатика. Учебник по базовому курсу”. М.: Лаборатория базовых знаний, 1998.
2. *Кузнецов А.А., Апатова Н.В.* Основы информатики. 8—9-е классы. М.: Дрофа, 1999.
3. Информатика. 7—8-е классы / Под ред. Н.В. Макаровой. СПб.: Издательство “Питер”, 2000.

План публикаций лекций курса “Современные педагогические технологии и частные методики обучения информатике” на “Страницах повышения квалификации”.

Номер лекции	Номер газеты
1	37/2001
2	39/2001
3	41/2001
4	43/2001
5	45/2001
6	47/2001
7	5/2002
8	7/2002
9	9/2002
10	11/2002
11	13/2002
12	15/2002

Общая количественная оценка используемых понятий такова:

Семакин [1]	Кузнецов [2]	Макарова [3]
332 определения	109 определений	178 определений

Много это или мало? Если по этим учебникам учить в 10-м классе, где на информатику отводится 2 часа в неделю, т.е. всего 64 часа за учебный год, то, как несложно подсчитать, за один урок придется вводить от 2 до 5 определений.

Не менее интересно понять круг необходимых, основных определений школьного курса информатики. На мой взгляд, этот круг значительно уже. К основным понятиям школьного курса информатики можно отнести такие понятия (см. следующую страницу):

1. Информатика	27. Структуры данных
2. Информация	28. Операционная система
3. Информационные процессы	29. Многозадачная операционная система
4. Алгоритм	30. Корпоративная и вытесняющая многозадачность
5. Команда	31. Многопользовательская операционная система
6. Исполнитель	32. Программное обеспечение
7. Вычислительный процесс	33. Файловая система
8. Сложность алгоритма	34. Система программирования
9. Система команд исполнителя	35. Компьютер
10. Алгоритмический язык	36. Архитектура ЭВМ
11. Синтаксис языка	37. Основные принципы работы компьютера
12. Семантика языка	38. Способы представления информации в ЭВМ
13. Программа	39. Современные информационные технологии
14. Оператор	40. Телекоммуникация
15. Системы счисления	41. Компьютерная сеть
16. Алфавит системы счисления	42. Глобальная сеть Интернет
17. Базис системы счисления	43. Гипертекст
18. Алгебра логики	44. Web-страница
19. Высказывание	45. Web-сервер
20. Логические операции	46. Электронная почта
21. Булева функция	47. Информационный ресурс
22. Способы задания булевых функций	48. База данных
23. Основные законы алгебры логики	49. Система управления базами данных
24. Моделирование	50. Текстовый редактор
25. Модель	51. Графический редактор
26. Информационная модель	52. Электронные таблицы

Закон “обратного отношения содержания и объема понятий” имеет ограниченный характер, он применим лишь в случае, когда одно понятие входит в объем другого.

С учащимися обязательно надо заниматься изучением связи между содержанием и объемом понятия. Начинать, естественно, надо с простых, известных им примеров, быть может, не связанных с информатикой. В младших классах нельзя непосредствен-

Каждый учитель может дополнить или изменить этот список в соответствии со своим курсом, кроме того, практически любое из этих понятий требует ввода дополнительных определений, помогающих разъяснить основное понятие.

Содержание и объем понятия

Каждое понятие можно охарактеризовать совокупностью признаков тех предметов, которые отражены в данном понятии. Совокупность основных признаков объектов, охватываемых понятием, называется *содержанием понятия*. Например, содержанием понятия “позиционная система счисления” являются: 1) базис системы счисления; 2) алфавит системы счисления; 3) основание системы счисления, если базис представляет собой геометрическую прогрессию; 4) возможность записи любого действительного числа.

Но понятие характеризуется не только совокупностью основных признаков, составляющих содержание понятия, в каждом понятии различается *объем*. Объем понятия — это множество объектов, на которые распространяется данное понятие. Так, например, объем понятия “позиционная система счисления” составляется из множества систем счисления, базисом которых являются геометрические прогрессии (традиционные позиционные системы счисления), а также из нетрадиционных позиционных систем счисления: не имеющих основания (фибоначчиева, факториальная), использующих в качестве значений символов алфавита не натуральные числа (уравновешенные системы счисления), систем счисления, имеющих несколько оснований (...секунда, минута, час, сутки, месяц...).

Чем шире содержание понятия, тем уже его объем, и, наоборот, чем уже содержание понятия, тем шире его объем. Например, если в понятие “позиционная система счисления” ввести требование о постоянном соотношении веса соседних разрядов, т.е. мы расширим понятие, то под это определение не попадут нетрадиционные системы счисления, т.е. мы сузим объем понятия.

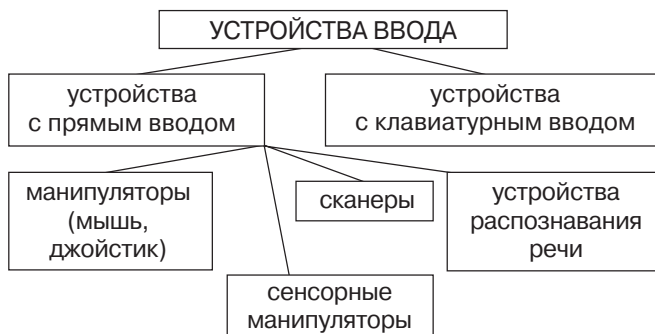
но говорить учащимся о содержании и объеме понятий, но разбором соответствующих примеров необходимо постепенно подводить учащихся к пониманию вопросов, связанных с содержанием и объемом понятий.

Например, в [2] дается следующее определение. *Мышь — коробочка с кнопками, внизу или вверху которой выступает часть шара. Поворот шара соответствует движению курсора по экрану. При работе с мышью курсор приобретает вид стрелки или другой фигуры, движущейся по экрану непрерывно. С помощью мыши можно попасть в любую точку экрана.* Насколько просто учащийся может понять из данного определения, что *мышь* — это устройство ввода информации, и включить *мышь* в объем определения “устройство ввода — техническое устройство для ввода информации в ЭВМ”?

В [3] дается достаточно исчерпывающая классификация устройств ввода: “Многообразие устройств ввода определяется принципом ввода и видом вводимой информации. Устройства ввода можно разделить на два основных класса:

- с клавиатурным вводом, при котором осуществляется ручной ввод с клавиатуры;
- с прямым вводом, при котором данные считываются непосредственно компьютерными устройствами. Среди устройств с прямым вводом данных выделяются подклассы устройств.

И далее приводится схема, поясняющая данное определение.



Но можете ли вы, прочитав эти два определения, ответить на вопросы: “А почему все же мышь является устройством ввода? Какую информацию она вводит?”? Почему интуитивно очевидное понятие *устройства ввода* оказалось таким сложным для формулирования?

Давайте ответим сами себе на вопрос: “Что мы хотим получить от учащегося, вводя определение *мышь* как устройства ввода?” Если мы будем обучать наших учеников только пользовательским навыкам, то с понятием, построенным на ассоциации “мышь — курсор” ([2]), можно успешно работать. Если же мы хотим донести до школьников принципы ввода информации в ЭВМ, то учащимся надо дать определение *мышь*, из которого они бы поняли, что же происходит, когда *мышь* “вводит” информацию в компьютер. За попыткой разяснить принцип ввода информации *мышью* потянутся понятие *объекта в операционной системе* (например, в Windows), понятие *иконки, ярлыка, выполняемого файла, активного окна, фокуса ввода*.

Освоив все эти понятия, умея оперировать с ними, имея практические навыки работы в операционной среде Windows, учащийся сможет верно ответить на вопрос: *Какое из окон в Windows, расположенных на Рабочем столе, является активным?*

1. Окно, заголовок которого выделен инверсным цветом;
2. Окно, расположенное поверх остальных окон;
3. Окно, расположенное на панели задач;
4. Окно, на которое направлен фокус ввода.

Соотношение между понятиями

Каждое понятие находится в известном отношении, известной связи со всеми остальными.

1. При определении новых понятий мы опираемся на уже известные понятия. Так, определение понятия *алгоритмический язык* предполагает знание об алгоритме, исполнителе, командах исполнителя, способах записи алгоритмов.
2. Изучение свойств понятий всегда сопровождается введением новых определений. Для изучения свойств понятия *алгоритмический язык* мы вводим понятия *синтаксис* и *семантика* алгоритмического языка, *алфавит*, *элементы языка*, *транслятор (компилятор и интерпретатор)*, *оператор*, *операция* и т.д.
3. Существуют понятия, которые невозможно мыслить вне отношений к другим понятиям. Например, понятие *компилятор* поясняет один из типов трансляции алгоритма, записанного на языке высокого уровня, в язык машинных кодов.

Несмотря на большое многообразие понятий в смысле их содержания, число возможных отношений между объектами понятий невелико и строго ограничено.

Будем называть два понятия *сравнимыми*, если в содержании их имеются общие признаки. Сравнимые понятия принято делить на две категории:

- совместимые понятия;
- несовместимые понятия.

Совместимыми называются понятия, объемы которых полностью или частично совпадают. Между совместимыми понятиями могут существовать следующие виды отношений:

- отношение тождества;
- отношение частичного совпадения;
- отношение подчинения.

1. *Отношение тождества*. В таком отношении находятся понятия, объемы которых полностью совпадают, сами понятия называются “тождественными”, или “равнозначными”. Примеры таких понятий:

- программа на Турбо Паскале — формальный способ записи алгоритма на языке Турбо Паскаль;
- Дж. Буль — основатель алгебры высказываний.

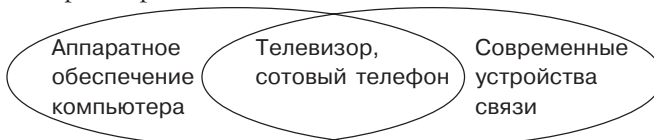
Следует заметить, что хотя равнозначные понятия имеют один и тот же объем, но содержание их различно. И различие это является чисто логическим, т.е. оно заложено не в объекте, а в мыслях (представлении) об этом объекте.

Так, например, в понятии *Дж. Буль* мыслится английский ученый XVIII века. А в понятии *основатель алгебры высказываний* в первую очередь мыслится Дж. Буль как математик, сформулировавший основные законы алгебры логики.

В понятии *программа на Турбо Паскале* мыслится текст, записанный на языке Турбо Паскаль, который надо отладить. В понятии *формальный способ записи алгоритма на языке Турбо Паскаль* мыслится, что существуют разные способы записи алгоритмов и алгоритм, записанный на языке Турбо Паскаль, является формальным предписанием исполнителю — компьютеру.

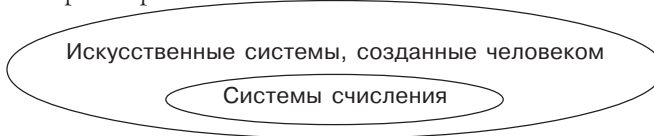
2. *Отношение соподчинения*. Определения, объемы которых частично совпадают, называются *перекрещивающимися*.

Пример



3. *Отношение подчинения*. Одно понятие называется подчиненным другому, если объем первого понятия входит как часть в объем второго понятия.

Пример



Работа с перекрещивающимися и подчиненными понятиями дает учителю возможность понять, насколько четко учащиеся ориентируются в изучаемой теме. Например, учащимся 11-х классов СУНЦ МГУ был задан вопрос “Чем операторы в Турбо Паскале отличаются от операций?”. Для того чтобы сформулировать верный ответ (“Операции — элементарные действия исполнителя, оператор — законченная последовательность действий”), мы с учащимися должны были пройти длинный путь: вспомнить, какие операторы есть в Турбо Паскале, какие действия есть в Турбо Паскале, для этого в качестве подсказки вспомнили, что действия имеют приоритет выполнения и делятся на 4 группы (унарную, мультипликативную, аддитивную и группу операций отношения). Все это показывает, как непросто в головах школьников выстроить логически стройное восприятие изучаемой темы.

❓ Как из нескольких похожих на первый взгляд определений выбрать наиболее удачное?

Определяя понятие, мы вводим новое слово (термин), обозначающее данное понятие. Но, как известно, для многих понятий вводятся в разных учебниках похожие, но различно воспринимаемые определения. Например, неоднозначно до сих пор обстоит дело с определением понятия *алгоритм*.

Определение по [1]. *Алгоритм* — последовательность команд, управляющих работой какого-либо объекта.

Определение по [2]. Под *алгоритмом* понимают понятное и точное предписание (указание) исполнителю совершить определенную последовательность действий, направленных на достижение указанной цели или решение поставленной задачи.

Определение по [3]. *Алгоритм* — совокупность правил выполнения определенных действий, обеспечивающих решение задачи.

Очевидно, что в школьном курсе информатики можно дать только интуитивное понятие алгоритма, более или менее формализованное. Но при этом мы на основе данного определения должны выяснить основные свойства алгоритма. Какое из этих определений выбрать?

Каждое из приведенных определений требует уточнений, например, что число команд у исполнителя конечно, что и сама последовательность команд конечна (а иначе как записать бесконечную последовательность указаний?). Другое дело, будет ли эта записанная последовательность команд исполнителя выполнена за конечное число шагов, т.е. не заикнется ли наш алгоритм при выполнении?

Мы в СУНЦ МГУ пользуемся следующим определением алгоритма:

Определение по [7]. *Алгоритм* — строгая и четкая конечная система правил, которая определяет последовательность действий исполнителя над некоторыми объектами и после конечного числа шагов приводит к поставленной цели. *Объекты*, над которыми исполнитель может совершать действия, образуют *среду исполнителя*.

Правила построения (выбора, анализа) определений

1. Определение должно быть соразмерным, т.е. объемы определяемого и определяющего понятий должны быть равны.
2. Определение не должно заключать в себе “порочного круга”, т.е. не должно ссылаться на определяемое понятие.
3. Определение не должно быть отрицательным.
4. Из определения должны вытекать все свойства определяемых объектов, составляющих объем определения. То есть определение должно быть исчерпывающим.
5. Определение должно использовать разъясняющую терминологию, доступную уровню развития учащихся. То есть определение должно быть “узнаваемо”.
6. Определение, вводимое в качестве подготовки, постепенного привыкания к изучаемым объектам, может определяться на бытовом, ассоциативном уровне.

Работающие и неработающие понятия

Отвечая на первоначально поставленный вопрос: “Перегружен ли курс информатики определениями?” — следует взглянуть на проблему вот с какой точки зрения: а все ли вводимые понятия являются “работающими”?

В учебной литературе вводится достаточно много понятий, терминов (особенно в разделах, относящихся к содержательной линии “Современные информационные

технологии”), с которыми на уроках активно не работают, кроме того, использование терминов из “взрослой, настоящей” информатики не улучшает восприятие школьниками изучаемой темы.

Приведем статистику используемых терминов (по мере появления их в учебнике) по некоторым разделам (см. с. 7):

Следует также отметить, что наибольшее количество терминов используется при изложении раздела “Устройство компьютера”:

Семакин [1]	Кузнецов [2]	Макарова [3]
37 определений	23 определения	47 определений

При этом та техническая информация, которая предъявляется учащимся, скорее всего большинством востребована не бывает. Обилие новых терминов с одновременным отсутствием логической связи с другими разделами (и в силу ограниченности времени, отводимого на информатику, и в силу ненужности ее установления в рамках средней школы) рождает отношение к учебнику, как к “Краткому энциклопедическому словарю”.

И здесь мы подходим к основной проблеме: какие определения вводить в курсе информатики? какие понятия можно вводить на “бытовом” уровне без ущерба для целостности курса (или для улучшения его восприятия)? насколько важно вводить или не вводить установившуюся терминологию науки информатики в школьную информатику?

Сегодняшний курс информатики явно делится на две крупные части: теоретическая информатика и пользовательский курс, обучающий основам офисной технологии обработки информации. Попытки изучать основы теоретической информатики через продукты Microsoft Office приводят, с нашей точки зрения, к резкому “утяжелению” курса. Попытка рассказывать ученикам 10–11-х и тем более 7–8-х классов принципы построения OLE-технологий, как нам кажется, преждевременна, хотя использованию в практической деятельности OLE-технологии они обучаются, но без постоянной практики приобретенные навыки быстро теряются. Можно согласиться, что введенные понятия *объект OLE*, *сервер OLE*, *клиент OLE* относятся к “неработающим”, а следовательно, в проверочные работы нельзя выносить вопросы на понимание этих понятий.

Все понятия, формируемые при изучении разделов, относящихся к основам теоретической информатики, должны быть построены на вышеизложенных методических принципах. Понятия, используемые в этих разделах, логически связаны между собой, что облегчает работу по вводу и усвоению понятий.

Овладение понятиями, формируемыми в разделах, связанных с пользовательскими технологиями, как правило, требуют большого практического тренинга. В этом случае часть понятий достаточно сформулировать на “бытовом” уровне, их понимание и дальнейшая формализация возможны только после выполнения длительных практических работ. При невозможности проведения практических работ эти понятия автоматически переходят в разряд “неработающих” понятий, а следовательно, их непедagogично выносить в контрольные и тестовые задания.

Именно наметившимся разделением курса информатики на две части объясняется разное количество определений, вводимое в школьных учебниках. Ориентация учебника (на теоретическую информатику или на освоение

Семакин [1]	Кузнецов [2]	Макарова [3]
Компьютерные телекоммуникации		
<ol style="list-style-type: none"> 1. Компьютерная сеть 2. Локальные сети 3. Файл — сервер 4. Рабочие станции 5. Компьютер — сервер 6. Сетевая операционная система 7. Глобальная сеть 8. Телекоммуникационная сеть 9. Телекоммуникация 10. Хост-машина 11. Терминал 12. Отраслевые сети 13. Региональные сети 14. Шлюз 15. Мировая система компьютерных сетей 16. Электронная почта 17. Электронное письмо 18. Почтовый ящик 19. Электронный адрес 20. Составное имя узла сети 21. Телеконференция 22. Линии связи 23. Терминал абонента 24. Модем 25. Демодуляция 26. Протокол работы сети 27. Технология "клиент/сервер" 28. Клиент — программа 29. Сервер — программа 30. Internet 31. Web-страница 32. Web-сервер 33. Гипертекст 34. World Wide Web 35. Web-браузер 36. Поисковые программы 37. Гиперпространство 	<ol style="list-style-type: none"> 1. Средства телекоммуникации 2. Модем 3. Скорость передачи данных 4. Компьютерная телекоммуникационная сеть 5. Компьютер — сервер 6. Компьютер — абонент 7. Режим on-line 8. Режим off-line 9. Маршрутизация 10. Электронная почта 11. Электронная доска объявлений 12. Телеконференция 13. Распределенные базы данных 	В данном учебнике эта тема не раскрывается
Технология обработки числовой информации (электронные таблицы)		
Семакин [1]	Кузнецов [2]	Макарова [3]
<ol style="list-style-type: none"> 1. Вычисляемые (зависимые) поля 2. Независимые поля 3. Исходные данные 4. Табличные процессоры 5. Ячейки 6. Адрес ячейки 7. Вещественные числа с фиксированной точкой 8. Вещественные числа с плавающей точкой 9. Мантисса 10. Порядок 11. Фрагмент 12. Статистические функции 13. Графический режим 	В данном учебнике эта тема не раскрывается	<ol style="list-style-type: none"> 1. Табличный процессор 2. Ячейка 3. Строка 4. Столбец 5. Блок ячеек 6. Текущий объект электронной таблицы 7. Ссылка 8. Формулы 9. Операнд 10. Форматирование 11. Общий формат 12. Фиксированный формат 13. Процентный формат 14. Денежный формат 15. Экспоненциальный (научный) формат 16. Однотипные (подобные) формулы 17. Абсолютная ссылка 18. Относительная ссылка 19. Полная абсолютная ссылка 20. Частичная абсолютная ссылка 21. Диаграмма 22. Линейчатая диаграмма 23. Поверхностная диаграмма 24. Круговая диаграмма 25. Столбчатая диаграмма 26. Смешанная диаграмма 27. База данных 28. Структурирование данных 29. Поле 30. Запись 31. СУБД

современных информационных технологий) с необходимостью приводит к разным системам используемых понятий. Но с методической точки зрения строгая система основных понятий не может быть построена в сложившейся ситуации, так как попытка воспринимать информатику и как науку, и как искусство (ремесло), а преподавать ее как логически стройную дисциплину изначально обречена на неудачу.

Хорошего гончара учат в гончарной мастерской, а не на уроках стереометрии в школьном классе. И в гончарной мастерской своя система понятий, отличная от системы понятий курса геометрии и стереометрии.

Мне кажется, что мы подошли к рубежу, за которым надо принять решение о формальном разделении курса информатики на два фактически сложившихся курса.

Литература

1. Семакин И., Залогова Л., Русаков С., Шестакова Л. "Информатика. Учебник по базовому курсу". М.: Лаборатория базовых знаний, 1998.
2. Кузнецов А.А., Апатова Н.В. Основы информатики. 8—9-е классы. М.: Дрофа, 1999.
3. Информатика. 7—8-е классы / Под ред. Н.В. Макаровой. СПб.: Издательство "Питер", 2000.
4. Казиев В.М. Информатика: понятия, виды, получение, измерение и проблема обучения // ИНФО № 4, 2000.
5. Солсо Р. Когнитивная психология. М.: Триволта, 1996.
6. Никитин В.В., Рутасов К.А. Определения математических понятий в курсе средней школы. Пособие для учителей. М.: Учпедгиз, 1963.
7. Андреева Е.В., Котик У.В., Фалина И.Н., Чернокожин Е.В. Информатика (Пособие для поступающих в вузы). М.: Изд-во ВМиК МГУ, 2000.

Азы информатики

А.А. ДУВАНОВ

МАТЕРИАЛЫ
Роботландского
университета

Книга 1. См. № 1, 2/2002


Книга 2. См. № 5, 6/2002

Книга 2. В мире информации (продолжение)

Книга для ученика

3 Что можно делать с информацией
Читальный зал Роботландии
Переславль-Залесский RU 2001
РОБОТЛАНДИЯ.RU © А.А.ДУВАНОВ

Информацию можно хранить, передавать и обрабатывать.



3. ЧТО МОЖНО ДЕЛАТЬ С ИНФОРМАЦИЕЙ

Читальный зал

Мы живем в информационном мире

В последнее время Вася Кук заметил: все, что он делает, так или иначе связано с информацией. Вот три истории, которые приключились с Васей за один только день — прошлую субботу.

История первая. Вася читает библиотечную книгу

Вася взял в школьной библиотеке книгу про животных. Он сразу открыл главу о кошках. Вася прочитал, что были случаи, когда кошку увезли за сотни километров и она все равно возвращалась назад. Вася понял, что кошки очень хорошо запоминают дорогу. В своем читательском дневнике Вася записал эту мысль.



Обсуждение первой истории

— О чем получил информацию Вася Кук? В каком виде представлена эта информация?

Кук узнал о случаях, когда кошки возвращались, будучи увезенными далеко от дома. Эта информация представлена в текстовом виде.

— Где находилась эта информация?

Местом **хранения** информации была библиотечная книга о животных. Еще точнее — глава, посвященная кошкам.

— Как Вася получил эту информацию?

Для **передачи** информации (из книги в память) Вася Кук вос-



пользовался глазами. У этой передачи информации есть точное направление: из книги — в Васину голову. Информация как бы вводится в память.

— Где оказалась информация, когда Вася ее получил?

Новое место **хранения** информации — Васина память.

— Какое заключение сделал Кук, прочитав книгу?

Результат **обработки** полученной информации — новая информация о способностях кошек



запоминать дорогу. Эта обработка информации выполнена в Васиной голове.

— Что сделал Вася с полученной информацией?

Произошла **передача** информации — Вася записал ее в свой дневник.

Эта передача представляет собою вывод обработанной информации на бумагу.

Итак, в этой истории легко увидеть, что происходило с информацией:

— сначала **хранение** (в книге),

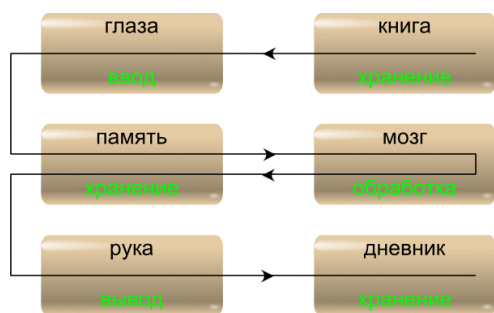
— потом **передача** (ввод в память),

— еще раз **хранение** (в памяти),

— затем **обработка** и получение новой информации (в голове),

— **хранение** новой информации (в памяти),

— наконец, еще одна **передача** (вывод на бумагу).



История вторая. Вася у классной доски

На уроке математики Кука вызвали к доске. Учитель написал на доске пример:

$$(12 + 13) \cdot 4$$

Вася, конечно, справился с этой задачей. А потом, вернувшись на место, задал себе вопросы:

— В каком виде была представлена информация?

— Как я получил информацию, необходимую для решения этой задачи?

— Где хранилась информация, пока я решал задачу?

— Что происходило с информацией во время решения?

— Где хранился результат после решения?

— Как результат стал известен учителю и всему классу? Как была выведена информация?

Кук нашел ответы на все вопросы. А вы?

История третья. Вася играет с Костей

Соседский малыш Костя обратился к Васе за помощью: ему никак не удавалось собрать самолет из нового “Конструктора”. Вася внимательно изучил чертеж, подумал немного и понял, что главная трудность — правильно собрать и прикрепить крылья. Он объяснил Косте, как это сделать. Тот увлеченно принялся за работу, а Вася опять стал задавать себе вопросы:

— Какую информацию я получил? Где она хранилась? В каком виде представлена?

— Как я обрабатывал информацию? Какая новая информация получилась после обработки?

— Что произошло с этой новой информацией? Как я передал ее?

— Мы с Костей смотрели на один и тот же чертеж. Почему он сам не сумел построить самолет, а после моей подсказки сделал это очень быстро?

Во всех трех историях можно выделить следующие этапы работы с информацией:

— хранение информации,

— передача информации,

— обработка информации.

Хранение, передача и обработка — это **информационные процессы**. Все, что происходит с информацией, всегда включает хранение, передачу и обработку.

Конспект

Информацию можно **хранить, передавать и обрабатывать**.

Хранение, передача и обработка — это **информационные процессы**.

Информация хранится в памяти человека и животных, в природе, на специальных устройствах (бумага, кинолента, магнитные и лазерные диски).

Информацию можно передавать и получать. Информация поступает от окружающей природы. Устройства для передачи информации: радио, телевидение, телефон, Интернет. Человек и животные получают информацию благодаря своим органам чувств.

Обрабатывая информацию, получают новую информацию. Человек и животные обрабатывают информацию, используя природный дар — работу собственного головного мозга. Устройства для механизации и автоматизации обработки информации: арифмометр, калькулятор, компьютер.

Вопросы

1. Ответьте на вопросы, которые поставил себе Вася во второй истории.

2. Ответьте на Васиные вопросы из третьей истории.

3. Какие информационные процессы приводят к появлению новой информации?

4. В каждом из следующих примеров укажите, о каком информационном процессе (хранении, передаче или обработке) идет речь:

— Игорь пишет диктант.

— Лида смотрит телевизор.

— На столе лежит учебник по информатике.

— Илья решает задачу.

— Решение было записано в тетради.

— В комнате работает радиоприемник.

— На одном лазерном диске — песни, а на другом — компьютерная игра.

— Иван набрал свое имя в редакторе строки.

— Руслан увидел на экране компьютера красивую кнопку.

— Нажму-ка я эту кнопку, — подумал Руслан.

— Руслан нажал кнопку и увидел на экране кукиш.

— Сам осел, — сказал Руслан компьютеру и выключил его.



Задания на дом



Вариант 1

1. Какие действия с информацией выполняются, если вы слушаете радио?

2. Вася читает письмо от друга. Укажите, на чем хранится, откуда и куда передается при этом информация.

3. Учитель записал условие задачи на доске. Ученики решили задачу в тетради. Какие информационные процессы при этом происходили (опишите по шагам)?

4. Радист на океанском лайнере принял радиограмму, переданную азбукой Морзе, и записал для капитана: “После 16 часов возможен 6-бальный шторм”. Произошла ли при этом обработка информации?



Вариант 2

1. Какие действия с информацией выполняете вы и компьютер, когда исправляете ошибки в набираемом тексте?

2. Вы смотрите демонстрационный ролик компьютерной игры. Укажите, на чем хранится, откуда и куда передается при этом информация.

3. Ваш лучший друг передал вам диск с новой компьютерной программой.

Вы установили программу на свой компьютер. Опишите информационные процессы, которые при этом происходили.



Вариант 3

1. Опишите реальную ситуацию, в которой с информацией происходят все информационные процессы: хранение, передача, обработка.

2. Придумайте и зарисуйте ребусы для названий информационных процессов.

3. Опишите, какие информационные процессы выполняются, когда вы ищете нужную вам информацию в библиотеке или в Интернете.



Практикум

Редактор строки

Сегодня потренируемся вводить в редактор строки знаки препинания и специальные символы.

Точки, запятые, скобки, кавычки и другие знаки препинания встречаются в текстах очень часто. В записи арифметических выражений не обойтись без знаков “+” и “-”. Для умножения и деления на клавиатуре ЭВМ тоже есть свои символы, хотя они и не очень привычны: в качестве знака умножения часто

пользуются звездочкой “*”, а деление обычно обозначают наклонной чертой “/”. Например, обычное арифметическое выражение

$$(2 \cdot 5 + 4 \cdot 3) : 6$$

часто приходится набирать на клавиатуре так:

$$(2 * 5 + 4 * 3) / 6.$$

Когда нужно записать сравнение двух чисел, используют символы “>” (больше), “<” (меньше) и “=” (равно).

На клавиатуре есть еще несколько “странных” символов: “#”, “\$”, “&”, “@”, “^”, “|” — и другие. Смысл некоторых из них вы узнаете позднее.

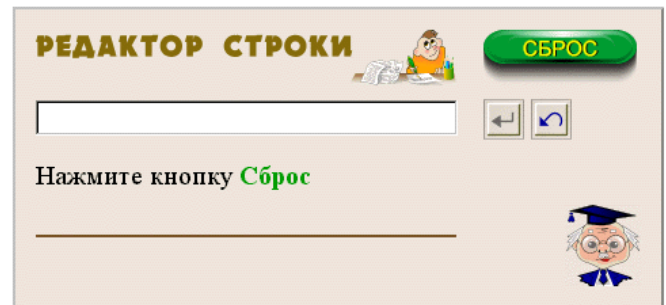
Все символы клавиатуры надо уметь набирать. Для этого надо сначала определить, на какой половинке клавиши — верхней или нижней — нарисован нужный символ. В зависимости от этого надо выбрать соответствующий регистр.



На некоторых клавишах можно увидеть не два, а даже три специальных символа.

Действие этих клавиш зависит не только от регистра (заглавные — малые), но и от алфавита (русский — латинский).

Часто бывает так, что на разных клавиатурах расположение символов немного иное, как и правила набора. Вам придется определить эти правила опытным путем, поработав с Редактором строки.



Задания исполнителя: наберите указанные символы.

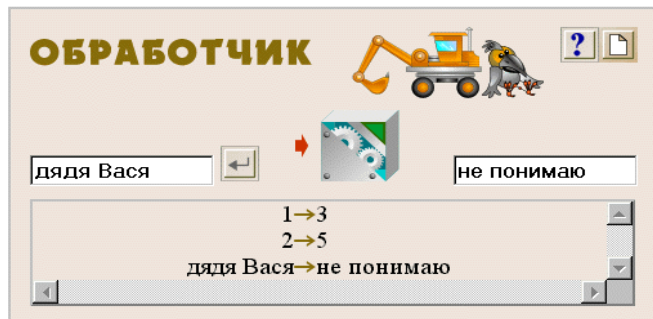
- 1) $(4 * 3 - 2 * 5) + 2 = 5 - 1$;
- 2) “Кто там?”;
- 3) 50% от 120 есть 60;
- 4) Оля, ау! Где ты? Я — тут!
- 5) — Не хрюкай, — посоветовала ему Алиса;
- 6) “Может, он захныкал, а не захрюкал?”;
- 7) :) — так обозначают улыбку;
- 8) :(— так обозначают огорчение;
- 9) $5 < 10$, а $8 > 3$;
- 10) $(*) + (@) + (\#) + (\^{\sim}) + (\$) + (\&)$.

Работа с информацией

Обработчик — это робот, который обрабатывает информацию по заложенному в него алгоритму.

Обработчик берет информацию из строки ввода (слева) и размещает результат в окошко вывода (справа). Свою работу он заносит в дневник, который виден на экране.

В качестве результата **Обработчик** может написать сообщение “Не понимаю”. Это означает, что алгоритм не обрабатывает ту информацию, которая была задана. Например, алгоритм, обрабатывающий числа, на ввод “дядя Вася” равнодушно сообщит: “Не понимаю”.



Попробуйте отгадать три алгоритма **Обработчика**.

1. Первый алгоритм **Обработчика**.

Вход	Выход
1	3
2	5
3	7
4	9
5	11
10	?

2. Второй алгоритм **Обработчика**.

Вход	Выход
1	1
2	1
10	2
кот	3
лиса	4
молоко	?

3. Третий алгоритм **Обработчика**.

Вход	Выход
1	3
2	3
10	4
кот	5
лиса	6
молоко	?

Вопросы

- Где хранится информация перед тем, как она передается **Обработчику**?
- Что нужно сделать, чтобы информация передалась в хранилище **Обработчика**?
- Сколько символов информации вмещает хранилище **Обработчика**?
- Как информация передается **Обработчику** из хранилища?
- Что делает **Обработчик** с информацией?
- Где хранятся результаты обработки?
- Как отгадать алгоритм **Обработчика**?



Зачетный класс

- Память компьютера состоит из ячеек, которые называются **байтами**. В одном байте может храниться один символ. Сколько байт нужно для хранения сообщения: “Что за шум?”?

2. Сколько байт нужно для хранения сообщения:

— Нет-нет! 2 + 2 не 5 :)

3. Напишите сообщение, которое можно разместить не более чем в 5 байтах.

4. Какой основной информационный процесс происходит в каждом описанном случае?

Петя читает сказку.

Ира нажимает клавиши на клавиатуре.

Ваня решает задачу.

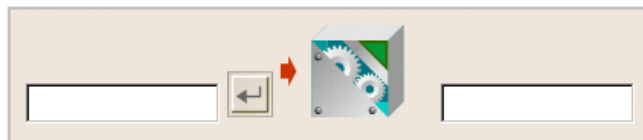
В блокноте записан номер телефона.

В библиотеке много книг.

5. Напишите сообщение, которое не поместится в 5 байтах.

6. В Интернете средняя скорость передачи информации равна примерно 1000 байтам в секунду. Сколько потребуется секунд для передачи страницы, содержащей четыре тысячи знаков, включая пробелы?

7. Исполнитель работает с любыми числами, но устройство ввода у него сломалось и может вместить только один знак:



Что получится в результате обработки числа 15 после того, как устройство ввода будет отремонтировано?

Ниже приводится возможный протокол работы с исполнителем:

Вход	Выход
1	2
2	3
3	4
4	5
15	?

8. Исполнитель с поврежденным устройством ввода из задания 7 настроен на новый алгоритм. Что получится в результате обработки числа 15 после того, как устройство ввода будет отремонтировано?

Ниже приводится возможный протокол работы с исполнителем:

Вход	Выход
1	2
2	5
3	8
4	11
15	?

9. Исполнитель с поврежденным устройством ввода немного отремонтирован. Теперь в него можно вводить до трех символов! Что получится в результате обработки слова “листик” после того, как устройство ввода будет полностью отремонтировано?

Ниже приводится возможный протокол работы с исполнителем:

Вход	Выход
1	1
12	21
сон	нос
листик	?

Книга для учителя

3. ЧТО МОЖНО ДЕЛАТЬ
С ИНФОРМАЦИЕЙ

Поход за информацией

Тема позволяет провести содержательный урок информатики на свежем воздухе, где-нибудь в лесу, парке, на берегу реки. А главное, вдали от душного компьютерного класса.

Цель такого урока: закрепить понятие информации, научить распознавать элементы информационных процессов в окружающем мире.

Такой урок-игру автор часто проводил в Международном детском компьютерном лагере в Переславле-Залесском. Ниже описан один из возможных сценариев, который учитель может взять за основу, привязав к местным условиям.

Сценарий игры

В поход идет вся детская группа, но ребята разбиваются на две команды.

Поэтому учителю потребуется помощник — вожакий, шеф-старшеклассник или кто-нибудь из родителей.

Подготовка к этому уроку требует значительно больше усилий, чем обычно. Учителю надо предварительно пройти по маршрутам обеих команд, тщательно их проверить, заготовить и разложить по тайникам записки-информацию, сделать необходимые метки, запастись компасами и лопатками, познакомить с маршрутом своего помощника и проинструктировать его.

Учитель начинает урок. Распахивается дверь. Появляется человек с запиской интригующего содержания. Сообщение зачитывается всему классу: “Срочно заканчивайте урок! Робот **Обработчик** в беде! Он забыл волшебное слово и не может больше работать. Волшебное слово найти непросто. Пусть две команды возьмут листки с информацией и немедленно отправляются в путь по двум разным маршрутам. Не забудьте взять компасы!”

Вслед за содержанием некоторых из приводимых ниже листков указаны комментарии. Это пояснения для учителя. Информация, которую будут разыскивать школьники, представляет собою волшебные слова “компьютер” и “информация”.

Идите к старому дереву.
С северной стороны копайте.

Листок 1а

(а — индекс первой команды).

Идите к выходу на полянку.
Направо у двух ближних деревьев копайте.

Листок 1б

(б — индекс второй команды).

Дети отправляются по маршруту и находят в указанных местах буквы: “п” — первая команда, “м” — вторая команда, а также новые листочки с информацией.

25 шагов на юг и
30 шагов на восток.
Ищите наверху.

Листок 2а.
Надо поднять головы
и поискать на дереве.

Новый этап поиска дает: “о” — для первой команды, “я” — для второй и следующую порцию информации для дальнейшего поиска.

Расшифруйте сообщение,
в котором поменялись
местами две пары букв.

Пе трепанку к езору.
Пед влажнйшам
ероухевым кустем
ащато.

Листок 2б.
Здесь поменялись местами
буквы “о” и “е”, “а” и “и”.

Идите вдоль забора.
Там, где тропинка
уходит от забора,
лежат три камня.
Ищите.

Листок 3а.
Записка лежит под одним
из камней.

Результаты поиска: “е” — первая команда, “р” — вторая и новая информация.

15 шагов на
юго-запад.
Остановитесь у
высокой сосны.
Ищите на дереве.

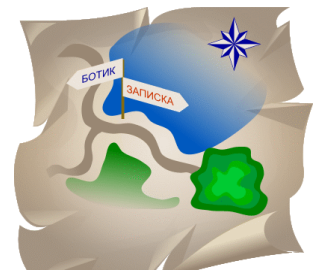
Листок 3б.
Записка в дупле дерева.

Следуйте по тропинке
дальше. У опушки
леса муравейник.
В п метрах от него на
юг - ямка.
 $p = (24 \cdot 6 \cdot 2) - (14 / (21 - 24))$

Листок 4а.
Нужно вычислить значение *p*.

На этом этапе было обнаружено: “т” — первая команда, “и” и еще “и” — вторая команда.

Ребята из второй группы удивлены: сразу две буквы “и”. Что бы это значило?



Листок 4б.
Схема с тропинкой и местом,
где лежит очередная записка.

Идите к старому дереву.
С северной стороны копайте.

Листок 1а

(а — индекс первой команды).

Идите к выходу на полянку.
Направо у двух ближних деревьев копайте.

Листок 1б

(б — индекс второй команды).

В двадцати метрах
от муравейника
на север мост
через ручей.
Ищите на дне.

Листок 5а.
Записка нацарапана на доске,
которая положена под камень
на дне ручья прямо под мостом.

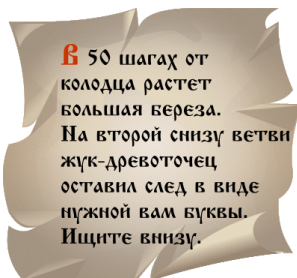
Около домика
лесника стоят
три березы. Ваша -
самая высокая.

Листок 5б.
Записка приколота
у основания ствола.

Вот что нашли ребята на сей раз: “к” — первая команда, “н” — вторая.

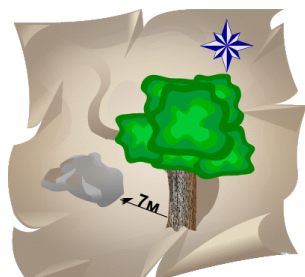


Листок 6а.
Записка под перевернутой лодкой.



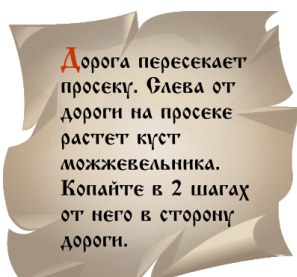
Листок 6б.
Листок с дальнейшим маршрутом — в траве точно под указанной веткой.

Информация этого этапа поиска: “ю” — первая команда, “о” — вторая команда.

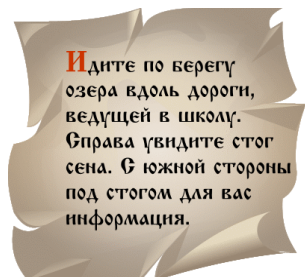


Листок 7а.

Найденные буквы: “м” — первая команда, “а” — вторая.

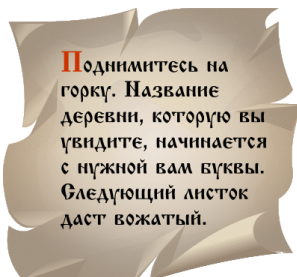


Листок 7б.

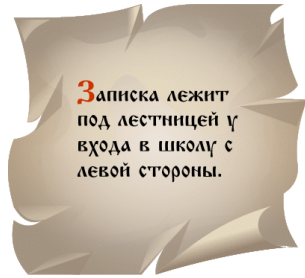


Листок 8а.

В результате этого этапа получено: “р” — первая команда, “ф” — вторая (деревня называется Федорово, заранее этого сказать было нельзя, так как рядом, но вне поля зрения находится Орехово).

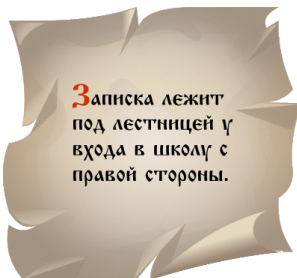


Листок 8б.



Листок 9а.

Ребята догадались, конечно, какое “заклинание” им следовало разыскать, и, найдя последнюю букву в указанном месте, составляют слова: первая команда — “компьютер”, вторая команда — “информация”.



Листок 9б.

Обсуждение игры

Во время азартной игры, конечно, не до обсуждений информационных процессов, которые сопровождали детей в их занимательной беготне на свежем воздухе.

Однако цель игры не будет достигнута, если не проанализировать ее с точки зрения информатики. В качестве домашнего задания предложите детям озаглавить в тетради три страницы названиями информационных процессов и записать на них краткое содержание игры.

Записи, вероятно, будут многочисленными. Помимо работы с записками, дети вспомнят и другие информационные происшествия.

— Когда я полез на дерево, каркнула ворона. Сначала я испугался и чуть не свалился с сосны. Потом понял, что бояться нечего!

— Вода в реке морщилась на солнце от маленьких водяных брызг. Я понял, что это играет рыба.

— Мы увидели на просеке пеня и стали считать его кольца. Дереву исполнилось 30 лет, когда его сразил топор дровосека.

И так далее. В итоге дети запишут информационную картину своего приключения.

Хранение	Передача	Обработка
Информация хранилась в записке	Информация поступала в наши головы как зрительный образ	Мы обрабатывали информацию и получили новую: как найти следующую записку
....

Ответы на вопросы

1. Учитель написал на доске пример:

$$(12 + 13) \cdot 4$$

Вася, конечно, справился с задачей. А потом, вернувшись на место, задал себе вопросы:

— В каком виде была представлена информация?
— Как я получил информацию, необходимую для решения этой задачи?

— Где хранилась информация, пока я решал задачу?
— Что происходило с информацией во время решения?
— Где хранился результат после решения?

— Как результат стал известен учителю и всему классу? Как была выведена информация?

Ответы.

— Информация была представлена в числовом виде.
— Условие задачи Вася увидел на доске (зрительный канал приема информации) и запомнил (“записал” в свою память). Знания о правилах математических действий уже хранились в его памяти.

— Информация хранилась в Васиной памяти.
— Во время решения информация Васей обрабатывалась.

— Результат был получен в Васиной голове и хранился в его памяти.

— Вася записал результат мелом и таким образом передал информацию из своей памяти на школьную доску.

2. Костя не смог сам разобраться с “Конструктором”. Вася изучил чертеж, подумал немного и понял, что главная трудность — правильно собрать и прикрепить крылья. Вася объяснил Косте, как это сделать, и опять стал задавать себе вопросы:

— Какую информацию я получил? Где она хранилась? В каком виде представлена?

— Как я обрабатывал информацию? Какая новая информация получилась после обработки?

— Что произошло с этой новой информацией? Как я передал ее?

— Мы с Костей смотрели на один и тот же чертеж. Почему он сам не сумел построить самолет, а после моей подсказки сделал это очень быстро?

Ответы.

— Была получена зрительная информация (Вася смотрел на чертеж). Чертеж хранился на бумаге. Чертеж представлял собой смешанный вид информации (текстовая, графическая и числовая).

— Вася смотрел на чертеж и думал, а значит, обрабатывал информацию.

Новая информация, полученная в процессе обработки:

1) главная трудность — правильно собрать и прикрепить крылья;

2) Вася понял, как надо собрать и прикрепить крылья.

— Новая информация была передана Косте словами. Он воспринял ее ушами.

— Костин “обрабатывающий” прибор (его голова) недостаточно опытен! Ведь Костя еще маленький. Но Костя способный, он быстро понял объяснения Васи.

3. Какие информационные процессы приводят к появлению новой информации?

Ответы.

1) Передача информации. Новую информацию получает тот, кто принимает информацию.

2) Обработка информации.

4. В каждом из следующих примеров укажите, о каком информационном процессе (хранении, передаче или обработке) идет речь:

— Игорь пишет диктант.

— Лида смотрит телевизор.

— На столе лежит учебник по информатике.

— Илья решает задачу.

— Решение было записано в тетради.

— В комнате работает радиоприемник.

— На одном лазерном диске — песни, а на другом — компьютерная игра.

— Иван набрал свое имя в редакторе строки.

— Руслан увидел на экране компьютера красивую кнопку.

— Нажму-ка я эту кнопку, — подумал Руслан.

— Руслан нажал кнопку и увидел на экране кукиш.

— Сам осел, — сказал Руслан компьютеру и выключил его.

Ответы.

— Игорь пишет диктант. Учитель передает информацию детям в классе.

Это звуковая информация. Она запоминается в памяти учеников, потом передается из памяти детей на бумагу.

На самом деле в этом информационном примере задействована и обработка. Когда дети слышат “малако”, они преобразуют звук в текст “молоко” по правилам русского языка.

— Лида смотрит телевизор. Прежде всего это передача информации от телевизора (звуковая и зрительная информация) в память Лиды. Конечно, это не просто прием информации. Когда Лида смотрит передачу, она думает, делает выводы, а значит, обрабатывает информацию. Например, Лида увидела, как герой мультфильма попал в неприятную историю из-за двоек в школе. Лида подумала и решила, что учиться не просто интересно, но и очень полезно для жизни.

— На столе лежит учебник по информатике. В нем хранится информация.

Ее никто не обрабатывает и никто не передает. Она просто хранится в учебнике.

— Илья решает задачу. Идет обработка информации.

— Решение было записано в тетради. Передача информации.

— В комнате работает радиоприемник. Передача звуковой информации.

— На одном лазерном диске — песни, а на другом — компьютерная игра.

Это хранение информации.

— Иван набрал свое имя в редакторе строки. Это передача информации.

— Руслан увидел на экране компьютера красивую кнопку. Это передача информации. Изображение с экрана передается в память Руслана.

— Нажму-ка я эту кнопку, — подумал Руслан. Обработка полученной информации.

— Руслан нажал кнопку и увидел на экране кукиш. Передача информации, обработка ее компьютером, передача новой информации Руслану.

— Сам осел, — сказал Руслан компьютеру и выключил его. Обработка полученной информации и попытка передать результат обработки компьютеру. Компьютер смог бы получить в свою память слова Руслана, если бы у него был микрофон, специальная программа, которая обрабатывала бы звуковую информацию, и если бы Руслан компьютер не выключил!

Ответы на вопросы практикума

Алгоритмы Обработчика:

1) **Обработчик** работает только с числовой информацией. Всякую другую (текстовую, смешанную) он отказывается обрабатывать и сообщает “не понимаю”. Обработчик умножает введенное число на два и прибавляет один.

2) Обработчик подсчитывает число введенных символов.

3) Обработчик подсчитывает число введенных символов и к полученному числу добавляет два.

Ответы на вопросы.

1. Где хранится информация перед тем, как она передается **Обработчику**?

Информация хранится в нашей голове.

2. Что нужно сделать, чтобы информация передалась в хранилище **Обработчика**?

Нужно щелкнуть мышкой по редактору строки и набрать информацию на клавиатуре.

3. Сколько символов информации вмещает хранилище **Обработчика**?

20 символов. Это легко определяется экспериментальным путем.

4. Как информация передается **Обработчику** из хранилища?

Информация поступает в **Обработчик** при нажатии на клавишу  или на экранную кнопку

 (изогнутая стрелка).

5. Что делает **Обработчик** с информацией?

Обрабатывает по заранее заложенному в него алгоритму. Алгоритмы в **Обработчик** заложил автор книги.

6. Где хранятся результаты обработки?

Результаты обработки отображаются в поле справа от исполнителя.

7. Как отгадать алгоритм **Обработчика**?

Нужно провести серию опытов и подумать (обработать полученную информацию). Затем проверить догадку новыми опытами.

Если **Обработчик** выводит результаты такими, как вы предполагали, то скорее всего алгоритм отгадан.

3. Напишите сообщение, которое можно разместить не более чем в 5 байтах.

Любой набор, содержащий от 1 до 5 символов, например, слово “кот”.

4. Какой основной информационный процесс происходит в каждом описанном случае?

Петя читает сказку.	Передача
Ира нажимает клавиши на клавиатуре.	Передача
Ваня решает задачу.	Обработка
В блокноте записан номер телефона.	Хранение
В библиотеке много книг.	Хранение

5. Напишите сообщение, которое не поместится в 5 байтах.

Например, сообщение из 6 символов “молоко”.

6. В Интернете средняя скорость передачи информации равна примерно 1000 байтам в секунду. Сколько потребуются секунд для передачи страницы, содержащей четыре тысячи знаков, включая пробелы?

Ответ: 4

7. Исполнитель работает с любыми числами, но устройство ввода у него сломалось и может вместить только один знак. Что получится в результате обработки числа 15 после того, как устройство ввода будет отремонтировано?

Исполнитель работает с числами и добавляет одно к входному числу. Если на вход подать 15, то на выходе получится 16.

8. Исполнитель с поврежденным устройством ввода из задания 7 настроен на новый алгоритм. Что получится в результате обработки числа 15 после того, как устройство ввода будет отремонтировано?

Исполнитель работает с числами, умножает входное число на 3 и отнимает один. Если на вход подать 15, то на выходе получится 44.

9. Исполнитель с поврежденным устройством ввода немного отремонтирован. Теперь в него можно вводить до трех символов! Что получится в результате обработки слова “листик” после того, как устройство ввода будет полностью отремонтировано?

Исполнитель переворачивает входную строку. Если на вход подать слово “листик”, то на выходе получится слово “китсил”.

Решения зачетного класса

1. Память компьютера состоит из ячеек, которые называются байтами. В одном байте может храниться один символ. Сколько байт нужно для хранения сообщения: “Что за шум?”?

Сообщение содержит 13 символов (8 букв, два символа пробела, две кавычки и знак вопроса). Для хранения потребуется 13 байт.

2. Сколько байт нужно для хранения сообщения:

— Нет-нет! 2 + 2 не 5 :)

Сообщение содержит 22 символа (8 букв, 5 пробелов, 2 знака “—”, один знак “+”, двоеточие, скобку, знак восклицания и три цифры). Для хранения потребуются 22 байта.

ИНФОРМАТИКА

Газета “Информатика” и Роботландский сетевой университет продолжают совместную акцию

RU Роботландский университет
WWW.botik.ru-robot
kurs@robotland.botik.ru

“Подписчикам везде у нас дорога... и скидка!”

Данный купон дает право на скидку в размере 10% при приобретении:

- интерактивного учебника-лаборатории “Знакомство с компьютером”;
- программного пакета “Конструирование”;
- интерактивного учебника-лаборатории “HTML-конструирование” (см. № 21, 22/2000);
- программного пакета “Зимние вечера” (см. № 1, 5—11, 13—18/2001);
- интерактивного учебника-лаборатории “JavaScript-конструирование” (см. № 21, 25, 29, 33/2001).

Для получения скидки необходимо выслать заявку на приобретение того или иного продукта по адресу: 152025, г. Переславль-Залесский, ул. Октябрьская, д. 43, кв. 112, Дуванову Александру Александровичу.

В письмо необходимо вложить оригинал данного купона или ксерокопию купона вместе с ксерокопией подписной квитанции на “Информатику”. Для быстрого получения программ рекомендуется дополнительно отправить электронное письмо с заявкой по адресу: kurs@robotland.botik.ru. В электронном письме требуется указать дату отправки бумажного письма с купоном или ксерокопиями.

Данное предложение действительно до 30 апреля 2002 г.



Привод CD-ROM — это сложное устройство для считывания информации с лазерных дисков (CD-дисков). Он состоит из одной или нескольких электронных плат управления, шпиндельного узла, оптической считывающей головки с приводом ее перемещения и механического устройства загрузки диска. В электронную часть, кроме схем непосредственного управления механизмом, входят процессор декодирования информации, получаемой при чтении диска, буферная память, интерфейс контроллера, интерфейс выхода звукового сигнала с разъемными блоками переключателей (джамперов).

Внешне можно различить следующие узлы CD-привода (рис. 1):

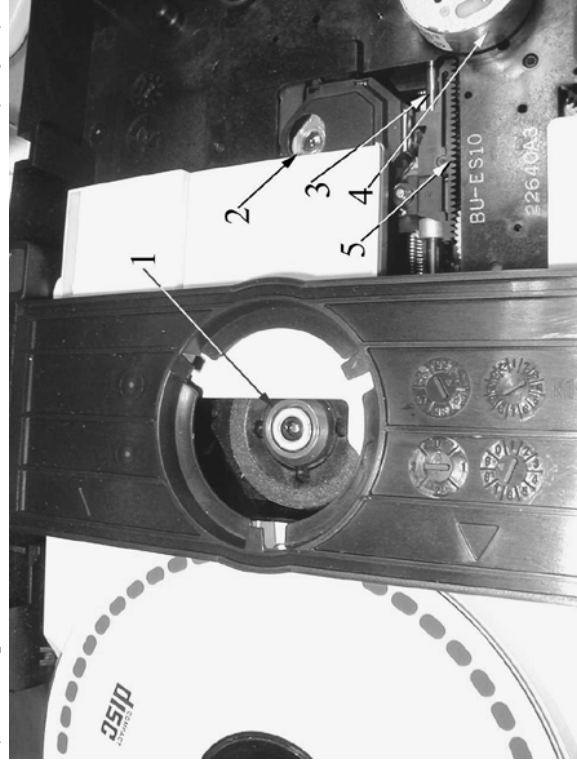


Рис. 1. Внутреннее устройство привода CD-ROM

1. Шпиндель с держателем диска. Прямо под ним располагается электродвигатель. Он служит для вращения диска. Односкоростными являются дисководы в музыкальных центрах. Они обеспечивают скорость обмена в 150 Кб/с, двухскоростные — 300 Кб/с и т. д. Однако если вы думаете, что в случае 48-скоростного устройства надо 150 Кб/с умножить на 48, то вы глубоко заблуждаетесь. Впрочем, обо всем — по порядку.

Во всех лазерных дисководах — от одно- до 12-скоростных — использовалась технология CLV (*Constant Linear Velocity* — постоянная линейная скорость). При этом данные с края компакт-диска считываются с такой же скоростью, как и данные с внутренних дорожек. В более скоростных CD-ROM используется иная технология — комбинация CLV и CAV (*Constant Angular Velocity* — постоянная угловая скорость), и данные внешних дорожек считываются намного быстрее внутренних.

Использование только постоянной линейной скорости привело бы к тому, что при чтении внутренних областей диска он вращался бы безумно быстро, что привело бы к разрушению или неправильной работе дисковода. Таким образом, внутренние области считываются в режиме CAV, а внешние — в режиме CLV.

Впрочем, особого выигрывает эта технология не дает, и большую часть времени 48-скоростной CD-ROM работает со скоростью 12-скоростного. Это, кстати, отмечается и в маркировке устройств. Не "48X", а "48X max". То есть максимальная скорость считывается иногда (на несколько долей секунды) достигает 150 × 48 Кб/с. Правда, высокоскоростные CD-приводы снабжаются, как правило, достаточно большой буферной памятью, что позволяет все же увеличить скорость считывания информации. Некоторые CD-ROM-приводы при многократных ошибках чтения могут автоматически снижать скорость вращения диска, однако большинство из них после этого не могут вернуться к своей максимальной скорости вплоть до смены диска.

Двигатель шпинделя используется как для разгона, так и для торможения диска. На оси шпин-

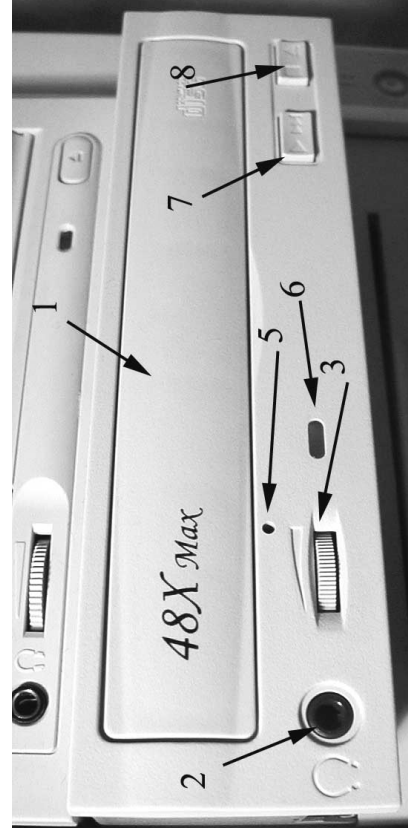


Рис. 2. Передняя панель привода CD-ROM

дельного двигателя (или в собственных подшипниках) закреплен собственно шпиндель, к которому после загрузки прижимается диск. Поверхность шпинделя иногда покрыта резиной или мягким пластиком для устранения проскальзывания диска (что хорошо видно на рис. 1). Диск прижимается к шпинделю при помощи верхнего прижима, расположенного с другой стороны диска. 2. Система оптической головки. Она состоит из самой головки (2) и узла ее перемещения, состоящего из суппорта (3). В головке размещены: лазерный излучатель на основе светодиода, система фокусировки, фотоприемник и частотный преобразователь. Система усиливает сигнал фокусировки, обычно представляющий собой подвижную линзу, которая с помощью особого электромагнитного устройства производит перемещение лазерного луча, компенсируя вертикальные биения диска даже при значительных скоростях вращения.

Привод перемещения головки имеет собственный электродвигатель (4), приводящий в движение каретку с оптической головкой при помощи зубчатой, ленточной либо червячной передачи (5). В качестве двигателя обычно используется шаговый двигатель. Чаще всего система загрузки диска происходит через выдвижной лоток (tray) (1 — см. рис. 2), на который кладется сам диск, и с использованием втяжного механизма. Такие системы содержат специальный двигатель, обеспечивающий выдвижение лотка. В довольно неплохих приводах фирмы Pioneer используется система с втяжным механизмом. Он содержит двигатель для втягивания и выброса дисков через узкую зарядную щель. В приводах, допускающих размещение диска в вертикальном положении, предусматриваются фиксаторы, удерживающие его при выдвинутом лотке.

На передней панели привода (рис. 2), кроме лотка (1), обычно расположены кнопка "Eject" (8) для загрузки/выгрузки диска, индикатор обращения к приводу (6) и гнездо для подключения наушников (2) с электронным или механическим регулятором громкости (3). В ряде моделей добавлена кнопка "Play/Next" для запуска проигрывания звуковых дисков и перехода между звуковыми дорожками (7). Большинство приводов также имеют на передней панели небольшое отверстие (5), предназначенное для аварийного извлечения диска в тех случаях, когда обычным способом это сделать невозможно, — например, при выходе из строя привода лотка или всего CD-привода, при исчезновении питания и т. п. В отверстие обычно нужно вставить шпильку или распорную скрепку и аккуратно нажать. При этом снимается блокировка лотка и его можно выдвинуть вручную.

Проигрывание звуковых дисков является побочной для CD-приводов функцией, и поэтому изменяемые для этого технические решения и элементная база обычно заимствуются у самых дешевых карманных проигрывателей — так что ожидать качества не приходится. Правда, некоторые производители добавляют в конструкцию своих приводов специальные электронные схемы, резонансные звуковые сигналы. Иногда такие устройства даже снабжаются пультами дистанционного управления, чтобы подчеркнуть близость к стационарным проигрывателям музыкальных компакт-дисков. Многие современные приводы имеют на задней стенке (рис. 3) дополнительный цифровой выход звука (5) в профессиональном формате SPDIF (*Sony/Philips Digital Interface Format* — формат цифрового интерфейса Sony/Philips), который можно подключить к студий-

ной или Hi-End-аппаратуре (например, высококачественному внешнему цифро-аналоговому преобразователю (ЦАП), имеющему вход SPDIF или AES/EBU), что позволяет воспроизводить звук с диска на самом высшем уровне. Существуют и специальные звуковые карты для PC, имеющие SPDIF-вход и отличного качества ЦАП, что позволяет получить на PC от CD-ROM звук профессионального качества.

Практически все современные приводы подключаются через IDE-интерфейс, которым пользуются и жесткие диски. Несмотря на то, что IDE CD-привод использует интерфейс IDE, он не является HDD-совместимым устройством и использует собственный протокол обмена, обычно отвечающий стандарту ATAPI (*ATA Packet Interface*). Стандарт ATAPI — это новый, очень мощный протокол обмена данными и командами между устройствами и их драйверами через IDE-интерфейс.

Для того чтобы подключить CD-привод, необходимо:

- Установить его в стандартное место для 5,25-дюймовых устройств.
- Подключить питание (разъем 1, рис. 3). Свободный разъем находится без проблем, причем подсоединить его неправильно попросту невозможно.
- Подключить к CD-приводу такой же IDE-кабель, что и к жесткому диску (разъем 2, рис. 3). Здесь возможны варианты.

1. Установить переключатель-джампер (разъемы 3, рис. 3) на CD-ROM в положение Slave и подключить его ко второму разъему кабеля, идущего к винчестеру. (Красный край кабеля должен оказаться на контакте № 1!)

2. Установить переключатель-джампер на CD-ROM в положение Master и подключить его вторым кабелем ко второму разъему для IDE-устройств на материнской карте.

Второй вариант более предпочтителен, поскольку для передачи данных используется другой канал PCI-шины.

Теперь остался один шлейф — от CD-ROM к Sound Blaster (звуковой карте) (разъем 4, рис. 3). Зачем он нужен? Без Sound Blaster для прослушивания музыкальных записей необходимо подключить колонки или наушники к разьему на передней панели дисковода. Со звуковым шлейфом можно подключить их к выходящему разьему звуковой карты и наслаждаться одновременно и компьютерными звуками, и музыкой с компакт-диска.

Практически все современные CD-приводы автоматически определяют операционными системами, хотя производители и прилагают на дискете драйверы. Они нужны при работе в уже подзабытой MS-DOS. Мало того, большинство

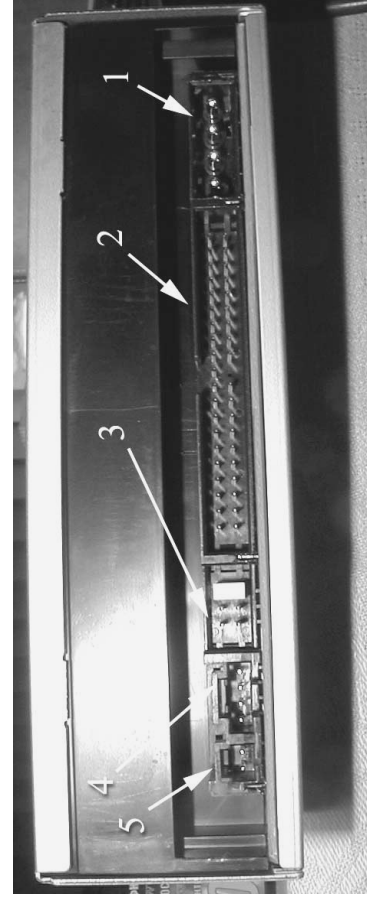


Рис. 3. Задняя панель привода CD-ROM

CD-приводов определяются самой материнской платой, что позволяет использовать эти устройства для загрузки операционной системы непосредственно с CD-диска.

В отличие от винчестеров CD-приводы обладают значительно большей способностью к изнашиванию, и порой не помогают никакие специальные чистяще-полирующие диски: чтение информации с компакт-дисков у старых устройств может оказаться из рук вон плохим. Иногда удается их реанимировать, аккуратно (!) протерев линзу (2) жидкостью для мытья стекол. Диск необходимо аккуратно вставлять в дисконвод, не допуская его застревания при закрытии. Нежелательно вынимать диск, если с ним работает какая-либо программа. Это, с одной стороны, может привести к зависанию системы, а с другой — некоторые модели CD-ROM в этом случае не тормозят раскрученный диск, и он при открывании дверки страшно вращается, царапая свою поверхность, а способен и просто вылететь. Кстати, такой CD-привод лучше сразу же поменять на другой под предлогом, что это изделие вам не подходит. Можно даже не объяснять причины. (В течение месяца вы это можете сделать по Закону о защите прав потребителя.)

Литература

1. Гук М. Дисконвая подсистема ПК. СПб.: Питер, 2001, 336 с.
2. Евгений Музыченко (www.ixbt.ru).

Программно-методические комплексы для преподавания информатики в 1–6-х классах

В.А. Козлова,

г. Пермь

Этот материал подготовлен на основе доклада, сделанного автором на конференции “Рождественские чтения”, которая состоялась в дни зимних каникул в Перми. В приведенных таблицах обобщены фактическая информация и авторские оценки различных программных продуктов, предназначенных для использования в начальной и младшей школе. В частности, помимо информации о том, что можно использовать и где взять тот или иной программный продукт (соответствующая информация приведена в первой таблице), учителю будет интересно познакомиться с авторской точкой зрения на то, как можно (и можно ли вообще!) использовать рассматриваемые программы. Учитывая, что точка зрения автора основана на многолетнем практическом опыте, такая информация особенно ценна.

Наименование программного продукта	Направленность	Источники информации о программном продукте
Развивающие программы фирмы “Никита” (комплексы “Малыш”, “Вундеркинд” и др.)	Подготовка к школе, 1—2-е классы. Развитие основных мыслительных операций, основы счета, азбука, словарный запас в русском и английском языках	http://www.nikita.ru ПМК на CD и дискетах в продаже
Программы ассоциации “Компьютер и Детство” (“Климат”, “Карта”, “Дед Мазай и зайцы”, “Незнайка грамота” и др.)	Подготовка к школе, 1—2-е классы. Развитие творческой активности и основных мыслительных операций, основы счета, азбука, природоведение	http://javaworld.osp.ru/school/1998/03/19.htm Программы на CD в продаже
ПМК “Путешествие в страну Букварию”	Подготовка к школе, 1-й класс. Азбука, чтение	http://admin.smolensk.ru/web_dis/KotovAA/metodica/teach3.htm
Мультимедийные развивающие игры фирмы NMG: “Башня знаний”, “Веселая Азбука” и др.	1—2-е классы. Развитие основных мыслительных операций, основы счета, азбука, природоведение	http://www.nmg.ru Программы на CD в продаже
Комплекс программ “Радауга в компьютере”	1—4-е классы. Поддержка предметов начальной школы	Варченко В.И. ПМК “Радауга в компьютере”. ИНФО № 3/2001.
Безмашинный методический комплекс А.В. Горячева и др. “Информатика в играх и задачах” (Куликова Г.Н. Компьютерная поддержка курса для 1-х классов)	1—4-е классы. Развитие логического мышления, основы алгоритмизации и системологии	http://www.iro.yar.ru:8101/resource/distant/earlyschool_education/gr/okurs.htm Комплект учебников-тетрадей в продаже и в Интернет-магазинах (CD в продаже)
ПервоЛого (Лого в картинках)	1—5-е классы. Развитие логического мышления, пропедевтика понятий угла и фигуры, компьютерные технологии, основы алгоритмизации. Реализация мультимедиа-проектов	http://school.edu.ru/int/logo/products.html С сайта можно получить демо-версию
ПМК “Роботландия”	2—5-е классы. Формирование представления об информации и информационных процессах, развитие логического мышления, компьютерные технологии, основы алгоритмизации и программирования	ООО “Роботландия” (http://www.botik.ru/-robot). Руководитель — Первин Юрий Абрамович (pervin@robot.botik.ru)

Гипертекстовый учебник А.А. Дуванова "Азы информатики"	3—6-е классы. Формирование представления об информации и информационных процессах, развитие логического мышления, основы алгоритмизации, компьютерные и сетевые технологии	Роботландский сетевой университет (http://www.botik.ru/~robot/ru). Руководитель — Дуванов Александр Александрович (kurs@robotland.botik.ru). С сайта можно получить демо-версию этого и других гипертекстовых учебников А.А. Дуванова
ЛогоРайтер	5—11-е классы. Основы алгоритмизации и программирования, машинной графики и моделирования. Реализация проектов по предметным дисциплинам	http://school.edu.ru/int/logo/products.html
ЛогоМиры	5—11-е классы. Компьютерные технологии, основы алгоритмизации и программирования, машинной графики и моделирования. Реализация мультимедиа-проектов	http://school.edu.ru/int/logo/products.html С сайта можно получить демо-версию

Характеристики программно-методических комплексов для преподавания информатики в 1–6-х классах

Наименование программного продукта	Интерфейс	Оценка эффективности обучения	Планирование урока	Связь с курсом информатики для 7—11-х классов
Развивающие программы фирмы "Никита" (комплексы "Мальш", "Вундеркинд" и др.)	DOS, выбор из меню, работа с мышью	Отсутствует	Жесткое (обычно 1 пункт меню на урок)	Нет
Программы ассоциации "Компьютер и Детство" ("Климаг", "Топография", "Деа Мазай и зайцы", "Незнайка грамота" и др.)	DOS, выбор из меню, работа с клавиатурой и мышью	Отсутствует	Жесткое (обычно 1 игра на урок)	Нет
ПМК "Путешествие в страну Букварию"	DOS, выбор из меню, работа с клавиатурой и мышью	Отсутствует	Жесткое (обычно 1—2 пункта меню на урок)	Нет
Мультимедийные развивающие игры фирмы NMG: "Башня знаний" -1, 2, "Веселая Азбука" и др.	Windows, выбор из меню, работа с клавиатурой и мышью	Опосредованно, через количество набранных очков в игре	Очень плохо регламентируется время работы с компьютером. Больше подходит для дополнительных занятий	Пропедевтика некоторых методов информатики и компьютерных технологий
Комплекс программ "Радуга в компьютере"	DOS, выбор из меню, работа с клавиатурой и мышью	Отсутствует	Вариативное, предусмотрена возможность создания плана урока учителем и настройки параметров	Нет
Куликова Г.Н. Компьютерная поддержка курса А.В. Горячева и др. "Информатика в играх и задачах" для 1-го класса	Windows, оконный интерфейс, работа с мышью. <i>Интерфейс программы не соответствует требованиям для детей программным продуктам для детей</i>	Отсутствует	Обусловлено тематикой уроков в рабочих тетрадях для учащихся	Пропедевтика основных понятий и методов информатики и компьютерных технологий

Наименование программного продукта	Интерфейс	Оценка эффективности обучения	Планирование урока	Связь с курсом информатики для 7—11-х классов
ПервоЛото (Лото в картинках)	Windows, оконный интерфейс, работа с мышью и клавиатурой, графическим и музыкальным редакторами, создание гипертекстовых информационных ресурсов	Отсутствует	Плохо регламентируется время работы с компьютером. Больше подходит для дополнительных занятий	Продемонстрация некоторых понятий и методов визуального и объектного программирования
ПМК "Роботландия"	DOS, но с оригинальным оконным кнопочным интерфейсом, выбор из меню, работа с клавиатурой и мышью	Некоторые исполнители пре-дусматривают оценку в виде системы званий ("Профессор", "Студент", "Торговец", "Незнайка")	Соблюдение санитарных норм работы за компьютером предусмотрено и обеспечено тщательной методической подготовкой занятий в ПМК	Продемонстрация основных понятий и методов информатики, компьютерных технологий и процедурного программирования
Гипертекстовый учебник А.А. Дуванова "Азы информатики"	Windows, графический оконный интерфейс, обработка текста, работа с мышью и клавиатурой, программой-браузером	Каждый урок завершается автоматизированной проверкой усвоения пройденного материала с оценкой в "роботландской" системе званий	Вариативное. Для реального урока учитель может выбрать подмножество разделов таким образом, чтобы обеспечить компьютерную поддержку теоретического материала, не нарушая санитарных норм	Продемонстрация основных понятий и методов информатики, компьютерных и сетевых технологий
ЛогоРайтер (Logo Writer)	DOS, среда со своеобразным нестандартным текстовым интерфейсом. Обработка текста, простейший графический редактор и пиктограмм	Отсутствует	Соблюдение санитарных норм работы за компьютером возможно только при условии предварительного безмашинного разбора задач	Освоение основных понятий и методов процедурного программирования
ЛотоМиры	Windows, графический оконный интерфейс, обработка текста, работа с мышью и клавиатурой, графическим и музыкальным редакторами, создание гипертекстовых информационных ресурсов	Отсутствует	Соблюдение санитарных норм работы за компьютером возможно только при условии предварительного безмашинного разбора задач. Плохо регламентируется время работы с компьютером. Больше подходит для дополнительных занятий	Знакомство с компьютерными технологиями. Освоение основных понятий и методов процедурного и объектного программирования

Универсальная модель движения

Н.Н. Паньгина,

г. Сосновый Бор, Ленинградская обл.

Использование компьютерных технологий позволяет значительно повысить эффективность преподавания школьных дисциплин. Межпредметные связи, основанные на моделировании процессов и явлений из самых разных областей науки и техники, позволяют многократно усилить мотивацию в изучении как компьютерных технологий, так и материалов других предметов.

Вообще тема “Моделирование” является одной из наиболее интересных в программе курса “Основы информатики и вычислительной техники”. Хорошо воспринимается учащимися задача А.Г. Гейна “О выборе места строительства железнодорожной станции” с использованием принципов экономической целесообразности, социальной справедливости и принципа компромисса [1]. Но, пожалуй, самым понятным и наглядным для учащихся является моделирование тех видов движения, которые они изучают на уроках физики и которые знакомы им из повседневной жизни.

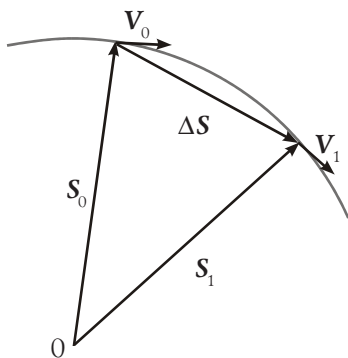
Модели движения относятся к динамическим моделям, т.е. моделям, в которых состояние объекта зависит от времени. В качестве объекта моделирования мы выберем материальную точку, состояние которой описывается, в частности, координатами в пространстве. В качестве же конечного результата моделирования нас будет интересовать траектория, описываемая этой материальной точкой. Чтобы построить динамическую модель, нужно задать систему координат и описать в ней траекторию движения тела.

На уроках физики в средней школе даются аналитические модели движения (дифференциальные модели движения изучаются в высшей школе), а вот для компьютерной реализации на уроках информатики наиболее удобными являются разностные модели.

Теоретические основы универсальной модели движения

При моделировании на компьютере мы будем использовать разностные модели движения, то есть рассматривать положение движущейся материальной точки через некоторые заданные промежутки времени. При этом непрерывное движение заменяется последовательностью отдельных равномерных прямолинейных шагов [2].

Основная идея построения универсальной модели движения основывается на известном соотношении, связывающем вектор перемещения ΔS материальной точки с вектором ее средней скорости V_{cp} .



$$\Delta S = V_{cp} \cdot \Delta t, \quad (1)$$

где $V_{cp} = (V_0 + V_1)/2$

Δt — промежуток времени между двумя последовательными рассматриваемыми состояниями материальной точки, в которых известны векторы скорости $V_0(t)$ и $V_1(t + \Delta t)$.

Основная формула (1) получается из известных формул равноускоренного движения (полагается, что a — вектор ускорения изменяется незначительно за время Δt):

$$\begin{aligned} V_1 &= V_0 + a \cdot \Delta t; \\ \Delta S &= V_0 \cdot \Delta t + a \cdot \Delta t^2/2. \end{aligned} \quad (2)$$

Ниже представлена детальная разработка приложения в среде Visual Basic для моделирования и демонст-

рации различных видов движения: и таких простых, как равномерное прямолинейное, и более сложных, как движение тела, брошенного под углом к горизонту, и таких достаточно сложных, как движение бумажного самолетика или движение спутника Земли. “Универсальность” модели состоит в том, что будет использоваться одна и та же формула (1) для различных видов движения. Усложнение модели требуется лишь при учете дополнительных факторов в определении вектора скорости в формуле (2).

Как обычно, начинать надо с простого, чтобы заинтересовать учащихся в моделировании самых простых видов движения, тех, с которыми не возникает особых сложностей на уроках физики. Затем можно перейти к сложным видам движения, давая возможность учащимся самим выполнить моделирование на компьютере.

Общий алгоритм для моделирования отдельного вида движения можно записать в следующем виде:

Начало

Ввод начальных данных (для момента времени $t = t_0$);

Повторять

Вычислить новые значения скорости и координат в момент времени $t + \Delta t$;

Нарисовать отрезок траектории;

Переписать значения скорости и координат для нового момента времени t ;

Конец повтора;

Конец

Равномерное движение

Равномерное движение характеризуется тем, что в формуле (2) вектор ускорения нулевой и соответствующее слагаемое отсутствует.

МОДЕЛЬ 1. РАВНОМЕРНОЕ ПРЯМОЛИНЕЙНОЕ ДВИЖЕНИЕ В ОДНОМ ИЗМЕРЕНИИ

Рассмотрим движение в плоскости вдоль оси абсцисс.



Скорость постоянна $V_1 = V_0 = \text{const}$;
 $S_1 = S_0 + \Delta S$;

Следовательно, $S_1 = S_0 + (V_0 + V_1)/2 \cdot \Delta t$ или
 $X_1 = X_0 + V_0 \cdot \Delta t$,

то есть траектория в данном случае — это прямая линия.

МОДЕЛЬ 2. РАВНОМЕРНОЕ ПРЯМОЛИНЕЙНОЕ ДВИЖЕНИЕ С ОТРАЖЕНИЕМ

Таким же образом можно промоделировать движение шара по прямой до стенки и обратно. В алгоритм после действия переприсваивания координаты X вводится проверка достижения шаром стенки и замена вектора скорости на противоположный по направлению (абсолютно упругий удар):

Если (стенка достигнута) **то** $V_0 = -V_1$;

МОДЕЛЬ 3. РАВНОМЕРНОЕ ПРЯМОЛИНЕЙНОЕ ДВИЖЕНИЕ В ДВУХ ИЗМЕРЕНИЯХ

Движение в плоскости рассматривается как составное движение на прямой соответственно по оси X (абсцисс) и оси Y (ординат).

Скорость постоянна $V_1 = V_0 = V = \text{const}$, но задается направление скорости — угол α (угол между вектором скорости и осью абсцисс). Вектор скорости раскладывается на две составляющие — V_{0x} и V_{0y} :

$$V_{0x} = V_0 \cdot \cos\alpha;$$

$$V_{0y} = V_0 \cdot \sin\alpha.$$

Следовательно, по формуле (1) $S_1 = S_0 + V_{cp} \cdot \Delta t$ или в координатах

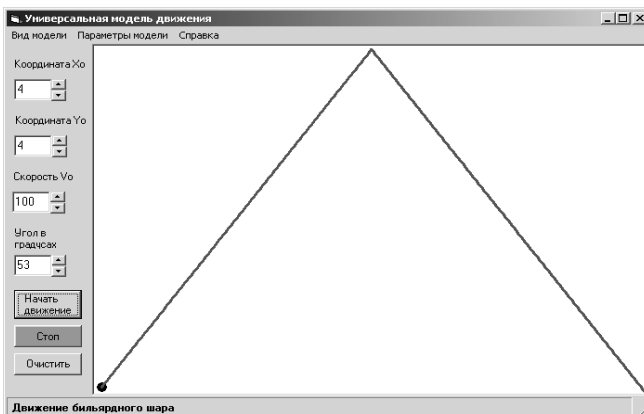
$$X_1 = X_0 + V_{0x} \cdot \Delta t,$$

$$Y_1 = Y_0 + V_{0y} \cdot \Delta t,$$

т.е. траектория в данном случае — это прямая линия, направленная под углом α к оси абсцисс.

МОДЕЛЬ 4. ДВИЖЕНИЕ БИЛЬЯРНОГО ШАРА

В качестве самостоятельной работы учащимся предлагается смоделировать движение бильярдного шара по столу с отражением от стенок. В готовом проекте предлагается из заданного положения шара попасть в лузу с однократным отражением от борта.



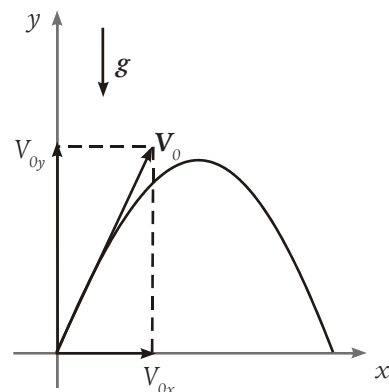
Равноускоренное движение

МОДЕЛЬ 5. РАВНОУСКОРЕННОЕ ДВИЖЕНИЕ ТЕЛ В ПОЛЕ СИЛЫ ТЯЖЕСТИ

Какова же будет траектория движения тела, брошенного под углом к горизонту, при отсутствии сопротивления воздуха, но с учетом силы тяжести? В формуле (2) вектор ускорения a есть вектор ускорения свободного падения g , т.е. получаем векторную формулу

$$V_1 = V_0 + g \cdot \Delta t.$$

При таком движении (см. рисунок) составляющая вектора скорости V_x по оси X не меняется, а составляющая вектора скорости V_y по оси Y будет уменьшаться, так как вектор ускорения направлен противоположно.



$$V_{1x} = V_{0x}$$

$$V_{1y} = V_{0y} - g \cdot \Delta t$$

Из формулы (1) следует, что точка траектории во время $t + \Delta t$ имеет следующие координаты:

$$\begin{cases} X_1 = X_0 + (V_{0x} + V_{1x})/2 \cdot \Delta t \\ Y_1 = Y_0 + (V_{0y} + V_{1y})/2 \cdot \Delta t \end{cases}$$

С помощью основанных на этих формулах программ можно решать реальные физические задачи, такие, как:

- вычисление дальности полета,
- вычисление высоты полета,
- определение времени подъема тела до самой высокой точки траектории,
- определение общего времени полета тела.

Можно усложнить задачи, моделируя полет резинового мяча с абсолютно упругим ударом о горизонтальную поверхность и отражением от нее. Наиболее способные и заинтересованные учащиеся могут разработать простейшие игры на этой основе (попасть в цель из орудия или мячом в баскетбольную корзину).

МОДЕЛЬ 6. ДВИЖЕНИЕ ТЕЛ В ПОЛЕ СИЛЫ ТЯЖЕСТИ ПРИ СОПРОТИВЛЕНИИ ВОЗДУХА

При относительно малых скоростях движения сила сопротивления воздуха пропорциональна скорости движения тела и направлена в противоположную сторону

$$F_c = -c \cdot V,$$

где c — коэффициент, который пропорционален площади поперечного сечения тела и плотности воздуха.

Учащиеся могут сами рассказать об этом, опираясь на следующие примеры из жизни:

- при езде на велосипеде сила сопротивления воздуха тем больше, чем больше скорость движения;
- при бросании различных предметов сила сопротивления воздуха зависит от формы данных предметов.

Следовательно, ускорение a можно выразить следующим образом:

$$a = a_c = \alpha_c \cdot V,$$

где $\alpha_c = c/m$ (m — масса тела).

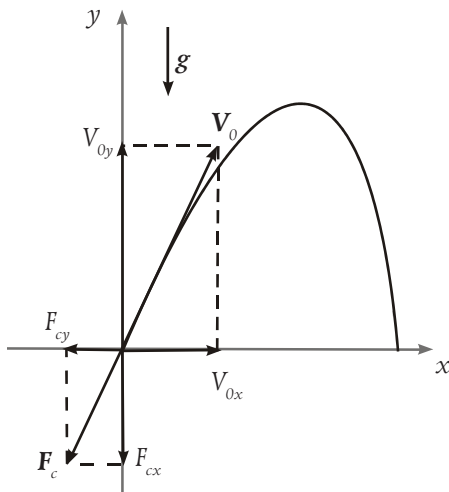
По формуле (2)

$$V_1 = V_0 + (g + a_c) \cdot \Delta t \text{ (векторная форма).}$$

Следовательно, в проекциях:

$$V_{1x} = V_{0x} - \alpha_c \cdot V_{0x} \cdot \Delta t,$$

$$V_{1y} = V_{0y} - g \cdot \Delta t - \alpha_c \cdot V_{0y} \cdot \Delta t.$$



В отличие от предыдущей задачи траектория материальной точки представляет собой несимметричную кривую относительно вертикали, проведенной через высшую точку полета, причем нисходящая ветвь круче восходящей. Очень к месту вспомнить, что в спортивном бадминтоне этим пользуются очень широко. Высокую подачу, например, подают настолько высоко, насколько позволяют высота потолка и силы игрока. После такого удара волан приземляется на заднюю линию поля противника почти вертикально, что делает невозможным сильный ответный удар, так как этому мешает оперение.

Моделируя данный вид движения, учащиеся также решают реальные физические задачи определения высоты, дальности и времени полета. Точно так же можно и усложнить ситуацию, рассмотрев отскакивающий от поверхности резиновый мяч. Здесь будут интересны заходящие отскоки.

Модель 7. ПОЛЕТ БУМАЖНОГО САМОЛЕТИКА

Эксперименты с самолетиком, запускаемым под разными углами и с разными начальными скоростями, показывают, что его траектория гораздо сложнее, чем тра-

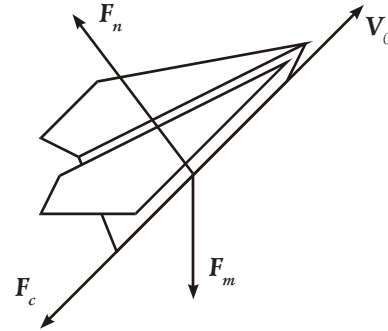
ектория камня или волана. Угадать ее практически невозможно, однако это вовсе не означает, что поведение кувыркающегося самолетика не поддается никакому анализу. Эту задачу поставил и решил основоположник аэродинамики Н.Е. Жуковский (1847—1921), назвавший ее “Задачей о планере”.

На бумажный самолетик действуют три силы:

F_m — сила тяжести,

F_c — сила сопротивления воздуха,

F_n — подъемная сила, перпендикулярная вектору скорости V_0 .



По теореме Жуковского величина подъемной силы пропорциональна квадрату скорости

$$F_n = \beta \cdot V^2.$$

Рассуждения, аналогичные предыдущему случаю, приводят к следующим формулам:

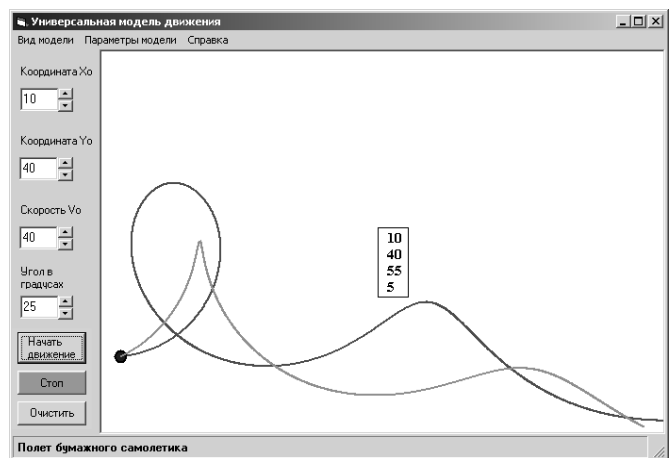
$$V_{1x} = V_{0x} - \alpha_c \cdot V_{0x} \cdot \Delta t - \alpha_p \cdot V \cdot V_{0y} \cdot \Delta t;$$

$$V_{1y} = V_{0y} - g \cdot \Delta t - \alpha_c \cdot V_{0y} \cdot \Delta t + \alpha_p \cdot V \cdot V_{0x} \cdot \Delta t;$$

$$V = \sqrt{V_{0x}^2 + V_{0y}^2}.$$

Здесь α_p — коэффициент, учитывающий подъемную силу.

На рисунке показаны траектории полета бумажного самолетика при различных начальных параметрах.



Очень оживляет урок натурное моделирование, когда наряду с компьютерными экспериментами проводятся настоящие физические опыты с бумажными самолетиками. Варьируя такими исходными данными, как начальная скорость и угол наклона, можно добиться различных результатов и получить разнообразные траектории движения.

МОДЕЛЬ 8. ДВИЖЕНИЕ СПУТНИКА ЗЕМЛИ

Последняя модель уже не только физическая, но и астрономическая. Она показывает, что расчету поддаются не только траектории шарика, волана или самолета, но и таких крупных объектов, как планеты и даже звезды. Мы смоделируем движение спутника Земли [3].

На спутник действует единственная сила — сила притяжения Земли. Ее можно вычислить по формуле закона всемирного тяготения

$$F = G \frac{M \cdot m}{R^2},$$

где G — гравитационная постоянная.

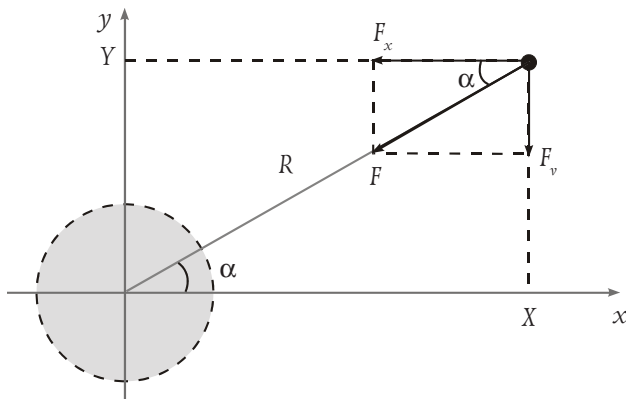
Она равна $6,67 \cdot 10^{-11} \text{ м}^3 / (\text{кг} \cdot \text{с}^2)$;

M — масса Земли, равная $5,98 \cdot 10^{24} \text{ кг}$;

m — масса спутника.

Если известны координаты X , Y спутника, то расстояние от центра Земли до спутника:

$$R = \sqrt{X^2 + Y^2}.$$



Из рисунка видно, что

$$F_x = F \cdot \cos\alpha,$$

$$F_y = F \cdot \sin\alpha.$$

Если на спутник действует только притяжение Земли, то второй закон Ньютона можно записать в виде:

$$m \cdot a_x = -G \frac{M \cdot m}{R^2} \cdot \frac{x}{R},$$

$$m \cdot a_y = -G \frac{M \cdot m}{R^2} \cdot \frac{y}{R}.$$

Сократив на m и подставив значение ускорения в известные нам формулы (1) и (2), получим формулы для вычисления новых значений, составляющих скорости через время Δt :

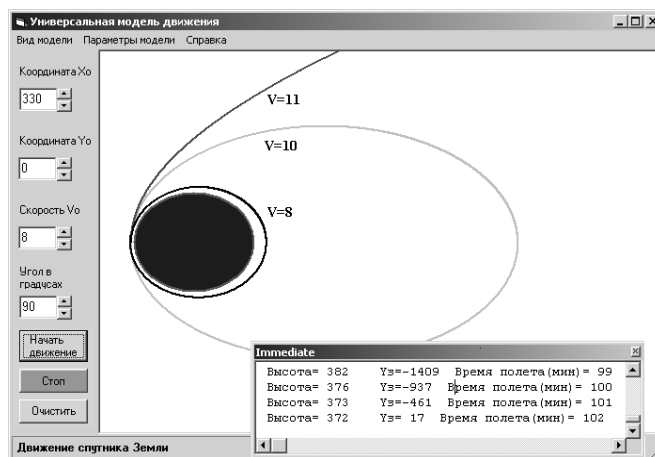
$$V_{1x} = V_{0x} - G \cdot M \cdot X_0 \cdot \Delta t / R^3,$$

$$V_{1y} = V_{0y} - G \cdot M \cdot Y_0 \cdot \Delta t / R^3.$$

При разностном моделировании весьма важен правильный выбор шага по времени. Не вдаваясь в специальные методы его расчета, просто будем уменьшать шаг в два раза и, если решение для нового Δt отличается от старого не более чем на заданную величину, будем считать, что процедура выбора шага закончена.

На следующем рисунке приведен результат работы программы. В задаче заданы реальные физические дан-

ные для полета космического корабля с космонавтом Юрием Гагариным, который облетел Землю за 108 минут. В задаче получено хорошее приближение — 102 минуты. На рисунке приведены также орбиты спутников при различных начальных скоростях.



Разумеется, программа допускает разнообразные улучшения и дополнения. Первая их группа касается внешнего оформления. Можно через определенные промежутки времени выводить на экран текущие значения скорости и расстояния до Земли. Можно найти время, за которое спутник совершит полный оборот. Или если, например, спутник упал (расстояние до спутника стало меньше радиуса Земли), то можно в точке падения изобразить взрыв, да и сам спутник изобразить более реалистично.

Помимо художественных улучшений, можно изменить и саму модель. Так, если в формулах для V_x и V_y заменить минусы на плюсы, то закон всемирного тяготения превратится в закон всемирного отталкивания. Тем самым модель спутника превратится в модель взаимодействия одноименно заряженных частиц. А как будут выглядеть траектории во Вселенной с другим законом всемирного тяготения? У нас сила тяготения обратно пропорциональна квадрату расстояния. А что получится, если она будет пропорциональна кубу или первой степени расстояния? Несложно смоделировать и эту ситуацию — достаточно в формулах заменить R^3 на R^4 или на R^2 .

Построение универсальной модели движения

Создадим программу в среде Visual Basic 6.0 для демонстрации универсальной модели движения (предполагается знакомство с понятиями и конструкциями этого языка).

1. Визуальное программирование

1.1. Поместим в *Toolbox* — стандартный блок инструментов Visual Basic — дополнительные элементы управления и контроля. Для этого выберем опцию *Components* в меню *Project*. В появившемся списке вкладки *Controls* поставим галочку против элементов *Microsoft Windows Common Controls 6.0* (содержит, в частности, объект *StatusBar* — Строка состояния),

Microsoft Windows Common Controls-2 6.0 (содержит объект UpDown — Счетчик), Microsoft Common Dialog Control 6.0 (для отображения в программе стандартного окна выбора цвета — Color).

1.2. Разместим слева на форме (Form) массив из четырех Текстовых окон (TextBox) для задания начальных параметров модели: координаты X_0 , координаты Y_0 , скорости V_0 , начального угла в градусах, для обозначения каждого Текстового окна используется соответствующий массив Меток (Label).

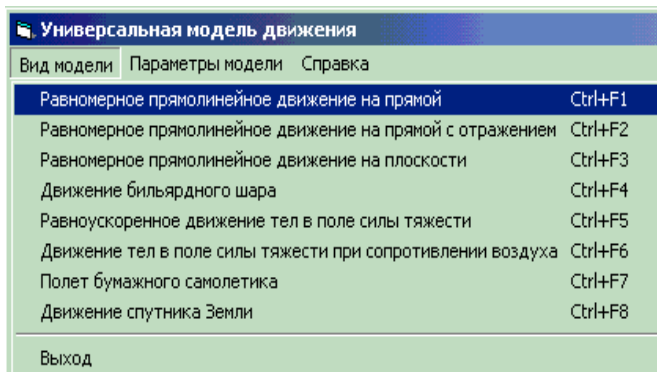
1.3. Каждому текстовому полю присоединим свой элемент Счетчик (UpDown). Соответствие указывается в свойстве BuddyControl или выбирается автоматически при AutoBuddy = True. При установленном значении BuddyProperty = Default элемент UpDown будет автоматически обновлять стандартное свойство Text связанного с ним элемента управления TextBox. Определите минимальное (Min), максимальное (Max) значения и величину изменения (Increment) числовых значений, определяемых каждым объектом UpDown.

1.4. Разместим по нижнему краю формы объект Строка состояния (StatusBar) для отображения необходимой информации в процессе работы программы (о выбранной демонстрационной модели). Щелкнув правой кнопкой мыши на объекте StatusBar, выберем вкладку Панели (Panels) диалогового окна Properties Pages и установим свойство AutoSize = sbrSpring для первой панели, в результате чего она заполнит все пространство строки состояния. Выберем жирный шрифт (Bold) на вкладке Font.

1.5. Поместим на форму объект Диалоговое окно (CommonDialog) и присвоим ему имя cdlColor.

1.6. Нанесем на форму три Командные кнопки, изменив их свойство Caption на “Начать движение”, “Стоп” и “Очистить” соответственно.

1.7. Создадим меню из трех главных пунктов “Вид модели”, “Параметры модели” и “Справка”. В первом пункте создадим массив из восьми подпунктов, обозначающих вид конкретной модели, присвоив каждому элементу массива еще и комбинацию горячих клавиш.



Там же, после разделительной черты, поместим и подпункт “Выход”. В пункте меню “Параметры модели” создадим два подпункта — “Движение с траекторией” и “Цвет траектории”. В подпункте “Движение с траекторией” используем свойство Checked (при активизации данного подпункта рядом с названием бу-

дет ставиться галочка). В пункте “Справка” можно поместить краткую информацию о данной программе и авторе.

1.8. Нанесем на форму объект Таймер, присвоив свойству Interval значение 5 (это означает, что каждые 5 миллисекунд будет срабатывать событие Timer).

1.9. И, наконец, нанесем на форму основной элемент управления Окно рисунка (Picture Box), присвоив свойству Align значение 4 — Align Right (выравнивание по правому краю формы), а свойству AutoRedraw — значение True. Для свойства FillColor выберем синий цвет.

2. Написание кода программы

2.1. В данном приложении будут использоваться следующие константы: ускорение свободного падения (g), число Пи (π), гравитационная постоянная (G_r), масса Земли (M), радиус Земли (R_3), максимальные значения координат по оси абсцисс (X_{max}) и по оси ординат (Y_{max}) для масштабирования Окна рисунка. Кроме того, понадобятся следующие глобальные переменные: координаты начального положения модели (x_0 , y_0); начальная скорость и ее составляющие по осям координат (v_0 , v_{0x} , v_{0y}); начальный угол (α); переменная, обозначающая вид модели (vid); коэффициент сопротивления воздуха (ac); коэффициент подъемной силы (ap); расстояние от центра Земли до спутника (R_s); шаг по времени (dt); время обращения вокруг Земли (T); радиус модели (r); величина GrM , равная произведению G_r на M ; номер цвета для траектории (col). Описываем их в общем разделе описаний General Declarations.

```
Const pi = 3.1415926, g = 9.8 '[м/сек^2]
Const Gr = 6.67E-20 '[км3/(кг*сек^2)]
Const M = 6E+24 '[кг]
Const R3 = 6370 '[км]
Const Xmax = 300, Ymax = 200
Dim x0, y0, v0, v0x, v0y, alf, vid, ac, ap,
Rs, dt, T, r, GrM, col
```

2.2. Первое событие после загрузки Формы — выбор модели движения в пункте меню “Вид модели”. В процедуре обработки данного события в статусную строку помещаем название выбранной модели, в переменной vid запоминаем номер этой модели, равный индексу соответствующего подпункта меню, и вызываем общую процедуру ввода данных Input_Data.

```
Private Sub mnu_TypeOfMotion_Click(Index As Integer)
StatusBar1.Panels(1).Text =
mnu_TypeOfMotion(Index).Caption
vid = Index
CmdStop.Enabled = False
Input_Data
End Sub
```

2.3. Описание процедуры Input_Data:

- с помощью свойства Scale масштабируем Окно рисунка;
- присваиваем начальные значения следующим переменным: шагу по времени, координатам центра и радиусу модели, начальной скорости и углу;

- в зависимости от вида модели, в операторе выбора либо меняем некоторые параметры, либо вводим через Окно ввода (InputBox) новые значения таких параметров, как коэффициент сопротивления воздуха или коэффициент подъемной силы;

- при выборе модели 8 (спутник Земли) задаются специфические параметры и новое масштабирование;

- в Текстовых окнах выводятся соответствующие параметры модели, а в Окне рисунка появляется модель в виде небольшого шарика.

Sub Input_Data()

```
Picture1.Scale (0, Ymax) - (Xmax, 0)
dt = 0.01
r = 2
v0 = 50: alf = 0
x0 = 5: y0 = 10
Select Case vid
  Case 3 To 7
    alf = 25
    ac = 0.1: ap = 0.02
    If vid >= 6 Then
      ac = InputBox("Коэф. сопротивления воздуха",
        "Окно ввода", ac)
    If vid = 7 Then
      ap = InputBox("Коэф. подъемной силы",
        "Окно ввода", ap)
  Case 8
    GrM = Gr * M
    dt = 2
    v0 = 8: alf = 90
    x0 = 330 '[км] - высота спутника
    y0 = 0
    Picture1.Scale (-10000, 25000) - (50000, -25000)
End Select
Text1(1).Text = x0
Text1(2).Text = y0
Text1(3).Text = v0
Text1(4).Text = alf
End Sub
```

2.4. В процедуре обработки события Щелчок на пункт меню “Движение с траекторией” необходимо выполнить следующие действия:

- сделать отмеченным галочкой (либо снять пометку) данный пункт меню,

- при выборе движения с траекторией задать режим рисования обычный (поверх), а без траектории — режим с наложением цветов.

Private Sub mnu_Track_Click()

```
mnu_Track.Checked = Not mnu_Track.Checked
If mnu_Track.Checked Then
  Picture1.DrawMode = vbCopyPen
Else
  Picture1.DrawMode = vbNotXorPen
End If
End Sub
```

2.5. В процедуре обработки события Щелчок на пункт меню “Цвет траектории” необходимо отобразить диалоговое окно выбора цвета. В его свойстве Color будет размещаться числовой код выбранного цвета, который необходимо запомнить в переменной col.

Private Sub mnu_Color_Click()

```
cdlColor.ShowColor
col = cdlColor.Color
End Sub
```

2.6. Следующее событие — щелчок на Командную кнопку “Начать движение”. В процедуре обработки данного события из текстовых полей считываются начальные параметры движения, которые могут быть предварительно изменены “исследователем”. Для модели со спутником (vid = 8) рисуется Земля (голубая планета) и выбирается система отсчета относительно центра Земли. Дается зеленый цвет “светофора” кнопке “Стоп”, а свойству Enabled объекта Timer присваивается значение True, чтобы “включить” демонстрацию движения для выбранной модели.

Private Sub CmdStart_Click()

```
x0 = Text1(1).Text
y0 = Text1(2).Text
v0 = Text1(3).Text
alf = Text1(4).Text * pi / 180
If vid = 8 Then
  T = 0
  x0 = -(R3 + x0)
  Picture1.Circle (0, 0), R3, QBColor(3), , , 0.8
End If
Picture1.Circle (x0, y0), r
v0x = v0 * Cos(alf)
v0y = v0 * Sin(alf)
CmdStop.Enabled = True
CmdStop.BackColor = &HFF00&
Timer1.Enabled = True
```

End Sub

2.7. По событию от таймера перерисовывается объект модели (круг) в состоянии, соответствующем изменению времени на промежуток dt, а для этого производится расчет изменения компонент скорости. Затем определяются координаты нового положения и выбранным цветом рисуется (согласно меню) либо сам объект, либо его траектория. Учитываются: для моделей 2 и 4 — отражение от границ, для моделей 5, 6, 7 — достижение нижней границы, для модели 8 в окне отладки Immediate печатается дополнительная информация о положении спутника и времени полета (таким образом можно определять, например, апогей и период обращения).

Private Sub Timer1_Timer()

```
Select Case vid
  Case 1 To 4
    v1x = v0x
    v1y = v0y
  Case 5
    v1x = v0x
    v1y = v0y - g * dt
  Case 6
    v1x = v0x - (ac * v0x) * dt
    v1y = v0y - (g + ac * v0y) * dt
```

```

Case 7
  V = Sqr(v0x ^ 2 + v0y ^ 2)
  v1x = v0x - (ac * v0x + ap * V * v0y) * dt
  v1y = v0y - (g + ac * v0y - ap * V * v0x) * dt
Case 8
  Rs = Sqr(x0 ^ 2 + y0 ^ 2)
  v1x = v0x - (GrM * x0 / Rs ^ 3) * dt
  v1y = v0y - (GrM * y0 / Rs ^ 3) * dt
End Select
  x = x0 + (v0x + v1x) * 0.5 * dt
  y = y0 + (v0y + v1y) * 0.5 * dt
If mnu_Track.Checked Then
  Picture1.Line (x0, y0)-(x, y), col
Else
  Picture1.Circle (x0, y0), r
  Picture1.Circle (x, y), r
End If
'
If vid = 4 Or vid = 2 Then
  If x > Xmax - r Or x < r Then v1x = -v1x
  If y > Ymax - r Or y < r Then v1y = -v1y
End If

If vid >= 5 And vid <= 7 Then
  If y < r Then Timer1.Enabled = False
End If

If vid = 8 Then
  T = T + dt
  If CLng(T) Mod 60 = 0 Then
    Debug.Print "Высота="; CLng(Rs - R3), _
      " Yз="; CLng(y); " Время полета (мин)=";
    CLng(T / 60)
  End If
'
  v0x = v1x: v0y = v1y
  x0 = x: y0 = y
End Sub

```

2.8. Следующее событие — щелчок на Командную кнопку “Стоп”. В процедуре обработки данного события Таймер либо “выключается”, то есть приостанавливается движение модели, либо “включается”, то есть продолжается движение модели. Кнопке “Стоп” дается либо зеленый цвет “светофора”, либо красный.

```

Private Sub CmdStop_Click()
Timer1.Enabled = Not Timer1.Enabled
If Timer1.Enabled Then
  CmdStop.BackColor = &HFF00&
Else
  CmdStop.BackColor = &HFF&
End If
End Sub

```

2.9. В процедуре обработки события Щелчок на Командную кнопку “Очистить” очищается Окно рисунка.

```

Private Sub CmdCls_Click()
Picture1.Cls
End

```

2.10. Последнее событие — щелчок на пункт меню “Выход”. В процедуре обработки этого события при подтверждении выхода завершается работа программы.

```

Private Sub mnu_Exit_Click()
If MsgBox("Вы действительно желаете выйти?",
vbYesNo, "Выход") = vbYes Then End
End Sub

```

Последней частью работы является запуск программы на выполнение, ее тестирование и отладка.

Как уже говорилось выше, разработанные модели можно использовать для изучения траекторий при разных начальных условиях. Кроме того, весьма интересным оказывается и сравнительный анализ разных моделей между собой.

Заключение

Компьютерные модели — эффективное средство познавательной деятельности учащихся, которое открывает для учителя широкие возможности по совершенствованию урока. Используя учебные компьютерные модели, учитель может представить изучаемый материал более наглядно, продемонстрировать его новые и неожиданные стороны, что, в свою очередь, повышает интерес учащихся к изучаемому предмету и способствует лучшему пониманию учебного материала. Созданные самими учащимися, модели позволяют им не просто наблюдать на экране компьютера различные виды движений, но и управлять моделируемыми процессами, изменяя соответствующие параметры. Кроме того, компьютерное моделирование, будучи востребованным в конкретной предметной области, позволяет существенно экономить время при подготовке к урокам и на самих уроках.

Предложенная универсальная модель движения является очень простой в реализации, наглядной в обучении и очень полезной для развития межпредметных связей информатики и физики.

Несомненным достоинством данной модели является и то, что ее можно реализовать не только на предложенном языке программирования Visual Basic, но и на простейшем “школьном” языке QBasic, языке Паскаль, объектно-ориентированном языке Delphi и т.п.

Литература

1. Гейн А.Г. и др. “Основы информатики и вычислительной техники, 10—11-е классы”. М: Просвещение, 1991, 1992, 1996.
2. Хилькевич С.С., Зайцева О.А. “Как построить траекторию. Векторные уравнения в кинематике”, журнал “Квант” № 7, 1987.
3. Носков Н.Н., Столбоушкин С.К. “Спутник на дисплее”, журнал “Квант” № 10, 1989.

Простая гостевая книга для школьного сайта

Л.О. Сергеев,
Москва

Готовя материал, посвященный языку Perl, для рубрики “Начните с простого”, мы прежде всего определили для себя, что простота не должна подразумевать полную тривиальность и бесполезность рассматриваемых примеров. Исходя из этих соображений, мы остановились на задаче, обладающей двумя основными свойствами: во-первых, данная задача является типичной для языка Perl, а, во-вторых, ее решение может быть непосредственно использовано на практике, например, при создании школьного web-сайта. Речь идет о создании простой гостевой книги.

Что представляет собой язык Perl?

В настоящее время Perl представляет собой универсальный язык программирования высокого уровня. Особенности этого языка таковы, что имеются классы задач, которые на нем решать удобно (часто много удобнее, чем на других языках программирования). Например, Perl очень эффективен для задач, связанных с обработкой строк. Perl также часто используется при решении задач, связанных с программированием для Интернета (одну такую задачу мы рассмотрим в данной статье). Perl — интерпретируемый язык, причем интерпретаторы Perl имеются практически для всех популярных операционных систем (отметим, что большинство интерпретаторов Perl распространяются свободно). Те, кто работает в операционной системе Windows, обычно используют Active Perl, который можно получить, например, на сайте разработчиков www.activestate.com. Думаем, что те, кто работает под Linux и другими клонами Unix, в наших советах по выбору интерпретатора Perl не нуждаются.

Мы не можем описать типичные проблемы, возникающие при установке Perl под Windows, поскольку нам таковые не известны. Perl устанавливается легко, и конфигурация “по умолчанию”, как правило, не требует изменений. В этой конфигурации Perl устанавливается в каталог C:\Perl, а собственно интерпретатором является файл C:\Perl\bin\perl.exe. При установке путь к интерпретатору Perl добавляется в переменную PATH, а файлы с расширением pl ассоциируются с интерпретатором Perl.

Первая программа на языке Perl

После установки Perl можно немедленно проверить его в работе. Для этого создайте файл, содержащий одну строчку:

```
print "\nHello!\n";
```

Назовите это файл, например, hello.pl. Это и будет первая простая программа.

Для запуска программы передайте ее на вход интерпретатора:

```
perl hello.pl
```

На экран будет выведена строка Hello! (символами “\n” обозначается переход на новую строку).

CGI-программы

Аббревиатура CGI расшифровывается как *Common Gateway Interface*. Суть работы CGI-программ довольно проста. Эти программы исполняются на стороне web-сервера (под управлением web-сервера). Как правило, CGI-программы располагаются в специальном каталоге web-сервера (часто этот каталог называется cgi-bin). Запуск CGI-программ производится сервером при получении пользовательского запроса. Например, если CGI-программа hello.pl расположена в каталоге cgi-bin, на сервере www.domain.ru, она будет запущена при доступе пользователя к следующему URL:

```
http://www.domain.ru/cgi-bin/hello.pl.
```

При организации web-сервера в локальной сети часто используют не имена хостов, а непосредственно IP-адреса. Если, например, web-сервер установлен на компьютере с IP-адресом 192.168.0.1 (это типичный случай), URL может выглядеть так:

```
http://192.168.0.1/cgi-bin/hello.pl.
```

Вывод CGI-программы сервер возвращает пользователю. Приведем пример CGI-программы на языке Perl, которая формирует HTML-страницу и выводит на нее одно слово “Hello!”.

```
#Простая CGI-программа
print "Content-type: text/html\n\n";
print "<html><head><title>Hello!</title>
</head><body>Hello!</body></html>";
```

Посредством первого оператора print выводится заголовок HTML-документа. Этот заголовок сервер всегда отправляет браузеру, для того чтобы тот знал тип данных, которые ему придется получать. Например, GIF-файлу соответствует заголовок "Content-type: image/gif". Заголовок отделяется от документа пустой строкой (это требование стандарта CGI). Дело в том, что заголовок может состоять из нескольких строк (в нашем случае строка одна) и пустая строка является признаком окончания заголовка и начала раздела данных. Именно для вывода пустой строки строка заголовка завершается двумя символами “\n”. Второй оператор print формирует простейший HTML-документ.

Организация гостевой книги

Наша простая гостевая книга будет состоять из двух файлов: `guestbook.pl` и `guestbook.html`. Второй файл будет хранить записи, т.е. фактически он является файлом данных, а первый представляет собой программу на языке Perl для управления гостевой книгой.

Приведем содержимое двух этих файлов и подробно прокомментируем его.

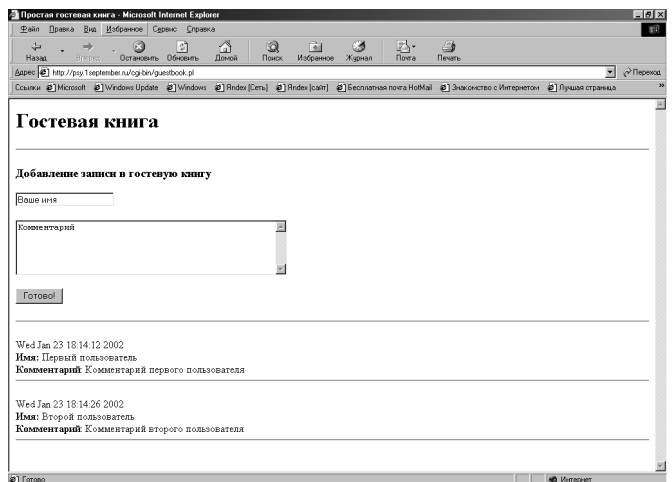
Начальное содержимое файла `guestbook.html` (содержимое этого файла будет меняться, поскольку в него будут добавляться записи).

```
<h3>Добавление записи в гостевую книгу</h3>
<form method=post
  action=/cgi-bin/guestbook.pl>
<p><input type=text name=name
  value='Ваше имя'></p>
<p><textarea name=comment rows=5
  cols=50>Комментарий</textarea></p>
<p><input type=submit value='Готово!'>
```

Файл `guestbook.pl`

```
# Подключим модуль CGI.pm
use CGI qw/:standard/;
# Функция вывода части HTML-файла до раздела
# данных
sub outhead {
  print "<html>";
  print "<head>";
  print "<title>Простая гостевая книга</title>";
  print "</head>";
  print "<body>";
  print "<h1>Гостевая книга</h1><hr>";
}
# Функция вывода части HTML-файла после
# раздела данных
sub outtail {
  print "</body>";
  print "</html>";
}
# Функция замены HTML-тегов
sub antihack {
  $_[0] =~ s/</&lt;/g;
  $_[0] =~ s/>/&gt;/g;
}
# Основная программа
# Если параметры имеются, дописываем их
# в файл
if (param()){
  open GB, ">> guestbook.html";
  $name=param('name'); antihack($name);
  $comment=param('comment'); antihack($comment);
  print GB "\n<p>";
  $lt=localtime; print GB "\n$lt<br>";
  print GB "\n<b>Имя:</b> $name<br>";
  print GB "\n<b>Комментарий</b>: $comment";
  print GB "\n<hr>";
  close GB;
}
# В любом случае выводим содержимое файла
print "Content-type:text/html\n\n";
&outhead;
open GB, "< guestbook.html";
while (<GB>) { print $_; }
close GB;
&outtail;
```

Страница с гостевой книгой в браузере



Сначала прокомментируем содержимое файла `guestbook.html`. Как мы уже говорили, этот файл является файлом данных, в котором хранятся записи гостевой книги. Кроме этого, в начале файла находится форма, посредством которой происходит добавление записей. Так, после добавления в гостевую книгу двух записей файл `guestbook.html` будет содержать следующий текст:

```
<h3>Добавление записи в гостевую книгу</h3>
<form method=post action=/cgi-bin/
guestbook.pl>
<p><input type=text name=name
  value='Ваше имя'></p>
<p><textarea name=comment
  rows=5 cols=50>Комментарий</textarea></p>
<p><input type=submit value='Готово!'>
</form>
<hr>
<p>
Wed Jan 23 18:14:12 2002<br>
<b>Имя:</b> Первый пользователь<br>
<b>Комментарий</b>: Комментарий первого
пользователя

<hr>
<p>
Wed Jan 23 18:14:26 2002<br>
<b>Имя:</b> Второй пользователь<br>
<b>Комментарий</b>: Комментарий
второго пользователя
```

В форме указано, что при нажатии на кнопку 'Готово!' следует вызвать программу `guestbook.pl`. При этом, в соответствии с правилами обработки форм, данной программе передаются значения двух полей — `name` и `comment`.

Перейдем к рассмотрению программы `guestbook.pl`. В первой строке подключается модуль `CGI.pm`, содержащий функции, используемые в CGI-программах. На самом деле нам понадобится единственная функция `param`, и большинство программистов предпочли бы в такой ситуации обойтись без подключения модуля. Но это несколько усложнит текст, поэтому мы избрали не самый изысканный, но простой вариант.

Функции (подпрограммы) `outhead` и `outtail` служат для вывода начала (части до раздела данных) и конца HTML-документа. Подпрограммы могут располагаться в любой части программы; то, что мы разместили их в начале, не является обязательным.

Следующая подпрограмма `antihack`, на наш взгляд, является самой интересной. Какой бы простой не была наша гостевая книга, мы все же не хотим наступать на общеизвестные “грабли”. “Грабли” эти заключаются в предоставлении возможности пользователю использовать HTML-теги. Это позволит пользователям, например, помещать в гостевую книгу скрипты и прочие потенциально опасные фрагменты. Для исключения такой возможности достаточно заменить в тексте, вводимом пользователем, символы "<" и ">" на строки "<" и ">" соответственно. При этом отображение введенного пользователем текста не изменится, но HTML-теги будут интерпретироваться как обычный текст. Для того чтобы понять, как работает функция `antihack`, нужно разобраться с целым рядом вопросов.

Вопрос первый — о скалярных переменных и массивах. В языке Perl имеются не только скалярные переменные, но и массивы (в Perl имеются и другие структурированные типы, но они сейчас нас не интересуют). Имена скалярных переменных всегда начинаются с символа “\$” (примеры: `$a`, `$a1`, `$_`), имена массивов — с символа “@” (примеры: `@a`, `@a1`, `@_`). Следует обратить внимание на то, что элементы массивов являются скалярами, поэтому при обращении к элементу массива следует использовать символ “\$” (примеры: `$a[0]`, `$a1[0]`, `$_[0]`). Мы не случайно привели примеры “странных” имен (`$_` и `@_`). Такого рода имена в Perl допустимы, причем некоторые переменные с подобными “странными” именами используются для специальных целей. Пример этого мы сейчас увидим.

Вопрос второй — о передаче параметров в подпрограммы. Передача параметров в подпрограммы осуществляется через специальный массив `@_`. Если мы вызываем подпрограмму `f` от трех переменных `$a`, `$b`, `$c`: `f($a, $b, $c)`, этим переменным будут соответствовать элементы массива `@_`: `$_[0]`, `$_[1]` и `$_[2]`. Передача параметров производится по ссылке (т.е. все изменения элементов массива `@_` отражаются на фактических параметрах подпрограммы).

Вопрос третий — об операции замены в строке. Это наиболее сложный вопрос. Он имеет отношение к так называемым “регулярным выражениям”, и для его рассмотрения не хватит никакой статьи (впрочем, в ближайших номерах “Информатики” готовится к печати статья, посвященная регулярным выражениям). Поэтому мы поступим следующим образом: объясним ровно то, что написано в функции `antihack`.

Операция замены в строке в языке Perl имеет следующий вид:

```
$строковая_переменная =~ s/что_заменять/на_что_заменять/модификаторы.
```

В нашем случае (в функции `antihack`) используются:
`$строковая_переменная` — `$_[0]`;
`что_заменять` — символы "<" (в первом случае) и ">" (во втором случае);
`на_что_заменять` — строки "<" и ">" соответственно.

Модификаторы не являются обязательными, но мы использовали один модификатор “g” (*global*), для того чтобы замена производилась по всему тексту, а не единжды.

Как уже было сказано выше, в результате такой замены в тексте, введенном пользователем, будут “подавлены” все HTML-теги, например, строка "<script>" превратится в строку "<script>".

Закончив с функциями, перейдем к рассмотрению кода основной программы. В условном операторе несколько раз используется функция `param`. Эта функция содержится в модуле CGI (мы уже отмечали, что только из-за этой функции мы подключаем модуль CGI). При вызове без параметров эта функция возвращает логическое значение — были ли переданы в программу параметры. Мы не проверяем имена переданных параметров и считаем, что если уж параметры были, то это те параметры, которые нам нужны, — `name` и `comment`. По факту наличия или отсутствия параметров мы определяем и то, каким образом была вызвана программа `guestbook.pl`. Она может быть вызвана при запросе пользователем соответствующего URL (в этом случае программа не получает параметров и, пропуская раздел модификации файла `guestbook.html`, лишь выводит содержимое гостевой книги) или при заполнении формы — при нажатии на кнопку `submit` (в этом случае программа получит параметры `name` и `comment` и выполнит модификацию файла `guestbook.html`). Отметим, что избранный нами способ проверки механизма вызова программы, мягко говоря, не совсем правильный. Было бы лучше проверить метод вызова программы `guestbook.pl` — GET или POST. Но это, как и многое другое, может увести нас слишком далеко.)

В настоящей гостевой книге такую проверку следует выполнять хотя бы из соображений безопасности. В операторе

```
open GB, ">> guestbook.html";
```

файл `guestbook.html` открывается для дозаписи (*append*). (Режимом доступа к файлу можно управлять посредством символов, которые ставятся перед его именем. В частности, файл можно открыть для чтения — "<", для записи — ">" и, как в нашем случае, для дозаписи — ">>".)

Первый параметр оператора `open` — `GB` — идентификатор файла.

В следующих операторах мы получаем (снова посредством функции `param`) значения параметров `name` и `comment`, обрабатываем их функцией `antihack` и записываем в файл `guestbook.html`, добавляя элементы HTML-разметки.

После модификации файла `guestbook.html`

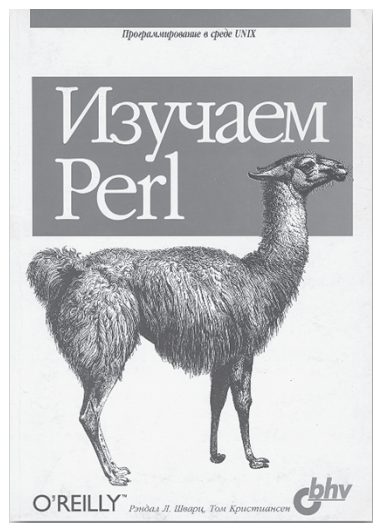
мы приступаем к формированию вывода. В первом операторе `print` выводится заголовок, после него, как уже было сказано выше, требуется вывести пустую строку. Далее вызывается подпрограмма для вывода части HTML-файла до раздела данных. Обратите внимание на способ вызова подпрограмм в Perl. Затем открывается для чтения файл `guestbook.html` и его содержимое выводится построчно. Здесь мы еще раз встречаемся с использованием специальной переменной с именем `$_`. Обсудим этот цикл подробнее. Чтение строки в переменную `$s` из файла, связанного с идентификатором `ID`, выполняется следующим образом: `$s = <ID>;`. Одно из многочисленных умолчаний языка Perl заключается в том, что, если имя переменной не указано, чтение производится в переменную с именем `$_`. Таким образом, развернутая запись оператора цикла в нашей программе следующая:

```
while ($_  
<GB>) { print $_; }
```

Мы использовали сокращенную запись потому, что, в первых, хотели продемонстрировать использование спе-

циальной переменной `$_`, а во-вторых, именно так и пишут на Perl. Именно такую запись вы скорее всего и встретите в книгах.

Кстати, о книгах. Для начального знакомства с языком Perl мы можем порекомендовать классическую и, возможно, лучшую из имеющихся книгу Р.Л. Шварца и Т.Кристиансена "Изучаем Perl". Эта книга вышла на русском языке в 1998 г. (на языке оригинала она была издана в 1997 г.) и выдержала множество переизданий. В магазинах книгу "Изучаем Perl" легко узнать по замечательному изображению ламы на обложке.



Калейдоскоп

Падение спроса на домены

По сообщениям Associated Press, количество доменных имен, заведенных в прошлом году, показало, что спрос на домены в COM, NET, ORG постепенно падает. Компания VeriSign Inc, "главный хранитель" этих трех суффиксов, сообщила, что количество доменов в прошлом году увеличилось всего на два процента. Для справки приводятся данные 2000 года, когда количество доменов утроилось, и 1999 год, где увеличение достигло ста процентов.

Приобретения доменов пришелся на июнь, когда их было зарегистрировано 32,4 миллиона, а затем это число стало уменьшаться: от 32 миллионов в сентябре до 28,8 миллиона в декабре.

Как считают специалисты, не последнюю роль в этом сыграли спекулянты, приобретающие "звучные" имена по дешевке за тридцать долларов и продающие их порой по баснословным ценам. Примером такой спекуляции может служить BUSINESS.COM, обошедшийся владельцу в 7,5 миллиона долларов.

Но и не последнюю роль сыграл тот факт, что в настоящее время стали разрабатываться такие суффиксы, как INFO и BIZ.

По материалам информационного агентства "Associated Press"

Подведены итоги прошедшего года

Журнал "NewScientist" подвел итоги прошедшего года. При этом все было разделено на множество подгрупп, среди которых есть и довольно необычные: "Поведение", "Наука о еде". Но мы остановимся на "Математике и Компьютерах".

Среди примечательных событий в этой области журнал выделил следующие:

1. Была обнаружена связь между цифрами, входящими в запись числа Пи. Оказывается, каждая десятичная цифра появляется с одинаковой частотой.
2. Была продемонстрирована возможность использования компьютерных мощностей машин, подключенных к Интернету, без согласия пользователей.
3. Американское правительство выступило с инициативой развития компьютерных мощностей до 12 триллионов операций в секунду, что помогло бы, например, исследователям из НАСА.
4. Были сделаны существенные шаги в теории чисел на пути разработки формул чисел, представимых в виде суммы квадратов.
5. Участвовавшие случаи заражения вирусами компьютеров в июле и августе прошлого года заставили ученых по-другому посмотреть на эту проблему.
6. Была создана программа, позволяющая анализировать аминокислотные последовательности.
7. Новая интерпретация глиняных табличек из Месопотамии, называемых Плимптон-322. Высказывается мнение, что это список примеров, который преподаватель использовал при опросе учащихся по теме "Квадраты и последовательности чисел".
8. Участниками сетевого проекта было получено очередное число Мессена (магическое), состоящее из 4 053 946 цифр.

Другие итоги были подведены на поисковом сервере Google. Эти исследования основывались на запросах пользователей Интернета в США, поступавших на протяжении всего 2001 года. Наибольшим интересом среди американцев пользовался великий Нострадамус. Оказывается, как утверждают некоторые источники Интернета, у предсказателя говорится о террористической атаке на США. И, как следствие, список возглавляют Усама бен Ладен, башни Нью-Йорка и Талибан.

Однако во всем остальном американцы остались верными самим себе. Очень их интересовала личная жизнь певцов и певиц. Компьютерные игры интересуют их меньше, чем Бритни Спирс и Майкл Джексон. А еще меньше волновала сфера техники, где лидирующие места занимают компания Nokia, Sony, BMW, Palm Inc., Adobe Systems Inc. и Dell Computer Corp.

По материалам журнала "NewScientist", газеты "Метро", поискового сервера Google

Некоронованный король



В 1830—1840 годах Гаусс занимался главным образом теоретической физикой. Значительную часть результатов он получил тут благодаря сотрудничеству с Вильгельмом Эдуардом Вебером (1804—1891).

Еще в 1829 году Гаусс сформулировал принцип наименьшего принуждения (*принцип Гаусса*).

В 1832 году он ввел понятие системы единиц физических величин. В качестве основных Гаусс выбрал единицы длины, массы и времени, а именно — миллиметр, миллиграмм и секунду. На их основе, используя закон Кулона для магнитных масс, он образовал единицы магнитных величин.

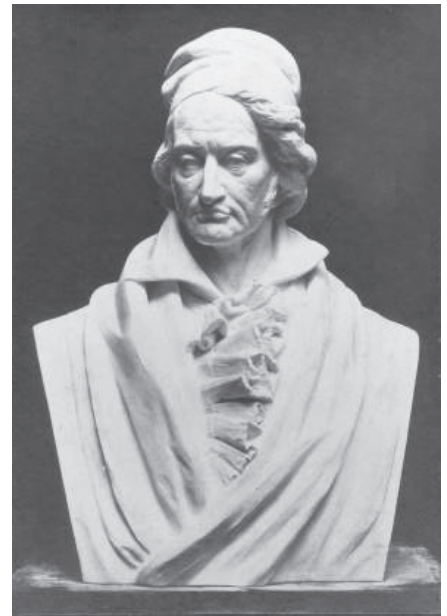
Примерно через год Гаусс и Вебер сконструировали первый в Германии электромагнитный телеграф [уже после устройства Павлом Львовичем Шиллингом (1786—1837) первого в мире телеграфа в России].

В 1839 году в небольшом сочинении «Общая теория сил притяжения и отталкивания, действующих обратно пропорционально квадрату расстояния», Гаусс изложил основы теории потенциала, в частности, ряд положений и теорем, например, основную теорему электростатики (*теорема Гаусса — Остроградского*).

К числу его физических работ относятся и «Диоптрические исследования»

(1840 г.), где даны основы теории построения изображений в системах линз.

Гаусс один из первых высказал (в 1818 г.) предположение о возможности существования неевклидовой геометрии [1—4]. Широко используется так называемое *распределение Гаусса* (нормальное распределение) — важ-



ный в теоретическом и практическом отношении закон распределения вероятностей.

Вообще трудно указать такую отрасль теоретической и прикладной математики, в которую Гаусс не внес бы существенного вклада. При этом следует отметить, что многие его исследования остались неопубликованными [2].

В 1855 году (уже после смерти ученого) в Германии изготовили медаль, на которой под барельефом Гаусса было написано: «*Mathematicorum princeps*» (Король математиков) [1]. Его именем названа единица измерения магнитной индукции (*гаусс*) в системе СГС.

Литература

1. Гиндикин С.Г. Рассказы о физиках и математиках. М.: Наука, 1985.
2. Гаусс // Большая советская энциклопедия. Изд. 2-е. М.: Гл. науч. изд-во «Большая советская энциклопедия», 1952. Т. 10.
3. Стройк Д.Я. Краткий очерк истории математики: Пер. с нем. 4-е изд. М.: Наука, 1984.
4. Храмов Ю.А. Физики. Биографический справочник. Изд. 2-е. М.: Наука, Гл. редакция физико-математической литературы, 1983.

Окончание. Начало на с. 1

ниям прикладного характера [2]. Сперва он занимался астрономией [в связи с астрономическими вычислениями ученый занялся исследованиями вопроса о сходимости бесконечных рядов, и эти исследования благодаря основанным уже на них работам Огюстена Коши (1789—1857) и Нильса Хенрика Абеля (1802—1829) привели к прогрессу в общей теории рядов], а затем занялся геодезией. Здесь он фактически создал науку, которая носит название *высшей геодезии* и имеет своей задачей установление формы земной поверхности не в упрощенном, а в действительном ее виде.

Чтобы установить значения тех или иных величин (длин, координат, дуг и т.п.), в астрономии и геодезии производятся многочисленные измерения в различных местах, различными инструментами, разными наблюдателями. Для

обработки результатов наблюдений Гаусс создал так называемый *метод наименьших квадратов*, который и сегодня является одним из самых широко применяемых математических методов [независимо от Гаусса метод наименьших квадратов был предложен французским математиком Адриеном Мари Лежандром (1752—1833)].



Гл. редактор
С.Л. Островский
Зам. гл. редактора
А.И. Сенокосов
Редакция:
Е.В. Андреева
Н.Л. Беленькая
Л.Н. Картелишвили
Н.П. Медведева
Дизайн и верстка:
Н.И. Пронская
Корректоры:
Е.Л. Володина,
С.М. Подберезина

©ИНФОРМАТИКА 2002
выходит четыре раза в месяц
При перепечатке ссылка
на ИНФОРМАТИКУ обязательна,
рукописи не возвращаются

Адрес редакции
и издателя:
121165, Киевская, 24
тел. 249-48-96
Отдел рекламы
тел. 249-98-70

Учредитель: ООО «Чистые пруды»

Зарегистрировано в Министерстве РФ по делам печати. ПИ № 77-7230 от 12.04.2001.
Отпечатано в ОИД «Медиа-Пресса»,
125993, ГСП-3, Москва, А-40, ул. «Правды», 24.
Тираж 7000 экз.
Срок подписания в печать по графику 06.02.2002.
Номер подписан 06.02.2002.
Заказ №
Цена свободная

ИНДЕКС ПОДПИСКИ
для индивидуальных подписчиков 32291
комплекта изданий 32744

Тел.: (095)249-31-38, 249-33-86. Факс (095)249-31-84

Internet: inf@1september.ru
WWW: http://www.1september.ru

ИЗДАТЕЛЬСКИЙ
ДОМ «ПЕРВОЕ
СЕНТЯБРЯ»,
ГЛАВНЫЙ
РЕДАКТОР —
А. СОЛОВЕЙЧИК

Газеты ИЗДАТЕЛЬСКОГО ДОМА: *Первое сентября* — гл. ред. Е.Бирюкова, *Английский язык* — гл. ред. А.Громушкина, *Библиотека в школе* — гл. ред. О.Громова, *Биология* — гл. ред. Н.Иванова, *Воскресная школа* — гл. ред. монах Киприан (Яценко), *География* — гл. ред. О.Коротова, *Дошкольное образование* — гл. ред. М.Аромштам, *Здоровье детей* — гл. ред. А.Лекманов, *Информатика* — гл. ред. С.Островский, *Искусство* — гл. ред. Н.Исмаилова, *История* — гл. ред. А.Головатенко, *Литература* — гл. ред. Г.Красухин, *Математика* — гл. ред. И.Соловейчик, *Начальная школа* — гл. ред. М.Соловейчик, *Немецкий язык* — гл. ред. М.Бузова, *Русский язык* — гл. ред. Л.Гончар, *Спорт в школе* — гл. ред. Н.Школьникова, *Управление школой* — гл. ред. А.Адамский, *Физика* — гл. ред. Н.Козлова, *Французский язык* — гл. ред. Г.Чесновицкая, *Химия* — гл. ред. О.Блохина, *Чудесная газета* — гл. ред. М.Аромштам, *Школьный психолог* — гл. ред. М.Сартан.