

С 1 по 30 апреля 2002 года в Московском городском доме учителя состоится

Московский педагогический марафон учебных предметов



• 1 апреля – День учителя русского языка • 2 апреля – День учителя литературы • 3 апреля – День учителей мировой художественной культуры, музыки и ИЗО • 4 апреля – День учителя истории • 5 апреля – День школьного библиотекаря • 8 апреля – День учителя географии • 9 апреля – День учителя биологии • 10 апреля – День учителя химии • 11 апреля – День учителя физики • 12 апреля – День учителя математики • 15 апреля – **ДЕНЬ УЧИТЕЛЯ ИНФОРМАТИКИ** • 16 апреля – День учителя английского языка • 17 апреля – День учителя немецкого языка • 18 апреля – День учителя французского языка • 19 апреля – День учителей технологии, профориентации и ОБЖ • 22 апреля – День учителя физкультуры • 23 апреля – День здоровья детей • 24 апреля – День дошкольного образования • 25 апреля – День учителя начальной школы • 26 апреля – День логопедов и коррекционных педагогов • 29 апреля – День школьного психолога • 30 апреля – День школьной администрации •

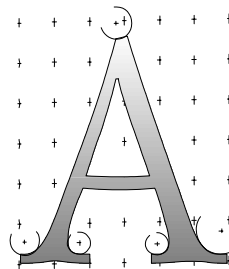
Подробную информацию о Дне учителя информатики см. на с. 8

№ 13 (350) 1–7 апреля 2002

ПОДПИСКА: (095) 249-47-58

Еженедельная газета Издательского дома «ПЕРВОЕ СЕНТЯБРЯ»

ИНФОРМАТИК



Java и JavaScript

10 лет назад, в начале 1990-х годов, появился язык программирования Java, предназначенный для создания используемых в сети Интернет прикладных программ („приложений“)

“Java – язык “Ява” (Ява – название главного острова Индонезии, на жаргоне *java* означает кофе, излюбленный напиток программистов). Высокоуровневый язык программирования. Разработан компанией Sun Microsystems. Является объектно-ориентированным языком, похожим на Си++, но упрощенным для устранения некоторых особенностей, которые ведут к ошибкам программирования... Один из основных языков программирования в сети Интернет”.

“**JavaScript** – язык программирования, предназначенный для расширения функциональности HTML-документов (создания активных динамических клиентских web-страниц)... Разработан фирмой Netscape совместно с Sun. Изначально назывался LiveScript (переименован из чисто коммерческих соображений. – Прим. ред.). Является как бы “младшим братом” языка Java. Хотя он содержит многие особенности и структуры языка Java, развитие этих языков происходило независимо. Основное отличие – JavaScript не требует никаких специальных средств разработки: программы на нем встраиваются непосредственно в документы HTML. Интерпретатором в данном случае выступает браузер”.

И.Л. Мостицкий. Новейший англо-русский толковый словарь по современной электронной технике (М.: Лучшие книги, 2000).

Программы, создаваемые компилятором языка Java, называются апплетами (applets), поскольку они обычно представляют собой небольшие прикладные (application) программы. Язык Java некоторое время имел название Oak (дуб), которое появилось в связи с тем, что за окнами кабинета Джеймса Гослинга, начавшего в 1990 году создавать этот язык (причем – для контроллеров, встроенных в бытовую технику), рос дуб [1]. Однако оказалось, что слово “Oak” часто употребляется в торговых марках.

Языки Java и JavaScript разрабатывались независимо друг от друга. Любой из них пригоден для решения

многих задач [2]. Java – полноценный язык программирования, для которого разработаны компилятор и инструментальные средства создания пользовательского интерфейса. Его применение не ограничивается обслуживанием web-страниц. JavaScript – более простой язык. Он тесно связан с HTML и в целом обеспечивает лишь дополнительные возможности формирования страниц.

Языки программирования Java и JavaScript имеют много общего.

Оба языка наследуют синтаксические конструкции Си и Си++. Это упрощает изучение Java и JavaScript.

Окончание читайте на с. 32

Читайте в номере

Страницы повышения квалификации 3–7

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

Списки, очередь, стек. Обычно ребята с удовольствием решают задачи с использованием этих структур — на уровне составления алгоритмов.

Попробуйте хотя бы ради “горящих” глаз ваших учеников ввести в курс информатики тему “Использование динамических структур данных в алгоритмах обработки информации”.

Информация..... 8, 26

Программа Дня учителя информатики, который пройдет 15 апреля в Московском городском доме учителя.

План серии тематических выпусков “Жаркое лето-2002”

Экзамены..... 9–13

Е.А. Еремин, А.П. Шестаков. Примерные ответы на примерные билеты

Сумеют ли ваши ученики объяснить, что такое информация, перечислить ее свойства, а также немного рассказать об информационных процессах, замкнутых и разомкнутых системах управления?

Примерные билеты (и ответы на них) для проведения итоговой аттестации выпускников 9-х классов. Сегодня представлены седьмой и восьмой билеты.

Новые информационные технологии 14–15

Т.С. Северова. Об использовании в образовании трехмерных виртуальных миров

“Виртуальные миры — это трехмерные модели реального или вымышленного мира, спроектированные с помощью специального языка и отображенные на экране компьютера”.

Познакомьте ребят с этой перспективной технологией, используя разработки компании Parallel Graphics.

На стенд в кабинете информатики 16–17

Электронная почта, или Как знак @ докатился до “собачьей жизни”

Дуг Энгельбарт, Рэй Томлинсон, Лоуренс Робертс. С именами этих людей связывают появление электронной почты. А кто из них ввел в обиход известный теперь всем значок @, какова история его происхождения и везде ли @ называют “собачкой”?

Задачи..... 18–20

О.В. Туркин. Создание модели соударения шариков с помощью электронной таблицы

Знаете ли вы, что модель такого рода можно построить с классами “средней подготовки” всего за два часа?

Уроки 21–26

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

При передаче информации могут возникнуть помехи, и тогда она поступает от источника к приемнику в искаженном виде. Кстати, пусть дети вспомнят игру “испорченный телефон”.

Продолжение второй книги, рассказывающей о способах хранения, передачи и обработки информации (“бумажная” версия первой книги — “Знакомство с компьютером” — представлена в № 1, 2/2002).

Информатика всюду 27–31

Эрик Фрэнк Рассел. Дышите... не дышите

Окончание рассказа, открывающего новую рубрику, который подтверждает, что информатика действительно всюду, причем не только на планете Земля...



День учителя информатики в Москве

15 апреля,

*Московский городской дом учителя
(улица Пушкинская, 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

Лекция 11. Алгоритм вычисления арифметических выражений и другие алгоритмы, использующие динамические структуры данных

Наш опыт преподавания информатики в школе показывает, что учащиеся старших, да и средних классов достаточно успешно усваивают понятия таких динамических структур данных, как списки, очередь и стек. Учащиеся с удовольствием решают задачи (на уровне составления алгоритмов) на использование этих структур. Конечно же, реализация этих алгоритмов на каком-либо алгоритмическом языке требует уже высокой культуры программирования. Но если поставить перед собой цель — познакомить школьников с некоторыми динамическими структурами данных и показать, как с использованием этих структур красиво решаются часто выполняемые задачи обработки информации, то, поверьте, цель эта реально выполнима.

Тема “Использование динамических структур данных в алгоритмах обработки информации” — один из немногих моментов, когда ученик соприкасается с “настоящей” в его понимании наукой, когда перед ним поднимают занавес абстрактного мышления, высокой формализации. А для любого школьника, даже для потерявшего интерес к учебе, заглядывание за этот занавес поднимает самооценку, они пока еще зачаровываются игрой слов “стек”, “кольцевой список”, они с гордостью (быть может, и не высказываемой, но глаза-то выдают!) осознают, что они тоже **могут**. Ради этих горящих глаз (а ведь именно это является педагогической целью) эту тему целесообразно вводить в курс школьной информатики.

Изложение темы “Алгоритмы, использующие динамические структуры данных”, достаточно ограничить разработкой алгоритмов без реализации их на алгоритмическом языке. Начинаем мы, как правило, с решения следующей задачи.

Задача-считалка. В кругу стоят N пронумерованных игроков и играют в игру-считалку. Счет начинается с первого игрока, и M -й игрок выбывает из круга. Игра-считалка продолжается с $(M + 1)$ -го игрока, каждый M -й по счету игрок выбывает из круга. Игра-считалка продолжается до тех пор, пока останется один человек. Выписать номера игроков, последовательно выбывших из круга.

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

В процессе обсуждения решения этой задачи естественно учащимся подвести к использованию структуры “кольцевой список”.

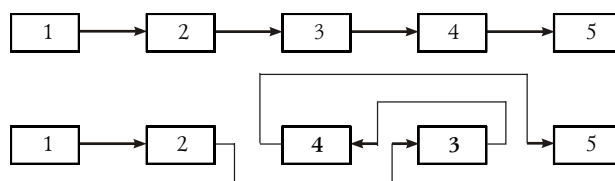
Динамические структуры данных

Развитие теории алгоритмов, а также появление автоматических вычислителей, устройств передачи, хранения и обработки информации вызвали интерес к таким структурам данных, которые сами изменяются в процессе выполнения алгоритма. Такие структуры данных стали называть *динамическими*.

Как правило, динамические структуры данных рассматривают как набор *элементов*, некоторым образом связанных друг с другом. Для динамических структур определяют операции, которые изменяют именно организацию (связи) элементов. Дадим определения некоторых динамических структур данных.

Линейный список — совокупность линейно связанных элементов, для которых разрешается добавлять элементы между любыми двумя другими и удалять любой элемент.

Пример графического изображения линейных списков



В линейном списке для каждого элемента, кроме *первого*, есть *предыдущий* элемент; для каждого элемента, кроме *последнего*, есть *следующий* элемент. Таким образом, все элементы списка *упорядочены* или связаны.

В линейном списке можно обойти все элементы, двигаясь последовательно от текущего элемента к следующему, начиная с первого. Аналогично линейный список можно обойти, двигаясь по предыдущим элементам, начиная с последнего. Элементы линейного списка можно *зачаровать*.

Пример 1

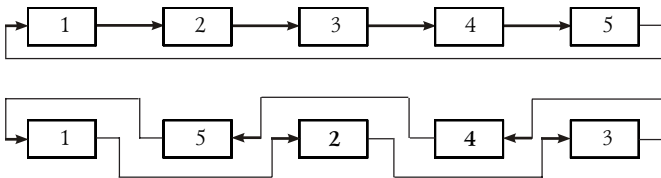
На уроке физкультуры класс выстраивается по росту; это список, в котором отношение “предыдущий — следующий” отражает отношение роста учеников “выше — ниже”.

Список дежурных по классу определяет отношение дней дежурств “раньше — позже”.

В классном журнале приведен список учеников. Порядок следования записей в журнале определяет отношение “предыдущий — следующий” между учениками. Как правило, сами записи в журнале имеют *дополнительное свойство* — они упорядочены по алфавиту.

Кольцевой список — такая же структура, как и линейный список, но имеющая дополнительную связь между последним и первым элементами.

Пример графического изображения кольцевых списков



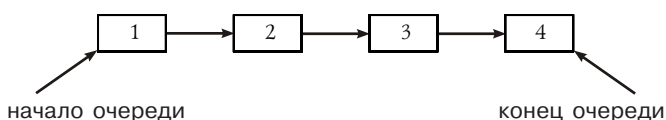
В кольцевом списке, в отличие от линейного, *все элементы равноправны* (поскольку для каждого элемента определены и предыдущий, и следующий элементы). Выделение “первого” и “последнего” элементов в кольцевом списке весьма условно, так как собственно *структура списка не имеет явно выделенных элементов!*

Пример 2

Во многих играх дети используют считалочки, чтобы выбрать ведущего, разделиться на команды и т.п. Как правило, считалочки длинные, и дети (сами того не зная) организуют кольцевой список. Отношение “предыдущий — следующий” определяется тем, кто считает.

Очередь — частный случай линейного односвязного списка, для которого разрешены только два действия: добавление элемента в *конец (хвост)* очереди и удаление элемента из *начала (головы)* очереди.

Пример графического изображения очереди

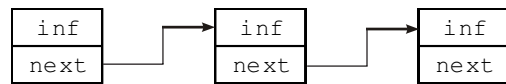
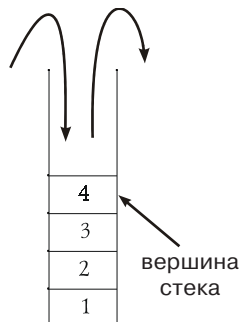


Стек — также частный случай линейного односвязного списка, для которого разрешены только два действия: добавление или удаление элемента только с одного конца стека, который называется *вершиной (головой)* стека.

Структуру “очередь” часто называют структурой типа FIFO (*First In First Out*): первый пришел — первый ушел. Структуру “стек” называют структурой типа LIFO (*Last In First Out*): последний пришел — первый ушел.

Для организации связей между элементами динамических структур данных требуется, чтобы каждый элемент, кроме информационных значений, содержал как минимум один указатель. Отсюда следует, что в качестве элементов таких структур необходимо использовать *записи*. В простейшем случае элемент динамической структуры может состоять из двух полей: *информационного* и *указательного*.

Пример графического изображения стека



Для создания и работы со списком надо иметь минимум два указателя: на начало очереди и на текущий (вставляемый/удаляемый) элемент.

Для создания и работы с очередью надо иметь минимум три указателя: на начало очереди, на конец очереди и на текущий (вставляемый/удаляемый) элемент.

Для создания и работы со стеком надо иметь минимум два указателя: на вершину стека и на текущий (вставляемый/удаляемый) элемент.

Работа с динамическими структурами данных достаточно подробно, с большим количеством графических иллюстраций и примеров описана в [1].

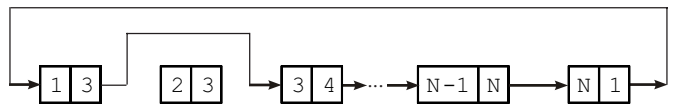
Алгоритм решения задачи “Считалка”

Отообразим информацию “игроки, стоящие по кругу”, на структуру “кольцевой список”. Каждый игрок будет соответствовать одному элементу кольцевого списка. Элемент списка состоит из двух полей: *inf* и *next*. В поле *inf* содержится номер игрока на момент начала игры. В поле *next* содержится указатель на следующего (по направлению счета) игрока. В качестве значения поля *next* будем использовать начальный порядковый номер следующего игрока.

Пусть в кругу на момент начала игры стояли N человек, тогда кольцевой список примет следующий вид:



Пусть в результате первого расчета считалка закончилась на втором игроке, т.е. игрок под номером 2 выбывает из круга. Тогда наш кольцевой список примет следующий вид:



Запишем алгоритм решения задачи-считалки по шагам.

Будем использовать следующие обозначения:

P — номер элемента, с которым работаем в данный момент. Содержимое его полей будем обозначать $P.inf$ и $P.next$, в каждом из этих полей содержится номер: в $P.inf$ — собственно номер элемента, в $P.next$ — номер его последующего соседа в кольцевом списке.

N — количество элементов в списке в начальный момент.

M — количество слов в считалке.

I — счетчик игроков внутри одной считалки.

① Перенумеруем игроков, т.е. заполним все поля: *inf* и *next*.

② Выберем любого игрока в качестве “водящего”, т.е. с него будет начинаться счет. (Для простоты реализации алгоритма выберем элемент с номером 1.) На “водящего” будет указывать указатель P .

③ Если $P.next = P$, т.е. в списке остался только один элемент, который “по кругу” указывает сам на себя, то печатаем номер элемента $P.inf$; конец алгоритма.

④ $I = 1$.

⑤ Если $I \neq M - 1$, то переход к следующему элементу: $P := P.next$. Операция перехода заключается в том, что мы сдвигаем указатель к следующему в кольцевом списке элементу.

⑥ $I := I + 1$, переход на п. ⑤.

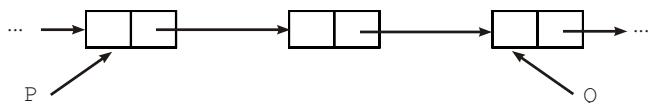
⑦ Иначе, т.е. если следующий элемент выбывает из списка (считалка почти закончилась), то

1) печатаем номер выбывающего игрока:

$P.next.inf$;

2) запоминаем в указателе Q номер элемента, с которого будет начинаться считалка в следующий раз:

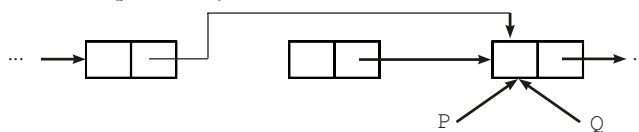
$Q := P.next.next$;



3) удаляем выбывший элемент из списка, для этого в текущем элементе (на него указывает P) изменяем значение поля $next$: $P.next := Q$;



4) делаем начальные установки для нового цикла считалки, т.е. переносим указатель P : $P := P.next$ или $P := Q$.



5) переход на п. ③.

Слева приведена блок-схема алгоритма решения задачи-считалки.

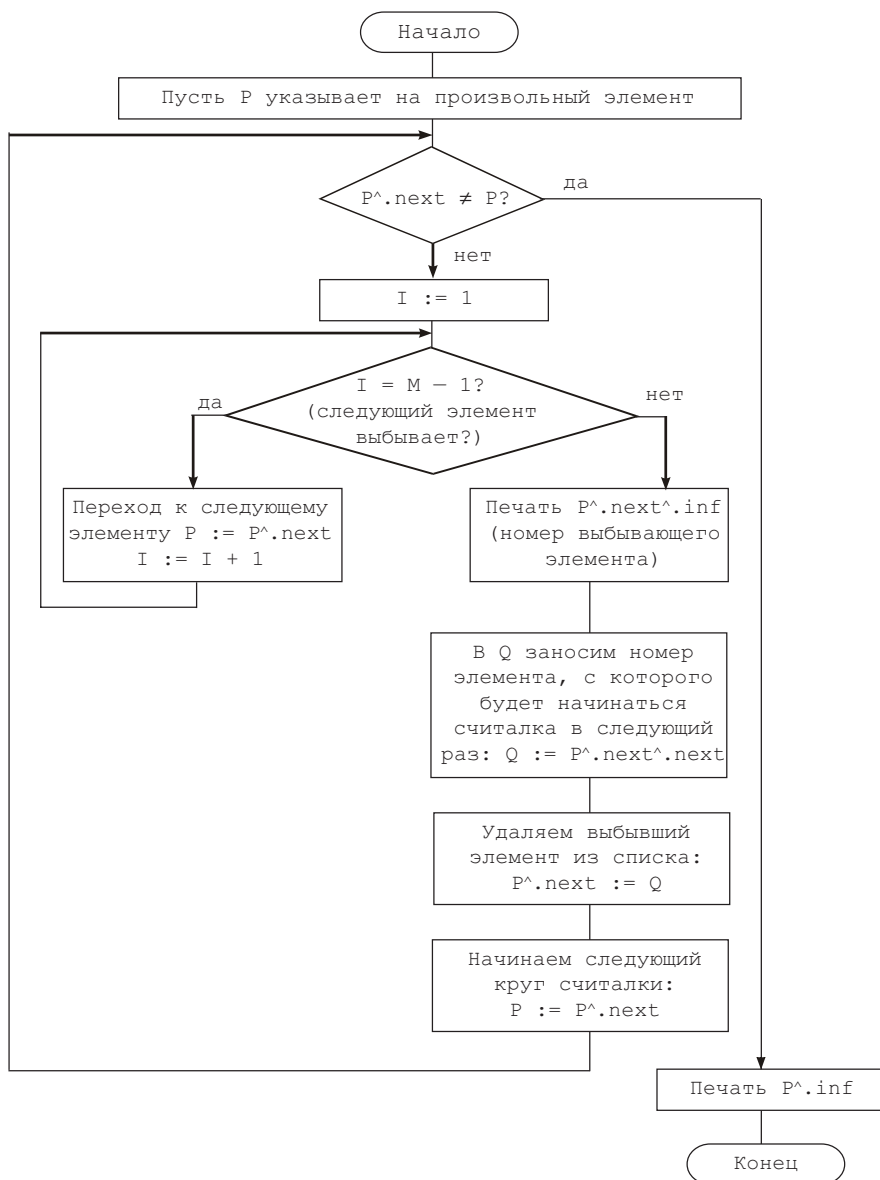
В ней использованы следующие обозначения:

P — указатель на текущий элемент в списке;

M — количество слов в считалке;

Q — элемент, с которого будет начинаться считалка в следующий раз;

I — счетчик игроков внутри одной считалки.



Задача определения сбалансированности скобок

После рассмотрения этой (или ей подобной) задачи целесообразно рассмотреть задачу определения сбалансированности скобок разных видов в арифметическом выражении. Например, проанализировать, сбалансированы ли круглые и квадратные скобки в выражении

$$[(a + b) - (a - c)] + ([[a - b]])$$

Для решения этой задачи будем использовать динамическую структуру *стек*. Исполнитель должен уметь выполнять следующие команды:

- 1) анализ символа из арифметического выражения;
- 2) анализ символа на вершине стека;
- 3) поместить символ в стек;
- 4) удалить символ из стека;
- 5) анализ стека на пустоту.

Приведем алгоритм решения этой задачи по шагам.

Будем использовать следующие обозначения:

i — счетчик анализируемого символа;

n — количество символов в выражении.

- ① $i = 0$.
- ② $i = i + 1$.
- ③ Если $i \leq n$, то переход на п. ⑤.
- ④ Если стек пуст, то выдаем сообщение “скобки сбалансированы”, в противном случае выдаем сообщение “скобки не сбалансированы”. Конец алгоритма.
- ⑤ Если i -й символ равен “(”, то помещаем его в стек, переход на п. ②.
- ⑥ Если i -й символ равен “)”, то проверяем вершину стека: если на вершине стека находится “(”, то извлекаем ее из стека; переход на п. ②, иначе скобки не сбалансированы, конец.
- ⑦ Если i -й символ равен “[”, то помещаем его в стек, переход на п. ②.
- ⑧ Если i -й символ равен “]”, то проверяем вершину стека: если на вершине стека находится “[”, то извлекаем ее из стека; переход на п. ②, иначе скобки не сбалансированы, конец.
- ⑨ Если i -й символ отличается от символов скобок, то переход на п. ②.

Справа приведена блок-схема алгоритма решения задачи определения сбалансированности скобок.

Задача вычисления значения арифметического выражения

После разбора этих двух задач попросите ребят составить алгоритм вычисления значения арифметического выражения. Вы увидите, как много различных алгоритмов они предложат, и все предложенные алгоритмы будут содержать ошибки (скорее всего каждый алгоритм будет решать какую-то частную, конкретную задачу). Но эти попытки учащиеся выполняют с интересом, иногда с азартом. Далее ничего не остается, как рассказать “теорию” решения этого вопроса.

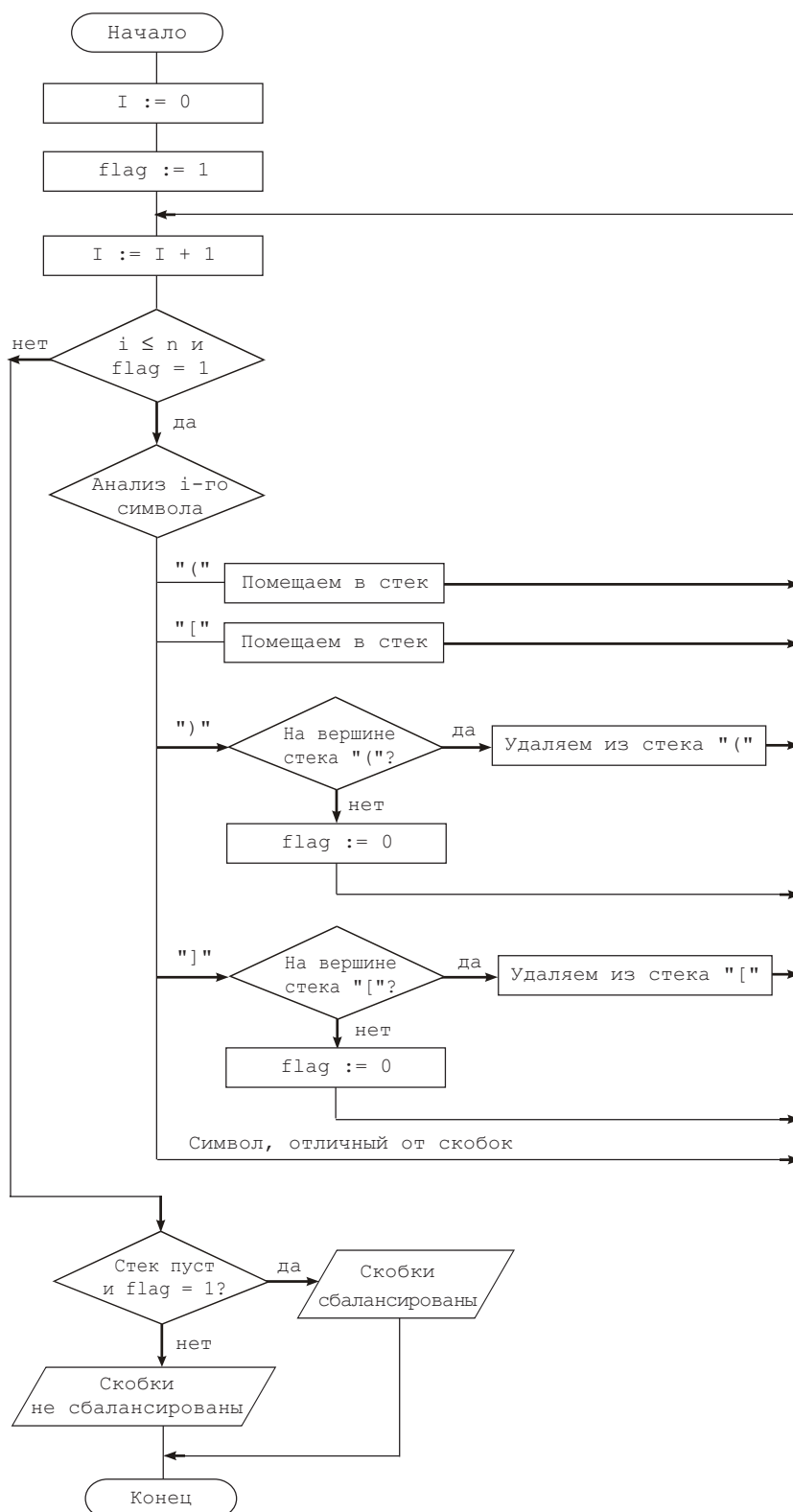
Для решения этой задачи сначала привычная нам (*инфиксная*) запись арифметического выражения переводится в *постфиксную* (“обратная польская запись”) форму. Затем по полученной записи производится вычисление выражения с использованием определенных правил.

Определение. Постфиксной формой записи выражения $a \Delta b$ называется запись, в которой знак операции размещен за операндами: $ab\Delta$.

Примеры

$$\begin{aligned} a - b &\rightarrow ab - \\ a * b + c &\rightarrow ab * c + \\ a * (b + c) &\rightarrow abc + * \end{aligned}$$

Далее необходимо на конкретных примерах показать решение поставленной задачи (два этапа: пере-



вод выражения в “обратную польскую запись” и вычисление выражения по “обратной польской записи”). Упражнения для таких заданий можно взять из задачника [2].

Алгоритм перевода арифметического выражения в “обратную польскую запись”

Приведем алгоритм перевода арифметических выражений в “обратную польскую запись” по шагам.

Будем использовать следующие обозначения:
 k — счетчик анализируемого элемента очереди;
 $St[k]$ — анализируемый элемент очереди;
 n — количество элементов в исходной очереди.

Алгоритм решения этой задачи строится на основе использования динамических структур данных — стека и двух очередей (исходной и результирующей). Считаем, что запись арифметического выражения расположена в элементах исходной очереди, первый элемент выражения (операнд, или знак унарной операции, или открывающая скобка) находится в начале очереди. В стек будем помещать только знаки операций и открывающие скобки. Выражение в форме “обратной польской записи” будет получено в результирующей очереди.

① $k = 0$.

② $k = k + 1$.

③ Анализируем k -й ($St[k]$) элемент очереди:

1) если $St[k]$ является переменной, то помещаем этот элемент в результирующую очередь; переход на п. ④;

2) если $St[k]$ является числом, то помещаем этот элемент в результирующую очередь; переход на п. ④;

3) если $St[k]$ является открывающей скобкой, то этот символ переносится в стек; переход на п. ④;

4) если $St[k]$ является закрывающей скобкой, то она удаляется из очереди. При этом из стека извлекаются все знаки операций до первой открывающей скобки и по порядку переносятся в результирующую очередь, а открывающая скобка из вершины стека изымается и уничтожается; переход на п. ④;

5) если $St[k]$ является знаком операции, то
 а) если в вершине стека находится знак операции с более низким приоритетом, или в вершине стека находится открывающая скобка, или стек пуст, то $St[k]$ заносится в стек, знак унарной операции “минус” при этом следует заменить, например, на “подчеркивание”; переход на п. ④;

б) в противном случае из стека изымаются знаки операций до тех пор, пока их приоритет больше или равен приоритету данного знака, и заносятся в результирующую очередь; затем элемент $St[k]$ заносится в стек; переход на п. ④.

④ Если исходная очередь пуста ($k = n$), то оставшиеся в стеке знаки операций по порядку переносятся в результирующую очередь — и конец алгоритма, иначе переход на п. ②.

Алгоритм вычисления значения арифметического выражения, записанного в форме “обратной польской записи”

Приведем алгоритм вычисления значения арифметических выражений, записанных в форме “обратной польской записи” по шагам.

Будем использовать следующие обозначения:
 k — счетчик анализируемого элемента;

$St[k]$ — анализируемый элемент очереди (им может быть операнд или знак операции);
 n — количество элементов в очереди.

Алгоритм решения этой задачи строится на основе использования динамических структур данных — стека и очереди. Считаем, что запись арифметического выражения поэлементно расположена в элементах очереди. Элементами “обратной польской записи” могут быть только переменные, числа и знаки операций. В стек будем помещать только операнды. Искомый результат (одно число) по окончании алгоритма будет находиться в стеке.

① $k = 0$.

② $k = k + 1$.

③ Анализируем k -й ($St[k]$) элемент очереди.

1) если $St[k]$ является операндом (переменная или число), то помещаем этот элемент в стек; переход на п. ④;

2) если $St[k]$ является знаком бинарной операции, то из стека извлекаются два верхних элемента и над ними выполняется операция, результат которой записывается в стек; переход на п. ④;

3) если $St[k]$ является знаком унарной операции, то из стека извлекается только верхний элемент и над ним выполняется унарная операция, результат которой записывается в стек; переход на п. ④.

④ Если исходная строка не исчерпана ($k \neq n$), то переход на п. ②, иначе конец алгоритма, в стеке находится искомый результат.

Замечания:

1) унарными операциями являются “+” и “—”, но при программной реализации алгоритма для обозначения этих операций разумно использовать другие символы;

2) при выполнении алгоритма все операции должны иметь приоритет, например, общепринятый в алгебре.

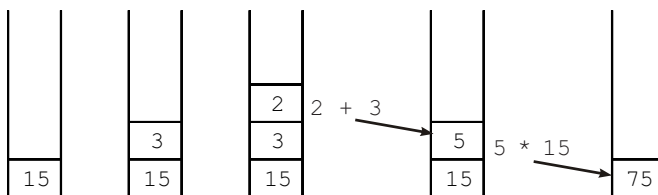
Пример

Вычислить выражение $15 * (3 + 2)$, используя вышеприведенный алгоритм.

Решение

1. Перевод в “обратную польскую запись” нам даст
 $15 \ 3 \ 2 \ + \ *$.

2. Подсчет по “обратной польской записи” даст следующие состояния стека:



Литература

1. Марченко А.И., Марченко Л.А. Программирование в среде Turbo Pascal 7.0. М.: Бином Универсал, Киев: ЮНИОР, 1997.

2. Пильщикова В.Н. Сборник упражнений по языку Паскаль: Учебное пособие для вузов. М.: Наука, 1989.

План проведения Дня учителя информатики в Московском городском доме учителя

15 апреля, ул. Пушкинская, дом 4, строение 2,
станция метро "Лубянка" или "Кузнецкий мост"

Время проведения мероприятия	БОЛЬШОЙ ЗАЛ Секция "Преподавание информатики в начальной, средней и старшей школе"	МАЛЫЙ ЗАЛ Секция "Страницы повышения квалификации"	ПАРКЕТНЫЙ ЗАЛ Секция "Мастер-классы"
10 ⁰⁰ — 10 ³⁰	Открытие Пленарное заседание Тема пленарного заседания: "Место курса информатики в действующем и экспериментальном учебных планах"		
10 ³⁰ — 10 ⁵⁰	Перерыв		
10 ⁵⁰ — 12 ¹⁰	Семинар "Информатика в начальной школе" Семинар проводит Ю.А. Первин	Семинар "Олимпиады по информатике. Пути к вершине" Семинар проводит Е.В. Андреева	Семинар "Виртуальная школа — всепроникающие информационные технологии" Семинар проводит А.И. Сенюков
12 ¹⁰ — 12 ³⁰	Перерыв		
12 ³⁰ — 13 ⁵⁰	Семинар "Базовый курс информатики" Семинар проводит Н.Д. Угринович	Семинар "Современные педагогические технологии и частные методики обучения информатике" Семинар проводит И.Н. Фалина	Семинар "Роботландский сетевой университет" Семинар проводит В.А. Козлова
13 ⁵⁰ — 14 ¹⁰	Перерыв		
14 ¹⁰ — 15 ³⁰	Семинар "Профильные курсы информатики" Семинар проводит Е.К. Хеннер	Семинар "Введение в специальность "учитель информатики" Семинар проводит А.Г. Гейн	Семинар "Олимпиады по базовому курсу информатики" Семинар проводит С.В. Русаков
15 ³⁰ — 15 ⁵⁰	Перерыв		
15 ⁵⁰ — 16 ⁵⁰	Пленарное заседание. Тема пленарного заседания: "Обновленное содержание курса информатики, стандарты, итоговая аттестация" Закрытие		
16 ⁵⁰ — 17 ¹⁰	Перерыв		
17 ¹⁰ — 18 ⁰⁰	Концерт		

В течение всего дня в Педагогической гостиной Дома учителя работает музей истории вычислительной техники, а в Голубой гостиной – выставка-продажа учебной литературы.

Примерные ответы на примерные билеты

Е.А. Еремин, А.П. Шестаков,
г. Пермь

Продолжение. См. № 9, 10, 11/2002

Билет № 7

1. Информация и информационные процессы в природе, обществе, технике. Информационная деятельность человека. Примеры.

2. Решение задачи на упорядочивание данных в среде электронных таблиц или в среде системы управления базами данных.

* * *

1. Информация и информационные процессы в природе, обществе, технике. Информационная деятельность человека. Примеры.

Информация относится к фундаментальным, неопределяемым понятиям науки информатика. Тем не менее смысл этого понятия должен быть разъяснен. Предпримем попытку рассмотреть это понятие с различных позиций.

Термин “информация” происходит от латинского слова *informatio*, что означает *сведения, разъяснения, изложение*. В настоящее время наука пытается найти общие свойства и закономерности, присущие многогранному понятию *информация*, но пока это понятие во многом остается интуитивным и получает различные смысловые наполнения в различных отраслях человеческой деятельности:

- **в быту** информацией называют любые данные, сведения, знания, которые кого-либо интересуют. Например, сообщение о каких-либо событиях, о чьей-либо деятельности и т.п.;

- **в технике** под информацией понимают сообщения, передаваемые в форме знаков или сигналов (в этом случае есть источник сообщений, получатель (приемник) сообщений, канал связи);

- **в кибернетике** под информацией понимают ту часть знаний, которая используется для ориентирования, активного действия, управления, т.е. в целях сохранения, совершенствования, развития системы;

- **в теории информации** под информацией понимают сведения об объектах и явлениях окружающей среды, их параметрах, свойствах и состоянии, которые уменьшают имеющуюся степень неопределенности, неполноты знаний о них.

Применительно к компьютерной обработке данных под информацией понимают некоторую последовательность символических обозначений (букв, цифр, закодированных графических образов и звуков и т.п.), несущую смысловую нагрузку и представленную в понятном компьютеру виде. Каждый новый символ в такой последовательности символов увеличивает информационный объем сообщения.

Информация может существовать в виде [7]:

- текстов, рисунков, чертежей, фотографий;
- световых или звуковых сигналов;
- радиоволн;
- электрических и нервных импульсов;
- магнитных записей;

- жестов и мимики;
- запахов и вкусовых ощущений;
- хромосом, посредством которых передаются по наследству признаки и свойства организмов, и т.д.

Свойства информации (с точки зрения бытового подхода к определению информации):

релевантность — способность информации соответствовать нуждам (запросам) потребителя;

полнота — свойство информации исчерпывающе (для данного потребителя) характеризовать отображаемый объект или процесс;

своевременность — способность информации соответствовать нуждам потребителя в нужный момент времени;

достоверность — свойство информации не иметь скрытых ошибок. Достоверная информация со временем может стать недостоверной, если устаревает и перестанет отражать истинное положение дел;

доступность — свойство информации, характеризующее возможность ее получения данным потребителем;

защищенность — свойство, характеризующее невозможность несанкционированного использования или изменения информации;

эргономичность — свойство, характеризующее удобство формы или объема информации с точки зрения данного потребителя.

Под *информационным* понимают процесс, связанный с определенными операциями над информацией, в ходе которого может измениться содержание информации или форма ее представления. В информатике к таким процессам относят *получение, хранение, передачу, обработку, использование* информации.

Получение информации основано на отражении различных свойств объектов, явлений и процессов окружающей среды. В природе такого рода отражение выражается в *восприятии* с помощью органов чувств. Человек пошел дальше по этому пути и создал множество приборов, которые многократно усиливают природные способности к восприятию.

Человек воспринимает с помощью органов чувств следующую информацию:

- визуальную (восприятие зрительных образов, различные цвета и т.д.) — с помощью зрения;
- звуковую (восприятие музыки, речи, сигналов, шума и т.д.) — с помощью слуха;
- обонятельную (восприятие запахов) — с помощью обоняния;
- вкусовую (восприятие посредством вкусовых рецепторов языка) — с помощью вкуса;
- тактильную (посредством кожного покрова восприятие информации о температуре, качестве предметов и т.д.) — с помощью осязания.

Примеры получения информации:

1) динамик компьютера издает специфический звук, хорошо знакомый Васе, — следовательно, пришло новое сообщение по ICQ;

2) с вертолета пожарной охраны в глубине леса замечен густой дым — обнаружен новый лесной пожар;

3) всевозможные датчики, расположенные в сейсмологически неустойчивом районе, фиксируют изменение обстановки, характерное для приближающегося землетрясения.

Хранение информации имеет большое значение для многократного использования информации, передачи информации во времени. С точки зрения человека, различная информация, в зависимости от степени ее важности и ценности, может иметь разное по длительности время хранения. Некоторую информацию он хранит в течение всей жизни (а может и передать потомкам), другую же — от нескольких секунд до нескольких дней. Память человека не способна хранить всю получаемую информацию (следует иметь в виду, что получение информации не прекращается ни на секунду). Для долговременного хранения используются книги, в настоящее время — компьютерные носители, устройства внешней памяти и др. Следует заметить, что информация чаще всего хранится для неоднократной дальнейшей работы с ней. В этом случае для ускорения поиска информация должна быть тем или иным образом упорядочена. В библиотеках это картотеки, при хранении с использованием компьютера — базы данных, информационно-поисковые системы и т.д., в простейшем случае — размещение информации в определенных папках. При указанных способах хранения информация хранится в знаковой форме.

Передача информации необходима для того или иного ее распространения. Простейшая схема передачи такова: **источник информации — канал связи — приемник (получатель) информации**

Для передачи информации с помощью технических средств необходимо *кодирующее устройство*, предназначенное для преобразования исходного сообщения источника информации к виду, удобному для передачи, и *декодирующее устройство* для преобразования закодированного сообщения в исходное.

При передаче информации необходимо учитывать тот факт, что информация при этом может теряться или искажаться, т.е. при передаче информации могут присутствовать *помехи*. Для нейтрализации помех при передаче информации зачастую используют помехоустойчивый избыточный код, который позволяет восстановить исходную информацию даже в случае некоторого искажения. Другой случай — преднамеренное искажение информации злоумышленниками. На этот счет тоже предусмотрены свои средства. Существует специальная наука, которая разрабатывает способы защиты информации, — криптология.

Основными устройствами для быстрой передачи информации на большие расстояния в настоящее время являются телеграф, радио, телефон, телевизионный передатчик, телекоммуникационные сети на базе вычислительных систем.

Обработка информации подразумевает преобразование ее к виду, отличному от исходной формы или содержания информации.

Наиболее общая схема обработки информации такова: **входная информация — преобразователь информации — выходная информация**

Процесс изменения информации может включать в себя, например, такие действия: численные расчеты, редактирование, упорядочивание, обобщение, систематизацию и т.д.

Примеры изменения формы информации при обработке: перевод с одного языка на другой, двоичное кодирование изображения и т.д. Вообще чаще всего изменение формы информации предполагает наличие процесса кодирования и декодирования.

Частные примеры обработки информации:

1) в приведенном выше примере с сейсмологической станцией после получения информации о приближающемся землетрясении все полученные данные обобщаются, процесс землетрясения моделируется и прогнозируются возможные его ход и последствия;

2) учителю приносят личные дела учащихся, поступивших в первый класс. На основе анализа этих материалов учитель составляет классный журнал, где список учащихся составлен в алфавитном порядке, заполнена необходимая справочная информация об учащихся и т.д.;

3) ученый-математик доказывает новую теорему, т.е. на основе имеющейся системы аксиом, определений и ранее доказанных теорем обосновывается новое утверждение.

Живые организмы и растения обрабатывают информацию с помощью своих органов и систем. Компьютер является устройством, которое по разработанным человеком программам производит автоматическую обработку информации. Чаще всего, с точки зрения человека, он действует по принципу “черного ящика”, т.е. для определенных наборов данных по указанной выше схеме позволяет получить соответствующие им выходные результаты (алгоритм обработки при этом неизвестен).

Результаты обработки информации в дальнейшем *используются* в тех или иных целях.

Деятельность человека, которая связана с процессами получения, преобразования, накопления, передачи и использования информации, управления, называют *информационной деятельностью*.

Вот **основные вехи** в процессе развития и совершенствования информационной деятельности человека.

Появление речи. Значительно расширило возможности информационной деятельности человека, в особенности передачи информации.

Возникновение письменности. Дало возможность долговременного хранения информации и передачи накопленных знаний и культурных ценностей последующим поколениям.

Изобретение книгопечатания. Революция в мире тиражирования знаний, хранящихся в письменном виде. Расширение научной информации, развитие художественной литературы и т.д.

Возможность быстрого тиражирования книг приводит к росту количества библиотек, архивов, аккумулирующих знания человечества. Целенаправленная обработка информации по-прежнему остается прерогативой человека.

Изобретение ЭВМ — универсальных инструментов информационной деятельности. В последние десятилетия рост объема информации настолько велик, что это стало объективной предпосылкой появления такого рода инструментов. Практически во все сферы информационной (да и не только) деятельности человека внедряются компьютеры.

Разработка способов и методов представления информации, технологий решения повседневных и научных задач с использованием компьютеров стала важным аспектом деятельности людей многих профессий.

Основная литература

1. Гейн А.Г., Сенокосов А.И., Шолохович В.Ф. Информатика: 7—9-е классы. Учебник для общеобразовательных учебных заведений. М.: Дрофа, 1998.

2. Каймин В.А., Щеголев А.Г., Ерохина Е.А., Федюшин Д.П. Основы информатики и вычислительной техники: Пробный учебник для 10—11-х классов средней школы. М.: Просвещение, 1989.

3. Кушниренко А.Г., Лебедев Г.В., Сворень Р.А. Основы информатики и вычислительной техники: Учебник для средних учебных заведений. М.: Просвещение, 1993.

4. Семакин И., Залогова Л., Русаков С., Шестакова Л. Информатика: учебник по базовому курсу (7—9-е классы). М.: Лаборатория базовых знаний, 1998.

5. Угринович Н. Информатика и информационные технологии. Учебное пособие для общеобразовательных учреждений. М.: Лаборатория базовых знаний, 2001, 464 с. (Введение в информатику, с. 13—16.)

6. Информатика. 7—8-е классы / Под ред. Н.В. Макаровой. СПб.: ПитерКом, 1999, 368 с.

Дополнительная литература

7. Шауцукова Л.З. Информатика: Учебник для 10—11-х классов. М.: Просвещение, 2000 г.

8. Гейн А.Г. Обязательный минимум содержания образования по информатике: и в нем нам хочется дойти до самой сути // Информатика № 24/2001. С. 3—9.

* * *

2. Решение задачи на упорядочивание данных в среде электронных таблиц или в среде системы управления базами данных.

Имеется файл данных, содержащий некоторую информацию (по усмотрению учителя это может быть файл данных табличного процессора или СУБД). Требуется упорядочить эту информацию по возрастанию (убыванию) значений, содержащихся в одном из полей.

Примечание. Учащимся, претендующим на оценку “отлично”, можно предложить упорядочить информацию по нескольким полям.

Приведем пример.

Пусть имеется файл данных KOLL.XLS (KOLL.MBD), содержащий информацию о результатах самостоятельных работ по информатике в 9-м классе. Вычислить средний балл учащихся и упорядочить информацию в порядке убывания среднего балла.

Пример исходного файла (электронной таблицы) приведен на рисунке.

A	B	C	D	E	F	
1	Результаты самостоятельных работ по информатике в 9 классе					
2						
3	№	Фамилия, имя	С.р. № 1	С.р. № 2	С.р. № 3	С.р. № 4
4	1	Варанкин Александр	4	4	4	5
5	2	Деменова Елена	5	5	5	5
6	3	Ермакова Екатерина	4	3	4	4
7	4	Карчинская Светлана	4	4	4	3
8	5	Марамыгина Светлана	4	4	3	2
9	6	Мокрушина Анна	4	3	3	4
10	7	Новикова Татьяна	5	5	5	5
11	8	Носкова Ирина	3	4	3	4
12	9	Сарачева Наталья	4	4	4	4
13	10	Углицких Татьяна	4	4	3	3
14	11	Узельман Ольга	5	5	4	4
15	12	Устинова Анна	4	4	3	3
16	13	Уточкин Алексей	4	4	3	3
17	14	Чупашева Елена	4	4	4	3

Для решения задачи (после вычисления среднего балла) можно выбрать один из вариантов:

1) установив курсор в одну из ячеек поля, по которому производится сортировка, выбрать на панели инструментов “Сортировка по убыванию”;

2) меню “Данные | Сортировка”, выбрать поля и способ сортировки.

Результат представлен на рисунке ниже.

A	B	C	D	E	F	G	
1	Результаты самостоятельных работ по информатике в 9 классе						
2							
3	№	Фамилия, имя	С.р. № 1	С.р. № 2	С.р. № 3	С.р. № 4	Ср. балл
4	2	Деменова Елена	5	5	5	5	5
5	7	Новикова Татьяна	5	5	5	5	5
6	11	Узельман Ольга	5	5	4	4	4,5
7	1	Варанкин Александр	4	4	4	5	4,25
8	9	Сарачева Наталья	4	4	4	4	4
9	3	Ермакова Екатерина	4	3	4	4	3,75
10	4	Карчинская Светлана	4	4	4	3	3,75
11	14	Чупашева Елена	4	4	4	3	3,75
12	6	Мокрушина Анна	4	3	3	4	3,5
13	8	Носкова Ирина	3	4	3	4	3,5
14	10	Углицких Татьяна	4	4	3	3	3,5
15	12	Устинова Анна	4	4	3	3	3,5
16	13	Уточкин Алексей	4	4	3	3	3,5
17	5	Марамыгина Светлана	4	4	3	2	3,25

Билет № 8

1. Управление как информационный процесс. Закрытые и разомкнутые системы управления, назначение обратной связи. Примеры.

2. Практическое задание на создание на диске архива файлов и раскрытие архива с использованием программы-архиватора.

* * *

1. Управление как информационный процесс. Закрытые и разомкнутые системы управления, назначение обратной связи. Примеры.

В повседневной жизни мы встречаемся с процессами управления очень часто: пилот управляет самолетом, а помогает ему в этом автоматическое устройство — автопилот; директор и его заместители руководят производством, а учитель — обучением школьников; процессор обеспечивает синхронную работу всех узлов компьютера, а каждым его внешним устройством руководит специальный контрол-

лер; без дирижера большой оркестр не может согласованно исполнить музыкальное произведение, а хоккейная или баскетбольная команда обязательно имеет одного или нескольких тренеров, которые организуют подготовку спортсменов к соревнованиям. В наиболее общем виде управление есть совокупность действий, осуществляемых человеком, группой людей или автоматическим устройством (автоматом), направленных на поддержание или улучшение работы некоторого объекта. Эта деятельность ведется либо по заданным извне правилам, либо планируется самостоятельно в соответствии с известной целью управления. Управлять — значит влиять на ход какого-либо процесса в живом организме, машинах, обществе или на состояние некоторого объекта. Управление обязательно ведется целенаправленно, а случайные хаотические воздействия на систему не имеют к этому прямого отношения.

Независимо от природы исполнительных и управляющих органов процесс управления имеет определенные об-

щие закономерности. Их изучением занимается специальная наука, которая называется **кибернетикой**.

Основоположником кибернетики является американский ученый Норберт Винер (1894—1964). В 1948 году вышла его книга “Кибернетика, или Управление и связь в животном и машине”, заложившая основы новой науки. В Советском Союзе большой вклад в развитие теоретической и прикладной кибернетики внесли академики Аксель Иванович Берг (1893—1979) и Виктор Михайлович Глушков (1923—1982).

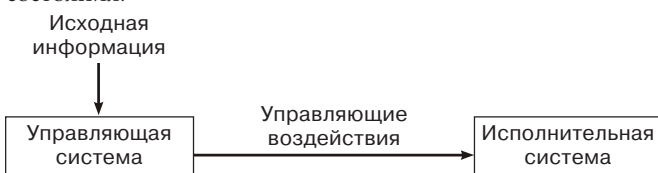
Кибернетические исследования заключаются в изучении наиболее общих свойств процессов управления в живых и неживых системах. При этом используются в основном абстрактные системы, т.е. некоторые обобщенные модели реальных систем различной природы — технических, биологических или социальных. Именно такой подход позволяет отразить наиболее общие закономерности, не обращая внимания на второстепенные детали устройства конкретных систем.

Управление является информационным процессом. Действительно, для принятия тех или иных решений и их реализации требуется постоянно производить различные действия с информацией. Сначала нужно *получить* и *обработать* данные о состоянии регулируемого объекта и окружающей его среды. Затем сформированные в результате проведенного анализа управляющие команды необходимо *передать* контролируемому объекту для изменения его состояния в соответствии с целью управления. Часто при организации процесса управления требуется также *хранить* поступающую информацию, для того чтобы использовать ее в дальнейшем. Таким образом, в ходе управления применяются все наиболее важные виды работы с информацией, фигурирующие в определении информационного процесса.

Итак, в рассматриваемой научной дисциплине основными являются понятия **управление** и **информация**. Необходимо отметить, что первоначально *все* исследования в области информации относились к кибернетике, так как считалось, что использование информации в системах управления является центральной проблемой. Однако с развитием науки область ее проблем расширялась, изучались все новые аспекты работы с информацией, и в конце концов проблемы управления выделились в самостоятельную область. В результате возникла более общая наука — **информатика**, в которую в качестве одной из составных частей вошла **кибернетика**.

Системы, изучаемые в кибернетике, могут быть очень сложными, включающими в себя множество взаимодействующих объектов. Однако для понимания базовых понятий теории можно обойтись простейшей из таких систем, которая содержит *всего два объекта* — *управляющий и исполнительный*. Практическим примером может служить, например, система, состоящая из светофора и автомобиля. Как ни удивительно, даже в таких небольших системах информационное взаимодействие может быть построено по разным схемам.

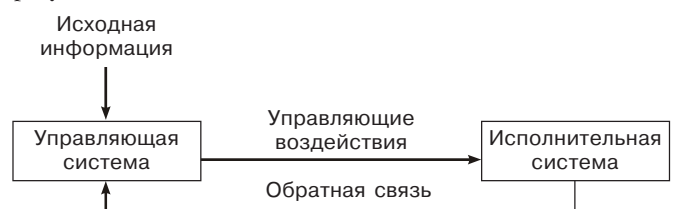
В простейшем случае управляющий объект посылает свои команды исполнительному объекту, никак не учитывая его состояния.



На схеме изображены управляющая и исполнительная системы, а также информационные потоки, которые они используют. В рассматриваемом нами сейчас случае каждая система состоит из единственного объекта, но это, разумеется, совсем

не обязательно. Например, на практике автомобилей на перекрестке обычно несколько. Для того чтобы управляющая система начала работать, в нее необходимо ввести некоторую исходную информацию о способах управления (в случае светофора она содержит программу переключения управляющих сигналов). Используя входную информацию, управляющая система начинает выдавать команды управления для исполнительной системы (объекта управления). Поскольку в описанной схеме воздействия передаются только в одном направлении, она называется **разомкнутой**. Помимо описанной выше, разомкнутыми системами являются всевозможные информационные табло на вокзалах и аэропортах, которые управляют перемещениями многочисленных пассажиров. К рассматриваемому классу систем можно отнести и всевозможные современные программируемые бытовые приборы. В обществе также существует способ управления, когда состояние его членов никак не влияет на принимаемые законы и решения, он называется диктатурой. Как правило, описанная схема управления не очень эффективна и нормально работает только до возникновения экстремальных условий. Так, при больших потоках транспорта возникают пробки, в аэропортах и вокзалах приходится дополнительно открывать справочные бюро, в микроволновой печи при неправильной программе может произойти перегрев, из-за которого кусочки продуктов разлетятся по стенкам; диктаторы чаще всего вынуждены бежать из своей страны, спасаясь от справедливого гнева населения.

Более совершенные системы управления отслеживают результаты деятельности исполнительной системы.



В таких системах дополнительно появляется еще один информационный поток, на этот раз от объекта управления к системе управления; в литературе его принято называть **обратной связью**. Именно по каналу обратной связи передаются сведения о состоянии объекта и степени достижения (или, наоборот, не достижения) цели управления. Из рисунка видно, почему такая система называется **замкнутой**. Главным принципом управления в замкнутой системе является выдача управляющих команд в зависимости от получаемых сигналов обратной связи. Легко понять, что управляющая система стремится скомпенсировать любое отклонение объекта от состояния, предусмотренного целями управления. Например, если на одной из улиц перекрестка машин накопилось много, а на другой их практически нет, то алгоритм поочередного перекрытия улиц требуется нарушить, чтобы по возможности пропустить максимум машин по перегруженной улице. Обратную связь, при которой управляющий сигнал стремится уменьшить (скомпенсировать) отклонение от некоторой поддерживаемой величины, принято называть **отрицательной**. Сущность управления с помощью обратной связи хорошо описал основоположник кибернетики Н.Винер в своей статье “Мое отношение к кибернетике. Ее прошлое и будущее”:

“Мысль моя заключалась в следующем. В устройствах управления применяется метод стабилизации действия, при котором какая-либо величина, зависящая от успеха действия, подается обратно на вход устройства как новая регулирующая порция информации. Так как каждое откло-

нение от заданного значения здесь компенсируется корректирующим действием в противоположном направлении, то подобная обратная связь называется отрицательной. Нам... пришло в голову, что такие простые человеческие действия, как вождение автомобиля, регулируются отрицательными обратными связями. Мы поворачиваем рулевое колесо автомобиля не по заранее составленной программе, а так, что если мы отклоняемся слишком влево, то правим вправо, и наоборот. Поэтому мы были убеждены, что отрицательная обратная связь участвует в человеческом механизме управления, и в частности, в том, при помощи которого мы следим взглядом за самолетом”.

Приведем еще несколько примеров замкнутых систем управления. Начнем с хорошо знакомого вам примера — обучения в классе. Здесь управляющую систему представляет учитель, а ученики являются объектом управления (авторы просят прощения, если применение принятой в кибернетике терминологии кому-то показалось обидным; помните, что научные термины всегда лишены эмоциональной окраски и никого не унижают и не обижают). Прямой канал передачи информации — это передача знаний учителем, а обратная связь — ответы учеников, результаты контрольных работ, сочинения и т.п. Благодаря обратной связи в результате анализа проведенной контрольной учитель может, например, провести дополнительный урок по данной теме или, наоборот, особо отличившихся учеников освободить от каких-то заданий.

Интересный пример замкнутой системы автоматического управления описан в книге известного писателя-сценариста Майкла Крайтона “Крылья”.

“Все современные самолеты нуждаются в компьютерах для стабилизации в полете — и не потому, что ими не могут управлять пилоты. Пилоты могут водить их, тут нет никаких сложностей. Однако авиакомпании желают иметь экономичные машины. Максимальная экономия топлива достигается при минимальной тяге.

...Чтобы минимизировать тягу, ... самолет должен очень точно выдерживать определенное положение в воздухе. Самое эффективное положение — легкий наклон носом вверх. При крейсерском полете самолет удерживается в этом положении при помощи компьютеров. Это самое обычное дело”.

К чему может приводить неправильное управление при наличии автопилота, прочтите в цитируемой книге сами. А те, кого заинтересовали принципы автоматизации управления полетом, могут обратиться к статье “Автоматическое управление” на с. 106—108 энциклопедического словаря [1].

Завершая обсуждение принципов работы управляющих систем, отметим, что их изучение не является самоцелью, а служит основой для разработки устройств автоматического управления. Кибернетика разделяет системы управления на три группы: **неавтоматические**, **автоматизированные** и **автоматические**. Как очевидно из названия, в **неавтоматических** системах управлением человек занимается самостоятельно. **Автоматизированные** системы управления (АСУ) занимают некоторое промежуточное место — сбор необходимой для принятия решения информации и ее обработка производится автоматически, а окончательное решение принимает человек. Наконец, в системах **автоматического** управления (САУ) все операции, связанные с процессами управления, происходят без участия человека. Разумеется, программы (алгоритмы) управления предварительно готовит человек. Количество автоматизированных и автоматических систем вокруг нас неуклонно расширяется.

Литература

1. Информатика: Энциклопедический словарь для начинающих / Сост. — Д.А. Пospelов. М.: Педагогика-Пресс, 1994, 352 с.
2. Информатика в понятиях и терминах: Книга для учащихся старших классов средней школы / Г.А. Бордовский, В.А. Извозчиков, Ю.В. Исаев, В.В. Морозов; Под ред. В.А. Извозчикова. М.: Просвещение, 1991, 208 с.
3. *Семакин И., Залогова Л., Русаков С., Шестакова Л.* Информатика. Учебник по базовому курсу (7—9-е классы). М.: Лаборатория базовых знаний, 1998, 464 с.

Дополнительная литература

Шилейко А.В., Шилейко Т.И. Кибернетика без математики. М.: Энергия, 1973, 144 с.

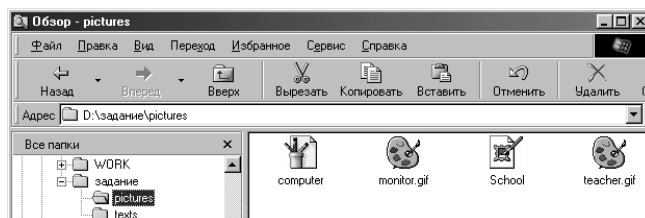
* * *

2. Практическое задание на создание на диске архива файлов и раскрытие архива с использованием программы-архиватора.

Задание. Создать на дискете архив предложенной учителем папки (каталога), сохраняя при этом ее внутреннюю структуру. Раскрыть архив в *другую* указанную учителем папку и убедиться в идентичности результата и исходной папки.

Комментарии к заданию

Пытаться описать выполнение задания для всех существующих архиваторов — занятие трудновыполнимое в силу их большого разнообразия. Здесь обратим только внимание читателя на содержание задачи. Представляется необходимым, чтобы оно было хотя и не очень сложным, но все же соответствующим тому, что бывает на практике. В связи с этим можно предложить, например, в качестве исходной информации для архивации следующую структуру исходной папки **задание**:



Как видно из рисунка, в данную папку вложены еще две, в одной из которых (pictures) находятся графические файлы нескольких типов, а в другой (texts) — тексты. Ученик при архивации должен сохранить эту структуру, в противном случае оценка должна быть снижена.

При выполнении второй части задания — распаковке архива желательно обратить внимание на умение ученика распаковать архив сразу в указанную папку (на нашем рисунке WORK), а не переписывать ее после проведения разархивации.

В качестве дополнительного вопроса можно попросить найти с помощью архиватора информацию о том, какова степень сжатия для каждого типа файлов, и объяснить полученный результат. Так, текстовые файлы и графические файлы типа BMP содержат много избыточной информации, поэтому сжимаются хорошо. Напротив, в файлах GIF и JPG уже используется сжатие при сохранении информации и, следовательно, их объем не может быть существенно уменьшен в ходе архивации.

Если учитель считает экзаменуемых достаточно подготовленными по описываемому вопросу, можно усложнить задание, потребовав, например, создать самораспаковывающийся или многотомный архив.

Об использовании в образовании трехмерных виртуальных миров

Т.С. Северова,
Москва

Виртуальные миры — это трехмерные модели реального или вымышленного мира, спроектированные с помощью специального языка и отображенные на экране компьютера. Язык моделирования виртуальной реальности *Virtual Reality Modeling Language (VRML)* позволяет создавать и размещать в сети Интернет разнообразные трехмерные пространства с использованием объектов, текстур, графических изображений.

Интерес к виртуальной реальности возник несколько лет назад [1]. Тогда казалось, что уже завтра на смену “плоскому” двумерному пространству Интернета придут трехмерные виртуальные миры. Вот как Дж. Хартман и Ж.Вернеске [2] описывали новые возможности: “Вы можете использовать VRML для создания интерактивных моделей, включающих анимацию, физическое движение, а также управление в режиме реального времени с участием многих пользователей. Виртуальные ландшафты, созданные вами, могут быть доступны пользователям Всемирной паутины, отображены на экранах компьютеров других пользователей, а также просмотрены удаленными пользователями в интерактивном режиме... Приложения для VRML подразделяются на серьезные (медицинские изображения, молекулярное моделирование, инженерные работы и дизайн, архитектура), занимательные (игры, приложения в любых формах, виртуальные тематические парки) и используемые в повседневной жизни (выбор и расположение мебели в гостиной, планирование воскресной прогулки в загородном парке, ремонт карбюратора)”. В книге Е.Якушиной [3] языку моделирования виртуальной реальности посвящена отдельная глава, в которой автор излагает историю создания языка, рассматривает его основные функции, примеры использования и направления развития.

Сегодня ясно, что стремительного перехода к VRML не получилось. Что же сдерживает его повсеместное распространение? Во-первых, для комфортной работы с трехмерными изображениями нужны достаточно мощные компьютеры, иначе перемещение в виртуальном мире будет скачкообразным. Во-вторых, это довольно сложный язык высокого уровня, на котором трудно программировать “вручную”, в отличие от более простого языка разметки гипертекста HTML. Нужны инструментальные средства разработки VRML-приложений [4]. И, наконец, для просмотра VRML-файлов требуются специальные программы.

Очевидно, что с развитием аппаратных и программных средств перечисленные проблемы будут решены, и это приведет к значительно более широкому использованию виртуальных миров во Всемирной паутине. Впрочем, уже сегодня вы можете познакомить своих учеников с этой перспективной технологией, используя разработки компании Parallel Graphics, ранее известной под именем ParaGraph (адрес в Интернете: <http://www.paragraph.ru>).

Наиболее простыми в использовании и в то же время эффективными средствами создания трехмерных сцен в формате VRML являются конструкторы виртуальных миров *Virtual Home Space Builder (VHSB)* и *Internet Space Builder (ISB)*, выпущенные этой компанией соответственно в 1996-м и 1998 годах. Используя весьма несложный интерфейс VHSB, можно построить здание или лабиринт из прямоугольных блоков, развесить на стенах картины, использовать различные цвета и текстуры, применить эффект движущихся поверхностей. Возможности конструктора ISB существенно богаче: работа с широким диапазоном базовых трехмерных форм (сферы, призмы, пирамиды, конусы, цилиндры) и с галереей готовых объектов, редактирование и трансформация текстур, вставка в сцену трехмерных строк текста из любого алфавита, использование коллекции готовых текстур, картин и анимированных роликов. Поэтому и на освоение ISB приходится тратить несколько больше усилий и времени.

В литературе приведено немало примеров использования виртуальной реальности в самых разных областях человеческой деятельности: музыка, архитектура, виртуальные магазины, искусство, развлечения, путешествия, техника. Специфика нашего Центра, в котором есть лицейские художественно-графические классы, определила приоритетное использование пакетов для обучения Интернет-дизайну. Как показал опыт, для работы вполне достаточно компьютеров Pentium I с тактовой частотой 75 МГц и оперативной памятью 48 Мб. Конструктор VHSB осваивали учащиеся восьмых классов в базовом курсе информатики. Тема была продолжена в ходе выполнения индивидуальных и групповых творческих проектов. Примером такого проекта явилась виртуальная картинная галерея лучших работ школьных художников (см. рис. 1 и 2).

Изучение конструктора ISB предусмотрено для учащихся 8-х и 9-х художественно-графических классов в профильном курсе информатики и информационных технологий. Краткое руководство пользователя на русском языке, встроенное в программу, а также оформленное в виде отдельной брошюры [5], позволяет начать работу без предварительной подготовки. Руководство, являясь строгим техническим документом, составлено как хорошее методическое пособие. Разъясняется смысл основных понятий: “сцена”, “объект”, “форма”. Описаны элементы экрана: окна плана и перспективы, галереи готовых форм, объектов, текстур и т.д. (см. рис. 3, 4). Подробные примеры показывают последовательность действий по шагам при строительстве простого дома и просмотре трехмерной сцены. Документация, встроенная в программу, содержит дополнительные примеры. В разделе “Краткий перечень команд” приведено описание команд меню и панели инструментов.

Активно используя встроенное руководство, учащиеся в течение двух-трех уроков обучаются приемам работы с новым программным продуктом и изучают его возможности. Затем выполняют индивидуальные творческие задания.

В отдельных случаях программа устанавливается на домашние компьютеры для самостоятельной работы над конкурсными проектами.

Большое значение имеет выбор темы индивидуального задания. В качестве примеров виртуальных проектов можно назвать: воссоздание памятников архитектуры по имеющимся чертежам, школьное здание, старинные замки, летающая тарелка, фантастические миры, театральные декорации (см. рис. 5, 6).

Создание трехмерных объектов и разнообразных сцен способствует развитию пространственного воображения, художественной фантазии и творческих способностей учащихся. Трехмерные виртуальные миры могут использоваться в специальных средних и высших учебных заведениях при обучении студентов специальностям художественно-графического и архитектурного профиля, например, в курсе черчения и начертательной геометрии или для построения

объемного архитектурного коллажа (“встраивания” моделей зданий в различную среду). По сравнению с профессиональными программами, такими, как 3D-Studio, конструкторы виртуальных миров имеют более простой интерфейс и не столь требовательны к аппаратным средствам.

Литература

1. Усенков Д.Ю. Виртуальная реальность: кто напишет книгу? “Информатика” № 44, 1998.
2. The VRML 2.0 handbook: building moving worlds on the web. Jed Hartman and Josie Wernecke. Copyright 1996 by Silicon Graphics, Inc.
3. Якушина Е. Изучаем Интернет, создаем web-страничку. СПб.: Питер, 2000.
4. Эд Титтел и др. Создание VRML-миров: Пер. с англ. Киев: Издательская группа BHV, 1997.
5. Internet Space Builder. Краткое руководство пользователя. Copyright Parallel Graphics, 1998.

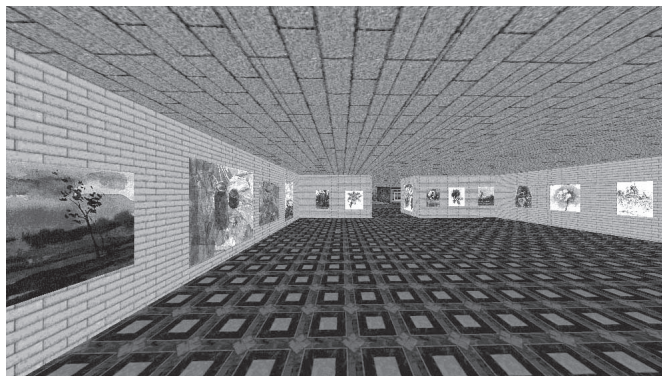


Рис. 1



Рис. 2

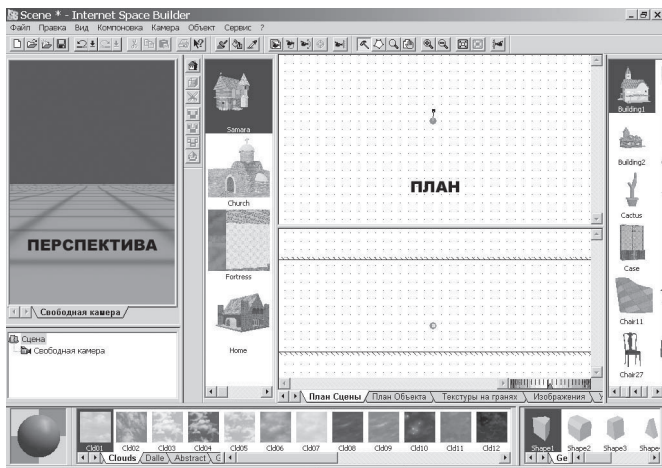


Рис. 3

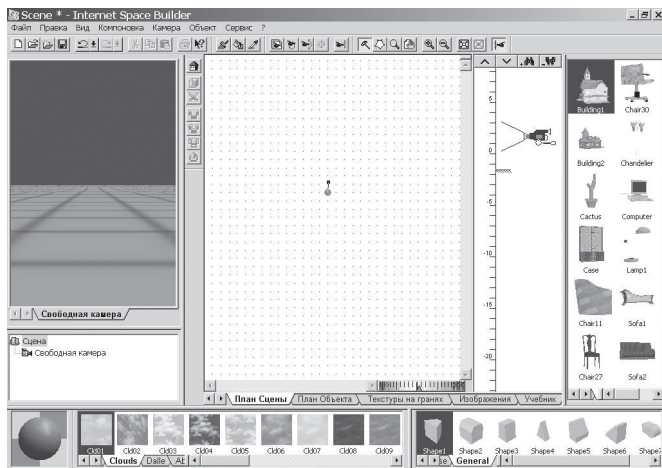


Рис. 4



Рис. 5



Рис. 6

Э Л Е К Т Р О Н Н А Я
И Л И К А К
Д О К А Т И Л С Я
Д О " С О Б А Ч Ь Е Й
Ж И З Н И "

Как известно, один из первых узлов ARPANET (см. "Очень краткую историю Интернета": "Информатика" № 9/2002) был образован в Станфордском исследовательском институте (SRI), где Дуг Энгельбарт (*Doug Engelbart*) руководил проектом "Наращивание человеческого интеллекта". Энгельбарт извещал еще двумя серьезными изобретениями. Он первый придумал компьютерную мышь и предложил гипертекстовую систему, идеи которой были в дальнейшем использованы при создании Всемирной компьютерной паутины. Кроме того, он разрабатывал для ARPANET программу общения между компьютерами.

Через месяц после подключения SRI из лаборатории Леонарда Клейнрока (*Network Measurement Center, UCLA*) туда было послано первое межкомпьютерное сообщение. Произошло это 29 октября 1969 года, и некоторые историки предлагают именно этот день считать днем рождения Интернета.

Тем не менее авторы "Краткой истории Интернета" (Дэвид Кларк, Роберт Кан, Леонард Клейнрок, Дэниэл Линч, Джон Постел, Лоуренс Робертс, Стефен Вулф, История Интернета, www.jetinfo.ru) считают, что электронная почта появилась в марте 1972 года. Отмечался же тридцатилетний юбилей в конце 2001 года, то есть рождение электронной почты отнесено к осени 1971 года. Почему возникли такие разногласия в датах? Дело в том, что Рэй Томлинсон (*Ray Tomlinson*), исследователь из американской компании BBN Technology, далеко не сразу написал и отладил базовые программы пересылки и чтения электронных сообщений. Напомним, что компания BBN Technology была одной из участниц проекта ARPANET — создания сети компьютеров, непосредственной предшественницы Интернета, — который велся по заказу министерства обороны США. Томлинсон как раз занимался разработкой почтовых программ и созданием виртуальных почтовых ящиков. Собственно, виртуальный почтовый ящик представлял собой файл, который отличался от обычного файла тем, что пользоваться им могли исправлять текст — только дограммы: SNDMSG — для отправления и READMAIL — для чтения.

Новая программа, которую написал Томлинсон, состояла из 200 строчек кода и представляла собой комбинацию SNDMSG, READMAIL и протокола CPYNET, использовавшегося в ARPANET для отсылки файлов на удаленный компьютер. Первое послание Томлинсона было отправлено с одного компьютера на другой, ставший в этой же лаборатории.

Вся работа над первой версией продолжалась полгода, и в рамках ее, естественно, проводилось промежуточное тестирование, то есть пересылка пробных сообщений. И лишь после этого программа стала активно использоваться для связи на самом деле удаленных компьютеров.

Результат усилий Томлинсона не был верхом совершенства, но годился для работы на первое время. Торжественного разрезания ленточки, банкета и пресс-релиза не было. Естественно, о произошедшем не сообщили дикторы CNN и никаких презентаций и награждений не было. О том, что у Томлинсона кое-что получилось, знали не больше нескольких сотен коллег, имевших доступ к "прасети".

Кроме того, творение Томлинсона было всего лишь промежуточным результатом длительной работы. Дело в том, что первую систему обмена текстовыми сообщениями до Томлинсона создал Дуг Энгельбарт. А Рэй Томлинсон придал ей вид почтового конверта с графами "Куда", "Кому" и самим текстом письма. Он ре-

шил также для удобства завести на каждом компьютере виртуальный почтовый ящик. Все это было реализовано в программе Send Message. Однако знакомый нам вид электронной почты приобрела только после серьезной доработки программы Лоуренсом Робертсом. Робертс предусмотрел просмотр списка всех писем, выборочное чтение нужного сообщения, сохранение письма в отдельном файле, пересылку другому адресу и возможность автоматической подготовки ответа.

И если уж непременно надо присваивать званию "отца электронной почты", то его в равной мере заслужил Энгельбарт, Томлинсон и Робертс. К сожалению, во время празднования тридцатилетнего юбилея электронной почты вовсе забыли Робертса, Энгельбарта вспомнили гораздо реже, чем он заслужил, а все лавры достались несколько озадаченному Томлинсону. Произошло это, видимо, потому, что имени он ввел в обиход всем теперь хорошо известный значок @.

История же знака @ берет начало якобы еще в средневековье, когда монахи — хранители древних знаний и рукописей занимались переводами и переписывали трактаты — в том числе и написанные на латыни. В латыни употребляется предлог "ad", что в переводе на современный английский означает "at" ("на", "в", "к") и указывает на принадлежность, направление и приближение. В шрифте, используемом монахами, буква "d" имела небольшую "хвостик", что делало ее похожей на цифру "6" в зеркальном отражении. Так "ad" довольно быстро превратилось в @.

В XV веке знак @ появляется вновь. Испанские купцы использовали этот знак в качестве сокращенного наименования меры весов — "agoba" (это приблизительно 11,52 кг, или 25,40 фунта). Любопытно, что эта мера использовалась для обозначения веса скота и вина.

В эпоху Возрождения @ стала использоваться для обозначения цены, а в эпоху индустриальной революции (время капитала, первых бирж, станков и так далее) @ стала неизменно встречаться в бухгалтерских отчетах. Так "собачка" вместе с \$, #, % и прочими "счетоводными" знаками с почти забытой семантикой перекочевала на клавиатуру.

"Собачка" мирно дождалась своего звездного часа, пока на нее случайно не упал взгляд Рая Томлинсона. Он использовал клавиатуру модели 33 Teletype, и в один прекрасный день он бросил на нее взгляд в поисках значка, котормени имени или названий, во-вторых, не мог бы встретиться ни в одном имени или названии, во-вторых, мог бы отделить имя пользователя от имени компьютера.

Помимо букв и цифр, на клавиатуре были и пунктуационные знаки, среди которых затесалась и "собачка". В те времена (до 1971 года, когда модель клавиатуры была изменена) подобные знаки располагались во втором ряду слева.

Символ @ удовлетворял всем требованиям Томлинсона. Как позже говорил сам Томлинсон, которого всю жизнь донимали журналисты, это был единственно возможный выбор. Говорят, именно в этот момент "собачка" явила светую свою морду, действи-

П О Ч Т А ,

тельно оказавшись простой, четкой и адекватной своей миссии.

Настоящее рождение @ пережила в 80-х, когда начались компьютерная революция — ПК вышли за пределы лабораторий, и в 90-х, когда появились первые web-браузеры. @ полюбилась пользователям, и даже рассказывают, что есть соответствующий дорожный знак.

Этимология слова "собачка" более запутанная. Считается, что только русские пользователи называют знак этим именем. Якобы @ по-корейски означает улитку, по-фински — спящую кошечку, по-венгерски — червяка, по-китайски — мышонка, по-шведски — булочку с корицей.

Несмотря на обрушившуюся на Томлинсона славу, он производит впечатление добродушного человека. Особенно пышные празднества с многочисленными интервью случились в связи с празднованиями 30-летия электронной почты осенью 2001 года, хотя очевидно, что его имя не мог появиться в один день, а "первое отправленное письмо" на самом деле было отправлено в пределах одного компьютера и вовсе не 30 лет назад.

История @ — это еще и забавная эпопея, связанная с содержанием первого послания. Существуют две легенды по этому поводу.

Первая гласит, что Томлинсон набрал QWERTYUIOP — верхний ряд букв слева направо в английской раскладке. По этому поводу на сайте "Радио Свобода" сказано: "Журналисты стали настойчиво добиваться. Рэй, человек не публичной профессии, не понял, что может сейчас сказать историческую фразу вроде: "В первом электронном письме было написано: "Мы верим в Бога" или "Я тебя люблю".

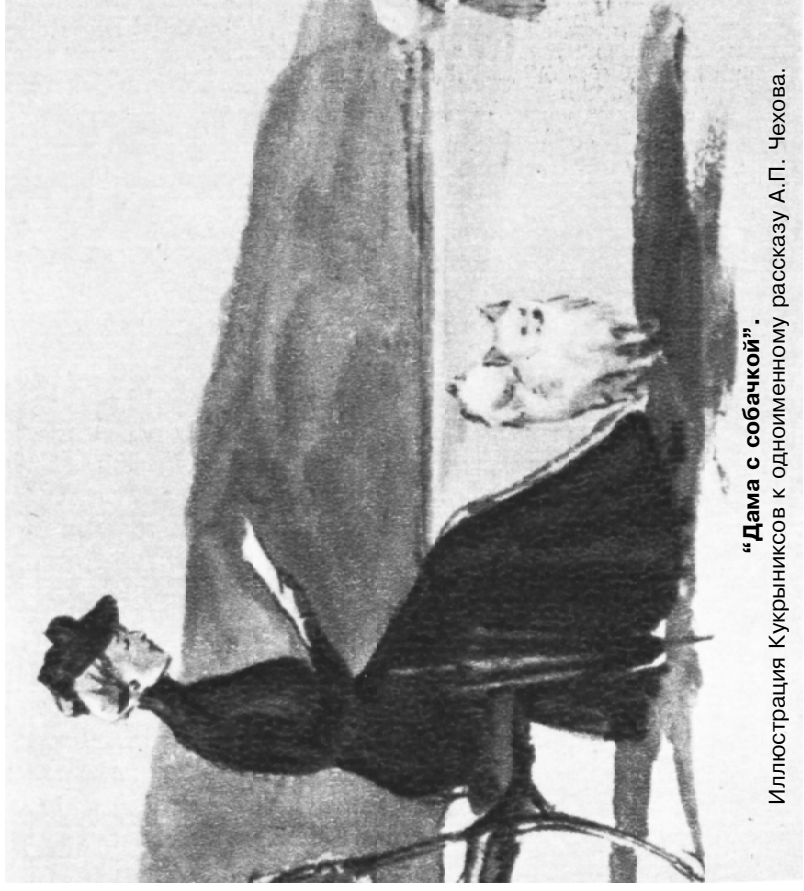
Он честно признался, что просто набрал на клавиатуре первые попавшиеся символы, ведь тогда он еще не мог знать, что письмо — историческое. Но журналисту для статьи нужны инстинкты. Не очень сильно звучит "ученый так и не вспомнил, что было в первом электронном письме". Поэтому посмотрел репортер на свою клавиатуру и нашел в верхнем ряду запоминающийся ряд главных английских букв — QWERTYUIOP". С тех пор инженер придерживается версии QWERTYUIOP и предпочитает не спорить.

По второй версии, Томлинсон заявил, что написал цитату из геттисбургской речи Авраама Линкольна по поводу открытия нового кладбища жертв гражданской войны 19 ноября 1863 года.

Считается, что во время первого телефонного сеанса связи Белл сказал в первую телефонную трубку своему ассистенту: "Ватсон, зайди, вы мне нужны". Обыденно и тривиально. А действительно, о чем думается в "исторические моменты"? Ну уж точно не о Линкольне...

Использованные источники информации

1. Как знак @ докатился до "собачьей жизни". www.membrana.ru.
2. Александр Костинский. 30-летие электронной почты. www.svoboda.org.



"Дама с собачкой".
Иллюстрация Кукрыникова к одноименному рассказу А.П. Чехова.

Создание модели соударения шариков с помощью электронной таблицы

О.В. Туркин,
Москва

В этой статье речь пойдет о создании компьютерной модели абсолютно упругого удара. Легко понять, что построение модели движения соударяющихся тел с использованием графики в любом языке программирования, даже в среде Visual Basic, достаточно сложная задача.

Использование электронной таблицы со встроенными средствами программирования, например Microsoft Excel с языком Visual Basic for Applications, позволяет, во-первых, осуществлять часть расчетов средствами самой электронной таблицы с оформлением диаграмм, а во-вторых, легко конструировать интерфейс программы с помощью различных свойств рабочего листа.

Итак, сначала о самой модели.

Как известно из школьного курса физики, при абсолютно упругом ударе двух тел массой M_1 и M_2 , движущихся соответственно со скоростями V_1 и V_2 , выполняются законы сохранения импульса

$$M_1 V_1 + M_2 V_2 = M_1 U_1 + M_2 U_2$$

и закона сохранения энергии

$$\frac{M_1 V_1^2}{2} + \frac{M_2 V_2^2}{2} = \frac{M_1 U_1^2}{2} + \frac{M_2 U_2^2}{2},$$

где U_1 и U_2 — скорости тел после соударения.

Совместное решение двух этих уравнений позволяет определить сначала значение скорости U_1

$$U_1 = \frac{(M_1 - M_2)V_1 + 2M_2 V_2}{M_1 + M_2}, \quad (1)$$

а затем и U_2

$$U_2 = V_1 + U_1 - V_2 \quad (2)$$

В нашей модели с помощью вышеприведенных формул будут рассчитываться координаты шариков, движущихся по одной прямой. Для того чтобы соударения происходили многократно, движения шариков ограничены боковыми стенками, при ударе о которые шарики изменяют свою скорость по направлению, сохраняя ее по величине.

Зададим необходимые для вычисления параметры в ячейках электронной таблицы.

Воспользуемся возможностью дать ячейкам уникальные имена. Для этого в первой строке, в ячейках A1:D1 соответственно, набираем имена ячеек, чтобы обозначить массы и начальные скорости каждого из шаров, а именно: mmm1, vvv1, mmm2, vvv2. После этого, выделив блок A1:D4, используем пункты меню “Вставка | Имя | Создать | строке ниже” и присваиваем ячейкам A2:D2 эти имена. Сразу, чтобы не запутаться в дальнейшем, можно в ячейки A1:D1 ввести пояснения, например, “масса1”, “скорость1” и т.д. (см. рисунок на с. 19).

В ячейки второй строки этого блока нужно ввести числовые значения. Массы шариков целесообразно выбрать в диапазоне от 1 до 10 кг, а скорости — в диапазоне от -10 до $+10$ м/с.

Теперь оформим блок E1:F2. В его вторую строку занесем начальные координаты шариков. Будем считать, что расстояние между ограничивающими стенками равно 10 метрам и, кроме того, шар 1 располагается левее шара 2, поэтому значение начальных координат равно соответственно 2 и 6. В ячейку G2 помещаем формулу — разность начальных координат первого и второго шариков. На этом создание области ввода закончено. Ее можно дополнительно оформить с помощью различных методов форматирования, снабдить примечаниями и т.д.

Переходим к созданию вычислительной области. Пусть эта область — A5:D5. Вписываем формулы — в A5 заносим =mmm1, в C5 — =mmm2. В четвертой строке (ячейки A4 и C4) поместим комментарии — “Первый шар”, “Второй шар”. В ячейках B5 и D5 в дальнейшем будут отображаться координаты движущихся шариков. Пока туда можно записать те же значения, что содержатся в E2 и F2. Теперь все готово для построения диаграммы.

Построим диаграмму, в которой положение шариков будет зависеть от координат (то есть от значений в ячейках B5 и D5), а размеры — от массы (значения в ячейках A5 и C5). Для этого необходимо выбрать пузырьковую диаграмму.

Выделим диапазон A5:B5 и начнем работу с мастером диаграмм.

На первом шаге работы мастера выбираем тип диаграммы — пузырьковая.

На втором шаге, в окне “Ряд”, определяем первый ряд (окно “Диапазон данных” пропускаем), для которого указываем значения X: =Лист1!\$B\$5, значения Y: 2 (вертикальная координата у шариков постоянна и меняться не будет) и размеры — =Лист1!\$A\$5. Сформировав ряд 1, добавляем ряд 2 с параметрами значения X: =Лист1!\$D\$5, значения Y: 2, размеры — =Лист1!\$C\$5.

На третьем шаге можно поместить заголовок диаграммы, отказаться от легенды и убрать ось Y.

На четвертом шаге размещаем диаграмму на “имеющемся листе 1”.

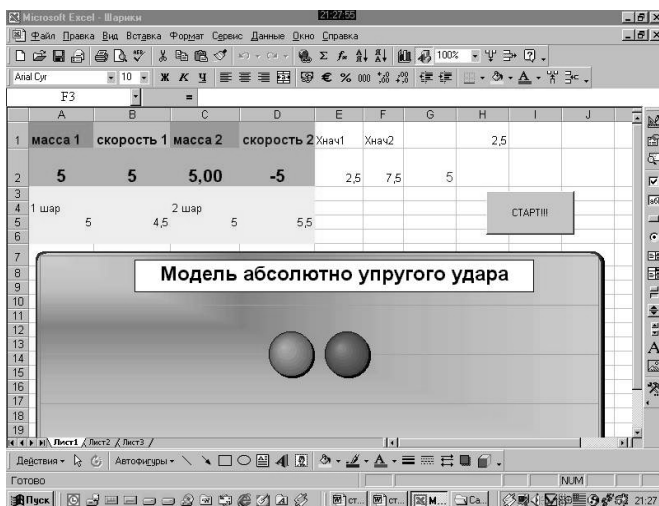
Диаграмма с изображением двух шариков готова. Используя возможности контекстного меню для диаграмм (области диаграммы, области построения и так далее), оформляем диаграмму по своему вкусу, но для оси X необходимо установить минимальное и максимальное значения шкалы (соответственно 0 и 10), отказав-

шись от режима “Авто” для этих параметров. На диаграмму можно поместить объект “Надпись” с соответствующим тестом (как на рисунке).

Остановимся на этом этапе. До сих пор мы не обращались к языку программирования. В этом не было необходимости. Меняя вручную значения координат шариков или их массы в расчетном блоке (строка 5 таблицы), можно заставить шарики перемещаться по диаграмме. Но для того чтобы заставить шарики двигаться в автоматическом режиме, нужно написать небольшую программу.

Создадим управляющую (командную) кнопку. Для этого в пункте меню “Вид | Панели инструментов” отметим “Элементы управления”. На появившейся панели элементов управления (ЭУ) нажимаем иконку “режима конструктора”, находим иконку “Кнопка” и размещаем этот элемент управления на нашем рабочем листе.

Далее, с помощью иконки “Свойства” на панели ЭУ открываем окно свойств этой самой “Кнопки” и изменяем внешний вид кнопки. Например, свойство `Caption` означает надпись на элементе. Заменяем стандартную надпись “CommandButton1” на слово “Старт”, также можно изменить стиль и цвет фона (`BackColor` и `BackColor`), тип, размеры и цвет шрифта (`Font` и `ForeColor`). При этом сам элемент “Кнопка” должен быть выделен как объект. Вообще в области среды программирования VBA, в которой мы оказались, действуют свои правила, и надо быть достаточно осторожным. Скажем, свойство `Name` лучше не менять.



Итак, кнопка имеет надпись “Старт”, окрашена выбранным цветом, теперь нужно заставить ее заработать: шарики должны двигаться по диаграмме и при столкновении менять свои скорости в соответствии с нашими формулами.

Начинаем составлять программу. Дважды щелкнем по нашей красивой кнопке и начнем работу в окне исходного текста Visual Basic. В этот редактор также можно попасть через панель ЭУ (иконка “Исходный текст”), через ярлычок рабочего листа (та же иконка в контекстном меню) или через “Меню | Сервис | Макрос | Редактор...”.

В окне уже есть две строки:

```
Private Sub CommandButton1_Click()
End Sub
```

В переводе — “локальная процедура (подпрограмма) для обработки щелчка по CommandButton1” и “конец процедуры”. Между этими строчками и будет размещаться наш код. В первую очередь позаботимся о том, чтобы данные из ячеек ЭТ попали в переменные программы. Нам нужны начальные координаты шариков. Это ячейки E2 и F2 (строка 2, столбцы 5 и 6), поэтому две первые строчки будут выглядеть так:

```
xxn1 = Cells(2, 5)
xxn2 = Cells(2, 6)
```

В этих двух строчках происходит передача данных из ячеек (объекты из коллекции `Cells`) листа в переменные программы. Конкретные ячейки определяются через номер строки и номер столбца.

Теперь определимся со скоростями. Начальные скорости расположены в ячейках B2 и D2. В программе мы их обозначим через переменные `dx1` и `dx2`. Координату шарика через промежуток времени t мы будем вычислять в соответствии с известной формулой физики

$$X = X_0 + V \cdot t$$

Полагаем $t = 1/10$, и тогда $dx = V/10$. Получаем следующие две строчки:

```
dx1 = Cells(2, 2) / 10
dx2 = Cells(2, 4) / 10
```

Очередные две строчки — это передача начальных координат в текущие координаты — переменные `x1` и `x2`, которые будут использованы в цикле `x1 = xxn1`

```
x2 = xxn2
```

Кроме того, в программе понадобятся и массы. Они располагаются в ячейках A2 и C2, поэтому

```
m1 = Cells(2, 1)
m2 = Cells(2, 3)
```

Теперь организуем цикл со стократным повторением, считая, что одно выполнение цикла соответствует $1/10$ с, то есть процесс происходит в течение 10 условных секунд. При выбранных скоростях и расстояниях такое значение можно считать оптимальным.

```
For i = 1 To 100
```

Позаботимся, чтобы шарики не вылетали за стенки. Первый шарик не должен улететь налево, и если он приблизился к левой стенке (ее координата равна 0), то величина его скорости должна поменять знак. Кроме того, чтобы зрительно шарик не пересекал стенку, запретим ему приближаться к ней ближе чем на 0,5. Поэтому

```
If x1 <= 0.5 Then dx1 = Abs(dx1)
```

Для второго шарика опасна правая стенка с координатой 10, и с учетом его размеров получаем:

```
If x2 >= 9.5 Then dx2 = -1 * Abs(dx2)
```

Следующий шаг — это процесс соударения. Когда первый шарик (левый) приблизился ко второму менее чем на 1 (размеры шариков), то скорости шариков меняются в соответствии с формулами (1) и (2).

Попутно заметим, что если мы работаем в системе СИ, то шарики у нас — метровые гиганты. Вот такой супербоулинг!

```
If x1 >= x2 - 1# Then
a = dx1 'это скорость V1 перед ударом,
dx1 = (dx1 * (m1 - m2) + 2 * dx2 * m2) / _
      (m1 + m2) 'скорость U1
dx2 = a + dx1 - dx2 'скорость U2
End If
```

Итак, значения скорости тел в случае соударений со стенками или друг с другом определены, теперь найдем новые координаты через очередные 1/10 с.

```
x1 = x1 + dx1
x2 = x2 + dx2
```

Отправим эти значения в ЭТ, чтобы изменить диаграмму. Вспомним, что положение шариков в диаграмме определяется ячейками B5 и D5:

```
Cells(5, 2) = x1
Cells(5, 4) = x2
```

Следующая строка кода очень важна. Она дает команду электронной таблице произвести перерасчет в соответствии с новыми данными, а также перестроить диаграмму (перевод команды — “делай события”).

```
DoEvents
  Завершаем цикл:
Next i
```

Полностью код выглядит следующим образом.

```
Private Sub CommandButton1_Click()
xxn1 = Cells(2, 5)
xxn2 = Cells(2, 6)
dx1 = Cells(2, 2) / 10
dx2 = Cells(2, 4) / 10
x1 = xxn1
x2 = xxn2
m1 = Cells(2, 1)
m2 = Cells(2, 3)
For i = 1 To 100
If x1 <= 0.5 Then dx1 = Abs(dx1)
If x2 >= 9.5 Then dx2 = -1 * Abs(dx2)
If x1 >= x2 - 1# Then
a = dx1
dx1 = (dx1 * (m1 - m2) + 2 * dx2 * m2) / _
      (m1 + m2)
dx2 = a + dx1 - dx2
End If
x1 = x1 + dx1
x2 = x2 + dx2
Cells(5, 2) = x1
```

```
Cells(5, 4) = x2
DoEvents
Next i
End Sub
```

Теперь вернемся в электронную таблицу, например, через значок таблицы в левом верхнем углу окна. Иконка “Режим конструктора” должна быть отжата, а ячейки во второй строке (начальные значения) должны быть заполнены. Не забудьте убрать выделение с диаграммы. Щелкаем по кнопке “Старт” и любуемся.

Если программа не работает, то в Office2000 это может быть связано с режимом безопасности. Заходим в “Меню | Сервис | Макрос | Безопасность” и выбираем средний уровень безопасности. Сохраняем рабочую книгу, выходим из приложения и запускаем эту рабочую книгу опять, но теперь при загрузке на вопрос о включении макросов отвечаем, что макросы должны быть включены.

Проверить программу можно следующим образом: при равенстве масс тел и одинаковых по абсолютному значению, но противоположно направленных скоростях должна получиться симметричная картинка. Задаем начальные значения — массы по 5, скорости 5 и —5, начальные значения — 2,5 и 7,5. Получаем достаточно знакомую картину соударения двух одинаковых шаров с одинаковыми скоростями. Именно такой вариант изображен на рисунке.

Кроме того, во время работы программы можно заметить, что в ячейках пятой строки меняются значения. Это изменяются координаты шариков. При желании можно вывести в какие-нибудь ячейки и значения скоростей. Таким образом, ячейки таблицы играют роль ячеек памяти и позволяют наблюдать за изменением значений переменных, что полезно для анализа работы программ или ее отладки.

Как видим, для создания движущейся модели понадобилось не так уж и много времени, и за 2 часа с классом средней подготовки ее вполне можно реализовать. И еще раз хочется обратить внимание на то, что программный код был использован только тогда, когда появилась необходимость в динамических расчетах. Такое продуманное использование VBA действительно открывает широкие возможности для творчества.

В следующий раз мы рассмотрим проект, в котором строятся одновременно 3 графика изопроецессов для идеальных газов с возможностью иллюстрации замкнутых газовых циклов. Эти процессы иллюстрируются диаграммой изменения количества теплоты, внутренней энергии и работы.

Азы информатики

А.А. ДУВАНОВ

Материалы
Роботландского
университета

Книга 1. См. № 1, 2/2002
Книга 2. См. № 5, 6, 7, 8, 9, 11/2002

Демо-версию гипертекстовой книги (700 Кб) можно скопировать с адреса:
<ftp://ftp.botik.ru/rented/robot/univer/azinfd.zip>

Книга 2. В мире информации (продолжение)

Книга для ученика

7. ИСКАЖЕНИЯ ПРИ ПЕРЕДАЧЕ

Искажения при передаче
Читальный зал Роботландии

РОБОТЛАНДИЯ.RU © А.А.ДУВАНОВ

© 2001

Ошибки при передаче бывают трех типов: неверный, лишний или пропущенный сигнал.



Читальный зал

Искажения при передаче информации

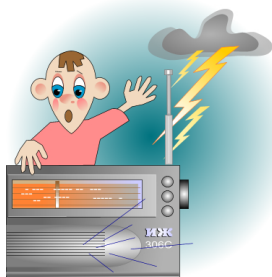
Очень часто при передаче информации возникают помехи. И тогда информация от источника к приемнику поступает в искаженном виде.

История 1. Радио и гроза

На дворе бушевала гроза, сверкали молнии, гремел гром. Но Васе очень хотелось послушать любимую радиопередачу. Он включил радио.

Из репродуктора понеслись рычания, треск, щелчки, мешающие услышать музыку и речь.

— Информация искажена, — сказал папа, — гроза создает помехи. Лучше выключи радио.



История 2. Событие в тридевятом королевстве

Вот такая история произошла при дворе грозного короля, который захотел казнить провинившегося слугу. Слуга просил пощады. Король гневно продиктовал приказ: “Казнить, нельзя помиловать!”.

Но, перенося приказ на бумагу, королевский писарь ошибся: поставил запятую в другом месте. Получилось: “Казнить нельзя, помиловать!”.

Министр, который получил это сообщение, вместо того чтобы выполнить суровый королевский приказ, отпустил слугу. Информация исказилась к счастью для слуги.



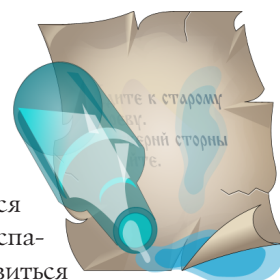
История 3. Путешествие вокруг света

Если вы еще не читали замечательную книгу Жюль Верна “Дети капитана Гранта”, прочитайте ее обязательно. Она вам понравится.

Морские волны передали людям информацию — записку в бутылке. Послали ее потерпевшие кораблекрушение, бросив бутылку в море.

К сожалению, вода размыла часть слов и тем самым исказила информацию.

Вместо того чтобы двинуться сразу к месту гибели корабля, спасатели вынуждены были отправиться



в кругосветное путешествие. Им самим пришлось испытать немало бедствий.

История 4. Телефонный разговор



Вася позвонил приятелю Коле, чтобы спросить, как тот решил задачу по информатике. Между ребятами произошел странный разговор.

— Алло, Коля! Ты сделал задание по информатике?

— Алло, кто говорит? Здесь нет Оли! Вы ошиблись номером!

— Алло, это я, Вася Кук! Ты решил задачу по информатике?

— Алло, какой еще “майский жук”? Какая такая “зачапка по шватике”? Вас очень плохо слышно, перезвоните!

Приятели так и не сумели понять друг друга, а все потому, что от повреждения телефонного кабеля на линии возникли помехи.

Типы ошибок передачи

Ошибки, возникающие при передаче информации, подобно ошибкам набора на клавиатуре компьютера, бывают трех типов:

1. Часть передаваемой правильной информации замещается на неверную.

Например, вместо слова “бидон” было получено слово “батон”.

2. К передаваемой информации добавляются лишние, посторонние сообщения. Например, вместо слова “кран” было получено слово “экран”.

3. Часть информации при передаче пропадает. Например, вместо слова “колбаса” было получено слово “колба”.

При передаче сообщений ошибки могут появляться в самой разной комбинации указанных выше типов.

Ошибки передачи данных в компьютере

Помехи могут возникнуть при передаче информации внутри компьютера.

Например, это может произойти при внезапном скачке напряжения в электрической сети. Программы начнут работать неверно или даже совсем перестанут реагировать на клавиатуру и мышь. Говорят: компьютер “повис”.

Когда информация пересылается по компьютерной сети, тоже могут возникнуть ошибки передачи. Приходится повторять пересылку. К счастью, это бывает не так часто.

Как борются с искажениями

Если вы пишете тест, внимательно следите за запятыми, не допускайте других грамматических ошибок. Одна

запятая, одна буква и даже один пробел могут совершенно исказить смысл вашего сообщения.

Вот в этом сообщении пропущена всего одна буква:

Предать котенка.

Как вы думаете, какая буква пропущена?

История 5. Передача чисел



Изобретатель Листик придумал устройство для передачи чисел. Его прибор передавал сообщения в виде цепочки коротких и длинных сигналов. В своих записях Листик обозначал короткий сигнал цифрой “0”, а длинный — цифрой “1”. При передаче чисел он использовал для каждой цифры следующий код:

Цифра	Код	Сигналы
0	0000	4 коротких сигнала
1	0001	3 коротких и длинный
2	0010	2 коротких, длинный, короткий
3	0011	2 коротких и 2 длинных
4	0100	короткий, длинный, 2 коротких
5	0101	
6	0110	
7	0111	
8	1000	
9	1001	

Число 12, например, Листик записывал для передачи так:
00010010

Аппарат передавал это сообщение цепочкой таких сигналов: три коротких, один длинный, два коротких, один длинный и один короткий.

Число 77 по системе Листика кодировалось так:
01110111



Аппарат передал это сообщение при помощи восьми длинных сигналов.

Другой аппарат, который принимал сообщения, посланные Листиком, каждую группу из четырех сигналов превращал в обычную цифру, и все сначала работало прекрасно.

Через некоторое время Листик заметил, что иногда при передаче возникают искажения: длинный сигнал приемником определяется как короткий, а короткий — как длинный.

Тогда Листик решил передавать каждый сигнал три раза. Приемник получал эти копии и оставлял то значение, которое встречалось чаще. Так, например, сигналы “010” считались сигналом “0”, а сигналы “110” — сигналом “1”.

Таким образом, Листик защитил передачу информации от помех. Правда, ему приходилось передавать в три раза больше сигналов. Так, если раньше число 5 он передавал как короткое сообщение:

0101

то теперь ему приходилось передавать длинную цепочку сигналов:

000111000111



Конспект

Три типа ошибок передачи информации:

- Неверные части.
- Лишние части.
- Отсутствующие части.



Вопросы

1. Приведите примеры искажения информации при передаче.
2. К чему могут привести ошибки передачи данных внутри компьютера?
3. Какие типы искажений возникают при передаче?
4. К каким типам искажений вы отнесете каждый из пяти примеров Читального зала?
5. Расскажите, как Листик кодировал числа для передачи.
6. Расскажите, как приемник Листика преобразовывал полученные сигналы в числа.
7. Какую защиту от помех придумал Листик?
8. Листик сначала хотел посылать по каналу связи не три копии сигнала, а только две, но потом понял, что так защититься от помех нельзя.
Объясните, почему.
9. Всегда ли надежна защита Листика? В каких случаях она не работает?



Задания на дом



Вариант 1

1. Никита, выслушав учителя, записал в тетради:
 - 1) Клавиатура выводит буквы, цифры и другие символы в компьютер.
 - 2) Монитор служит для ввода информации.
 - 3) Жесткий диск (или винчестер) — основная память компьютера.
 Исправьте допущенные Никитой ошибки и объясните, какие типы искажений произошли при передаче информации от учителя в тетрадь Никиты.
2. Происходит ли искажение информации в следующих примерах:
 - 1) Учитель записал на доске пример:

$$I + II = III$$
 Ученик записал в тетради:

$$1 + 2 = 3$$
 - 2) Переводчик, услышав от англичанина: “Hi!” — перевел встречающим:

“Привет!”

 Что же происходит с информацией в этих примерах?



Вариант 2

1. Выполните алгоритм:
 - 1) Включите домашний компьютер.
 - 2) Запустите программу *Блокнот*.
 - 3) Переключите указатель алфавитов на русский.
 - 4) Прижмите клавишу **Alt**.
 - 5) Не отпуская **Alt**, наберите на дополнительной цифровой клавиатуре цифры 0192.
 - 6) Отпустите клавишу **Alt**.
 - 7) Повторите пункты 4—6, набирая на цифровой клавиатуре последовательно:

0193, 0194, 0195, 0196.

Что появилось в рабочем поле *Блокнота*? Искажает ли компьютер вводимую вами информацию или происходит какой-то другой процесс?

2. Выясните, как с помощью дополнительной клавиатуры написать в *Блокноте* слово, пропущенное в стихотворении:

И компьютеры порой
Говорят между собой,
Но для этого одна
Им штуковина нужна —
Вещь, известная не всем!
Называется — ?.
К телефону подключил —
Сообщенье получил!
В.Агафонов. “Твой друг — компьютер”.



Вариант 3

1. Искажена ли информация в следующем стихотворении:

Необыкновенная девочка

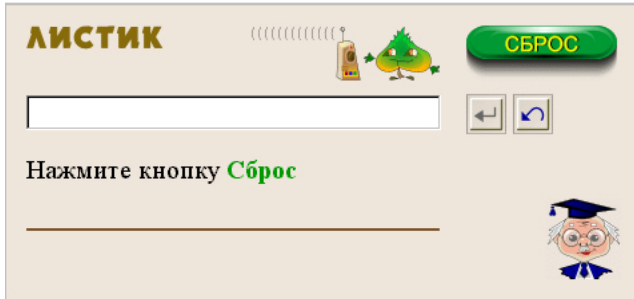
Ей было тысяча сто лет,
Она в сто первый класс ходила,
В портфеле по сто книг носила —
Все это правда, а не бред.
Когда, пыля десятком ног,
Она шагала по дороге,
За ней всегда бежал щенок
С одним хвостом, зато стоногий.
Она ловила каждый звук
Своими десятью ушами,
И десять загорелых рук
Портфель и поводок держали.
И десять ярко-синих глаз
Рассматривали мир привычно...
Но станет все совсем обычным,
Когда поймете мой рассказ.

2. Придумайте собственный необычный рассказ или стихотворение, описывающее обычные события без искажения информации.



Практикум

Код Листика



1. Листик вел передачу своим кодом без защиты от помех. Какое число было передано? Запишите ответы для всех заданий исполнителя.

- | | |
|--------------|--------------|
| 1) 0010 0110 | 4) 0011 0011 |
| 2) 0000 0111 | 5) 0001 0110 |
| 3) 1001 1000 | |

2. Зашифруйте числа кодом Листика.

- | | |
|-------|-------|
| 1) 50 | 4) 31 |
| 2) 21 | 5) 78 |
| 3) 79 | |

3. Листик использовал тройное повторение каждого сигнала при передаче. По полученному сообщению восстановите переданное число. Канал передачи работает с искажениями.

- | | |
|--------------------|--------------------|
| 1) 100 001 001 101 | 4) 100 110 011 101 |
| 2) 100 001 011 101 | 5) 101 001 001 110 |
| 3) 001 101 100 001 | |



Зачетный класс

1. Укажите тип искажения информации в следующих фрагментах стихотворений Агнии Барто.

- Уронили кошку на пол,
Оторвали кошке лапу.
- Наша Таня громко плачет:
Уронила мячик.
- Зайку бросила плохая хозяйка —
Под дождем остался зайка.

2. Укажите тип искажения информации в следующих фрагментах стихотворений Корнея Чуковского.

- Муха, Муха-Цокотух,
Позолоченное брюхо!
Вдруг навстречу мой хороший,
Мой любимый Крокодил Вася.
Он с Тотошей и Кокошей
По аллее проходил важно
И мочалку, словно галку,
Словно галку, очень быстро проглотил.
- Гуси начали опять
По-гусиному кричать:
Гы-гы-гы!

3. Листик вел передачу своим кодом без защиты от помех. Какое число было передано?

0111 1001

4. Зашифруйте число кодом Листика.

42

5. Листик использовал тройное повторение каждого сигнала при передаче. По полученному сообщению восстановите переданное число. Канал передачи работает с искажениями.

100 110 011 100 100 110 011 011

Книга для учителя

7. ИСКАЖЕНИЯ ПРИ ПЕРЕДАЧЕ

Игра в "испорченный телефон"

Готовя класс к теме "Искажение информации", учитель просит школьников назвать примеры передачи информации. Такую работу они уже выполняли, поэтому примеров будет много. Следует, подхватив один из примеров, показать, как при этой передаче информация может быть искажена.

Пример на передачу информации с возможными искажениями — игра "Испорченный телефон". Хорошо, если идею игры выдвинут дети. Тогда учителю останется только поддержать это начинание и организовать игру.

Слишком длинная цепочка (весь класс) не очень удобна.

С другой стороны, нельзя делить класс на играющих и не играющих, так как зрелищная сторона "Испорченного телефона" не впечатляюща. Поэтому ребят надо поднять из-за парт и построить в две цепочки (даже в три, если детей много). Правила игры можно детям напомнить, хотя обычно школьники с этой игрой знакомы очень хорошо. Начинает игру учитель.

Единое начальное слово для двух (и тем более трех) цепочек придаст игре дополнительный интерес. Важно поддержать высокий темп игры.

Комментируя ход игры и поведение игроков, полезно использовать терминологию информационных процессов — "информация", "передача", "источник", "приемник", "канал передачи", "помехи".

Код Листика

После игры дети подготовлены к рассказу о важности предохранения информации от искажений при передаче.

Нужно обязательно показать конкретный способ защиты информации и поработать с ним на практике. В качестве такого примера Читальный зал предлагает защитный код Листика. Защищаясь от помех, Листик передает длинные сигналы (код "1") и короткие (код "0") с утроением. Перевод каждой тройки в один сигнал решается "голосованием" кодов: сигнал считается длинным, если в тройке больше единиц, и коротким, если в тройке больше нулей.

Тройки

111 011 101 110

переводятся в длинные сигналы (код "1"), а тройки

000 100 010 001

переводятся в короткие (код "0").

Кодирование сообщений по методу Листика подкреплено заданиями Практикума и вопросами Зачетного класса.

При выполнении заданий на расшифровку искаженного сообщения можно использовать записи в тетради. Сначала каждая тройка преобразуется в одну цифру двоичного кода (и это записывается в тетради), а затем выполняется перевод числа в обычную десятичную запись.

Например, дети видят на экране пример кода, поступившего в приемник:

010 101 111 011 100 000 110 111

В тетради появляется запись:

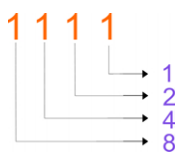
0111 0011

И, наконец, эта запись преобразуется в число 73.

В основе помехоустойчивого кода Листика лежит известное в практической информатике двоично-десятичное кодирование (каждой десятичной цифре соответствуют четыре бита).

Можно рассказать детям простое правило перевода двоичного кода в десятичную цифру.

Единица в коде на первом месте дает число 8, на втором — 4, на третьем — 2, на четвертом — 1. Для получения десятичной цифры числа складываются. Например, код “1001” переводится в цифру 9 (сумма чисел 8 и 1).



Этим же правилом можно пользоваться и при декодировании. Например, цифра 6 записывается как сумма чисел 4 и 2, значит, ее код будет “0110”.

С двоичным кодированием дети познакомятся более подробно в уроке 10 “Кодирование информации”.

Игра в кодирование по методу Листика

В коды Листика можно поиграть в классе без компьютера.

Дети разбиваются на пары (“передатчик”, “приемник”). Учитель каждой паре выдает список чисел для передачи. Передатчик кодирует числа и передает листочек с кодом приемнику. Приемник расшифровывает число, после чего партнеры сверяют результаты и ищут ошибки. Затем дети меняют свои функции: “приемник” становится “передатчиком”, а “передатчик” — “приемником”.

Для моделирования передачи с помехами нужны три человека: “передатчик”, “приемник” и “канал”, который будет вносить допустимые искажения (не более одного искажения в каждой тройке сигналов).

Ответы на вопросы

Ответ на вопрос 1. Помехи в радиосвязи: прием радио- и телевизионных передач, мобильные телефоны, оперативная связь. Помехи возникают из-за шумов космоса, погодных условий, промышленных радиоизлучений, из-за экранирующих препятствий (железобетонная плита), особенностей прохождения волн (передача на очень коротких волнах может вестись только по прямой, короткие волны почти не огибают земную поверхность), из-за наложения отраженных волн на основной сигнал, из-за собственных шумов приемника.

Помехи приводят к посторонним звукам в громкоговорителе при приеме речевых и музыкальных передач,

опечаткам при приеме текста телеграмм, искажениям изображения на экране телевизора.

Помехи в проводной связи: телефонный разговор и передача данных по модему в Интернете. Помехи обусловлены внешними электромагнитными воздействиями, погодными условиями, качеством линий, повреждениями проводов. Помехи проявляются в виде шорохов, тресков, в плохой разборчивости речи абонентов и слышимости разговоров, в виде ошибок при передаче данных.

Передача данных внутри компьютера так же подвержена помехам, например, во время скачков напряжения электрической сети или мощного электромагнитного излучения вблизи компьютера. Могут быть и внутренние причины: ошибки в микросхемах компьютера и его программном обеспечении.

Если в передаче участвует человек, то он может внести свои собственные ошибки, которые накладываются на ошибки, возникающие в каналах связи. Например, оператор отправляет телеграмму по каналу с ошибками, допущенными при наборе текста на клавиатуре.

Ответ на вопрос 2. Компьютер может “повиснуть”, то есть перестать реагировать на действия пользователя (через мышь, клавиатуру).

Ответ на вопрос 3. Ошибки, возникающие при передаче информации, подобно ошибкам набора на клавиатуре компьютера, бывают трех типов:

- Часть передаваемой правильной информации заменяется на неверную.
- К передаваемой информации добавляются лишние, посторонние сообщения.
- Часть информации при передаче пропадает.

Ответ на вопрос 4.

1) Радио и гроза. Часть передаваемой информации заменяется на неверную.

2) Событие в тридевятом королевстве. Пропал символ (правильная запятая), появился лишний (запятая на неверном месте).

3) Путешествие вокруг света. Часть информации заменилась на неверную (пустые места, которые появились от воздействия воды).

4) Телефонный разговор. Возможны искажения всех трех типов, но преобладает замена правильного сигнала на неверный.

5) Передача чисел. Замена верного кода на неверный.

Ответ на вопрос 5. Каждую десятичную цифру исходного числа Листик передавал при помощи четырех сигналов, комбинируя короткие (код “0”) и длинные сигналы (код “1”) по следующей кодовой таблице:

Код Листика	Десятичная цифра
0 — 0000	0
1 — 0001	1
2 — 0010	2
3 — 0011	3
4 — 0100	4
5 — 0101	5
6 — 0110	6
7 — 0111	7
8 — 1000	8
9 — 1001	9

Ответ на вопрос 6. Приемник принимал сигналы четверками. Каждую четверку он переводил в одну десятичную цифру, используя таблицу Листика.

Ответ на вопрос 7. Листик решил каждый сигнал, посылаемый по каналу связи, утраивать.

Таким образом, для передачи одной десятичной цифры теперь требовались не четыре сигнала, а двенадцать. Приемник, наоборот, каждые три сигнала заменял одним, тем, который в тройке встречался чаще.

Ответ на вопрос 8. Если в пришедшей двойке сигналы разные, то непонятно, какой из них считать правильным. Для того чтобы результат “голосования” сигналов был однозначным, нужно, чтобы их число было нечетным.

Можно, конечно, передавать сигналы, повторяя каждый из них пять, семь или большее число раз. Если канал передачи работает с сильными помехами, то так, вероятно, и надо делать. Но надо не забывать, что каждое повторение сигнала увеличивает общий объем передаваемой информации (передача может оказаться слишком медленной и дорогой).

Ответ на вопрос 9. Если канал передачи работает с сильными помехами, то две и даже все три копии сигнала могут быть искажены и тогда защита не сработает.

Защита Листика будет срабатывать только тогда, когда в каждой тройке сигналов искажению может быть подвержен только один сигнал.

Решения зачетного класса

Ответы к заданию 1.

- 1) Неверные знаки.
- 2) Пропущенные знаки.
- 3) Лишние знаки.

Ответы к заданию 2.

- 1) Пропущенные знаки.
- 2) Лишние знаки.
- 3) Неверные знаки.

Ответ к заданию 3. 79

Ответ к заданию 4. 0100 0010

Ответ к заданию 5. 67

“ЖАРКОЕ ЛЕТО-2002”

Уникальный комплект замечательных материалов к новому учебному году

А.А. Дуванов. DHTML-конструирование

“Бумажная” версия электронного учебника продолжает роботландский курс гипертекстового конструирования (ранее были опубликованы “HTML-конструирование” и “JavaScript-конструирование”). Новый учебник посвящен созданию динамических интерактивных приложений. В нем изложены основы CSS (каскадные таблицы стилей) и показаны способы управления содержимым страницы при помощи воздействий на гипертекстовую модель документа.

Д.М. Златопольский. Задачник по электронным таблицам (Excel)

Вопросы, связанные с обработкой информации с помощью электронных таблиц, занимают важное место в школьном курсе информатики. Но специализированного школьного задачника по электронным таблицам нет. Вернее, не было, а теперь есть. Его первая часть будет опубликована в летних номерах нашей газеты.

А.И. Сенокосов. Информатика и информатизация школы. Практические решения

Допустим, вы сделали школьный web-сайт. Протянули провода, построили локальную сеть, установили программное обеспечение. Все работает. А что же дальше? Как организовать информационное наполнение сайта? Как “встроить” web-сайт в школьную жизнь? Все, кто решал эти вопросы, знают, что они-то и являются настоящими **вопросами**.

А.А. Дуванов. Азы информатики. Книга 3 — “Пишем на компьютере”

“Бумажная” версия третьей книги нового интерактивного курса для малышей “Азы информатики” (первая книга — “Знакомство с компьютером” — была опубликована в № 1, 2, публикация второй — “В мире информатики” продолжается в текущих номерах). Заглавная тема третьей книги (современная обработка текстов) нагружена “анатомией” трех китов информатики: хранение, передача и обработка информации.

А.И. Терентьев. Организация школьного web-сайта

Как сделать школьный web-сайт “с нуля”? Как связать компьютеры в сеть, какое программное обеспечение выбрать и как его установить? Как, наконец, заставить все это “хозяйство” работать? В одном из летних номеров мы познакомим наших читателей с ответами на эти (и не только на эти!) вопросы.

Л.О. Сергеев. Уроки по теме “Базы данных”

В этом тематическом выпуске мы предложим вниманию читателей цикл уроков по теме “Базы данных”. В качестве основного инструмента для изучения этой темы автор предлагает использовать язык SQL. Теоретический материал подкрепляется большим количеством разноуровневых заданий.

Интеллектуальные игры

Что такое спортивное “Что? Где? Когда?” и чем оно отличается от телевизионного? Какие головоломки решают на чемпионатах мира? Во что можно поиграть без компьютера? Это и многое другое, а также вопросы, вопросы, вопросы... в специальном выпуске “Интеллектуальные игры”.

Д.М. Златопольский. Внеклассная работа по информатике. Избранные задания

Летом мы предложим подписчикам целый номер с заданиями, которые можно использовать на викторинах, конкурсах и других внеклассных мероприятиях по информатике. В него войдут множество новых заданий, а также лучшие из опубликованных ранее материалов популярной рубрики нашей газеты.

О.Г.А. Звенигородском. Редактор-составитель — Н.А. Юнрман

История становления школьной информатики не может быть написана без страниц, посвященных исследованиям и разработкам Г.А. Звенигородского. И сегодня остается актуальным воплощенное в них единство тематических и педагогических сторон работы с учащимися, живой практики и теоретического осмысления материала. 9 августа 2002 г. Г.А. Звенигородскому исполнилось бы 50 лет.

А.Г. Гейн. “Рыба” для учителя информатики

Авторы учебников обычно много говорят о том, что преподавать, и мало про то, как это делать. Свой взгляд на тематическое планирование предлагает один из авторов учебников по информатике.

А.И. Терентьев. Организация школьного web-сайта

Как сделать школьный web-сайт “с нуля”? Как связать компьютеры в сеть, какое программное обеспечение выбрать и как его установить? Как, наконец, заставить все это “хозяйство” работать? В одном из летних номеров мы познакомим наших читателей с ответами на эти (и не только на эти!) вопросы.

Уважаемые читатели!

Газета “Информатика” распространяется только по подписке, поэтому не забудьте подписаться на нашу газету.

Дышите... не дышите

Эрик Фрэнк Рассел

Окончание. Начало в № 10/2002

У официального представителя были громадный живот, как бурдюк, маленькие свиные глазки и елейная улыбка, казавшаяся приклеенной. Его манеры напоминали поведение циркового шпехштгалмейстера, собиравшегося объявить рекордный трюк.

— Ах, — сказал он, заметив книгу, — так ты изучил наши игры?

— Да.

— Надеюсь, ты не нашел ни одной подходящей.

— Ты надеешься? — Тейлор вопросительно посмотрел на него. — Почему?

— Приятно будет посмотреть на игру, основывающуюся на каких-то других правилах, не известных в нашем мире. Подлинно новая игра доставит всем много удовольствия. С тем условием, конечно, — добавил он поспешно, — что ее легко понять и что ты не выиграешь слишком быстро.

— Что ж, — сказал Тейлор, — должен признать, что я скорее справлюсь с тем, что знаю, чем с тем, чего не знаю.

— Отлично, отлично! — с энтузиазмом произнес гомбаринин. — Ты предпочитаешь играть в земную игру?

— Верно.

— На твой выбор накладываются ограничения.

— Какие же?

— Был у нас такой осужденный, убийца, так он хотел поиграть со своим противником в такую игру: кто вперед поймает луч и засунет его в бутылку. Это бессмысленно. Ты должен выбрать игру, которая несомненно имеет конец.

— Понимаю.

— Во-вторых, тебе нельзя выбрать игру, требующую сложной и дорогостоящей аппаратуры, которую нам придется долго изготавливать. Если приспособления нужны, то они должны быть простыми и дешевыми в изготовлении.

— Это все?

— Да... если не считать того, что полные правила игры должны быть описаны тобой недвусмысленно и четко. Как только игра начнется, нужно точно придерживаться этих правил, и никакие поправки вносить не разрешается.

— А кто санкционирует мой выбор после того, как я опишу игру?

— Я.

— Отлично. Вот что я хочу играть. — Тейлор все в деталях объяснил, взял ручку и сделал грубый набросок. Когда он закончил, гомбаринин забрал рисунок и сунул себе в карман.

— Странная игра, — произнес он, — но она, к сожалению, не кажется мне слишком сложной. Ты действительно считаешь, что сдаться по крайней мере целый день?

— Надеюсь.

— Может быть, даже два?

— Если повезет.

— Тебе очень пригодится везение! — Он задумчиво помолчал и с сомнением покачал головой. — Какая жалость, что ты не додумался до какой-нибудь более сложной и изощренной версии ализика. И аудитория была бы довольна, и ты получил бы себе больший срок жизни. Все были бы довольны, если бы ты побил рекорд затыжки перед казнью.

— В самом деле?

— Все ожидают нечто сверхъестественное от инопланетянина.

— Они и увидят новинку, верно?

— Да, конечно. — Гомбаринин все еще выглядел недовольным. — Да что там... жизнь твоя, ты сам и борись за то, чтобы продлить ее подольше.

— Мне некого будет проклинать, кроме себя, когда настанет конец.

— Верно. Игра начнется завтра, сразу же после полудня. Потом все будет в твоих руках.

Он ушел, тяжелые шаги замерли в коридоре. Через несколько минут появился тюремщик.

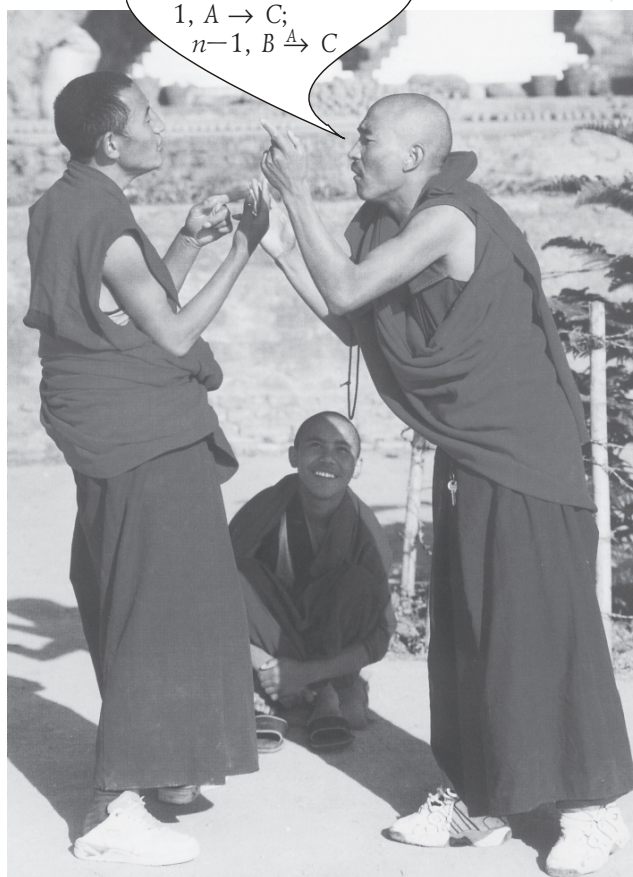
— Что ты выбрал?

— Арки-маларки.

— Как? Что это такое?

— Земная игра.

— Прекрасно, воистину прекрасно. — Тюремщик возбужденно потер руки. — Он санкционировал ее полностью?



— Да.

— Так что у тебя есть все основания продлить свое существование. Ты должен быть осторожным, чтобы избежать ловушки.

— Какой ловушки?

— Твой партнер будет играть на выигрыш так быстро и решительно, как только сможет. Этого от него и ожидают. Но как только он решит, что не сможет выиграть, он начнет играть на проигрыш. Ты не сможешь точно определить, когда он сменит тактику. Многие были пойманы на этом и обнаруживали, что игра кончена, прежде чем успели осознать это.

— Но он должен придерживаться правил, верно?

— Конечно. Ни тебе, ни ему не позволят нарушать правила. Иначе игра превратится в фарс.

— Мне это подходит.

Откуда-то снаружи донесся тонкий вопль, будто рысь налетела на кактус. Затем последовали шарканье ног, тупой удар и такие звуки, будто что-то волочили по полу. Дверь ввали со скрежетом открылась и с лязгом захлопнулась.

— Что происходит? — спросил Тейлор.

— Должно быть, кончилась игра Лягартина.

— А это кто такой?

— Политический убийца. — Тюремщик бросил взгляд на часы. — Он вы-

брал рамсид, карточную игру. Она длилась около четырех часов. С него хватит. Конец пришел этой дряни.

— А теперь его потащили на большую выжималку?

— Конечно. — Охранник спросил, не спуская с него глаз: — Нервничаешь?

— Ха-ха, — безрадостно хихикнул Тейлор.

Приготовления велись не в его камере, как он ожидал. Состязание, в котором будет участвовать инопланетянин, оказалось большим событием. Тейлора отвели по тюремным коридорам в большую комнату, где стоял стол с тремя креслами. Еще шесть кресел стояло вдоль стены, в каждом из которых сидел охранник в униформе и с оружием. Это был отряд вышибал, готовых к действиям в тот момент, когда игра подойдет к концу.

В углу комнаты стоял большой черный ящик с двумя прямоугольными отверстиями, в которых поблескивала пара линз. Вероятно, это и была видеокамера.

Пододвинув кресло к столу, Тейлор сел и одарил ледяным взглядом свою аудиторию. Узколицый субъект с круглыми, как бусинки, крысиными глазками занял кресло напротив. Чиновник с брюхом-бурдюком погрузился в оставшееся кресло. Тейлор и Крысоглаз обменялись взглядами. Землянин смотрел на противника с холодной уверенностью, тот — глубококомысленно и садистски.

На столе стояла доска, из которой торчали три длинных деревянных колышка. На левом колышке была колонка из

шестидесяти четырех дисков, постепенно уменьшающихся в диаметре. Большой — внизу, меньший — наверху. Все это очень походило на детскую пирамидку.

Не тратя времени даром, Бурдюк сказал:

— Эта земная игра называется “арки-маларки”. Колонку с диском нужно переложить с колышка, на котором они лежат, на любой из двух колышков. Они должны остаться в том же порядке: самый маленький — вверху, самый большой — внизу. Игрок, чей ход завершается построением пирамидки, выигрывает. Обоим понятно?

— Да, — сказал Тейлор.

Крысоглаз ответил ворчанием.

— Существуют три правила, — продолжал Бурдюк, — которые следует строго соблюдать. Ходы делаются по очереди. За один ход можно переложить только один диск. Нельзя класть диск на меньший по размеру, понятно?

— Да, — сказал Тейлор.

Крысоглаз снова хрюкнул.

Бурдюк достал из кармана крошечный белый шарик и осторожно бросил его на стол. Шарик подпрыгнул пару раз, покатился и упал со стороны Крысоглаза.

— Ты начинаешь, — сказал Бурдюк.

Без колебаний Крысоглаз взял самый маленький диск с вершины пирамиды на первом колышке и надел его на третий.

“Плохой ход”, — подумал Тейлор. Не меняя выражения лица, он переместил второй по размеру диск с первого колышка на второй.

Самодовольно ухмыльнувшись безо всякой причины, Крысоглаз снял самый маленький диск с третьего колышка и положил его поверх диска Тейлора на второй колышек. Тейлор тут же переложил верхний диск с пирамидки на первом колышке на освободившийся третий.

Через час Крысоглазу стало ясно, что первый колышек не только для того, чтобы держать пирамидку. Его тоже нужно использовать. Самодовольная улыбка сползла с его лица и заменилась все возрастающим раздражением, так как часы ползли, а ситуация стала во много раз более сложной.

К полуночи они все еще сидели за столом и перекидывали диски с колышка на колышек, как сумасшедшие, а игра далеко не продвинулась. Крысоглаз теперь ненавидел первый колышек, особенно если ему приходилось класть на него диск, а не снимать. Бурдюк, все так же не снимая своей приклеенной улыбки, объявил игру отложенной до утра следующего дня.

На следующий день долгая и напряженная игра продолжалась от рассвета до заката и прерывалась только два раза — игрокам подавали еду. Оба игрока играли быстро и сосредоточенно, не задумываясь делали ходы друг за другом и, казалось, соперничали друг с другом в стремлении скорее достигнуть конца игры. Ни один зритель не мог бы пожаловаться на медленную игру. Четыре раза Крысоглаз по ошибке пытался положить большой диск поверх маленького, и его тут же призывал к порядку рефер в тучном облике Бурдюка.

Прошли третий, четвертый, пятый и шестой день. Теперь Крысоглаз играл со смесью мрачной подозрительности и безнадежности. Колонка дисков на первом колышке вырастала так же часто, как и уменьшалась.

Хотя Крысоглаз и пришел в отчаяние, но дураком он не был. Он хорошо понял, что задача перекладки дисков с колышка на колышек выполняется. Но прогресс в выполнении этой задачи был ужасающе медленным. Более того, со временем он становился еще медленнее. В конце концов он совсем перестал понимать, как же проиграть эту игру, а еще меньше — как выиграть.

К четырнадцатому дню Крысоглаз сдал настолько, что слабыми автоматическими движениями перекладывал диски в бездумной, незаинтересованной манере создания, вынужденного выполнять противную и тяжелую работу. Тейлор оставался столь же невозмутимым, как бронзовый Будда, что тоже не способствовало улучшению душевного состояния Крысоглаза.

На шестнадцатый день опасность возросла, хотя Тейлор и не подозревал этого. Уже в момент, как он вошел в комнату, он почувствовал в воздухе атмосферу возросшего интереса и возбуждения. Крысоглаз выглядел еще более мрачным. Бурдюк сидел накупившись. Даже флегматичные, туповатые стражники проявляли слабые признаки оживления. К наблюдателям присоединились четверо тюремщиков, свободных от выполнения обязанностей. И видеоящик проявлял большую, чем обычно, активность.

Не обращая внимания ни на что, Тейлор сел на свое место, и игра продолжилась. Глупо потратить свою жизнь на перекладывание дисков с колышка на колышек, но позорный столб еще глупее. С любой точки зрения землянину стоило продолжать. Естественно, так он и делал, перекладывал диски, когда наступал его ход, и следил за своим противником.

В полдень Крысоглаз внезапно встал из-за стола, подошел к стене, пнул ее изо всех сил и громко высказался — и все это удивительно по-земному и по-дере-





венски. Бурдюк мягко упрекнул Крысоглаза за пустую трату времени, чтобы высказать свой патриотизм. Крысоглаз продолжал играть с угрюмым видом шалуна, которого позабыла поцеловать ма-тушка.

Поздно вечером Бурдюк остановил игру, повернулся к объективам видеокамеры и напыщенно сказал:

— Игра продолжится завтра — на семнадцатый день!

Когда на следующее утро тюремщик просунул сквозь решетку завтрак, Тейлор обратился к нему:

— Что-то ты запоздал. Я уже должен был играть.

— Сказали, что сегодня до обеда ты не понадобишься.

— Вот как? Что за суета тут поднялась?

— Вчера ты побил рекорд, — проинформировал тюремщик, неохотно выражая восторг. — До сих пор никто не затягивал игру до семнадцатого дня.

— Так что, мне подарили утро, чтобы я отпраздновал это событие? Да? Очень милосердно с их стороны.

— Понятия не имею, к чему эта поддержка, — сказал тюремщик. — Никогда не слышал, чтобы прерывали игру.

— Как ты полагаешь, игру остановили насовсем? — спросил Тейлор, чувствуя легкое удущье. — Как ты думаешь, они официально объявят игру законченной?

— Нет, такого быть не может. — Тюремщик будто даже ужаснулся этой мысли. — Мы не можем накликасть на себя проклятие мертвеца. Крайне важно, чтобы осужденный сам выбрал время своей казни...

— Почему?

— Потому что так всегда было, всегда.

Он ушел, чтобы раздать завтрак дру-гом заключенным, оставив Тейлора мучиться над этим объяснением. “Потому что так было всегда”. Недурная причина. На самом деле ее даже можно считать хорошей. Тейлор и сам может назвать несколько бессмысленных, нелогичных обычаев, существующих на Земле только потому, что “так всегда дела-

лось”. Таким образом, неизменяемые обычаи гомбариян ничуть не хуже и не лучше обычаев землян.

Хотя Тейлора несколько успокоили слова тюремщика, он ничего не мог с собой поделать и чувствовал себя все хуже и хуже по мере того, как шло время. Но ничего и не происходило. После шестнадцати дней игры он уже и во сне переключал диски с колышка на колышки... В отсрочке, которую ему предоставили, было что-то угрожающее.

Снова и снова Тейлор уловил себя на мысли о том, что инопланетяне ищут эффективный способ закончить игру, не пренебрегая в то же время условностями. Когда они найдут такой способ... если найдут... они тут же объявят игру законченной, уведут его и притянут к столбу очень тугим галстуком.

После обеда, когда он уже совсем погружился в пессимизм, за ним пришли. Его живенько, подбадривая толчками и пинками, отвели в ту же комнату, что и раньше. Игра возобновилась, будто и не прерывалась. Длилась она едва тридцать минут. Неожиданно Бурдюк велел остановиться, Тейлор вернулся в свою камеру.

Вечером его снова вызвали. Он шел неохотно, потому что эти внезапные и короткие приступы игры сказывались на его нервах гораздо сильнее, чем те, что продолжались целый день. Раньше он точно знал: его ведут играть в арки-маларки с Крысоглазом. Сейчас он не был уверен, что не станет главным персонажем в пьесе, от которой аж дух захватывает, причем в буквальном смысле.

Как только он вошел в комнату, то сразу осознал, что на сей раз все будет происходить совершенно по-другому. Доска с дисками и колышками все еще стояла в центре стола. Но Крысоглаз отсутствовал, так же, впрочем, как и вооруженный отряд. Его ждали трое: Бурдюк, Паламин и коренастый, крепко сколоченный субъект, у которого было какое-то специфическое отсутствующее выражение лица.

У Бурдюка был обиженный и недовольный вид. Словно он съел что-то недоброкачественное. Паламин выглядел особенно сердитым и выражал свое недовольство, фыркая, как горячая лошадь. Третий из присутствующих, казалось, был погружен в самосозерцание.

— Садись, — приказал Паламин, брызгая слюной. Тейлор сел.

— Теперь, Марникот, расскажи ему.

Коротышка очнулся, вспомнив, что находится на Гомбаре, и сказал Тейлору:

— Я редко смотрю видео. Это больше подходит черни, которой нечем заняться.

— К делу, — подстегнул Паламин.

— Но, услышав, что ты собираешься побить вековой рекорд, — невозмутимо продолжал Марникот, — прошлым вечером я включил видео. — Он сделал краткий жест, показывая, что сразу же чувствует, если что-то дурно пахнет. — Мне сразу же стало ясно: для того чтобы завершить твою игру, потребуется как минимум два в шестьдесят четвертой степени минус единица ходов. — Он снова на мгновение уплыл куда-то, вернулся и мягко добавил: — Это большое число.

— Большое! — сказал Паламин. Он фыркнул так, что колышки покачнулись.

— Предположим, — продолжал Марникот, — что ты будешь переключать диски один за одним так быстро, как только можешь, день и ночь, без пере-рывов на еду и сон. Знаешь ли ты, через сколько времени завершится игра?

— Примерно через шесть миллиардов земных столетий, — сообщил Тейлор, будто говорил о четверге на будущей неделе.

— Я не знаю земных способов измерять время. Но могу сказать, что ни ты, ни тысяча поколений твоих потомков не живут так долго, чтобы увидеть конец этой игры. Верно?

— Верно, — признался Тейлор.

— И все же ты говоришь, что это — земная игра?

— Да.

Марникот беспомощно развел руками. Чтобы показать, что ему больше сказать нечего.

Приняв угрожающе-хмурый вид, в разговор вступил Паламин.

— Игру нельзя признать за истинную игру, если в нее на самом деле не играют. Ты утверждаешь, что в эту так называемую игру действительно играют на Земле?

— Да.

— Кто?

— Жрецы одного храма в Бенаресе.

— И как долго они играют в нее? — спросил гомбаринин.

— Около двух тысяч лет.

— Поколение за поколением?

— Верно.

— Каждый игрок отдает игре всю свою жизнь, до конца дней, без надежды увидеть результат?

— Да.

Паламин кипел от злости.

— Так почему же они играют?

— Это часть их религиозной веры. Они верят, что в тот момент, когда будет положен последний диск, Вселенная взорвется.

— Они ненормальные?

— Не больше, чем те, кто играет в ализик и из-за столь же мизерной цели.

— Наша игра в ализик состоит из ряда отдельных игр, а не является единой бесконечной игрой. Пустую трату времени с недостижимым концом нельзя назвать игрой даже при самом богатом воображении.

— Арки-маларки не бесконечна. Она неопровержимо имеет конец. — Тейлор обращался к Марникоту, как к непроверяемому авторитету. — Верно?

— Да, она имеет конец, — произнес Марникот, не в силах отрицать этот факт.

— Так! — воскликнул Паламин более высоким тоном. — Ты думаешь, ты очень умный, да?

— Стараюсь, — сказал Тейлор, серьезно сомневаясь в этом.

— Но мы умнее, — объявил Паламин с самым тошнотворным выражением. — Ты нас обманул, а теперь мы ответим тебе тем же. Игра имеет конец. Ее можно завершить. А потому она будет продолжаться. Пока не подойдет к своему естественному концу. Ты будешь играть в нее днями, неделями, месяцами, годами, пока в конечном счете не сдохнешь от старости хронической фрустрации. Настанет время, когда сам вид этих дисков будет сводить тебя с ума, ты взмолишься о милосердной смерти. Но мы осчастливим тебя... ты будешь продолжать играть. — Он триумфально махнул рукой. — Заберите его.

Тейлор вернулся в свою камеру.

Тюремщик принес ужин.

— Мне сказали, что игра будет продолжаться регулярно с завтрашнего утра. Не понимаю, почему ее сегодня прервали.

— Они решили, что я достоин судьбы, которая хуже, чем смерть, — проинформировал Тейлор.

Надзиратель уставился на него.

— Я оказался ответным негодяем, — пояснил Тейлор.

Очевидно, Крысоглаза уведомили о создавшейся новой ситуации, поскольку он покрылся броней философского отношения и играл упорно, но без интереса. Тем не менее долгая скука повторяющихся движений раскрасила броню ржавчиной и постепенно проела насквозь.

Сразу же после полудня пятьдесят второго дня Крысоглаза оказался перед перспективой переложить большинство дисков обратно на первый колышек, один за одним. Он скинул башмаки, четырежды пробежался босиком по комнате и заблеял, как овца. Бурдюк потянул шейную мышцу, следя за ним. Двое стражников увели Крысоглаза, который продолжал блеять. Они забыли захватить башмаки.

Тейлор сидел за столом, не спуская глаз с дисков и пытался подавить внутреннюю тревогу. Что теперь будет? Если Крысоглаза окончательно сдался, то могут счесть, что он проиграл и пришло время познакомиться со столбом и шнурком. И с тем же основанием можно сказать, что неоконченная игра останется неоконченной игрой. Даже несмотря на то, что один из партнеров оказался в психиатрической лечебнице.

Если власти примут за истину первое мнение, ему для самообороны остается только настаивать на последнем. Он должен утверждать со всей оставшейся у него энергией, что раз он не выиграл и не проиграл, то его время еще не пришло. Не так-то будет легко, особенно если ему придется выражать протест, когда его будут волочить за ноги на лобное место. Его главная надежда базировалась на нежелании гомбаринан нарушать древние обычаи. Миллионы телезрителей косо посмотрят на правительство, которое попытается исказить любимое суеверие. Да, должен сказать, временами даже этот идиотский ящик бывает полезен.

Ему не стоило беспокоиться. Решив, что продолжение игры станет уточненной формой адских мучений, гомбаринане уже заготовили список сменных игроков, выбранных из мелких преступников, чьи амбиции никогда не простирались так далеко, чтобы быть достойными удушения. Так что через некоторое время появился другой противник.

Новичок оказался субъектом с длинным лицом и несколькими подбородками. Он напоминал чрезвычайно ленивую ищейку и, похоже, был способен выгова-

ривать едва ли три слова, а именно: “Не городи чущь”. Должно быть, потребовался целый месяц, чтобы научить его перекладывать за один раз только один диск и никогда, никогда, никогда не класть большой диск на меньший. Но каким-то образом его все же научили. Игра продолжалась.

Ищейка протянул неделю. Он играл медленно и задумчиво, будто боялся наказания за ошибку. Частенько его раздражала видеокамера, которая издавала тикающие звуки в краткие, но регулярные промежутки времени. Эти звуки обозначали то короткое время, на которое их выпускали в эфир.

По причинам, известным только ему одному, Ищейка питал отвращение к тому факту, что его облик транслируется на всю планету, и к концу седьмого дня сорвался. Без всякого предупреждения он вскочил с кресла, встал перед телекамерой и проделал ряд быстрых и стройных жестов. Эти жесты ничего не означали для наблюдавшего за ним Тейлора, но Бурдюк чуть не свалился со своего кресла. Стражники бросились вперед, схватили Ищейку за руки и за ноги и вынесли его из помещения.

Ищейку заменил свирепый субъект с тяжелой челюстью, который плюхнулся в кресло, посмотрел на Тейлора и пошевелил волосатыми ушами. Тейлор считал это искусство одним из своих достоинств и тут же пошевелил своими ушами в ответ. У его противника кровь бросилась в лицо.

— Эта змея землянин передразнивает меня, — проревел он Бурдюку. — Разве я должен мириться с этим?

— Прекрати паясничать, — приказал Бурдюк.

— Я только пошевелил ушами, — ответил Тейлор.

— Это одно и то же, — заявил Бурдюк. — Ты должен воздерживаться от подобных движений и сосредоточиться на игре.

Так все и продолжалось. Диски перекладывались с колышка на колышек, час за часом, день за днем. Противники Тейлора появлялись и исчезали. Где-то в районе двухсотого дня и сам Бурдюк принялся ломать свое кресло с явным намерением развести костер посреди пола. Стражники вывели и его. Появился новый рефери. Брюхо у него было еще больше, и Тейлор немедленно окрестил его Бурдюк-Два.

Как сам Тейлор сохранил здравый рассудок, он так и не понял. Но он продолжал играть, в то время как инопланетяне ломались один за другим. Чересчур уж велика была для него ставка. И все

же временами землянин просыпался в холодном поту от кошмаров, в которых он погружался в черные глубины инопланетного моря, а на шею ему был надет диск чудовищного размера. Он потерял счет дням и однажды обнаружил, что руки у него трясутся. Не добавляли ему спокойствия и раздававшиеся по ночам шум и гам. Однажды он спросил тюремщика о причине такого шума.

— Ясно, отказывался идти. Пришлось поколотить его, чтобы подчинился.

— Его игра закончена?

— Да. Этот идиот совместил пять якорей с пятью звездами. Как только он осознал, что наделал, он попытался убить своего противника. — Тюремщик покачал головой с грустным укором. — Такое поведение не приносит пользы. Таких доставляют к столбу избитыми. А если стражники разозлятся на них, те просят палача крутить помедленнее.

— Бр-р! — Тейлору было неприятно и думать об этом. — Удивительно, что никто не последовал моему примеру. Сейчас каждый должен знать об этой игре.

— Им не позволяют, — сказал тюремщик. — Теперь появился закон, что можно выбирать только известные гомбаранские игры.

Тюремщик неторопливо ушел. Тейлор растянулся на скамье в надежде на тихую, спокойную ночь. Какая сегодня дата по земному календарю? Как долго он находится здесь? Сколько ему еще здесь оставаться? Скоро ли он утратит контроль над собой и сойдет с ума? Что с ним сделают, когда он станет слишком безумным, чтобы играть?

Частенько, перед тем как заснуть, Тейлор обдумывал план бегства. Возможно, он смог бы выбраться из тюрьмы, несмотря на решетки, бронированные двери, замки, запоры, засовы и вооруженную охрану. Следовало лишь дожидаться редкой счастливой случайности и ухватиться за нее обеими руками. Но, предположим, он выберется. Что тогда? В любом месте на этой планете он будет столь же заметен, как кенгуру на тротуарах Нью-Йорка. Если бы он смог сделаться хотя бы слегка похожим на гомбаранина, у него появился бы какой-то шанс. Но это невозможно. Ему ничего не оставалось делать, кроме как выигрывать время.

Так он и поступал. Играл и играл, с перерывами только на еду и сон. К трехсотому дню он вынужден был признать себе, что чувствует себя дряхлым стариком. К четырехсотому ему стало казаться, что он играет по крайней мере пять лет и обречен играть вечно. Четыреста двадцатый день не отличался от всех других, если не считать одного: Тейлор не подозревал, что этот день — последний.

На рассвете четыреста двадцать первого дня никто не пришел за ним, чтобы отвести на игру. Тейлор прождал пару часов, но никто так и не появился. Может быть, его решили сломать, поиграв с ним в кошки-мышки? Вызывая на игру, когда он не ждет, или не вызывая вовсе. Не что вроде психологической пытки. Когда по коридору проходил тюремщик, Тейлор подошел к решетке и заговорил с ним. Тот тоже ничего не знал и был удивлен.

В полдень принесли еду. Тейлор только успел покончить с ней, как появился отряд стражников во главе с офицером. Они вошли в камеру и сняли с него окovy. Тейлор блаженно раскинул руки и забросал вопросами офицера и его свиту. На него не обращали внимания и вели себя так, словно он украл зеленый глаз маленького желтого бога. Потом его вывели из камеры, провели по коридорам, миновав комнату игр.

Через большую дверь они вышли на открытый двор. Посреди двора стояло шесть коротких стальных столбов, у каждого из которых было отверстие у вершины и подстилка у подножия. Стража флегматично шествовала прямо к столбам. Желудок Тейлора сжался. Отряд протопал мимо столбов к воротам. Желудок Тейлора благодарно расслабился и встал на место.

За воротами они сели в бронетранспортер, который рванул с места. Транспорт промчался по окраинам города и выехал к космопорту. Все вылезли и промаршировали мимо башни управления на бетонное поле. Тут они остановились.

На посадочном поле Тейлор увидел земной космический корабль. Он был слишком маленьким для боевого корабля, слишком коротким и толстым для разведчика. С недоверчивым восторгом Тейлор узнал в нем спасательную шлюпку линкора. Ему захотелось броситься к ней со всех ног, но вокруг стояли охранники. Они бы не допустили подобных эксцессов. Тейлор и его "свита" ждали четыре долгих, утомительных часа, не сходя с места. И вот на поле с ревом опустилась еще одна спасательная шлюпка и села рядом с первой. Из нее вышла группа существ, большей частью гомбараниан. Стражники толкнули Тейлора вперед.

Тот смутно осознавал, что на линии, означавшей полпути, происходит какая-то церемония обмена. Мимо него в противоположном направлении прошла цепочка угрю-

мых гомбараниан. Многие из них были украшены большим количеством латунных медалей и имели гневный вид генералов, только что пониженных в звании до полковников. Он узнал среди них одного гражданского, Боркора, и, проходя мимо него, с удовольствием пошевелил ушами.

Потом руки друзей помогли Тейлору залезть в люк, и он обнаружил себя сидящим в каюте взлетающего корабля. Юный и старательный лейтенант что-то рассказывал, но Тейлор понимал лишь половину речей лейтенанта.

— ...приземлились, захватили двадцать гомбараниан и смыслись с ними в пространство. Допросили их на языке знаков... были слегка удивлены, когда узнали, что вы еще живы... освободили одного с предложением обменяться пленниками. Девятнадцать бездельников гомбараниан на одного землянина — это честная сделка, верно?

— Да, — пробормотал Тейлор, оглядываясь.

— Скоро мы доставим вас на борт "Громовержца"... "Маклин" не мог, его отвлекли неприятности возле Лебеда... мы шли сюда со всей возможной скоростью... — Лейтенант с симпатией поглядывал на него. — Через несколько часов вы отправитесь домой. Голодны?

— Нет, вовсе нет. Вот уж чего они не хотели, так это уморить меня голодом.

— Хотите выпить?

— Спасибо, не пью.

Лейтенант засмутился, засуетился и спросил:

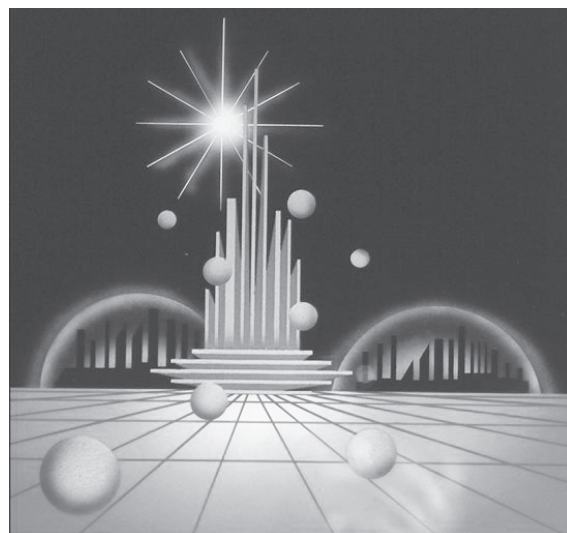
— А как насчет партии в шахматы?

Тейлор провел пальцем по внутренней стороне воротничка и ответил:

— Извините, я не умею играть и не хочу учиться. У меня аллергия к играм.

— Попробуйте, вам понравится.

— Если бы я попробовал сыграть в шахматы, меня бы повесили, — ответил Тейлор.



Java и JavaScript

Окончание. Начало на с. 1

И тот, и другой употребляются web-дизайнерами для оформления HTML-страниц.

В обоих языках есть объектно-ориентированные структуры данных, но в языке Java соответствующий принцип доведен до логического завершения: тут нельзя написать ни строчки без упоминания об объектах. В JavaScript таких строгих требований нет.

В обоих языках управление памятью полностью возложено на исполняющую систему. Программист не может распределять сегменты памяти и изменять содержимое ее ячеек произвольным образом. Данное ограничение обеспечивает безопасность и одновременно упрощает разработку программ.

Оба языка содержат "ограничивающие" правила, и программы не могут получить доступ к защищенной информации на "удаленном" компьютере.

В обоих языках есть средства обработки событий при запуске в "удаленном" браузере. Нажатие клавиш клавиатуры или мыши приводит к прерыванию программы (и к ответной реакции на эти события).

Что же касается различий, то главные из них состоят в следующем.

Java — полукомпилируемый язык, в то время как программы на JavaScript полностью интерпретируются (и ошибки в них выявляются только на этапе выполнения).

Язык Java является полностью объектно-ориентированным [например, он не дает возможности использовать переменную целого типа, не создав объект с элементом этого типа, так же, как и применить метод (правило), не связав его с объектом].

В JavaScript использование (или неиспользование) объектов зависит только от желания разработчика программы.

В языке Java можно создавать иерархически упорядоченные структуры классов, что позволяет повторно использовать фрагменты программы. Допускается создание классов, наследующих свойства родительского класса. В JavaScript не употребляются ни иерархические классы, ни принцип наследования свойств. Можно создать объект и связать с ним методы, но формальной системы классов не существует.



В языке Java тип данных указывается для любого элемента данных. В JavaScript проверка соответствия типов данных отсутствует.

Программа на языке Java поступает на "клиентский" компьютер в виде отдельного файла, в то время как программа на JavaScript содержится внутри HTML-документа.

Имеются отличия и в синтаксисе: каждый оператор в языке Java завершается точкой с запятой, тогда как в JavaScript новый оператор можно просто начать с новой строки. Существуют также другие (незначительные) отличия.

Вообще для решения больших и сложных задач целесообразно применять язык Java, а для создания небольших программ более удобен язык JavaScript. Но, разумеется, есть и исключения из этого правила...

Литература

1. Шниер М. Толковый словарь компьютерных технологий: Пер. с англ. Киев: ДиаСофт, 2000.
2. Вейнер П. Языки программирования Java и JavaScript: Пер. с англ. М.: ЛОРИ, 1998.

Гл. редактор
С.Л. Островский
Зам. гл. редактора
А.И. Сенокосов
Редакция:
Е.В. Андреева
Н.Л. Беленькая
Л.Н. Картвелишвили
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 20.03.2002.
Номер подписан 20.03.2002.

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.Бузоева, **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.