

ИНФОРМАТИК А

МАСТЕР МАТЕМАТИЧЕСКИХ ПРИБОРОВ

“Машина [фр. *machine*, от лат. *machina* — сооружение] — 1) механизм или сочетание механизмов, осуществляющие определенные целесообразные движения для преобразования энергии (машины-двигатели), изменения формы, свойств, состояния или положения предмета труда (машины-орудия, рабочие машины), сбора, передачи, хранения, обработки и использования информации (например, вычислительные машины); 2) автомобиль или какое-либо другое транспортное средство (например, мотоцикл, велосипед)”.

Словарь иностранных слов
Машина есть система связанных между собой частей из дерева... для передвижения тяжестей.

Витрувий, римский архитектор и инженер (I век до н. э.) [1].

Принято считать, что машиностроению немногим больше 200 лет, а у его истоков стоит паровая машина. Как

70 лет назад, в 1932 г., в Англии была учреждена Золотая международная медаль Джеймса Уатта, которая присуждается за крупные успехи в области машиностроения

известно, паровой двигатель придумал французский изобретатель Д. Папен в конце XVII — начале XVIII столетий, однако в прототип паровой машины его превратил английский инженер Т. Ньюкомен. Он создал работающее на пару устройство для откачивания воды из угольных шахт. Но подлинным создателем паровой машины является Джеймс Уатт (1736—1819).

Джеймс Уатт родился в Шотландии, неподалеку от Глазго. Его отец был местным судьей и одновременно занимался торговлей, строительством,

владел небольшой верфью, а также работал столяром, слесарем, кузнецом, ремонтировал музыкальные инструменты, делал морские навигационные приборы [2, 3]. Сын, помогавший отцу, многому у него учился.

Окончив начальную школу, Джеймс несколько лет обучался ремеслу механика в Глазго, а потом год совершенствовал свое мастерство в Лондоне. Вернувшись из Лондона, Уатт открыл мастерскую при университете в Глазго. Он принимал заказы на изготовление и ремонт музыкальных инструментов и всевозможных точных приборов.

В начале 1760-х годов Уатту пришлось заняться усовершенствованием машины Ньюкомена, принадлежавшей университету. Принцип ее работы был следующим. Пар поднимал поршень в цилиндре до самого верха, затем в цилиндр под поршень впрыскивалась вода. Большая часть пара, охладившись, конденсировалась, т.е. превращалась в воду. Под поршнем образовывался вакуум, и атмосферное давление гнало поршень вниз.

Машина действовала медленно, неравномерно, была громоздкой, “съедала” массу топлива. Ее можно было использовать только на угольных шахтах. Там она приводила в действие насосы. Вдали от угольных месторождений “прокормить” этот агрегат было очень трудно. Усовершенствовать машину Ньюкомена пытались многие изобретатели, но их старания не приводили к успеху.



Читайте в номере

Экзамены..... 3–7**Е.А. Еремин, А.П. Шестаков. Примерные ответы на примерные билеты**

Могут ли ваши ученики сказать, что такое компьютерная сеть, для чего нужны компьютерные сети, чем локальная сеть отличается от региональной и глобальной? Знают ли они, что такое сервер и что такое клиент (или рабочая станция)?

Пусть ребята попробуют рассказать о способах передачи информации, об организации и структуре локальных и глобальных компьютерных сетей, а затем попросите их перевести число из одной системы счисления в другую, например, из десятичной в двоичную или из двоичной в шестнадцатеричную.

Примерные билеты (и ответы на них) для проведения итоговой аттестации учеников 9-х классов. Сегодня представлен девятнадцатый билет.

Олимпиады..... 8–12**Решения задач XIV Всероссийской олимпиады по информатике**

А теперь попытайтесь решить с ребятами три олимпиадные задачи: “Дождик” (где нужно определить максимальную глубину в образовавшихся после дождя водоемах), “ПрограмМИСТИКА” (самая простая задача на олимпиаде), “Система защиты” (имеется в виду защита ценных бумаг).

Методика..... 13–15**А.Н. Минченко, М.М. Минченко. Основы программирования на QBasic: задания и упражнения**

Изучение алгоритмизации и программирования, особенно на начальных этапах, часто сопряжено с различными сложностями, связанными с идейной новизной рассматриваемых понятий. За примерами далеко ходить не надо; чего стоит, например, такой простой для профессиональных программистов оператор присваивания! Как много необходимо осознать школьнику, для того чтобы понять, почувствовать его суть. Поэтому такое значение придается методике начального обучения программированию.

Предлагается цикл упражнений для отработки и проверки понимания таких вопросов, как “Оператор присваивания” и “Условный оператор”. Особенность предлагаемых заданий заключается в том, что они должны выполняться на бумаге, без использования компьютера.

На стенд в кабинете информатики 16–17**Плазменные дисплеи**

Если среди плоскпанельных компьютерных дисплеев сегодня преобладают жидкокристаллические устройства, то “новый перспективный рынок настенных телевизионных дисплеев... наиболее успешно осваивается производителями плазменных экранов”. Кстати, знаете ли вы, что работа плазменной панели основана на управляемом холодном разряде разреженного газа, находящегося в ионизированном состоянии (холодная плазма)?

Уроки 18–27**А.А. Дуванов. Азы информатики. Материалы Роботландского университета**

“Объектом в информатике, как и всюду, называют все, что угодно: число, таблицу, исполнителя, программу, кнопку на экране...”

Однако информатика рассматривает объекты с точки зрения информационных процессов”.

Продолжение второй книги (“В мире информации”). В этом номере представлен раздел, посвященный информационным объектам.

Внеклассная работа 28–29**Н.Г. Сербяева. Учебный чайнворд**

Предлагаемый чайнворд может использоваться как при проведении внеклассных мероприятий в рамках недели информатики в школе, так и непосредственно на уроках, если требуется дать возможность ребятам немного отдохнуть, совмещая полезное с приятным. Можно также использовать чайнворд для оформления стендов в кабинете информатики и школьных газет.

Внимание! В серии “Жаркое лето-2002” выйдет номер “Внеклассная работа по информатике”, целиком состоящий из разнообразных разноуровневых заданий, которые можно использовать при организации внеклассных мероприятий с учащимися различного возраста.

Мнения 30–31**Компьютер для самых маленьких. Интервью с М.С. Цветковой**

Компьютерные технологии с неизбежностью вошли в нашу жизнь и входят в жизнь наших детей. С этим, во-первых, бесполезно, а во-вторых, даже вредно бороться. Правильнее приложить определенные усилия, для того чтобы минимизировать вред от раннего приобщения детей к компьютерам и максимизировать пользу. Как? Свое мнение на эту тему высказывает главный специалист по информатике Министерства образования РФ М.С. Цветкова.

Примерные ответы на примерные билеты

Е.А. Еремин, А.П. Шестаков,
г. Пермь

Продолжение. См. № 9, 10, 11, 13, 14, 15, 16, 17, 18/2002

Билет № 19

1. Способы передачи информации. Организация и структура локальных и глобальных компьютерных сетей.

2. Решение задач на представление чисел в десятичной, двоичной и других системах счисления.

1. Способы передачи информации. Организация и структура локальных и глобальных компьютерных сетей.

Как отмечалось в билете № 7 [“Информатика” № 13/ 2002, с. 9—13], *передача информации* необходима для того или иного ее распространения. Общая схема передачи такова:

*источник информации — канал связи —
приемник (получатель) информации*

Для передачи информации с помощью технических средств используются *кодировующее устройство*, предназначенное для преобразования исходного сообщения источника информации к виду, удобному для передачи, и *декодировующее устройство*, необходимое для преобразования кодированного сообщения в исходное.

При передаче информации необходимо учитывать тот факт, что информация при этом может теряться или искажаться, т.е. присутствуют *помехи*. Для нейтрализации помех при передаче информации зачастую используют помехоустойчивый избыточный код, который позволяет восстановить исходную информацию даже в случае некоторого искажения.

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

Передача информации между компьютерами существует с самого момента возникновения ЭВМ. Она позволяет организовать совместную работу отдельных компьютеров, решать одну задачу с помощью нескольких компьютеров, совместно использовать ресурсы и решать множество других проблем.

Под *компьютерной сетью* понимают систему распределенных на территории аппаратных, программных и информационных ресурсов (средств ввода/вывода, хранения и обработки информации), связанных между собой каналами передачи данных. При этом обеспечивается совместный доступ пользователей к информации (базам данных, документам и т.д.) и ресурсам (жесткие диски, принтеры, накопители CD-ROM, модемы, выход в глобальную сеть и т.д.).

По типу используемых ЭВМ выделяют *однородные* и *неоднородные* сети. В неоднородных сетях содержатся программно несовместимые компьютеры (чаще так и бывает на практике).

По территориальному признаку сети делят на *локальные*, *региональные* и *глобальные*. *Локальные* сети (LAN, *Local Area Network*) охватывают ресурсы, расположенные друг от друга не более чем на несколько километров (чаще всего это одно-два здания и прилегающая к ним территория — например, локальная сеть школы, вуза, компьютерного клуба и т.д.). *Региональные* сети охватывают город, район, область, небольшую республику (например, сеть Департамента образования Пермской области). *Глобальные* сети охватывают всю страну, несколько стран и целые континенты (например, сеть Интернет). Иногда выделяют *корпоративные* сети, где важно защитить информацию от несанкционированного доступа (например, сеть Министерства обороны).

По методу передачи информации различают сети с коммутацией каналов, сообщений, пакетов и со смешанной коммутацией. Чаще используются сети с коммутацией пакетов.

В зависимости от того, являются ли все компьютеры локальной сети равноправными или имеется выделенный центральный компьютер (сервер), сети подразделяют на *одноранговые*, или *сети с выделенным сервером*. Сеть с выделенным сервером является более производительной. Вообще *сервером* называется узел сети, который предоставляет свои ресурсы другим узлам (компьютерам и т.д.), но сам при этом не использует их ресурсы. *Клиентом* называется узел сети, который только использует сетевые ресурсы, но сам свои ресурсы в сеть не отдает (часто его еще называют *рабочей станцией*).

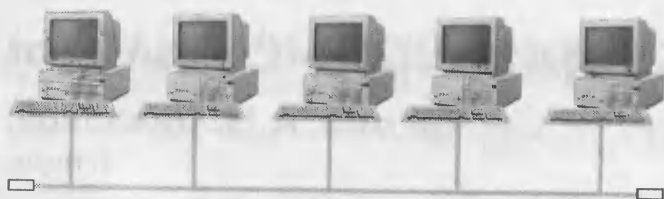
Основными свойствами локальной сети являются:

- высокая скорость передачи, большая пропускная способность;
- низкий уровень ошибок передачи;
- эффективный, быстродействующий механизм управления обменом;
- ограниченное, точно определенное число компьютеров, подключаемых к сети.

Очень важным является вопрос топологии локальной сети. Под *топологией* компьютерной сети обычно понимают физическое расположение компьютеров сети относительно друг друга и способ соединения их линиями. Топология определяет требования к оборудованию, тип используемого кабеля, методы управления обменом, надежность работы, возможность расширения сети.

Существует три основных топологии сети.

1. Шина (bus), при которой все компьютеры параллельно подключаются к одной линии связи, и информация от каждого компьютера одновременно передается ко всем остальным компьютерам.



Согласно этой топологии создается одноранговая сеть. При таком соединении компьютеры могут передавать информацию только по очереди, так как линия связи единственная.

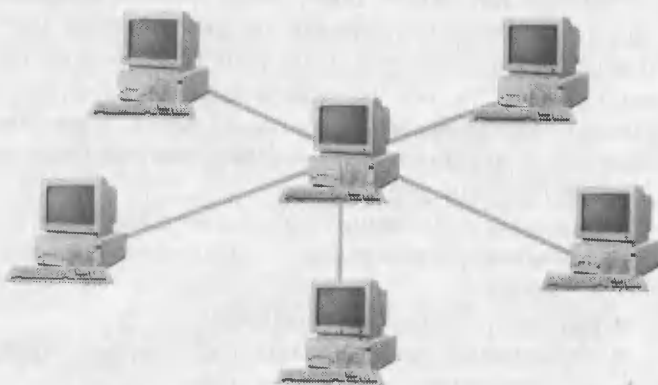
Достоинства:

- простота добавления новых узлов в сеть (это возможно даже во время работы сети);
- сеть продолжает функционировать, даже если отдельные компьютеры вышли из строя;
- недорогое сетевое оборудование за счет широкого распространения такой топологии.

Недостатки:

- сложность сетевого оборудования;
- сложность диагностики неисправности сетевого оборудования из-за того, что все адаптеры включены параллельно;
- обрыв кабеля влечет за собой выход из строя всей сети;
- ограничение на максимальную длину линий связи из-за того, что сигналы при передаче ослабевают и никак не восстанавливаются.

2. Звезда (*star*), при которой к одному центральному компьютеру присоединяются остальные периферийные компьютеры, причем каждый из них использует свою отдельную линию связи.



Весь обмен информацией идет исключительно через центральный компьютер, на который ложится очень большая нагрузка, поэтому он предназначен только для обслуживания сети.

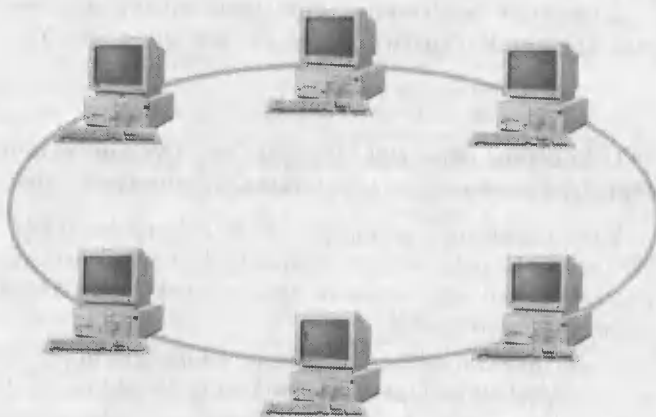
Достоинства:

- выход из строя периферийного компьютера никак не отражается на функционировании оставшейся части сети;
- простота используемого сетевого оборудования;
- все точки подключения собраны в одном месте, что позволяет легко контролировать работу сети, локализовать неисправности сети путем отключения от центра тех или иных периферийных устройств;
- не происходит затухания сигналов.

Недостатки:

- выход из строя центрального компьютера делает сеть полностью неработоспособной;
- жесткое ограничение количества периферийных компьютеров;
- значительный расход кабеля.

3. Кольцо (*ring*), при котором каждый компьютер передает информацию всегда только одному компьютеру, следующему в цепочке, а получает информацию только от предыдущего в цепочке компьютера, и эта цепочка замкнута.



Особенностью кольца является то, что каждый компьютер восстанавливает приходящий к нему сигнал, поэтому затухание сигнала во всем кольце не имеет никакого значения, важно только затухание между соседними компьютерами.

Достоинства:

- легко подключить новые узлы, хотя для этого нужно приостановить работу сети;
- большое количество узлов, которое можно подключить к сети (более 1000);
- высокая устойчивость к перегрузкам.

Недостатки:

- выход из строя хотя бы одного компьютера нарушает работу сети;
- обрыв кабеля хотя бы в одном месте нарушает работу сети.

В отдельных случаях при конструировании сети используют комбинированную топологию. Например, *дерево (tree)* — комбинация нескольких звезд.

Каждый компьютер, который функционирует в локальной сети, должен иметь *сетевой адаптер* (сетевую карту). Функцией сетевого адаптера является передача и прием сигналов, распространяемых по кабелям связи. Кроме того, компьютер должен быть оснащен сетевой операционной системой.

При конструировании сетей используют следующие виды кабелей:

- *неэкранированная витая пара*. Максимальное расстояние, на котором могут быть расположены компьютеры, соединенные этим кабелем, достигает 300 м. Скорость передачи информации — от 10 до 155 Мбит/с;
- *экранированная витая пара*. Скорость передачи информации — 16 Мбит/с на расстояние до 90 м. Обладает лучшей по сравнению с неэкранированной витой парой помехозащищенностью;

— *коаксиальный кабель*. Позволяет передавать информацию на расстояние до 2000 м со скоростью 2—44 Мбит/с;

— *волоконно-оптический кабель*. Позволяет передавать информацию на расстояние до 10 000 м со скоростью до 10 Гбит/с.

Теперь поговорим о глобальных сетях. На сегодняшний день их насчитывается в мире более 200. Из них наиболее известной и самой популярной является сеть Интернет.

В отличие от локальных сетей в глобальных сетях нет какого-либо единого центра управления. Основу сети составляют десятки и сотни тысяч компьютеров, соединенных теми или иными каналами связи. Каждый компьютер имеет уникальный идентификатор, что позволяет “проложить к нему маршрут” для доставки информации. Обычно в глобальной сети объединяются компьютеры, работающие по разным правилам (имеющие различную архитектуру, системное программное обеспечение и т.д.). Поэтому для передачи информации из одного вида сетей в другой используются *шлюзы (gateway)* — устройства (компьютеры), служащие для объединения сетей с совершенно различными протоколами обмена.

Протокол обмена — это набор правил (соглашение, стандарт) передачи информации в сети. Протоколы условно делятся на *базовые* (более низкого уровня), отвечающие за передачу информации любого типа, и *прикладные* (более высокого уровня), отвечающие за функционирование специализированных служб.

Для работы в глобальной сети пользователю необходимо иметь соответствующее аппаратное и программное обеспечение.

Программное обеспечение можно разделить на два класса:

- *программы-серверы*, которые размещаются на узле сети, обслуживающем компьютер пользователя;
- *программы-клиенты*, размещенные на компьютере пользователя и пользующиеся услугами сервера.

Глобальные сети предоставляют пользователям разнообразные услуги (об этих услугах и программах-клиентах, их поддерживающих, будет рассказано в ответе на билет № 20).

Подключение к глобальной сети может осуществляться одним из способов:

- *удаленный доступ по коммутируемой телефонной линии*. В этом случае в распоряжении пользователя должны быть *модем*, который преобразует подаваемую на него компьютером цифровую информацию в аналоговый сигнал (*модуляция*), и телефон. Аналоговый сигнал передается по телефонной линии, а модем на принимающей стороне совершает обратное преобразование информации (*демодуляцию*). Скорость, с которой будет производиться обмен информацией, определяется прежде всего скоростью передачи модема пользователя и качеством телефонной линии. Для предупреждения

искажения информации в процессе ее передачи и приема модем обычно работает в режиме коррекции ошибок, когда информация передается маленькими порциями, вычисляется контрольная сумма, которая также передается. Если отмечается искажение какой-то порции информации, ее передача повторяется;

- *прямой доступ по выделенному каналу*. Данный способ дороже, чаще его используют те или иные организации. В качестве выделенных каналов могут использоваться коаксиальные и оптоволоконные кабели, радиорелейные линии, спутниковая связь.

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, 240 с.
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. М.: Лаборатория базовых знаний, 1998. (Глава 5. Передача информации в компьютерных сетях, с. 107—125.)
5. Угринович Н. Информатика и информационные технологии. Учебное пособие для общеобразовательных учреждений. М.: БИНОМ, 2001, 464 с. (Глава 6. Компьютерные сети, с. 209—234.)
6. Информатика. 7—8-е классы / Под ред. Н.В. Макаровой. СПб.: ПитерКом, 1999, 368 с.

Дополнительная литература

7. Новиков Ю.В., Кондратенко С.В. Локальные компьютерные сети: архитектура, алгоритмы, проектирование. М.: Изд-во “ЭКОН”, 2001, 312 с.

2. Решение задач на представление чисел в десятичной, двоичной и других системах счисления.

Использованы материалы [9], [10].

Системой счисления называется совокупность приемов наименования и записи чисел. В любой системе счисления для представления чисел выбираются некоторые символы (их называют цифрами), а остальные числа получаются в результате каких-либо операций над цифрами данной системы счисления.

Система называется *позиционной*, если значение каждой цифры (ее вес) изменяется в зависимости от ее положения (позиции) в последовательности цифр, изображающих число.

Число единиц какого-либо разряда, объединяемых в единицу более старшего разряда, называют *основанием позиционной системы счисления*. Если количество таких цифр равно P , то система счисления называется

P -ичной. Основание системы счисления совпадает с количеством цифр, используемых для записи чисел в этой системе счисления.

Запись произвольного числа x в P -ичной позиционной системе счисления основывается на представлении этого числа в виде многочлена

$$x = a_n P^n + a_{n-1} P^{n-1} + \dots + a_1 P^1 + a_0 P^0 + a_{-1} P^{-1} + \dots + a_{-m} P^{-m}.$$

Арифметические действия над числами в любой позиционной системе счисления производятся по тем же правилам, что и в десятичной системе, так как все они основываются на правилах выполнения действий над соответствующими многочленами. При этом нужно только пользоваться теми таблицами сложения и умножения, которые соответствуют данному основанию P системы счисления.

При переводе чисел из десятичной системы счисления в систему с основанием $P > 1$ обычно используют следующий алгоритм:

- 1) если переводится целая часть числа, то она делится на P , после чего запоминается остаток от деления. Полученное частное вновь делится на P , остаток запоминается. Процедура продолжается до тех пор, пока частное не станет равным нулю. Остатки от деления на P выписываются в порядке, обратном их получению;
- 2) если переводится дробная часть числа, то она умножается на P , после чего целая часть запоминается и отбрасывается. Вновь полученная дробная часть умножается на P и т.д. Процедура продолжается до тех пор, пока дробная часть не станет равной нулю. Целые части выписываются после двоичной запятой в порядке их получения. Результатом может быть либо конечная, либо периодическая двоичная дробь. Поэтому, когда дробь является периодической, приходится обрывать умножение на каком-либо шаге и довольствоваться приближенной записью исходного числа в системе с основанием P .

Задача 1. Перевести данное число из десятичной системы счисления в двоичную:

- а) $464_{(10)}$;
- б) $380,1875_{(10)}$;
- в) $115,94_{(10)}$ (получить пять знаков после запятой в двоичном представлении).

Решение

Для перевода чисел в двоичную систему счисления целой части чисел используется, согласно описанному выше алгоритму, их последовательное деление на основание системы счисления 2, остатки от деления выписываются слева; дробная часть умножается последовательно на 2, целые части выписываются справа, после чего отбрасываются. При записи ответа в целой части выписываются остатки от деления в порядке, обратном их получению, в дробной части — целые части от умножения в порядке их получения.

а)

Частное	Остаток
464	0
232	0
116	0
58	0
29	1
14	0
7	1
3	1
1	1

$$464_{(10)} = 111010000_{(2)};$$

б)

Частное	Остаток	Целая часть	Дробная часть
380	0		1875
190	0	0	375
95	1	0	75
47	1	1	5
23	1	1	0
11	1		
5	1		
2	0		
1	1		

$$380,1875_{(10)} = 10111100,0011_{(2)};$$

в)

Частное	Остаток	Целая часть	Дробная часть
115	1		94
57	1	1	88
28	0	1	76
14	0	1	52
7	1	1	04
3	1	0	08
1	1	0	16

$$115,94_{(10)} \approx 1110011,11110_{(2)}$$

(в настоящем случае было получено шесть знаков после запятой, после чего результат был округлен).

Если необходимо перевести число из двоичной системы счисления в систему счисления, основанием которой является степень двойки, достаточно объединить цифры двоичного числа в группы по столько цифр, каков показатель вышеуказанной степени, и использовать приведенный ниже алгоритм. Например, если перевод осуществляется в восьмеричную систему, то группы будут содержать три цифры ($8 = 2^3$). Итак, в целой части будем производить группировку справа налево, в дробной — слева направо. Если в последней группе недостает цифр, дописываем нули: в целой части — слева, в дробной — справа. Затем каждая группа заменяется соответствующей цифрой новой системы. Соответствия приведены в таблицах.

P = 2	00	01	10	11
P = 4	0	1	2	3

P = 2	000	001	010	011	100	101	110	111
P = 8	0	1	2	3	4	5	6	7

P = 2	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
P = 16	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Задача 2. Перевести из двоичной системы в шестнадцатеричную число 1111010101,11₍₂₎.

Решение

$$0011\ 1101\ 0101,1100_{(2)} = 3D5,C_{(16)}.$$

При переводе чисел из системы счисления с основанием P в десятичную систему счисления необходимо пронумеровать разряды целой части справа налево, начиная с нулевого, и в дробной части, начиная с разряда сразу после запятой слева направо (начальный номер — 1). Затем вычислить сумму произведений соответствующих значений разрядов на основание системы счисления в степени, равной номеру разряда. Это и есть представление исходного числа в десятичной системе счисления.

Задача 3. Перевести данное число в десятичную систему счисления.

а) 1000001₍₂₎.

$$1000001_{(2)} = 1 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 64 + 1 = 65_{(10)}.$$

Замечание. Очевидно, что если в каком-либо разряде стоит ноль, то соответствующее слагаемое можно опускать.

б) 1000011111,0101₍₂₎.

$$1000011111,0101_{(2)} = 1 \cdot 2^9 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-2} + 1 \cdot 2^{-4} = 512 + 16 + 8 + 4 + 2 + 1 + 0,25 + 0,0625 = 543,3125_{(10)}.$$

в) 1216,04₍₈₎.

$$1216,04_{(8)} = 1 \cdot 8^3 + 2 \cdot 8^2 + 1 \cdot 8^1 + 6 \cdot 8^0 + 4 \cdot 8^{-2} = 512 + 128 + 8 + 6 + 0,0625 = 654,0625_{(10)}.$$

г) 29A,5₍₁₆₎.

$$29A,5_{(16)} = 2 \cdot 16^2 + 9 \cdot 16^1 + 10 \cdot 16^0 + 5 \cdot 16^{-1} = 512 + 144 + 10 + 0,3125 = 656,3125_{(10)}.$$

Литература к практическому заданию

8. Шауцукова Л.З. Информатика: Учебник для 10—11-х классов. М.: Просвещение, 2000 г. (Глава 4. Арифметические основы компьютеров.)

9. Введение в информатику. Лабораторные работы. / Авт.-сост. — А.П. Шестаков; Пермский университет. Пермь, 1999. Ч. I, 56 с.

10. <http://comp-science.narod.ru>. Раздел “Дидактические материалы по информатике”, статьи “Позиционные системы счисления”, “Задачи по позиционным системам счисления”.

“ЖАРКОЕ ЛЕТО-2002”

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, настроили 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 информации” — 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.

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.

Уникальный комплект замечательных материалов к новому учебному году

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. Юнерман

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.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.

Уважаемые читатели! Газета “Информатика” распространяется только по подписке, поэтому не забудьте подписаться на нашу газету. Информацию о подписке вы найдете на с. 31.

Решения задач XIV Всероссийской олимпиады по информатике

Задача "Дождик"

Автор — М.Бабенко

Имя входного файла: rain.in

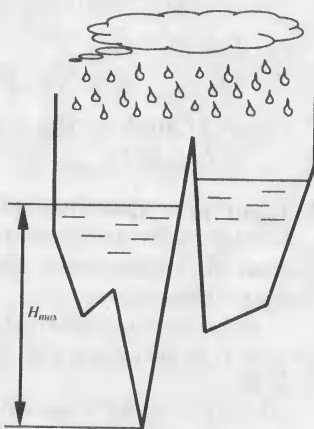
Имя выходного файла: rain.out

Ограничение времени: 3 секунды на тест

Ограничение памяти: 16 Мб

Максимальная оценка за задачу: 33 балла

В НИИ метеорологии решили изучить процесс образования водоемов на различных рельефах местности во время дождя. Ввиду сложности реальной задачи была создана двумерная модель, в которой местность имеет только два измерения — высоту и длину. В этой модели рельеф местности можно представить как N -звенную ломаную с вершинами $(x_0, y_0), \dots, (x_N, y_N)$, где $x_0 < x_1 < \dots < x_N$ и $y_i \neq y_j$ для любых $i \neq j$. Слева в точке x_0 и справа в точке x_N рельеф ограничен вертикальными горами огромной высоты.



Если бы рельеф был горизонтальным, то после дождя вся местность покрылась бы слоем воды глубины H . Но поскольку рельеф — это ломаная, то вода стекает и скапливается в углублениях, образуя водоемы.

Требуется найти максимальную глубину в образовавшихся после дождя водоемах.

Входные данные

В первой строке файла rain.in расположены натуральное число N ($1 \leq N \leq 100$) и H — действительное число, заданное с тремя цифрами после десятичной точки ($0 \leq H \leq 10^9$). В последующих $N+1$ строках — по два целых числа x_i, y_i : $-10\,000 \leq x_i, y_i \leq 10\,000$ ($0 \leq i \leq N$).

Числа в строках разделены пробелами.

Выходные данные

Выходной файл rain.out должен содержать единственное число — искомую глубину с точностью до четырех знаков после десятичной точки.

Пример

Входной файл rain.in	Выходной файл rain.out
7 7.000	15.8446
-5 10	
-3 4	
-1 6	
1 -4	
4 17	
5 3	
9 5	
12 15	

Решение

Решение задачи начнем с поиска внутренней точки i ломаной, такой, что $y_{i-1} < y_i > y_{i+1}$, с максимально возможным среди таких точек значением y_i . Если такой точки нет, то рельеф образует всего один водоем и его глубину подсчитать не очень сложно, например, методом деления пополам. При этом для каждого предполагаемого значения искомой высоты H_{max} считается площадь соответствующего выпуклого многоугольника путем суммирования площадей N трапеций, высота которых равна $(x_i - x_{i-1})$, а основания выражаются через y_i, y_{min} и H_{max} (см. рис. 1). Полученное значение площади сравнивается с числом $H(x_N - x_0)$. В зависимости от результатов сравнения искомое значение H_{max} уменьшается или увеличивается по правилу алгоритма бинарного поиска.

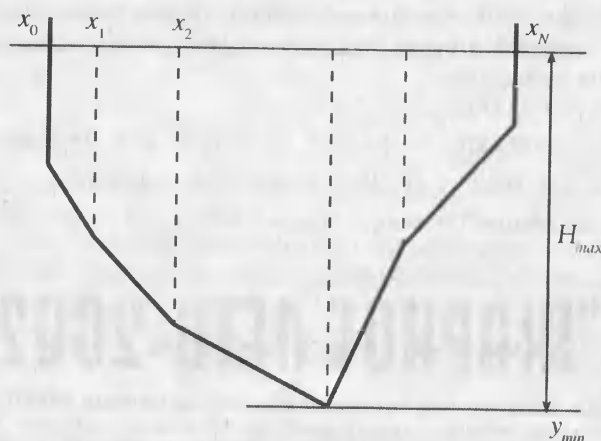


Рис. 1

Если же точка рельефа с искомым y_i существует, то следует определить, окажется она под водой после дождя или нет. Для этого проведем горизонтальную прямую $y = y_i$ и опять подсчитаем площадь уже двух выпуклых многоугольников, ограниченных этой прямой и ломаной. Если общая площадь окажется меньше, чем $H(x_N - x_0)$, то найденная верхняя точка рельефа — под водой и водоем по-прежнему один. Его глубину можно, как и ранее, искать методом деления пополам. В противном случае водоемов как минимум два и задачу будем решать рекурсивно для каждого из них.

Сложность при этом заключается лишь в анализе "переливов" воды из одного водоема в другой, в случае недостаточной емкости первого из них. Обнаружить наличие такой ситуации нетрудно. Если нам известно, что площадь одного из многоугольников, ограниченных прямой $y = y_i$ и частью ломаной, меньше, чем количество воды, попавшее на данную часть рельефа, то "лишняя" вода перелилась в соседний водоем (так как по условию $y_i \neq y_j$ для $i \neq j$, то такой водоем один). Однако распределить ее равномерно по всему соседнему рельефу нельзя. Требуется

найти ближайший к y_i участок рельефа вида “спуск-подъем” и сократить его соответственно количеству перелитой в него воды путем проведения горизонтальной прямой, площадь под которой соответствует перелитой воде (см. рис. 2). Если же площадь всего соседнего водоема опять окажется меньше, чем нужно, то процедуру следует повторить для следующего участка рельефа вида “спуск-подъем”. После того как рельеф будет окончательно модифицирован, решение задачи для него можно осуществлять рекурсивно, так же, как это было описано выше.



Рис. 2

Так как описанная рекурсивная процедура в процессе работы находит глубину каждого из образовавшихся водоемов, после ее окончания мы будем знать и максимальную глубину. Задача, таким образом, полностью решена.

Задача “ПрограмМИСТИКА”

Автор — С.М. Окулов

Имена входных и выходных файлов указаны в условии задачи

Максимальная оценка за задачу: 33 балла

В Перми становится популярной игра “ПрограмМИСТИКА”.

Для игры требуются плоские квадратные фишки четырех видов, представляющие собой поле 3×3 с вырезанной центральной клеткой. В остальных клетках каждой фишки записаны числа от 1 до 8. Все виды фишек показаны на рисунке. Количество фишек каждого вида не ограничено.

Игра проводится на поле размером $N \times N$. Первоначально все клетки поля заполнены единицами.

В начале игры Магистр несколько раз случайным образом помещает произвольные фишки на игровое поле так, что фишка попадает на поле целиком, а ее центральная клетка совпадает с одной из клеток поля. После помещения очередной фишки все числа в восьми клетках игрового поля, которые перекрывает фишка, умножаются на соответствующие числа в клетках фишки, и результаты становятся новыми значениями этих клеток игрового поля.

1	2	3
8		4
7	6	5

7	8	1
6		2
5	4	3

5	6	7
4		8
3	2	1

3	4	5
2		6
1	8	7

Таким образом, после окончания процесса размещения фишек игровое поле оказывается заполненным полученными произведениями. Далее Магистр передает получившееся поле игроку, которому необходимо установить для каждой клетки поля, сколько раз Магистр в нее помещал центральные клетки фишек.

Требуется для каждого входного файла, содержащего полученное Магистром поле, сформировать соответствующий ему выходной файл, в N строках которого содержится по N чисел, показывающих, сколько раз в соответствующую клетку помещались центральные клетки фишек.

Входные данные

Входные файлы имеют имена `game.01`, `game.02`, ..., `game.07`.

В первой строке каждого из этих файлов записано N ($3 \leq N \leq 100$). В последующих N строках записано по N разделенных пробелами натуральных чисел, являющихся значениями полученных Магистром произведений, каждое из которых не превышает $2 \cdot 10^9$.

Выходные данные

Каждый выходной файл должен состоять из N строк по N разделенных пробелами чисел, показывающих для каждой клетки поля, сколько раз в нее помещались центральные клетки фишек.

Вам требуется представить на тестирование 7 выходных файлов с именами:

`P<номер участника>_4.<номер теста>`

Пример

Входной файл
`game.0x`

```
5
1 2 3 1 1
8 1 140 48 7
7 6 120 1 16
1 1 15 8 3
1 1 1 1 1
```

Выходной файл
`P21111_4.0x`

```
0 0 0 0 0
0 1 0 0 0
0 0 0 2 0
0 0 0 0 0
0 0 0 0 0
```

Решение*

Эта задача была самой простой на олимпиаде, не требовала никаких специальных знаний и решалась путем простых умозаключений, прийти к которым можно было путем несложного анализа чисел в углах фишек каждого вида.

Поиск решения начнем с одной из угловых клеток игрового поля, например, левой верхней. Заметим, что в нее могли попасть только левые же верхние угловые клетки фишек, которые содержат числа 1, 3, 5 и 7: Последние три числа — простые. Следовательно, для того чтобы определить, сколько соответствующих им фишек попало

* Решение подготовлено Е.В. Андреевой.

в угол игрового поля (то есть центр фишек оказывался в клетке (2, 2)), нужно просто разложить на простые сомножители число, находящееся в клетке (1, 1) игрового поля. При получении очередного простого сомножителя в разложении следует “снять” соответствующую ему фишку с поля. А именно, разделить значения всех клеток игрового поля, расположенных вокруг клетки (2, 2), на значения соответствующих чисел обнаруженной фишки. И так до тех пор, пока значение в клетке (1, 1) не станет равным 1.

Остается определить, сколько именно фишек первого вида (с единицей в левом верхнем углу) попадали в левый верхний угол поля, причем таких фишек может и не быть совсем. Сделать это можно, рассмотрев значение второго поля в первой строке, т.е. клетки (1, 2). После “снятия” фишек второго, третьего и четвертого вида с левого верхнего угла поля четными сомножителями в значении этой клетки могут быть только двойки (нечетные же сомножители соответствуют размещению фишек с центром в (2, 3)). Поэтому количество фишек первого вида в левом верхнем углу поля равно степени двойки в разложении на простые сомножители числа, оставшегося после процедуры снятия фишек в клетке (1, 2). Определив количество таких фишек, также “снимем” их с поля. Таким образом, мы нашли ответ для клетки (2, 2) (очевидно, что для всех клеток первой и последней строк и первого и последнего столбцов ответ равен нулю).

Теперь аналогичную процедуру уже можно проводить, рассматривая текущее (т.е. полученное после снятия всех фишек с левого верхнего угла) значение произведений в клетке (1, 2). В результате чего будет получен ответ для клетки (2, 3) и т.д. После того как первая строка будет рассмотрена и соответствующие ей фишки “сняты” (а это означает, что в первой строке игрового поля остались одни единицы), можно перейти к анализу второй строки. После анализа всех строк задача окажется полностью решенной.

Задача “Система защиты”

Автор — Г.Корнеев

Имена входных и выходных файлов указаны в условии задачи

Максимальная оценка за задачу: 33 балла

“ПермьОлимпБанк” разработал сверхнадежную систему защиты ценных бумаг. В правом верхнем углу любой бумаги должно присутствовать прямоугольное изображение в виде черно-белого рисунка. Для определения подлинности документов была создана библиотека контрольных элементов. Если документ подлинный, то в изображении на документе заданное количество раз присутствует каждый контрольный элемент из библиотеки. В библиотеке нет совпадающих элементов.

Система контроля после сканирования получает изображение в виде цифрового массива $N \times M$ ($N \leq 100\,000$, $M \leq 1000$), в котором цифра 1 соответствует черному цвету, а цифра 0 — белому. Затем система ищет контрольные элементы в полученном массиве.

Контрольный элемент представляется массивом размером $L \times L$ ($L \leq 50$) цифр, каждая из которых равна 0 или 1. В библиотеке — K контрольных элементов ($K \leq 20$). Элемент библиотеки должен точно совпадать с какой-либо частью изображения. При сравнении изображения и контрольных элементов повороты не допускаются.

Требуется для каждого входного файла сформировать соответствующий ему выходной файл, в котором записано, сколько раз каждый элемент библиотеки встречается в изображении, описанном во входном файле.

Входные данные

13 входных файлов имеют имена CONTROL.01, CONTROL.02, ..., CONTROL.13.

В первой строке каждого из этих файлов записаны через пробел числа N , M , K , L .

Далее следуют по порядку K блоков, соответствующих элементам контрольного образца в библиотеке. Каждый блок состоит из L строк по L цифр (ноль или единица). После каждого блока следует пустая строка.

В последующих N строках записаны по M цифр в каждой, соответствующих изображению.

Выходные данные

Вам требуется сдать 13 выходных файлов с именами:

$P < \text{номер участника} > _6 _ \text{номер теста} >$

Каждый выходной файл должен содержать K строк.

В каждой строке содержатся два числа: номер контрольного элемента из библиотеки и число его обнаружений (0 — если элемент не обнаружен).

Номер контрольного элемента из библиотеки в первой строке равен 1, во второй — 2 и т.д.

Все числа в строках должны быть разделены пробелом.

Например, изображению на рис. 1 и контрольным элементам на рис. 2 соответствуют представленные ниже входной и выходной файлы.



Рис. 1

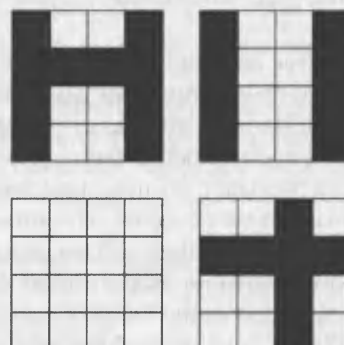


Рис. 2

Входной файл

CONTROL.xx

8 7 4 4

1001

1111

1001

1001

1001

1001

1001

1001

0000

0000

0000

0000

0010

1111

0010

0010

1001001

1001001

1001001

1111111

1001001

1001001

1001001

1001001

Выходной файл

Pnnnnn_6.xx

1 2

2 2

3 0

4 1

На рис. 2в изображено дерево после добавления в него всех 6 строк двух образцов. При этом третьей строке соответствует узел номер 8, четвертой (она первая во втором образце) — 9, пятой снова 3, шестой — 10.

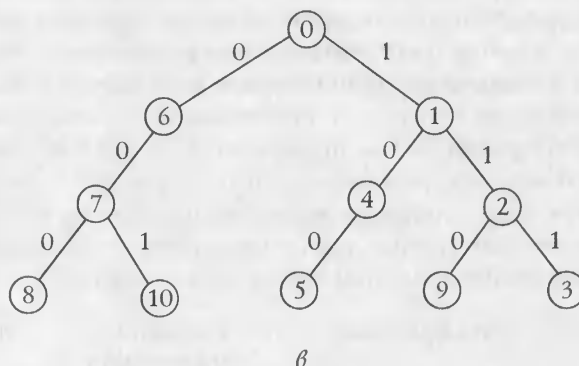
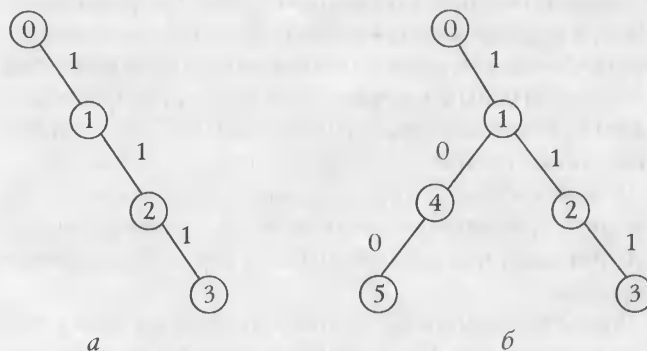


Рис. 2

Решение*

Рассмотрим решение задачи на примере поиска двух контрольных элементов 3×3 (рис. 1).

1	1	1
1	0	0
0	0	0

1	1	0
1	1	1
0	0	1

Рис. 1

Для поиска во входном файле одновременно всех 6 строк (длины 3) этих элементов построим двоичное дерево следующим образом. В начале дерево имеет только корень (узел номер 0). Цифре 0 в элементе будем ставить в соответствие левого сына текущего узла, а 1 — правого. Высота дерева равна L (в нашем случае 3). Для первой строки первого образца 111 имеем: правый (1-й сын корня, соответствующий первой единице в строке, становится узлом с номером 1, второй узел получаем как 1-й сын первого, а третий — 1-й сын второго (рис. 2а). При этом можно сказать, что первой строке в целом соответствует узел номер 3. Вторую строку 100 также включаем в дерево, начиная от корня: 1-й сын узла ноль уже существует, 0-й сын первого узла становится узлом номер четыре, а пятый узел — 0-м сыном четвертого (рис. 2б). Таким образом, второй строке соответствует листовой узел 5.

Определим для каждого узла дерева префикс-функцию F следующим образом. Пусть путь от корня до узла с номером N проходит по ребрам, помеченным символами $S_1 S_2 S_3 \dots S_L$ (где L — длина пути). Отбросим от пути первый символ и проверим, есть ли в дереве узел, путем до которого является $S_2 S_3 \dots S_L$. Если такой узел существует, то номер этого узла является префикс-функцией узла N , в противном случае отбросим от пути следующий символ и так далее. Если в результате отбрасывания был получен путь нулевой длины, то префикс-функция узла N равна 0.

Например, построим префикс-функцию узла номер 10: путь до него помечен символами 001. Отбросив первый символ, имеем путь 01, которому не соответствует ни один узел (у узла номер 6 нет 1-го сына). Отбросив следующий символ, имеем путь 1, которому соответствует узел 1, следовательно, $F(10) = 1$. Аналогичным образом получаем $F(1) = 0$, $F(2) = 1$, $F(3) = 2$, $F(4) = 6$, $F(5) = 7$, $F(6) = 0$, $F(7) = 6$, $F(8) = 7$, $F(9) = 4$.

Теперь покажем, как с помощью построенного дерева распознавать во входной строке одновременно все имеющиеся в контрольных элементах строки. Рассмотрим конечный автомат, в котором узлы дерева соответствуют состояниям. За исходное состояние примем состояние 0. Пусть автомат находится в состоянии N . Прочитаем следующий символ строки. Если у узла номер N есть соответствующий прочитанному символу сын, то новое состояние — номер этого сына, иначе проверим сына узла $F(N)$; если и его нет, то узла $F(F(N))$ и т.д. Если дошли до корня дерева и у него нет искомого сына, то новое

состояние — 0. Например, строке 010001011 изображения соответствует следующая последовательность состояний: 6, 1 (у узла номер 6 нет правого сына, но у узла номер $F(6) = 0$ он равен 1), 4, 5, 8 (у узла номер 5 сыновей нет, но у узла номер $F(5) = 7$ 0-й сын есть), 10, 4, 1 (здесь префикс-функция была использована два раза), 2. Так как по результатам каждой проверки или автомат переходит в новое состояние, или отбрасывается по крайней мере один символ входной строки, переход в новое состояние осуществляется за $O(1)$.

Если в процессе работы автомата мы попадаем в состояние, соответствующее одному из листьев дерева, то это означает, что найдена одна из строк контрольных образцов.

Будем применять полученный автомат ко всем строкам проверяемого изображения в отдельности. Заменим каждую цифру изображения на номер состояния автомата, в котором он находился при ее прочтении. Каждый же контрольный элемент при этом можно описать столбцом из L чисел — состояний автомата для каждой из его строк. В нашем примере это 3 5 8 и 9 3 10. Тогда для поиска контрольных элементов нужно просто в преобразованном двумерном массиве найти во всех столбцах, начиная с L -го, все последовательности чисел, соответствующие элементам библиотеки (см. рис. 3).

Изображение	Состояние 1-го автомата
0 1 1 0	6 1 2 9
1 1 1 1	1 2 3 3
1 0 0 1	1 4 5 10
0 0 0 1	6 7 8 10

Рис. 3

Данную операцию тоже можно производить быстро, построив дерево, аналогичное описанному выше, для последовательностей, соответствующих контрольным элементам, но в данном случае дерево будет уже не бинарное, а K -ичное, где K — количество листьев в бинарном дереве. Причем на каждом горизонтальном уровне этого дерева не может находиться более K узлов. Префикс-функция определяется, как и ранее.

В нашем случае дерево примет вид, показанный на рис. 4. Префикс-функция для всех узлов, кроме пятого, равна 0, а $F(5) = 1$. Первый контрольный элемент соответствует узлу номер 3, а второй — узлу номер 6. Применив полученный автомат к столбцам, содержащим номера состояний первого автомата, анализирующего строки, получим необходимый результат: если второй автомат находится в состоянии 3 или 6, значит, в анализируемой клетке находится правый нижний угол соответствующего элемента библиотеки.

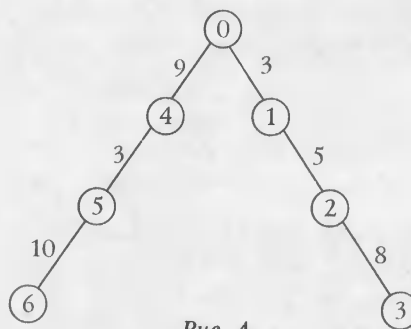


Рис. 4

Таким образом, подсчет количества появлений элементов из библиотеки производится за время линейное по площади проверяемого изображения. Данное решение легко обобщается для элементов библиотеки произвольных размеров, при этом для хранения деревьев необходимо $O(KL^2)$ ячеек памяти.

В этой задаче строятся две структуры данных, которые называются бор. С помощью этой структуры реализуется модификация алгоритма Кнута — Морриса — Пратта, предназначенная для одновременного поиска нескольких подстрок в строке (в том числе подстрок, разных по длине). Если бор строится для распознавания A строк, в которых встречается B различных символов, то возможны два варианта хранения сыновей узла: в виде массива, адресуемого символами (используется при $A \gg B$, как для первого построенного нами бора, в котором было всего два различных символа), тогда переход к новому символу осуществляется за $O(1)$, либо в виде упорядоченного списка, в котором производится бинарный поиск (используется при $A \ll B$, как во втором случае), при этом переход осуществляется за $O(\log A)$.



Очередная международная олимпиада по информатике пройдет в Южной Корее летом. Желаем успеха российской команде!

Основы программирования на QBasic: задания и упражнения

А.Н. Минченко, М.М. Минченко,

Москва

Изучение основ программирования в рамках школьного курса информатики, особенно на начальных этапах, как правило, сопряжено с множеством трудностей. Понимание сути процесса программирования приходит не сразу, что, в свою очередь, вызывает сложности в приобретении учащимися навыков осознанного использования изучаемых операторов и конструкций при выполнении практических заданий по написанию программ.

Эти проблемы зачастую подталкивают учителей к отказу от изучения программирования на уроках информатики. Но нашей задачей является не исключение из учебной программы той или иной составной части курса из-за плохого восприятия учащимися, а выработка таких методик и способов организации учебного процесса, которые будут улучшать это восприятие. При преподавании программирования учитель особенно остро сталкивается с необходимостью поиска, разработки и использования соответствующих методических приемов.

Наш опыт показывает, что развитию алгоритмического мышления, воспитанию алгоритмической культуры и внутреннему пониманию (а не механическому заучиванию порядка расстановки операторов) языка программирования в значительной мере, особенно на начальных этапах изучения программирования, может способствовать выполнение учащимися под руководством преподавателя, помимо практических заданий по написанию программ на компьютере, тренировочных упражнений и заданий по программированию в устной и письменной форме без компьютера.

Для отработки той или иной темы полезно использовать задания, в которых предлагается по готовой программе определить результат ее работы (задания типа "определить, что делает данная программа", "что будет выведено на экран" и т.п.) или указать имеющиеся в ней ошибки. Такие задания с успехом могут использоваться и при проведении самостоятельных и проверочных работ с целью выявления степени усвоения учащимися пройденного материала.

В качестве иллюстрации рассмотрим примеры такого рода упражнений, направленных на отработку начальных тем изучения языка программирования QBasic (оператор присваивания, операторы ввода-вывода, условный оператор, операторы циклов).

Операторы присваивания, ввода и вывода

Оператор присваивания, как правило, трудно воспринимается начинающими изучать программирование. Упражнения, в которых разным переменным присваи-

вается значение одного и того же выражения, помогают усвоить суть оператора присваивания.

Одна из трудностей также связана с тем, что учащиеся путают роль левой и правой частей оператора присваивания. И, например, когда одной переменной необходимо присвоить значение другой переменной (т.е. когда в качестве частного случая присваиваемого выражения выступает переменная), зачастую считают безразличным, какую из этих переменных записать в левой части оператора присваивания, а какую — в правой.

Основные трудности при использовании операторов ввода и вывода обычно связаны с их синтаксисом. Так, например, указав в приглашении оператора INPUT или поясняющей текстовой константе оператора PRINT имя переменной, учащиеся иногда не указывают следом соответствующую переменную. При использовании оператора PRINT учащиеся также не всегда представляют себе, как будет расположена на экране соответствующая выводимая информация в зависимости от употребленных в списке вывода разделителей.

Целью приведенных ниже упражнений является отработка и усвоение учащимися особенностей выполнения оператора присваивания и операторов ввода и вывода.

1. Какие из следующих операторов ввода записаны неверно? Объясните свой ответ. (Неверные операторы мы выделили жирным шрифтом, указав причину ошибки.)

- а) INPUT X
- б) INPUT "X="; X
- в) **INPUT "X="** Не указана переменная.
- г) INPUT "X, Y"; X, Y
- д) INPUT "X, Y"; X
- е) **INPUT "X="; X, "Y="; Y** В качестве приглашения можно использовать только одну строку.

2. Найдите ошибки в следующих фрагментах программ:

- а) A = 5
B = 2 * A
PRINT A : B Неверный разделитель в операторе PRINT.
PRINT "B="; 2*B
PRINT Конец Неправильно оформлена текстовая константа.
- б) X = 7
2 * X = Y Неверный формат оператора присваивания.
PRINT 'Y='; X^2 Неправильно оформлена текстовая константа.
PRINT Y + 3

3. Объясните назначение следующей программы:

```
INPUT "A="; A
INPUT "B="; B
A = A + B
B = A - B
A = A - B
PRINT "A="; A; "B="; B
```

Ответ. В этой программе вводятся значения переменных A и B, меняются местами и выводятся на экран.

4. Покажите на бумаге, что будет выведено на экран в результате работы следующего фрагмента программы:

```
X = 2
Y = 3
PRINT "X="; X, "Y="; Y
X = Y
Y = X
PRINT X, Y, 3
```

Ответ

```
X= 2      Y= 3
3          3          3
```

5. Что будет выведено на экран в результате работы следующего фрагмента программы:

```
A = 10 : B = 6
A = (A + B) / 2
B = (A + B) / 2
PRINT "A="; A, "B="; B
A = (A + B) / 2
PRINT : PRINT A, B
A = B : B = A
PRINT A, B
```

Ответ

```
A= 8      B= 7
Пустая строка
7.5       7
7         7
```

В данном фрагменте три оператора присваивания имеют одинаковую правую часть, а в левой части — одну из переменных вычисляемого выражения. Если учащиеся не понимают порядок выполнения оператора присваивания, в качестве ответа они, как правило, указывают одно и то же значение — число 8.

6. Что будет выведено на экран в результате работы следующего фрагмента программы:

```
X = 8 : Y = 3
X = 2 * X + 2 * Y
PRINT X, Y : PRINT
Y = 2 * X + 2 * Y
PRINT "X="; X, "Y="; Y,
X = 2 * X + 2 * Y
PRINT "X="; X, "Y="; Y
```

Ответ

```
22      3
Пустая строка
X= 50   Y= 22   X= 144   Y= 50
```

В этом упражнении сделан особый акцент на операторе вывода PRINT. В операторе PRINT пятой строки преднамеренно переставлены местами переменные X и Y (имена переменных не соответствуют указанным в кавычках текстовым константам). Кроме того, задание

требует от учащихся особого внимания к синтаксису оператора вывода. Например, в конце пятой строки стоит запятая, указывающая на то, что после выполнения данного оператора не произойдет перехода на следующую строку.

7. Дан фрагмент программы:

```
INPUT "Введите X и Y", X, Y
PRINT 2*X, Y
X = Y ^ 2 - X
Y = X ^ 2 - Y
PRINT "X="; X+1, "Y="; Y-1
PRINT
X = Y ^ 2 - X
Y = Y ^ 2 - X
PRINT X,
PRINT Y
```

Показать, что будет выведено на экран, если по запросу введены в X и Y соответственно числа 5 и 3.

Ответ

```
10      3
X= 5     Y= 12
Пустая строка
165     4
```

В этом упражнении особое внимание следует обратить на наличие вычисляемых выражений в операторах PRINT второй и пятой строк ($2 \cdot X$ и $X+1$, $Y-1$ соответственно). Учащиеся иногда считают, что значения X и Y в этих случаях изменятся.

Условный оператор

Целью следующих упражнений является усвоение учащимися порядка выполнения условного оператора. Упражнения составлены так, что в зависимости от конкретных вводимых (исходных) числовых значений проверяемые в условных операторах логические выражения в одних случаях истинны, а в других — ложны. В результате управление передается той или иной ветви условного оператора или следующей за условным оператором строке, что приводит к различным конечным результатам.

1. Объясните назначение следующей программы:

```
INPUT "X="; X
INPUT "Y="; Y
IF X > Y THEN Z = 0 ELSE Z = Y - X
M = X + Z
PRINT "M="; M
END
```

Ответ. В переменную M заносится наибольшее из чисел X и Y, затем значение этой переменной выводится на экран.

2. Дан фрагмент программы:

```
INPUT "M="; M
IF M < 10 THEN M = 10 ELSE M = 2 * M
N = M + 5
PRINT M, N
```

Покажите на бумаге, что будет выведено на экран, если по запросу введено:

а) Число 5

Ответ 10 15

6) Число 12

Ответ	24	29
--------------	----	----

в) Число 10

Ответ	20	25
--------------	----	----

3. Дан фрагмент программы:

```

INPUT "A="; A
INPUT "B="; B
A = (A + B) / 2
B = (A + B) / 2
PRINT "A="; A, "B="; B
IF A < B THEN A = 2 * A ELSE B = 2 * B
PRINT : PRINT A, B

```

Покажите на бумаге, что будет выведено на экран, если по запросу введены в А и В соответственно:

а) Числа 10 и 6

Ответ

A= 8	B= 7
Пустая строка	
8	14

б) Числа 4 и 12

Ответ

A= 8	B= 10
Пустая строка	
16	10

4. Дан фрагмент программы:

```

INPUT A, B
S = 0
10 IF A < B THEN A = A + B ELSE B = A + B
S = S + A
PRINT A, B
IF S < 40 THEN GOTO 10
PRINT "S="; S

```

Покажите, что будет выведено на экран, если по запросу введены в А и В соответственно:

а) Числа 2 и 4

Ответ

6	4
6	10
16	10
16	26
S= 44	

б) Числа 10 и 3

Ответ

10	13
23	13
23	36
S= 56	

Особые трудности вызывают ситуации, когда после THEN или ELSE должно следовать более одного операто-

ра, а также если вариантов выбора действий более двух. В этих случаях целесообразно использовать блочный формат условного оператора, для отработки которого предлагаются следующие упражнения.

5. Дана программная строка:

IF X > Y THEN X = A : Y = B ELSE Y = A : X = B
Замените ее блочным форматом условного оператора.

Ответ

```

IF X > Y THEN
  X = A
  Y = B
ELSE
  Y = A
  X = B
END IF

```

6. Объясните назначение следующей программы:

```

INPUT "X="; X
INPUT "Y="; Y
IF X > Y THEN
  PRINT "Большее число: "; X
ELSEIF Y > X THEN
  PRINT "Большее число: "; Y
ELSE
  PRINT "Числа равны"
END IF

```

Ответ. По введенным числам X и Y определяется и выводится на экран большее из них. В случае равенства чисел выводится соответствующее сообщение.

7. Объясните назначение следующей программы:

```

INPUT "X="; X
INPUT "Y="; Y
A = (X + Y) / 2
B = 2 * X * Y
IF X > Y THEN
  X = A
  Y = B
ELSEIF Y > X THEN
  X = B
  Y = A
END IF
PRINT "X="; X; "Y="; Y

```

Ответ. Большее из введенных двух чисел X и Y заменяется их полусуммой, а меньшее — их удвоенным произведением. В случае равенства исходных чисел значения переменных X и Y сохраняются неизменными. Результат выводится на экран.

ДАННЫЕ ДЛЯ ВЫПЛАТЫ АВТОРСКОГО ГОНОРАРА ЗА ПУБЛИКАЦИИ В "ИНФОРМАТИКЕ"

Дорогие авторы! Отправляя материалы для публикации в нашу газету, прикладывайте, пожалуйста, заполненный бланк с данными, необходимыми для выплаты гонорара.

Фамилия _____ Имя _____

Отчество _____

Приложение "ИНФОРМАТИКА"

Паспортные данные

серия _____

номер _____

когда выдан _____

кем выдан _____

Адрес прописки

индекс _____

город _____

улица _____

дом _____

корпус _____

квартира _____

Почтовый адрес для отправки гонорара

жители Москвы получают гонорар в редакции, все остальные должны обязательно указать данный адрес, даже если он совпадает с адресом прописки

индекс _____ город _____

улица _____

дом _____ корпус _____ квартира _____

телефон _____

Дата рождения _____

Место рождения _____

Необходимость почтового перевода (да/нет) _____

Номер страхового полиса Пенсионного фонда _____

Номер свидетельства о постановке на учет в налоговом управлении (если есть) _____

П л а з м е н н ы е

Д и с п л е и

Развитие персональных компьютеров стимулировало не только существенное улучшение параметров устройств отображения информации на базе электронно-лучевых трубок, но и дало новый импульс развития другим дисплейным технологиям, до той поры имевшим узкоспециальные применения. Этот импульс оказался настолько мощным, что дисплеи, выполненные по альтернативным технологиям, под общим названием "плоскопанельные дисплеи" [далее — FPD (Flat Panel Displays)] не только стали успешно конкурировать с электронно-лучевыми трубками в области компьютерных дисплеев, но и начали теснить их в традиционной телевизионной области.

Энтузиазм множества ведущих дисплейных фирм (в первую очередь японских и корейских), вкладывающих сотни миллионов долларов в развитие новых плоскопанельных дисплейных технологий, объясняется многомиллиардным рынком телевизионных и компьютерных дисплеев нового поколения.

Если среди плоскопанельных компьютерных дисплеев сегодня преобладают жидкокристаллические дисплеи, то новый перспективный рынок настенных телевизионных дисплеев (диагональ экрана — более 1 м, толщина — менее 10 см) наиболее успешно осваивается производителями плазменных дисплеев.

Еще лет пятьдесят назад было открыто одно интересное явление. Как оказалось, если катод заострить на манер швейной иглы, то электромагнитное поле в состоянии самостоятельно "выдергивать" из него свободные электроны. Необходимо только подать напряжение. По такому принципу работают лампы дневного света. Вылетающие электроны ионизируют инертный газ, чем заставляют его светиться.

Идея использования газового разряда в средствах отображения не нова. Подобные устройства выпускались много лет назад в СССР в Рязани в НПО "Плазма". Однако размер элемента изображения был достаточно велик, так что для получения приличного изображения было нужно создавать огромные табло. Изображение было некачественным, передавалось мало цветов, устройства были крайне ненадежными.

Трудность заключалась в отработке технологии получения "игольчатых матриц". Эту проблему решили в университете шт. Иллинойс в 1966 г., и в начале семидесятых годов компания Owens-Illinois довела проект до коммерческого состояния. В восьмидесятых годах эту идею пытались воплотить в реальный коммерческий продукт компании Burroughs и IBM, но тогда еще безуспешно.

Разработки дисплеев на основе плазменного эффекта начались в 1968 году. Первые образцы были монохромными и основывались на принципе газового разряда постоянного тока. Однако с тех пор, как был осуществлен переход на технологию разряда переменного тока, впервые примененную компанией Fujitsu, устройство дисплеев радикально не изменилось.

Надо сказать, что идея плазменной панели появилась вовсе не из чисто научного интереса. Ни одна из существовавших технологий не могла справиться с двумя простыми задачами: добиться высококачественной цветопередачи без неизбежной потери яркости и создать телевизор с широким экраном, чтобы он при этом не занимал всю площадь комнаты. А плазменные панели (PDP), тогда только теоретически, подобную задачу как раз могли решить. Первое время опытные плазменные экраны были монохромными (оранжевыми) и могли удовлетворить спрос только специфических потребителей, которым требовалась прежде всего большая площадь изображения. Поэтому первую партию PDP (около тысячи штук) купила нью-йоркская фондовая биржа.

Направление плазменных мониторов возродилось после того, как стало окончательно ясно, что ни ЖК-мониторы, ни ЭЛТ не в состоянии обеспечить получение недорогих экранов с большими диагоналями (более двадцати одного дюйма). Поэтому лидирующие производители бытовых телевизоров и компьютерных мониторов вновь вернулись к PDP.

Плазменная технология нового поколения идеально подходит для создания больших экранов. Она позволяет выпускать плоские и легкие мониторы глубиной всего 9 см. Поэтому, несмотря на большой экран, они могут быть установлены в любом месте — на стене, под потолком, на столе.

Благодаря широкому углу обзора изображение видно с любой точки. И, что самое главное, плазменные мониторы способны передать цвет и резкость, которые раньше были недостижимы при таком размере экрана.

Принцип работы плазменной панели состоит в управляемом холодном разряде разреженного газа (ксенона или неона), находящегося в ионизированном состоянии (холодная плазма).

PDP, так же, как и все остальные FPD, относятся к классу матричных индикаторов, в которых управление ячейками производится в простейшем случае с помощью двух перпендикулярных систем электродов.

При приложении к вертикальному и горизонтальному электродам напряжений, в сумме превышающих напряжение зажигания, в ячейке возникает газовый разряд, ограниченный барьерами.

Ультрафиолетовое излучение газового разряда возбуждает фотолюминофор, излучающий свет красного, зеленого или синего цвета. Люминофор обычно наносится на дно и боковые стенки ячейки, а иногда — тонким полупрозрачным слоем и на потолок ячейки. Верхняя система электродов делается прозрачной или достаточно узкой, для того чтобы не препятствовать выходу света. Рабочим элементом (пикселем), формирующим отдельную точку изображения, является группа из трех субпикселей, ответственных за три основных цвета соответственно.

Каждый субпиксель представляет собой отдельную микрокамеру, на стенках которой находится флуоресцирующее вещество одного из основных цветов (см. рис. 2). Пиксели находятся в точках пересечения прозрачных управляющих хром-медь-хромовых электродов, образующих прямоугольную сетку.

Три или четыре ячейки с разными цветами свечения образуют квадратный пиксель, как это показано на рис. 1.

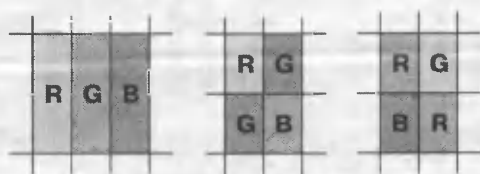


Рис. 1. Различные конфигурации пикселя

Для того чтобы "зажечь" пиксель, на два ортогональных друг другу питающий и управляющий электроды, в точке пересечения которых находится нужный пиксель, подается высокое управляющее переменное напряжение прямоугольной формы. Газ в ячейке отдает большую часть своих валентных электронов и переходит в состояние плазмы. Ионы и электроны попеременно собираются у электродов по разные стороны камеры, в зависимости от фазы управляющего напряжения. Для "поджига" на сканирующий электрод подается импульс, одноименные потенциалы складываются, вектор электростатического поля удваивает свою величину. Происходит разряд — часть заряженных ионов отдает энергию в виде излучения квантов света в ультрафиолетовом диапазоне (в зависимости от газа). В свою очередь, флуоресцирующее покрытие, находясь в зоне разряда, начинает излучать свет в видимом диапазоне, который и воспринимает наблюдатель. 97% ультрафиолетовой составляющей излучения, вредного для глаз, поглощается наружным стеклом. Яркость свечения люминофора определяется величиной управляющего напряжения, что, собственно говоря, и позволяет добиваться многомиллионного разнообразия цветов.

На рис. 2 показано схематическое изображение простейшей ячейки PDP постоянного тока.



Рис. 2. Ячейка постоянного тока

В PDP переменного тока используется более сложная ячейка, в которой электроды отделены от газового пространства диэлектрическими слоями. При наличии дополнительных электродов зажигание разряда в ячейке производится с помощью электродов 1 и 3, а поддержка горения так называемого "планарного разряда" — между электродами 1 и 2, как это показано на рис. 3.

Так как оба электрода в дисплеях переменного тока закрыты слоем диэлектрика, прямого разряда, как в моделях постоянного тока, не получается. Вместо этого каждый элемент (электрод-электрод) работает как очень емкий конденсатор. Этот принцип существенно продлевает срок жизни дисплея, оставаясь в сохранности электроды и увеличивая цветность и яркость. Однако в этом случае резко повышается цена устройства из-за усложняющейся управляющей электроники — для дисплеев переменного тока требуется более высокая частота.



42-дюймовый настенный PDP-телевизор фирмы NEC

Высокие яркость и контрастность, вдвое превышающие показатели обычного телевизора и в полтора раза — ЭЛТ-монитора, наряду с отсутствием дрожания являются большими преимуществами таких PDP-панелей. Высокая четкость изображения сохраняется на всей рабочей поверхности экрана. Кроме того, угол по отношению к нормали, под которым можно увидеть нормальное изображение на плазменных мониторах, существенно больше, чем у LCD-мониторов. К тому же плазменные панели не создают магнитных полей (что служит гарантией их безвредности для здоровья), не страдают от вибрации, как ЭЛТ-мониторы, а их небольшое время регенерации позволяет использовать их для отображения видео- и телесигнала.

Прозрачные электроды 1 и 2

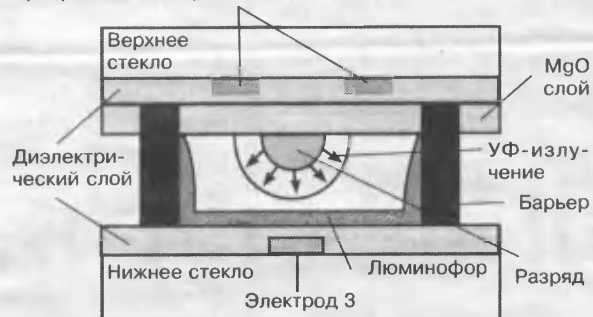


Рис. 3. Ячейка переменного тока

Отсутствие искажений и проблем сведения электронных лучей и их фокусировки присуще всем плоскопанельным дисплеям. Необходимо отметить и стойкость PDP-мониторов к электромагнитным полям, что позволяет использовать их в промышленных условиях, — даже мощный магнит, помещенный рядом с таким дисплеем, никак не повлияет на качество изображения. В домашних же условиях прямо на монитор можно поставить любые колонки, не опасаясь возникновения цветных пятен на экране.

Тем не менее плазменная технология отличается и некоторыми недочетами, иногда являющимися оборотной стороной достоинств. Тот факт, что размер коммерческих плазменных панелей обычно начинается с сорока дюймов, свидетельствует о том, что производство дисплеев меньшего размера пока что экономически нецелесообразно, поэтому мы вряд ли увидим плазменные панели, скажем, в портативных компьютерах. Это предположение подкрепляется и другим фактом: уровень энергопотребления "плазменных" подразумевает подключение их к сети и не оставляет никакой возможности работы от аккумуляторов. Еще один неприятный эффект, известный специалистам, — это интерференция, "перекрытие" микроразрядов в соседних элементах экрана. В результате подобного "смешивания" качество изображения, естественно, ухудшается. Кроме этого, свойство люминофорных элементов быстро ухудшаются, и экран становится менее ярким, поэтому срок службы плазменных мониторов ограничен 10 000 часов (это около 5 лет при офисном использовании).

Появление плазменных дисплеев сняло, пожалуй, последнее ограничение технологии плоских панелей — небольшой размер экрана, присущий LCD-панелям. Это позволило продвигать плазменную технологию прежде всего в самых прибыльных секторах — на рынке домашних кинотеатров, в секторе презентационного оборудования для корпораций и в секторе профессиональной графики.

Согласно прогнозам в ближайшие годы будет выпускаться от 2 до 4 миллионов телевизионных PDP. В настоящее время разработка и производство PDP сосредоточены в Японии и Юго-Восточной Азии. Наряду с повышением эффективности и качества важной задачей для производителей является снижение стоимости PDP (примерно со 150—200 до 30—60 долларов за один дюйм по диагонали). По прогнозам средняя стоимость 42-дюймового настенного телевизора снизится с 10 тысяч (1998 год) до 2,5 тысячи (2002 год) долларов США. Снижение стоимости связано с тремя факторами:

- с увеличением объема производства,
- с технологическими инновациями,
- со снижением стоимости материалов.

Использованные источники информации

1. Воронин А.А., Дедов В.П. Принципы построения и проблемы совершенствования плазменных дисплеев. "Оптический журнал" № 6, т. 6, июнь 1999, а также сайт <http://www.wpuls.net>.
2. Сергей Седых. Плазменные дисплеи. <http://www.computery.ru/upgrade>.
3. Антон Пермогоров. Плазменные дисплеи. <http://www.ferra.ru>.

Азы информатики

А.А. ДУВАНОВ

Материалы
Роботландского
университета

Книга 1. См. № 1, 2/2002

Книга 2. См. № 5, 6, 7, 8, 9, 11, 13, 14, 15, 16, 17, 18/2002

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

Книга 2. В мире информации (продолжение)

Книга для ученика

13. ИНФОРМАЦИОННЫЕ ОБЪЕКТЫ

Мир состоит из объектов

Можно считать, что мир вокруг нас состоит из объектов.

Однако термин “объект” не сообщает ничего нового о предметах и явлениях. Это просто удобное **обобщение**, которым заменяют названия конкретных предметов и явлений.

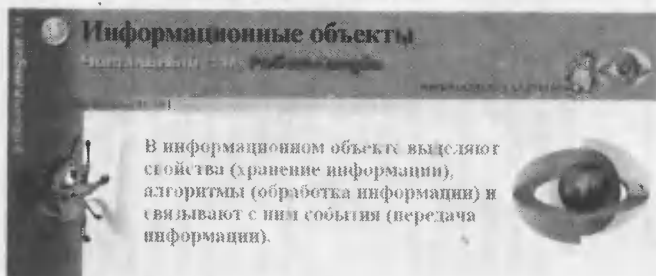
Свойство обобщения имеет не только термин “объект”.

Можно сказать: “посмотрите, там, на опушке, растут две березы и одна сосна”. А можно сказать: “на опушке растут три дерева”. Термин “дерево” заменяет собой названия конкретных видов деревьев, значит, он тоже является обобщающим термином.

Вам известны, конечно, и другие обобщающие термины. Словом “животные” можно воспользоваться в рассказе о том, что кукарекает, крикает и хрюкает у бабушки во дворе. А слово “кошки” можно употребить, описывая повадки льва, тигра и рыси.

И все же обобщающий термин “объект” имеет особое значение. Его можно использовать по отношению к любому предмету и явлению. Другие термины таким свойством не обладают, все они сами являются объектами.

Все явления и предметы окружающего мира можно изобразить очень большой (просто огромной) иерархической схемой, корнем которой является понятие объекта:



Читальный зал

Понятие объекта

Предметы и явления окружающего мира часто называют **объектами**.

Вы не раз работали с объектами на экране монитора (**экранные объекты**). Обычно это картинки, на которых изображены кнопки, значки, прямоугольные области.

Вы не раз слышали фразу, подобную такой: “Объектом исследования ученых стали северные сияния, необычайно сильные в этом году”.

Термин “объект” очень удобен: им можно заменять несколько слов сразу.

Говорят: “на экране мы видим три объекта”, вместо того чтобы длинно перечислять: “на экране находятся две нарисованных кнопки и одна прямоугольная область с текстом внутри”.



объект

животное

насекомое

стрекоза

комар

рыба

карась

плотва

земноводное

лягушка

тритон

птица

синица

воробей

зверь

волк

заяц

наука

математика

геометрия

алгебра

информатика

алгоритмика

программирование

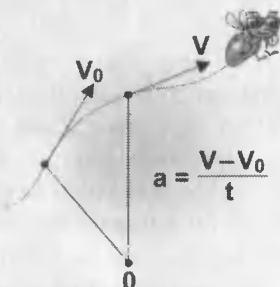
Объекты в информатике

Объектом в информатике, как и всюду, называют все, что угодно: число, таблицу, исполнителя, программу, кнопку на экране...

Однако информатика рассматривает объекты с точки зрения информационных процессов.

В таком подходе нет ничего удивительного: каждая наука рассматривает объекты со своей точки зрения.

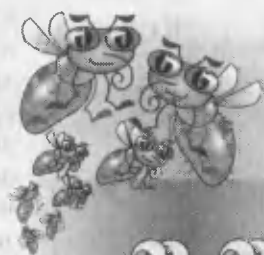
Физик, наблюдая полет пчелы, мысленно прикидывает формулы, которые описывают траекторию движения насекомого.



Биолог задумывается о числе глаз, крыльев и ножек.



ног — 6
крыльев — 4
глаз — 5



Информатик наблюдает за тем, как своим “танцем” пчела передает информацию сородичам о полях медоносных растений.

Обычно информатик выделяет в объекте три составляющие:

— Какую информацию объект хранит (**свойства объекта**).

— Как объект обрабатывает информацию (**алгоритмы объекта**).

— Какую информацию объект передает и принимает (**события объекта**).

Вася Кук решил исследовать свой телевизор как информационный объект.



Он взял три листочка. На одном написал: “свойства”, на другом — “алгоритмы”, на третьем — “события”. Вот что у него получилось:

СВОЙСТВА

цвет: черный
диагональ экрана: 53 см
вес: 20 кг



АЛГОРИТМЫ

включение/выключение: кнопка Ⓚ
переключение программ: кнопки ⊖ ⊕ блока PRG
регулирование звука: кнопки ⊖ ⊕ блока



СОБЫТИЯ

прием телепередачи
передача изображения и звука зрителю



Материальные и виртуальные объекты

Телевизор — это **материальный** объект. Слово “материальный” означает “реальный”, существующий на самом деле.

Корпус телевизора сделан из пластика, гладкого на ощупь. Телевизор достаточно тяжелый — не надо ронять его на ногу: будет больно. Если по телевизору стукнуть молотком, то скорее всего он выйдет из строя.

Телевизор нарисован в этой книге. Этот нарисованный телевизор, конечно, не является материальным (реальным) телевизором. Его нельзя потрогать, понюхать. Он не может упасть на пол. Нарисованный телевизор — это **виртуальный** объект, то есть не реальный, не существующий на самом деле.

С одним видом виртуальных объектов вы постоянно работаете на уроках информатики — это окна на Рабочем столе компьютера.

Информационные объекты

Понятие информационного объекта очень удобно для пользователей.

Для того чтобы посмотреть передачу по телевизору, не нужно знать, каким образом на экране возникает изображение и почему этот пластмассовый ящик говорит подобно человеку. Пользователю нужно только знать, как включить телевизор, как настроить его на нужную программу и как установить желаемую громкость звука. Иными словами, пользователь должен знать алгоритмы объекта и уметь ими пользоваться.

Полезно знать и свойства телевизора. Телевизор работает под очень высоким напряжением — вскрывать его опасно для жизни. Телевизор тяжелый — не надо ронять его на кошку. Телевизор хрупкий — не надо распрямлять на нем гвозди.

Информационные события, связанные с телевизором (это прием телепередачи, передача изображения и звука зрителю), позволяют использовать свойства и алгоритмы по назначению: посмотреть интересный мультфильм или узнать последние новости.

Менять свойства телевизора мы не можем, но можем их учитывать. А вот свойства виртуальных объектов менять легко. На странице электронной книги можно, например, изменить форму, цвет и размер геометрической фигуры:



Хорошо знаком вам виртуальный объект — окно на Рабочем столе компьютера. Вспомните, как можно ме-

нять такие его свойства, как активность, размер, расположение на экране.

Знакомы вам и алгоритмы окна:

- распаивание;
- восстановление;
- сворачивание;
- закрытие.

Некоторые события, связанные с окном:

- загрузка в окно нового документа;
- щелчок по кнопке внутри окна;
- изменение размера окна;
- закрытие окна.

Объектное программирование

Современное программирование — объектно-ориентированное.

Это означает, что программа, которой предстоит работать на компьютере, создается как набор информационных объектов.

Объекты имеют свойства — ячейки памяти для хранения информации и алгоритмы, которые обрабатывают **события**, — информацию, поступающую от других объектов.

Программа, составленная из объектов, не имеет одного общего алгоритма, а работает в результате информационного взаимодействия объектов.

Все происходит так, как в реальной жизни. Ведь мир вокруг нас состоит из отдельных объектов (людей, животных, предметов, явлений), которые существуют по своим собственным правилам. А общее развитие мира определяется тем, как отдельные объекты взаимодействуют друг с другом.

В Практикуме сегодняшнего урока вас поджидает исполнитель Конструктор. С его помощью вы создадите объектную программу и понаблюдаете ее в работе.

Программируют объекты так:

1. Сначала создают **описание** — шаблон, в котором определяют свойства и алгоритмы объекта.
2. Затем на базе готового описания создают столько **экземпляров** объекта, сколько требуется для программы.



Конспект

Объект — это обобщающий термин, которым обозначают любые предметы или явления, реальные или вымышленные.

Материальный объект — это реальный предмет или явление.

Виртуальный объект — это имитация реального или придуманного объекта (например, на экране компьютера).

Информационный объект — это объект с точки зрения информатики.

В информационном объекте выделяют **свойства** (хранение информации), **алгоритмы** (обработка информации) и связывают с ним **события** (передача информации).

Объектное программирование — это создание программы как совокупности отдельных программных объектов.

Для программирования объекта сначала составляют его **описание**, затем создают нужное число **экземпляров**.

Вопросы

Вопросы

1. Что называют объектом?
2. Что такое материальный объект?
3. Что такое виртуальный объект?
4. Что такое экранный объект?
5. Какие термины называют обобщающими?
6. Приведите пример иерархии обобщений.
7. Что понимают под терминами “физический объект”, “биологический объект”, “информационный объект”?
8. Какие три составляющие выделяют в информационном объекте?
9. Какие свойства и алгоритмы имеет информационный объект “окно на Рабочем столе компьютера”?
10. Какие информационные события связаны с объектом “окно на Рабочем столе компьютера”?
11. Расскажите, как можно менять свойства окна и запускать его алгоритмы.
12. Что такое объектное программирование?
13. Что понимают под термином “событие” в объектном программировании?
14. Как программируют объекты?

Задания на дом

Задания на дом

1. Придумайте примеры материальных или виртуальных объектов.

Представьте описания этих объектов в виде таблицы с графами Имя, Свойства, Алгоритмы, События.

2. В аналогичной таблице разместите сведения об объектах, о которых говорится в следующих загадках:

На потолке, в уголке,
Висит сито — не руками свито.

Четыре брата одним кушаком подпоясаны,
Под одной шляпой стоят.

Перья есть, а не летает;
Ног нет, а не догонишь.

Своих глаз нет, а другим видеть помогают.

Слабее ветра.
Ног нет, а идет.
Глаз нет, а плачет.

Ты от нее — она за тобой;
Ты за ней — она от тебя.

Бежит без ног,
Рукава без рук.

Держусь я только на ходу,
А если встану — упаду.

Расходится детина —
Удержу нет,
Уляжется — не видать,
Не слышать.

3. В аналогичной таблице разместите сведения об объектах, о которых говорится в следующем стихотворении (из В.Агафонова “Твой друг — компьютер”):

Оглянись, дружок, вокруг!
Вот ??? — верный друг.
Он всегда тебе поможет:
Сложит, вычитет и умножит!

Наверху машины всей
Размещается ??? —
Словно смелый капитан!
А на нем — горит экран.

Ну а рядом — главный блок:
Там бежит электроток
К самым важным микросхемам.
Этот блок зовут ???.

Он читает, он считает,
Сотни слов запоминает!
Он мыслитель — хоть куда!
Только знаешь, в чем беда?

Выключаешь — все забудет,
Ничего он знать не будет —
Прямо как Рассеянный
С улицы Бассейной!

Кто ж ему поможет? Вот,
Познакомьтесь — ???!
Коль с компьютером ты дружен —
То и справочник не нужен:

В упаковке, как конфета,
Быстро вертится ??? —
Там записаны программы
И для папы, и для мамы!

Это вот — ???.
Вот где пальцам физкультура
И гимнастика нужны!
Пальцы прыгать тут должны!

А вот это ???, братцы.
Тут нам надо разобраться,
Для чего же этот ящик?
Он в себя бумагу втащит,

И сейчас же буквы, точки,
Запятыя — строчка к строчке —
Напечатает в момент!
Очень нужный инструмент.

В зоопарке есть зайчишка.
У компьютера есть ???!
Эта мышка не простая,
Эта мышка вот такая:

Скромный серый коробок,
Длинный тонкий проводок,
Ну а на коробке —
Две или три кнопки.

Есть компьютер? Скажем прямо:
 Нам теперь нужна ???!
 Где живет она? В дискете!
 А порою — на кассете.

С ней компьютер очень дружен
 И всегда он ей послушен.
 И на нем ты скоро сможешь
 Составлять ??? тоже!

Вот теперь, дружок, немного
 Ты с компьютером знаком.
 Если дальняя дорога
 Приведет его в твой дом,
 То скажи ты всем вокруг:
 “Вот компьютер — верный друг!”

4. Запишите загадки или стихи с описаниями знакомых объектов.

Представьте сведения о них в таблице.

Практикум

Практикум

Визуальное программирование

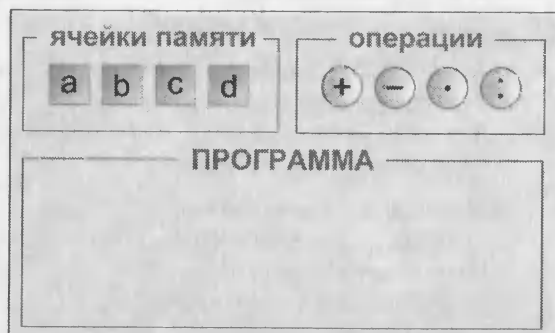
Как вы знаете, программы записываются на языке программирования.

Например, для нахождения произведения двух чисел можно записать такую программу на языке Малыша:

ввод (a)	Ввод первого числа
ввод (b)	Ввод второго числа
$d := a * b$	Нахождение произведения
вывод (d)	Вывод результата

Существуют специальные программные среды, они называются **визуальные редакторы**, которые способны освободить пользователя от записи программ на языке программирования.

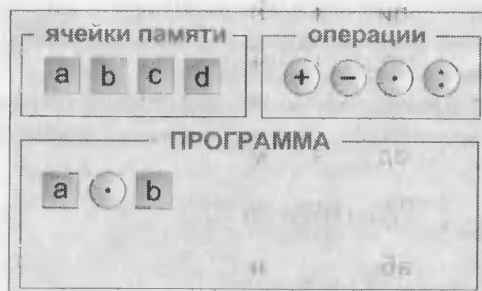
Вот пример такой среды для визуального программирования Малыша:



Визуальная среда Малыша состоит из трех областей:

- Область ячеек памяти.
- Область операций.
- Область для сборки программы.

Для визуальной сборки программы нахождения произведения двух чисел нужно перетащить мышкой в программную область следующие блоки:



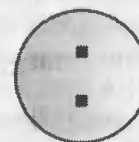
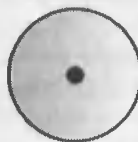
Теперь запускается конструктор, встроенный в среду, и он строит программу на языке программирования:

```

ввод (a)
ввод (b)
c := a * b
вывод (c)
  
```

Как видите, визуальный конструктор использовал для хранения результата ячейку c, а не d, как это сделано в “ручной” программе. И это понятно: конструктор взял для результата первую свободную ячейку памяти (a и b заняты под входные данные).

Заметьте, что пользователь работает с привычными знаками операций:



а визуальный конструктор автоматически переводит их в записи, принятые в языке программирования Малыша: знаки * и /.

Визуальное программирование объектов

Сегодня вы познакомитесь с исполнителем Конструктор.

Конструктор — это среда для визуального программирования объектов.



Далее приводится краткое описание экранных областей Конструктора.

Область "Описание объекта"

Щелчки по чувствительным областям задают свойства и алгоритмы объекта.

Кнопка "Создать экземпляр"

Экземпляр объекта создается в "Рабочей комнате".

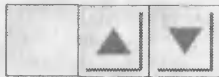
Область "Рабочая комната"

Щелчок по экземпляру объекта выводит панель с "паспортом" объекта и кнопкой удаления:

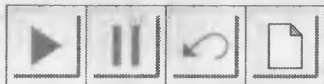


Панель управления

Выбор комнаты выполняется при помощи кнопок:



Команды "Пуск", "Пауза", "Начало", "Очистка" задаются при помощи кнопок:



Алгоритм работы с исполнителем

- Создайте несколько объектов в комнате по следующему алгоритму:
 - 1.1. Задайте свойства и алгоритмы на панели "Описание объекта".
 - 1.2. Создайте объект кнопкой "Создать экземпляр".
- Кнопкой "Пуск" запустите программу комнаты.

Задания

- Создайте два объекта, которые симпатизируют друг другу.
- Создайте два объекта, которые боятся друг друга.
- Создайте две группы объектов. В каждой группе объекты симпатизируют друг другу, но боятся членов другой группы.
- Попробуйте перекрасить пол комнаты в зеленый цвет.
- Создайте две группы объектов. Одна из них красит пол в зеленый, другая — в синий цвет.
- Создайте две группы объектов. Одна из них красит пол в зеленый цвет, а другая тщательно перекрашивает назад в белый.

Зачетный класс

Зачетный класс

- Постройте иерархию обобщений.
кошка зверь Мурка объект животное
- Постройте иерархию обобщений.
книга произведение
"Малыш и Карлсон" предмет объект
- Постройте иерархию обобщений.
устройство объект
Малыш компьютер исполнитель
- Постройте иерархию обобщений.
материальный объект объект рубанок слесарный
инструмент инструмент предмет
- Алгоритм объекта преобразует входное число, добавляет к результату свойство c и выдает ответ на выход.



Результаты опытов с объектом:

Вход	Выход
0	5
1	9
2	13
3	17

Угадайте алгоритм и свойство объекта.

- Алгоритм объекта преобразует вход, к которому предварительно добавляется свойство b , и выдает ответ на выход.

Результаты опытов с объектом:

Вход	Выход
0	6
1	8
2	10
3	12

Угадайте алгоритм и свойство объекта.

- Объект работает по такой схеме:

- 1) К входу добавляется свойство b .
- 2) К результату пункта 1 применяется алгоритм объекта.
- 3) К результату пункта 2 добавляется свойство b .
- 4) Результат пункта 3 отправляется на выход.

Результаты опытов с объектом:

Вход	Выход
0	8
1	11
2	14
3	17

Угадайте алгоритм и свойство объекта.

- Объект кодирует входную строку, используя алфавитный циклический сдвиг влево или вправо. Величина сдвига определяется свойством n .

Результаты опытов с объектом:

Вход	Выход
мама	ляля
папа	ояоя
лиса	кзря

Угадайте алгоритм и свойство объекта.

Книга для учителя

13. ИНФОРМАЦИОННЫЕ ОБЪЕКТЫ

Методология программирования

Программирование для компьютера предполагает у профессионального старателя наличие удивительной памяти (чтобы удерживать в голове все разветвления алгоритма) и потрясающей аккуратности (чтобы не допустить в ходе кодирования малейшей опечатки, простительной в обычной жизни).

Иными словами, программист должен сам быть немножко компьютером.

Такие позиции программирование занимало на заре развития компьютерной информатики, создавая профессии ореол элитности. На программиста смотрели как на шамана. Стать программистом было так же сложно, как космонавтом, и почти столь же почетно.

Понятно, что такое положение вещей совершенно не совместимо с бурным развитием компьютерных технологий, которое настоятельно требует пополнения программистского штата широкими народными массами.

Таким вот образом, развивая методы профессии и создавая полезный инструментарий, программисты собственными руками исключили себя из “отряда космонавтов” и записали в штат официантов привокзальной чебуречной.

Да... Есть о чем взгрустнуть тем, кто кропал программы под стрекот пишмаша лет 20 назад. А теперь каждый любитель может взять визуальный редактор, пощелкать в нем мышкой и: смотри ты! из-под его веселого пера на винт вылетает мегабайтный автоматический код, да такой, который и не снился самым шаманистым из ранних программистских гуров.

Ничего не поделаешь — прогресс! И на самом деле, очень хорошо, что этот прогресс есть. Теперь уже программистами называют не только тех, кто знает компьютер и язык программирования. Программист — это тот, кто способен провести формализацию и спроектировать быстрый и надежный алгоритм (впрочем, такая роль программисту отводилась всегда, еще классик Э.Дейкстра говорил, что самый главный язык, который должен знать программист, — это свой родной, на котором он изъясняется в повседневной жизни).

Рассмотрим кратко, что же случилось с программированием за эти быстро пролетевшие годы.

Структурное программирование

Все началось с того, что Э.Дейкстра предложил “структурный метод” проектирования программ. Этот подход предполагает разработку программы методом “сверху вниз” (метод постепенной детализации): сначала создается основной модуль, который описывает свойства и алгоритмы программы на верхнем иерархическом уровне, затем создаются подчиненные модули второго уровня, третьего и так далее, пока весь алгоритм не будет описан программным кодом.

Структурное программирование позволяет не просто разделять большой программный проект на небольшие простые модули, каждый из которых создается и отлаживается отдельно, но и проводить разработку, следуя иерархической сущности моделируемой системы. Нисходящая разработка позволяет обнаруживать и устранять серьезные ошибки на ранних стадиях работы над проектом (ведь ошибки на верхних этажах иерархии гораздо весомее тех, которые расположены на ее листьях).

Другой характерной чертой структурного программирования являются специальные языковые конструкции — структуры, которые позволяют конструировать собственные типы данных, отражающие информационную сущность исходной задачи (конструкция **struct** в Си, **record** в Паскале).

Объектное программирование

Объектно-ориентированное программирование (ООП) — современный способ создания программных кодов, который пришел на смену структурному программированию. Вернее, объектное программирование не заменило структурное, а довело его до логического совершенства.

В ООП-языках можно конструировать не только данные, но данные плюс процедуры (функции):

объект = данные + процедуры

В терминах Читального зала:

объект = свойства + алгоритмы

Итак, идея объекта очень проста — рассматривать данные и процедуры как одно целое.

Что продуктивного содержится в понятии объекта?

Инкапсуляция

Этим термином обозначают скрывание внутреннего устройства объекта. Объект рассматривается как “черный ящик”. Известны свойства объекта, то есть доступные извне переменные. Известны алгоритмы объекта, то есть доступные извне процедуры. Но как устроены эти процедуры, пользователю не сообщается. Пользователь не знает и о том, есть ли у объекта дополнительные внутренние вспомогательные процедуры и переменные и как они связаны с доступными свойствами и алгоритмами.

Вот телевизор — это сложное устройство. Но зачем пользователю знать, что внутри этого объекта? Достаточно уметь нажимать кнопки на пульте, чтобы заставить ящик показывать нужную программу с подходящей громкостью.

Инкапсуляция удобна, правда? Она позволяет слабому человеческому разуму справиться с огромными информационными потоками, закрывая глаза на несущественные в данный момент детали.

Объект и экземпляр объекта

Вообще говоря, конкретный телевизор в терминологии ООП — это не объект, а экземпляр объекта. Объектом является комплект документации на заводе, по ко-

торой изготавливаются телевизоры. Все телевизоры, которые сходят с конвейера, имеют одни и те же **свойства** и одни и те же **алгоритмы** управления.

То же и в программировании. Объект — это шаблон. Экземпляр объекта — это рабочая копия.

Событийное управление

Управление логикой программного кода при помощи событий — это вторая характерная черта ООП наряду с понятиями “свойства” и “алгоритмы” объекта.

Немного упрощенно можно так объяснить разницу между структурным и объектным программированием.

Программа, написанная без принципов ООП, представляет собой иерархию логической структуры, описанную на языке программирования:

```

главный модуль
  модуль01
    модуль0101
      модуль010101
        модуль010102
          ...
        ...
      ...
    ...
  ...
модуль02
  модуль0201
    модуль020101
      модуль020102
        ...
      ...
    ...
  ...
...

```

Данные, попадающие на обработку в главный модуль, следуя логике программы, проходят по ветвям логической структуры, пока не окажутся полностью обработанными в каком-нибудь листе иерархии.

Можно сказать, что в классическом программировании логика первична, а данные вторичны.

В объектном программировании нет единого алгоритма управления. Есть независимые друг от друга объекты:

```

объект01
объект02
объект03
...

```

Управляют работой программы события. События могут быть системными (мышь, клавиатура...), но их могут создавать и сами объекты.

Таким образом, в ООП первичны данные, а алгоритмы вторичны. Алгоритм начинает работать только тогда, когда его к этому побуждают полученные события (данные).

Методические замечания

Понятие объекта, разделение множества объектов на материальные, виртуальные, физические, биологические, информационные носители, конечно, довольно абстрактный характер.

Можно надеяться, что предыдущие уроки этой и первой книги подготовили ребенка к серьезному разговору, завершающему обзор информационной картины

мира. Если вы уверены, что это не так, можно исключить изучение этой темы из ваших планов: разговор об информационных объектах будет начинаться в курсе еще не раз.

Материал урока можно разделить на две части:

1. Общее представление об объекте, классификация объектов, рассмотрение объектов с разных профессиональных точек зрения.
2. Рассмотрение объектов с точки зрения информатики (информационные объекты), программирование объектов.

Первая часть имеет практическую поддержку заданиями Зачетного класса на построение иерархии сообщений.

Вторая часть темы поддержана Практикумом (исполнитель Конструктор) и заданиями Зачетного класса на угадывание свойств и алгоритмов объекта.

Авторское обоснование урока

Совершенно привычным для всех является понятие объекта как обобщающего термина. Объект — это корень иерархии всех на свете предметов (береза), понятий (дерево) и явлений (гроза в начале мая).

С такой трактовкой понятия объекта дети постоянно сталкивались на протяжении двух книг (первое определение объекта было дано в нулевом уроке первой книги). Теперь необходимо закрепить интуитивный опыт иерархической схемой с объектом в корне.

Итак, объект — это удобное обозначение для любого предмета, понятия или явления. Ничего нового этот термин в предмете, понятии или явлении не открывает. Это просто знаковый термин.

В первой части урока обобщаются те привычные представления, которые дети уже имеют. Они хорошо знакомы на практике с понятием иерархии, поэтому показать им еще раз иерархическую схему очень полезно: она соединит прежний интуитивный опыт и первое определение объекта с навыками иерархических построений, логично завершив дело, начатое еще в нулевом уроке первой книги.

Дети поймут, что объект — это корень всех иерархий на свете. Именно это я и хотел отразить в первой части Читального зала (материальные и виртуальные объекты — это первый уровень возможной классификационной схемы).

Далее я хотел показать, что каждая наука изучает объекты со своих точек зрения. Все науки изучают одно и то же (мир вокруг нас, то есть объекты), но подходят к предмету изучения с разных исходных позиций (отдельно рассматривая физику, химию, биологию, географию, социологию объекта).

Слабый человеческий разум не может открывать настоящие мировые законы, в которых учитывались бы одновременно все на свете стороны предметов, понятий и явлений. Он вынужден строить разные модели (подходы) и работать параллельно над одним и тем же в разных мировоззренческих плоскостях.

Вот почему возникли в Читальном зале три иллюстрированных точки зрения на полет пчелы.

Наконец, я хотел сосредоточить внимание читателя на точке зрения информатика.

Информатик смотрит на объект со своей “колокольни”: он пытается разобраться, как объект работает с информацией.

Но прежде всего, говоря “объект”, информатик, как и все остальные, использует этот термин как обобщающий заменитель других слов.

Например, еще до эры ООП информатик вполне мог назвать объектом нарисованную на экране “бутявку”, имея в виду всего лишь выделить этот рисунок среди других, но совсем не его информационную сущность (свойства и алгоритмы). Такого объекта (в понимании современного ООП), как “бутявка” у него в программе не было.

Бутявка перерисовывалась на новом месте программой, которая следила за клавишами управления курсора. Бутявка “кушала” других бутявок по каким-то правилам, но эти правила не были заложены внутри самой бутявки (хотя это выглядело именно так), а эти правила были заданы алгоритмом внешней (по отношению к бутявке) программы.

Иными словами, до эры ООП информатик моделировал информационную суть бутявки не в ней самой, а во внешней управляющей оболочке. Это не совсем согласовывалось с реальным мировым устройством, и такое моделирование было достаточно неудобным.

Философия ООП все вернула в естественное русло: у каждой “бутявки” появились свои свойства и алгоритмы, а управление было поручено событиям. Таким образом, ООП наконец поставило программирование с головы на ноги и программировать сложные системы стало гораздо проще.

Все это, конечно, рассказывать детям преждевременно. Моя цель была донести сам факт существования различных точек зрения на объект и выделить среди них точку зрения информатики.

Точка зрения информатики достаточно привычна: рассмотреть информационные процессы, связанные с объектом:

- хранение информации (свойства);
- обработка информации (алгоритмы);
- передача информации (события).

При рассмотрении конкретных объектов могут возникнуть трудности в выделении его свойств и алгоритмов. Например, объем и площадь мяча — это свойства или алгоритмы?

На самом деле такое разделение довольно условное. Например, объект Прямоугольник может задаваться свойствами “ширина”, “длина” и иметь алгоритм для вычисления своей площади.

Но объект Прямоугольник может задаваться свойствами “ширина”, “площадь” (что, конечно, немного неестественно) и иметь алгоритм для вычисления своей длины.

То есть, когда речь идет о программных объектах, разделение на алгоритмы и свойства выполняет программист, и делает это так, как считает нужным (удобным).

Рассматривая объекты из реальной жизни, мы фактически выделяем свойства и алгоритмы у тех информационных моделей, которые мысленно строим, наблюдая поведение реального объекта, значит, такое разделение может также носить неоднозначный характер.

Завершая разговор об информационных объектах, я хотел показать технологию построения их программных моделей (описание объекта, экземпляр объекта), принцип работы объектной программы (нет общего управляющего алгоритма), в отличие от обычной (есть общий управляющий алгоритм).

Этот раздел Читального зала ориентирован не столько на рациональное, сколько на чувственное восприятие. Расчет здесь на постепенное привыкание юного информатика к понятиям, разговор о которых будет продолжен в курсе.

Задания Зачетного класса

Для успешного выполнения заданий, связанных с угадыванием свойств и алгоритмов объекта, полезно предварительно обсудить в классе методику решений подобных задач.

Задание 5

Объект обрабатывает входную информацию по формуле:

$$\text{выход} = k * \text{вход} + c$$

Понятно, что свойство c однозначно определяется при подаче на вход нуля. Еще один опыт с ненулевым входом определяет алгоритм (множитель k).

Задание 6

Объект обрабатывает входную информацию по формуле:

$$\text{выход} = k * (\text{вход} + b)$$

Вход, равный нулю, образует выход, на котором свойство b умножается на коэффициент алгоритма k :

$$y_0 = k * b$$

Если теперь на вход поступит 1, то на выходе будем иметь:

$$y_1 = k + k * b = k + y_0$$

Коэффициент k определяется однозначно, он равен:

$$k = y_1 - y_0$$

Задание 7

Объект обрабатывает входную информацию по формуле:

$$\text{выход} = k * (\text{вход} + b) + b$$

Вход, равный нулю, образует выход:

$$y_0 = k * b + b$$

Единица на входе дает:

$$y_1 = k + k * b + b = k + y_0$$

Коэффициент k определяется так же, как и в предыдущей задаче:

$$k = y_1 - y_0$$

Найти b теперь не составляет труда.

Автор не настаивает на изложении детям этого формализма. Решить задачи вполне можно чисто опытным путем, учитывая конечность множества значений k в заданиях электронной книги.

Ответы на вопросы

1. Объект — это обобщающий термин, которым обозначают любые предметы или явления, реальные или вымышленные.

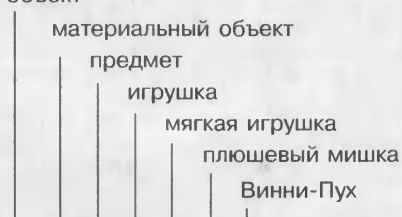
2. Материальный объект — это реальный предмет или явление.

3. Виртуальный объект — это имитация реального или придуманного объекта (например, на экране компьютера).

4. Экранными объектами называют объекты на экране компьютера. Это окна, кнопки, меню, строки ввода, полосы прокрутки...

5. Если одним термином можно заменить несколько других, то его называют обобщающим. Например, биологический термин “кошка” обобщает термины “рысь”, “лев”, “тигр”.

6. объект



7. Каждый объект можно рассматривать с разных точек зрения. Один и тот же объект физики называют физическим, биологи — биологическим, информатики — информационным объектом.

8. Информационный объект — это объект с точки зрения информатики. В информационном объекте выделяют **свойства** (хранение информации), **алгоритмы** (обработка информации) и **события** (передача информации).

9. Свойства:

- активность;
- размер;
- расположение на экране.

Алгоритмы:

- распаивание;
- восстановление;
- сворачивание;
- закрытие.

10.

- загрузка в окно нового документа;
- щелчок по области внутри окна;
- двойной щелчок по области внутри окна;
- зависание мыши над областью окна;
- изменение размера окна;
- изменение положения окна;
- закрытие окна.

11. Свойства:

- Активность: щелчок по окну или изображающей кнопке на Панели задач.

— Размер: протяжка за край окна.

— Расположение на экране: протяжка окна за заголовок.

Алгоритмы:

— Распаивание: кнопкой

— Восстановление: кнопкой

— Сворачивание: кнопкой

— Закрытие: кнопкой

12. Объектное программирование — это создание программы как совокупности отдельных программных объектов.

13. События управляют работой объектной программы. События могут быть системными (мышь, клавиатура...), их могут создавать и сами объекты.

14. Для программирования объекта сначала составляют его описание, затем создают нужное число экземпляров.

Решения зачетного класса

1.

объект
животное
зверь
кошка
Мурка

2.

объект
предмет
произведение
книга
“Малыш и Карлсон”

3.

объект
устройство
исполнитель
компьютер
Малыш

4.

объект
материальный объект
предмет
инструмент
слесарный инструмент
рубанок

5.

Алгоритм: умножение на 4
Свойство: 5

6.

Алгоритм: умножение на 2
Свойство: 3

7.

Алгоритм: умножение на 3
Свойство: 2

8.

Алгоритм: сдвиг влево
Свойство: 1

Учебный чайнворд

Н.Г. Сербаева,

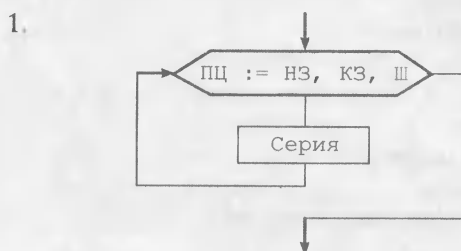
г. Усть-Илимск, Иркутская область

Напоминаем, что в чайнвордах буквы отгадываемых слов расположены последовательно, причем последняя буква каждого слова (кроме последнего) является первой буквой следующего.

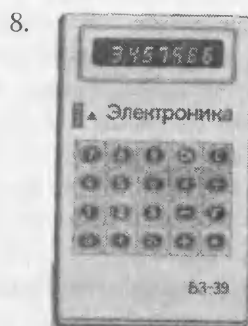
Ответы

1			2					3		
		9						10		
	16		21			22				4
									11	
				29			25			
								18	12	
				28	27	26				
	15		24			23				
8		20			19					
					14				13	5
		7					6			

1	Ц	И	К	2	Л	А	В	Л	Е	Й	3	С	С	Ы
	Т	О	9	Р	Е	Д	А	К	Т	О	10	Р	О	Л
	Я	Н	Т	Е	Р	Ф	Е	Й	17	С	К	Б	К	
	Л	16	И	Л	21	Л	И	С	22	П	Е	Р	А	О
	У	М	А	Ю	Н	К	Ц	И	Ф	Н	11	Т	Л	
	К	Е	Т	Ь	29	И	С	К	25	Я	О	Е	Е	Г
	Ь	Н	С	Н	Л	А	Р	Д	К	18	Р	12	Г	О
	Л	Е	И	О	28	И	27	С	26	О	Р	А	Е	Е
	А	15	К	Р	24	К	А	Б	23	А	Т	Р	К	Й
	8	К	И	20	К	Ы	З	19	Я	И	С	Р	У	Т
	И	Т	С	Й	О	Ж	14	Д	Й	А	Л	13	С	5
	Б	О	7	Р	О	Т	И	Н	О	6	М	Е	Д	О

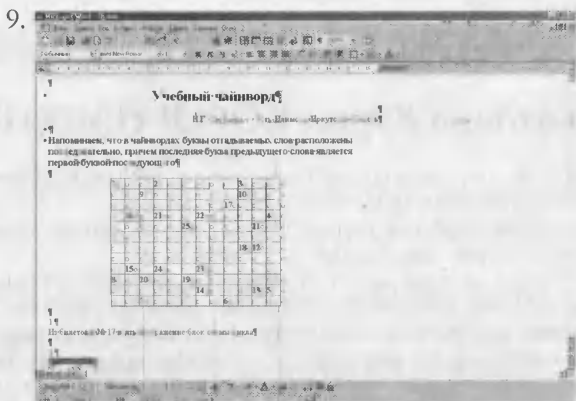


7. Учебный язык программирования, предназначенный для младших школьников.

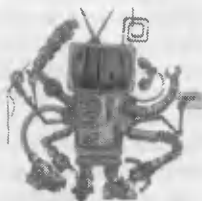


3. А1 — относительная ...;
 \$А\$1 — абсолютная ...;
 А\$1, \$А1 — смешанные ...

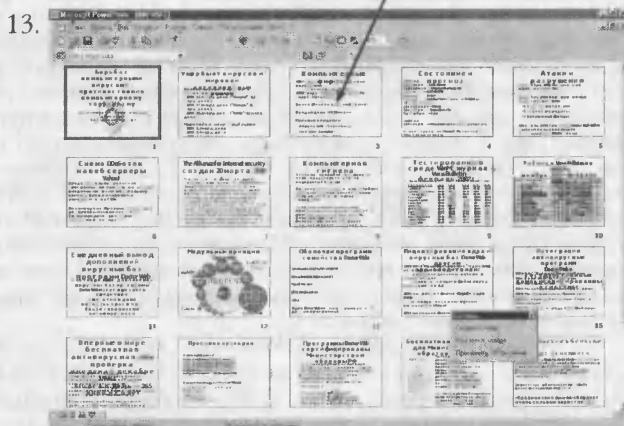
4. алг Знакомство (арг лит Имя, рез лит t)
нач
 вывод "Как тебя зовут ?"
 ввод Имя
 t := "Привет, " + Имя + "!"
 вывод t
кон



10. Впервые это слово ввел в оборот К.Чапек, а теперь оно знакомо даже малышам.



11. `<HTML> </HTML>`
`<HEAD> </HEAD>`



14.

15. Разработчик языка

```
10 INPUT "N = "; N
20 PRINT N
```

16. Средство взаимодействия, средство связи, сопряжения, согласования.

17.

18. Метод, использованный в алгоритме

алг КРАСИТЬ (цел X, Y, лит C)
арг X, Y, C

нач

если ЦВЕТ (X, Y, C) = ложь

то ТОЧКА (X, Y, C)

КРАСИТЬ (X+1, Y, C)

КРАСИТЬ (X+1, Y, C)

КРАСИТЬ (X+1, Y, C)

КРАСИТЬ (X+1, Y, C)

все

кон

19.

20. Основа современных микропроцессоров.

21. Язык программирования
(CAR' (1 2 3))



23.

24. Логическая операция A & B & C.

25. Центральная часть операционной системы.

26.

27. Язык, созданный Керниганом и Ритчи.

28. Логическая операция A or B or C.

29.

Компьютер для самых маленьких

Интервью подготовил Юрий КОРОГОДСКИЙ

Компьютер — важный инструмент в воспитании ребенка. Сейчас он необходим каждой семье, которая может использовать его не только как игровой центр, но и в качестве вспомогательного обучающего инструмента.

О различных аспектах использования компьютерных обучающих программ, предназначенных для самых маленьких детей дошкольного возраста, рассказывает главный специалист Департамента образовательных программ и стандартов общего образования Министерства образования Российской Федерации **Марина Серафимовна Цветкова**.

Марина Серафимовна имеет четырнадцатилетний опыт работы учителем информатики в школе. В 1998 году она стала лауреатом конкурса "Учитель года".

— *Марина Серафимовна, что нового принесли в жизнь маленьких детей информационные технологии?*

— На сегодня информационные технологии управляют процессом воспитания. Хотим мы этого или нет, но ребенка воспитывает внешний темп жизни. Не видеть этого просто неразумно. Мы делаем шаг, а то и два шага назад, когда отмахиваемся от процесса информатизации общества, в то время, когда ребенок уже готов к использованию этого нового качества жизни.

Новое качество жизни объективно присутствует сегодня, это так же, как в 50-е годы прошлого века непривычным фактором в жизни общества стали СМИ, то есть телевидение и радио. Тогда тоже были серьезные дебаты о том, можно ли ребенку смотреть телевизор и целесообразно ли вводить эти средства в процесс образования. Тот период своего развития общество пережило, и победил опыт, накопленный мировой педагогикой.

Сейчас у нас появилось такое понятие, как информационные технологии в семье. Информатизация семьи происходит как часть глобального процесса. Она не стала пока объектом пристального внимания психологов и педагогов. К сожалению, в нашем обществе информационные технологии долгое время работали только в сферах коллективного использования, и не каждая семья понимала их важность для процесса обучения. Но уже сейчас ни один родитель ребенка-школьника не скажет, что ему не нужен компьютер.

— *Что сам ребенок может ожидать от компьютера?*

— Чтобы использовать информационные технологии, нужно быть эмоционально к этому готовым. Здесь можно выделить три группы аспектов, регулирующих восприятие или отторжение того или иного ресурса. Первое — это мотив, то есть желание человека использовать данный ресурс. Второе — это интеллектуальная готовность (понимание того, чем и как нужно пользоваться). Третье — это практическая готовность (умение пользоваться этим ресурсом). Если мы нарушаем баланс между этими тремя группами, то мы тормозим процесс эффективного использования нового ресурса. Это касается любого человека и ребенка в том числе. Информационные технологии позволяют обозначить внутренние потенции не для мамы, а для самого ребенка. Нужно, чтобы ребенок, взаимодействуя с очень красивым наглядным устройством, сумел пробудить внутренние центры своих мотивов. В дошкольном возрасте нет задачи научить, есть задача пробудить интерес к обучению.

— *Может ли быть какой-то вред от чрезмерного использования малышом компьютерной техники?*

— Да, компьютер может влиять на ребенка в негативном плане. Тема вреда от компьютеров достаточно серьезно изучена психологами. Тут нужно помнить о том, что ребенок не защищен от вторжения чужой психики. Маленький ребенок воспитывается через живое общение. Он должен не просто слышать речь, он должен видеть выражение лица человека. Видеть, чтобы понимать, выражает тот информация со знаком "плюс" или же, наоборот, со знаком "минус". Маленький ребенок может самостоятельно работать на компьютере, но присутствовать рядом с ним обязательно должен или один из

родителей, или воспитатель. Очень опасно оставлять ребенка в обстановке безмолвия перед компьютером.

— *Сколько времени ребенок дошкольного возраста может проводить перед монитором?*

— Говоря на тему вреда от неправильного использования возможностей, которые предоставляет детям компьютер, мы выделяем два аспекта: психологический и физиологический (гигиенический). Но разделять эти аспекты нельзя, их нужно всегда рассматривать вместе. Прежде всего нужно сказать о нагрузке на скелет, которая вызвана неподвижным характером деятельности, а также о нагрузке на глаза. Для ребенка дошкольного возраста могут быть разрешены пяти-десятиминутные сеансы непрерывной работы на компьютере с дальнейшим достаточно большим перерывом. Малышу можно проводить три-четыре таких десятиминутных сеанса в день. И этого вполне достаточно, потому что работает психологическая нагрузка. Нагрузка эта связана с тем, что программа создана другим человеком, и у этого человека свое видение информации, свое представление о том, как она должна быть представлена на экране. Наконец, у этого человека свои ритмические особенности. Например, все помнят, что ребенок осознанно смотрит мультфильм только десять минут, а потом идет режим развлекательного просмотра. Малыш просто смотрит, как все, что находится на экране, движется.

Но ведь при просмотре мультфильма работают только уши, зрение и мозг, а при работе с компьютером нужно еще отвечать руками, реагировать на ту скорость, которую задает программа. В отличие от экрана телевизора, экран компьютера многозначен, каждая его картинка несет в себе интеллектуальную нагрузку. Если телевизор только подает информацию, то компьютер не просто ее предоставляет, но и приглашает к диалогу. На экране компьютера бывает до сотни обозначений. Для малыша это слишком много. Ему сложно просматривать сразу десять—двадцать фрагментов и понимать, что от него захочет программа.

— *С какого возраста можно начинать знакомство малыша с компьютером?*

— Начинать работать на компьютере ребенок может уже в возрасте трех-четырёх лет. Расстояние между малышом и монитором должно быть не менее чем пятьдесят сантиметров. Сидеть нужно на специальном стуле, который регулирует высоту сиденья. Поскольку ножки у ребенка будут висеть, нужна подставка для ног, чтобы скелет не уставал от этой нагрузки. Клавиатура и "мышь" должны быть расположены прямо под ладонями ребенка, чтобы он не тянулся. То есть столешница компьютера должна регулироваться по приближению к рукам ребенка. Спина у ребенка должна быть прямая, для этого у стула, на котором он сидит, должны быть высокая спинка и подголовник. Компьютер не должен вызывать у ребенка "чувство неудачника", и для этого необходимо, чтобы программа управлялась самим ребенком.

— *Марина Серафимовна, существуют ли обучающие программы для маленьких детей, на которые родителям стоит больше обращать внимание, когда они приходят в магазин?*

— В Министерстве образования Российской Федерации существует примерный перечень электронных средств, которые рекомендуются для использования при воспитании и обучении маленьких детей дошкольного возраста. Родителям стоит знать, что к таким средствам относятся: компьютерные азбуки и буквари для ознакомления работы с текстом, клавиатурные тренажеры с ненавязчивой скоростью работы, компьютерные раскраски и геометрические конструкторы, компьютерные лабиринты для управления объектом, компьютерные мозаики, логические игры, компьютерные энциклопедии, путешествия, топологические схемы, компьютерные учебники с иллюстрациями и заданиями (по технике безопасности и правилам движения), вычислительные игровые и алгоритмические среды, синтезаторы звука, игры-кресоворды и азбуки на иностранных языках.

Издательский дом Первое сентября

**23
ГАЗЕТЫ**



ИНФОРМАТИКА

Подписка по каталогу Роспечати «Газеты. Журналы» подписной индекс 32291

Во всех почтовых отделениях России продолжается подписка на 23 газеты Издательского дома «Первое сентября»

ПОДПИСНЫЕ ИНДЕКСЫ

Первое сентября

Первое сентября
Индекс 32024 и 32586



Английский язык
Индекс 32025 и 32587



Библиотека в школе
Индекс 33376 и 33377

БИОЛОГИЯ

Биология
Индекс 32026 и 32588

ВОСКРЕСНАЯ ШКОЛА

Воскресная школа
Индекс 32742 и 32743

ГЕОГРАФИЯ

География
Индекс 32027 и 32589



Дошкольное образование
Индекс 33373 и 33374

Здоровье детей

Здоровье детей
Индекс 32033 и 32590

ИНФОРМАТИКА

Информатика
Индекс 32291 и 32591

ИСКУССТВО

Искусство
Индекс 32584 и 32585

ИСТОРИЯ

История
Индекс 32028 и 32592

Литература

Литература
Индекс 32029 и 32593

МАТЕМАТИКА

Математика
Индекс 32030 и 32594



Начальная школа
Индекс 32031 и 32598

DEUTSCH

Немецкий язык
Индекс 32292 и 32599



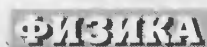
Русский язык
Индекс 32383 и 32601

Спорт в школе

Спорт в школе
Индекс 32384 и 32595

Управление школой

Управление школой
Индекс 32652 и 32653



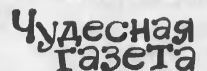
Физика
Индекс 32032 и 32596



Французский язык
Индекс 33371 и 33372



Химия
Индекс 32034 и 32597

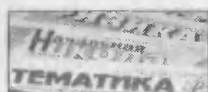


Чудесная газета
Индекс 35901 и 35902



Школьный психолог
Индекс 32898 и 32899

СПЕЦИАЛЬНОЕ ПРЕДЛОЖЕНИЕ



Комплект из 20 газет
Индекс 32744 и 32745

В комплект входят двадцать газет по цене восемнадцати.

Английский язык, Библиотека в школе, Биология, География, Здоровье детей, Информатика, Искусство, История, Литература, Математика, Начальная школа, Немецкий язык, Русский язык, Спорт в школе, Управление школой, Физика, Французский язык, Химия, Чудесная газета, Школьный психолог.

Льготы для организаций.

При подписке на комплект цена для организаций равна цене для индивидуальных подписчиков.

Издательский дом «Первое сентября»

Адрес: 121165, Москва, ул. Киевская, д. 24. Телефон: (095) 249-4758; факс: (095) 249-3184
E-mail: podpiska@1september.ru; Internet: www.1september.ru



Окончание. Начало на с. 1

Уатт решил прежде всего выяснить, почему машина Ньюкомена, поглощая так много топлива, производила такую незначительную работу. Он провел множество опытов, изучая свойства водяного пара. Настойчивость "мастера математических приборов" (так называлась должность Уатта) привела к цели. Ему стали ясны основные причины низкой производительности.

"Я пришел к твердому заключению, — писал Уатт, — что для построения совершенной паровой машины необходимо, чтобы цилиндр был всегда так же горяч, как и входящий в него пар; но, с другой стороны, конденсация пара для образования вакуума должна происходить при значительно более низкой температуре" [2].

Так появилась мысль разделить цилиндр на две части: горячую, куда будет впускаться пар и где будет совершаться рабочий ход, и холодную, где должен конденсироваться пар.

Конденсатор существенно увеличил экономичность машины. Уатт сделал еще несколько усовершенствований, создал соответствующий макет и вскоре получил патент на свое изобретение. Однако построить паровую машину ему долго не удавалось: то не хватало денег, то не было подходящих механизмов для обработки поршня и цилиндра, то не

получалось тонкой пригонки деталей. Дело сдвинулось с места только когда компаньоном Уатта стал крупный промышленник М.Болтон.

В 1774 году паровая машина Уатта была наконец построена и успешно прошла испытания. Как и следовало ожидать, она оказалась намного эффективнее машины Ньюкомена и быстро распространилась на шахтах.

А в 1784 году Уатт создал первую универсальную паровую машину двойного действия. В ней пар поступал в цилиндр попеременно, то с одной стороны поршня, то с другой. Поэтому поршень совершал и рабочий, и обратный ходы с помощью пара, чего не было в прежних машинах. Кроме того, Уатт сконструировал центробежный регулятор, который включил в конструкцию машины.

За эти годы Уатт изобрел также пять способов превращения возвратно-поступательного движения во вращательное. Один из них — замена кривошипно-шатунного механизма зубчатой передачей — являлся очень важным изобретением. Другой способ (а точнее, соответствующий механизм) и сейчас называют *параллелограммом Уатта*.

Уатт занимался не только усовершенствованием своей машины. Ему принадлежат такие важные изобретения, как паровой молот и паровое отопление. Он сконструировал также несколько счетных машин, копировальных прессов и других устройств [2, 4].

Свою изобретательскую деятельность Уатт закончил в 1785 году. Фирма "Болтон и Уатт" исправно приносила высокую прибыль.

Работы Джеймса Уатта получили исключительно высокую оценку. В 1784 году он был избран членом Эдинбургского королевского общества, через год стал членом Лондонского королевского общества, а несколько позже — членом Парижской академии наук.

По значимости достижений современники сравнивали Уатта с Архимедом. В 1932 году в Англии была учреждена Золотая международная медаль Джеймса Уатта (которую присуждают за крупные успехи в области машиностроения) [3]. Его именем названа единица мощности (*ватт*).

Литература

1. *Печерский Ю.Н.* Этюды о компьютерах. Кишинев: Штиница, 1989.
2. *Детская энциклопедия.* М.: Издательство Академии педагогических наук РСФСР, 1960. Т. 5.
3. *Самарин М.С.* Вольт, ампер, ом и другие. Единицы физических величин в технике связи. М.: Радио и связь, 1988.
4. *Храмов Ю.А.* Физики. Биографический справочник. 2-е изд. М.: Наука, Гл. редакция физико-математической литературы, 1983.

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 07.05.2002.
Номер подписан 07.05.2002.
Заказ № 12976
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
СЕНТЯБРЬ»,
ГЛАВНЫЙ
РЕДАКТОР —
А.СОЛОВЕЙЧИК

Газеты ИЗДАТЕЛЬСКОГО ДОМА: **Первое сентября** — гл. ред. Е.Бирюкова, **Английский язык** — гл. ред. А.Громушкина, **Библиотека в школе** — гл. ред. О.Громова, **Биология** — гл. ред. Н.Иванова, **Воскресная школа** — гл. ред. монах Киприан (Яценко), **География** — гл. ред. О.Коротова, **Дошкольное образование** — гл. ред. М.Аромштам, **Здоровье детей** — гл. ред. А.Лекманов, **Информатика** — гл. ред. С.Островский, **Искусство** — гл. ред. Н.Исмаилова, **История** — гл. ред. А.Головатенко, **Литература** — гл. ред. Г.Красухин, **Математика** — гл. ред. И.Соловейчик, **Начальная школа** — гл. ред. М.Соловейчик, **Немецкий язык** — гл. ред. М.Бузова, **Русский язык** — гл. ред. Л.Гончар, **Спорт в школе** — гл. ред. Н.Школьников, **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.