

Здравствуйте! Это первая колонка главного редактора за все время существования "Информатики". Я иногда с интересом читаю подобные колонки в других изданиях, но ничего не могу поделать с ощущением, что есть в них (и даже в самом названии — "Колонка главного редактора") что-то неправильное. Впрочем, "каждый выбирает для себя", и я не считаю возможным обсуждать, а тем более — осуждать мотивы коллег, считающих важным лично начинать каждый номер своего издания. Отмечу лишь, что и Елена Борисовна Докшицкая, которая была главным редактором "Информатики" с 1995 по 1999 год, и я всегда считали, что у главного редактора есть достаточно иных способов выразить себя на страницах своего издания. И все же я понял, что раз в году главный редактор не только может, но, наверное, и должен высказаться явно.

В прошлом учебном году на одной из встреч с коллегами-учителями я услышал вопрос, в котором были озвучены мои собственные сомнения. Наша постоянная читательница поинтересовалась: *на сколько еще нас хватит? Предмет-то не безграничен, тем не бесконечное множество. Конечно, какие-то информационные ресурсы из питающих "Информатику" являются возобновляемыми, но так ли их много? Ведь объем информации, который потребляет газета за год, огромен!* Я и сам много думал над этим вопросом и уже приготовился отвечать, когда та же читательница, сделав некоторую паузу, произнесла: *впрочем, ответ то я, конечно, знаю.* Далее я просто привожу расшифровку диктофонной записи.

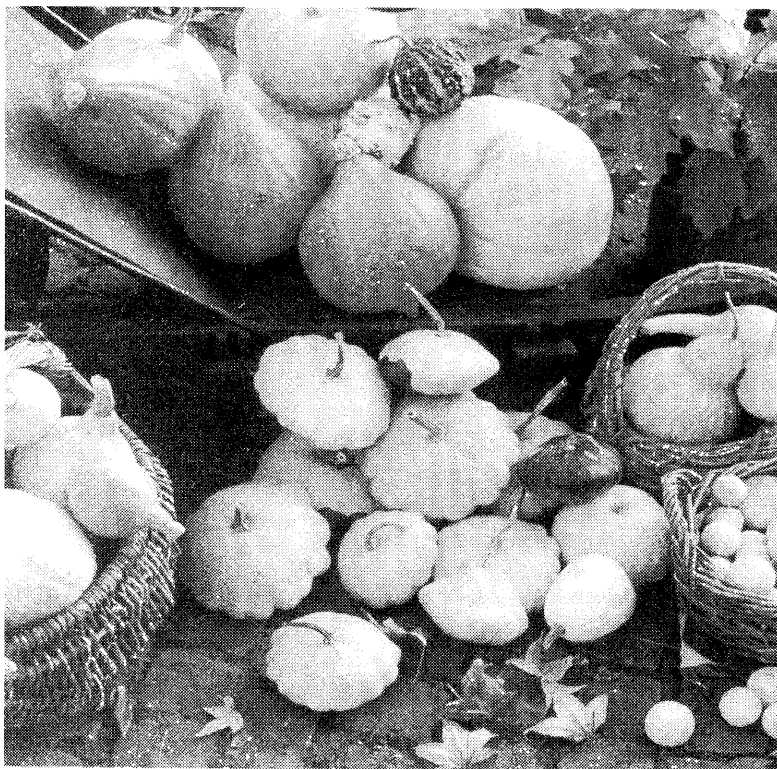
"Я записываю "Информатику" практически с самого начала, с 1995 года. И уже несколько лет меня занимал вопрос: когда же у них наконец темы кончатся. Выходить столько лет каждую неделю! Как же они это делают?! Ответ на этот вопрос я поняла, получив номера очередного "Жаркого лета". Газета научилась меняться вместе с предметом, вместе с нами, учителями. Ведь и меня несколько лет назад интересовали совсем другие вопросы, и я находила их на страницах "Информатики". А теперь я с удовольствием обнаруживаю в газете то, что интересует меня сейчас".

Нас ждет интересный год. Мы приложим все усилия, чтобы каждый подписчик в каждом номере находил не просто интересный (это уж обязательно), но необходимый ему материал. Мы будем меняться. И постараемся оставаться самими собой.

С.А. Островский
(so@1september.ru),
главный редактор

Еженедельная газета
Издательского дома «ПЕРВОЕ СЕНТЯБРЯ»

ИНФОРМАТИК А



*Урожайная осень пришла
на смену "Жаркому лету-2002"*

*Жаркое лето
2001-2002*

Основы вычислительной техники

Е.А. Еремин,
г. Пермь

ЛЕКЦИЯ 1

Введение

Для начала несколько слов о том, что будет изложено в рамках данного курса, каковы его цели и кому он будет полезен.

Предлагаемый вашему вниманию учебный курс **“Основы вычислительной техники”** посвящен принципам построения и функционирования вычислительных устройств. Прежде всего нас будет, конечно, интересовать современная вычислительная (компьютерная) техника¹, но определенное внимание будет уделено и историческим аспектам ее развития — в тех случаях, когда история позволяет лучше понять сегодняшние идеи и принципы. Главным направлением курса являются принципы работы основных блоков ЭВМ и их взаимодействия. Подробно будет описана логика функционирования главного узла компьютера — процессора; отдельная лекция будет посвящена организации памяти, ее видам, а также способам адресации данных, хранящихся там. Помимо указанных центральных устройств, предметом обсуждения будут и внешние устройства, принципы их работы и взаимодействия с другими узлами ЭВМ. Значительное внимание будет уделено роли программного обеспечения в работе компьютера, в первую очередь операционной системе. В заключение будет предложен методический материал по возможным способам изучения курса. В частности, будет проанализирован подход, связанный с использованием в преподавании учебных моделей ЭВМ, которым автор пользуется на своих занятиях уже много лет.

Сразу поспешу успокоить тех читателей, которых пугают термины “двоичная система”, “контроллер”, “шина”, “прерывание” и другие подобные слова, звучащие так устрашающе. Конечно, эти термины иногда будут использоваться (как же без них?), но обязательно с подробными и популярными разъяснениями, снабженными разнообразными примерами и аналогиями.

Хочется особо подчеркнуть, что данный курс призван не столько разъяснять детальное устройство компьютера какой-либо марки, сколько изложить наиболее общие базовые принципы и основную логику работы типичных машин. При этом автор старался отбирать такой материал, который наи-

более стабильно сохраняется в большинстве моделей ЭВМ вне зависимости от даты выпуска и фирмы-производителя. Надеюсь, читатели согласятся, что именно такие сведения и должны составлять основу хорошего учебного курса. Для изучения основ информатики такой подход является значительно более актуальным, чем для многих других дисциплин, из-за огромной скорости изменений в обсуждаемой области. Несколько подробнее этот вопрос будет рассмотрен в разделе 1.2 данной лекции.

Автор давно работает с вычислительной техникой и, анализируя свой опыт в этой области, пришел к твердому убеждению о полезности понимания основ теории для практической работы с ВТ. Подобный вывод признается не всеми — многие воинствующие “юзеры” принципиально не желают ничего слышать о том, “что находится внутри ящика”. Честно говоря, я сомневаюсь, что их можно переубедить. Поэтому давайте просто признаем за ними право иметь собственное мнение по этому вопросу и не будем при изложении пытаться “привлечь под свои знамена” абсолютно всех (кстати, маловероятно, что ярые пользователи вообще захотят заглянуть в лекции с подобным названием).

Главная цель курса состоит в том, чтобы дать читателю грамотные представления о логике работы компьютера и тем самым лучше подготовить его к практическому использованию этого достаточно сложного устройства. По этому поводу прекрасно сказал **Питер Нортона** в предисловии к одной из своих замечательных книг по устройству персонального компьютера IBM PC:

“Знать, как он работает, не менее важно, чем уметь работать с ПК. Вы можете вполне успешно пользоваться услугами компьютера, не понимая того, что в нем происходит. Однако чем глубже вы представляете процессы,

происходящие в ПК, тем лучше будете использовать его возможности... Если что-нибудь случится в процессе работы с компьютером, вероятность того, что вы примете правильное решение, а не сделаете глупостей и не испортите все окончательно, будет выше”.

Содержание лекций разрабатывалось в расчете **на максимально широкую аудиторию**, не обладающую какой-либо специальной предварительной подготовкой. В первую очередь это, конечно, преподаватели всех образовательных учреждений, которым приходится излагать те или иные аспекты устройства вычислительной техники. Хотелось бы, чтобы к ним присоединились старшеклассники и студенты, действительно

План публикаций лекций курса “Основы вычислительной техники” на “Страницах повышения квалификации” в первом полугодии 2002/2003 учебного года

Номер лекции	Номер газеты
1	33/2002
2	35/2002
3	37/2002
4	39/2002
5	41/2002
6	43/2002
7	45/2002
8	47/2002

Данный курс рассчитан на одно полугодие.

¹ В изложении обычно не делается особого различия между понятиями “компьютер” и “ЭВМ”, поскольку в основном речь идет о наиболее универсальных закономерностях.

интересующиеся информатикой. Вообще автор прилагал все возможные усилия, чтобы материал лекций свободно и легко мог быть прочитан *любим*, кто интересуется данной тематикой. Насколько это удалось — судите сами, уважаемые читатели.

При подготовке публикации, разумеется, использовалась всевозможная литература, и ссылки на нее будут обязательно приводиться. Тем не менее предлагаемые вашему вниманию материалы являются оригинальным авторским курсом. Подчеркиваю данный факт только для того, чтобы решительно отмежеваться от интенсивного процесса размножения одних и тех же материалов с многочисленных сайтов типа *www.referat.ru*, который в последнее время почему-то все больше ассоциируется с образованием².

1.1. Нужен ли такой курс?

В 1985 году, когда в школе появился новый современный учебный предмет, ему дали несколько длинное официальное имя — **Основы информатики и вычислительной техники** (ОИВТ). Конечно, такому подробному названию не суждено было прижиться хотя бы из соображений всевозможных практических неудобств, поэтому неудивительно, что от него в наше время сохранилось только одно слово **информатика**. Удивительно совсем другое — как с изменением *названия* попутно “укоротили” и *содержание* курса.

Существует большое количество самых разнообразных мнений, чему нужно учить в рамках курса информатики. По этой проблеме написано множество статей, книг и диссертаций, но, к великому сожалению, единого мнения до сих пор нет. Опасаясь втянуться в ненужную здесь продолжительную дискуссию по вопросу о самом факте необходимости изучения принципов работы вычислительной техники в курсе информатики, предлагаю в данном разделе ограничиться лишь кратким и по возможности наглядным обоснованием своей позиции по данной проблеме.

Начнем с того, что сам термин **информатика** происходит от соединения французских слов “**информация**” и “**автоматика**” (см., например, [1], с. 44). Таким образом, предметом исследования данной науки является именно *автоматическая* обработка информации. По моему мнению, из этого с достаточной очевидностью следует, что в содержании хорошего курса информатики должны быть рассмотрены не только вопросы, связанные с собственно информацией, но и с принципами представления различных ее видов в автоматических устройствах, а также основы функционирования таких устройств. Не случайно в основополагающем документе [2], который определяет содержание школьного курса информатики, изучение основ работы компьютера (так называемая “*линия компьютера*”) выделено в специальный раздел.

Далее, в основу учебных курсов желательно класть понятия и законы, которые являются *наиболее универсальными и не устаревают в течение коротких промежутков времени*. Это не значит, конечно, что учеников не надо информировать о новинках современной науки — просто не следует спешить немедленно все новое вносить в учебники. С указанной точки зрения основы ВТ выдержали “проверку временем”: скажем для примера, что многие классические идеи, предложенные фон Нейманом, Г.Голдстайном и А.Берксом еще в 1946 году в их знаменитой работе “Предварительное рассмотрение логической конструкции электронно-вычислительного устройства”, продолжают широко использоваться практически во всех современных компьютерах.

Хочется высказать и еще одно соображение. *Качественное образование есть не просто определенная сумма фактов*. А многие подходы к преподаванию курса информатики как прикладному фактически сводят его именно к набору готовых рецептов, как действовать в той или иной ситуации. Недостаток подобной методики не только в том, что число рецептов не может быть большим (иначе они просто “вылетят из головы”). Не менее важно, что конкретные советы не всегда в точности подходят к конкретной ситуации и часто устаревают. По поводу замены образования набором фактов замечательно писал американский фантаст Рей Бредбери в своей выдающейся книге “451° по Фаренгейту”:

“Устраивайте разные конкурсы, например: кто лучше помнит слова популярных песенок, кто может назвать все главные города штатов или кто знает, сколько собрали зерна в штате Айова в прошлом году. Набивайте людям головы цифрами, начинайте их безобидными фактами, пока их не затошнит, — ничего, зато им будет казаться, что они очень образованные. У них даже будет впечатление, что они мыслят, что они движутся вперед, хоть на самом деле они стоят на месте. И люди будут счастливы, ибо “факты”, которыми они напичканы, это нечто неизменное”.

Не пожалейте времени и перечитайте в очередной раз полностью эту очень глубокую по своему содержанию лекцию брандмейстера Битти, если вы действительно серьезно задумывается о содержании образования. Кстати, в связи с приведенной цитатой вспомните: а вам не приходилось встречать учеников, которые могут перечислить практически все особенности каждой версии программного обеспечения, включая достаточно мелкие и второстепенные, но которые тем не менее плохо решают задачи, теряются в нестандартной ситуации и немедленно обращаются к соседу за помощью?

И, наконец, еще одно обстоятельство. Большинство современных реформаций в нашей системе образования делается по образцу западных стран. Для многих из них, особенно для наиболее мощной державы — Соединенных Штатов, обучение носит прикладной характер. Именно в этом мне видится в значительной степени причина отказа от теоретических основ *во всех* учебных предметах. А между тем путь этот далеко не так хорош, как многие пытаются представить. Не считая себя специалистом в области систем образования за рубежом, обращусь за поддержкой к действительно компетентным и авторитетным людям. Вот, например, Алан Кэй: в трид-

² Зайдите в раздел “Образование” любого каталога интернет-ресурсов, и на первых позициях вы почти наверняка увидите сайты подобного содержания. Неужели еще остались преподаватели, которые продолжают задавать рефераты?!

цать лет — профессор Станфордского университета, автор первой объектно-ориентированной среды программирования Smalltalk; занимал пост вице-президента Walt Disney Company по исследованиям и разработкам в сфере визуальных технологий и более 25 лет проработал преподавателем в калифорнийских школах. В 1995 году Алан Кэй выступил на совместном заседании Комитета по науке и Комитета по экономике, образованию и занятости конгресса США по вопросам состояния системы образования в мире. Отрывки из стенограммы его выступления в русском переводе можно найти в журнале [3].

А.Кэй приводит простой, но поразительный по своим результатам пример. Эксперты Национального научного фонда США устроили собеседование со старшекурсниками и профессорами известного во всем мире Гарвардского университета. Испытуемым было предложено объяснить механизмы смены времен года и фаз Луны. К удивлению проверяющих, ответы **почти 95% (!!!)** испытуемых были весьма далеки от истины, которую ученые древности установили много веков назад. В частности, типичный ответ выглядел так: “Земля летом ближе к Солнцу, чем зимой”. Кэй провел аналогичное собеседование со своими студентами в университете Лос-Анджелеса и получил сходные результаты. Тогда он сделал следующий шаг. Он спросил: “А знаете ли вы, какое время года наступит в Южной Америке и Австралии, когда у нас в Штатах приходит лето?” “Зима, разумеется”, — был ответ. Но даже после этого понимание какого-то противоречия в сказанном приходило далеко не ко всем!

В чем же дело? Эксперты ожидали от студентов **знания фактов** — но кто же в состоянии удержать в голове их такое огромное количество? А вот чего здесь действительно не хватало, так это самых элементарных **навыков мышления**. Все, что нужно для правильного ответа, по отдельности было известно участникам испытания! Однако сопоставить факты и сложить из них правильный ответ большинству из них не удалось.

И еще одна очень поучительная цитата, в которой обучение работе на компьютере сравнивается с системой музыкального образования.

“Представим себе, что музыкальное образование объявлено одним из национальных приоритетов. Конгресс предлагает технологическое решение проблемы. “Пианино в каждой классной комнате к 2000 году! Не хватает бюджета, чтобы нанять для работы в школе профессиональных музыкантов? Будем проводить массовую переподготовку учителей, двухнедельные курсы помогут решить все проблемы!”.

Результат? Детям такая жизнь нравится, и общество выходит мало-помалу на новый уровень музыкальной культуры — каждый гражданин “разбирается в музыке” и может бегом сыграть “Собачий вальс”. Кому-то, правда, удастся получить и настоящее музыкальное образование.

“Разбираться в музыке”, “разбираться в науке”, “разбираться в компьютерах” — в этом ли наша цель?”

А вот мнение еще одного авторитетного человека — автора одного из первых российских школьных учебников информатики А.Г. Кушниренко. Обобщая свой опыт работы с российскими и американскими студентами, он писал следующее [4]:

“Если в инструкцию к программе вкралась опечатка и какая-то команда не работает, то российский студент скорее всего посмотрит на аналогичную команду, попытается сообразить, что делает команда, какие у нее должны быть по логике вещей аргументы, разберется, исправит опечатку и продолжит работу.

Американский студент в такой ситуации ни о какой логике вещей скорее всего и не подумает. Он найдет в руководстве на последней странице номер бесплатного телефона, позвонит на фирму, получит указание, что делать, выполнит его и продолжит работу. ...Не нужно сравнивать эти подходы. В одних ситуациях лучше один, а в других — другой.

Важно только понимать, что эта вот привычка к пониманию логики вещей — сильная сторона российского образования. На эту привычку, умение, желание понять, разобраться в любом предмете в Америке сегодня есть большой спрос.

Я полагаю, что такой спрос и в Америке, и в России, и во всем мире с годами будет только увеличиваться.

И потому с чисто прагматической точки зрения мы должны стараться сохранить в российском образовании то, что успешно работало раньше и что оказывается конкурентоспособным на мировом рынке сейчас, — стремление к фундаментальности”.

Хочется добавить, что заголовок раздела, из которого выписана цитата, может служить кратким и глубоким выводом по данному пункту; он сформулирован так: **“умение понимать не менее важно, чем умение делать”**.

Читатель, наверное, уже устал от общих рассуждений. А нет ли более конкретных аргументов о пользе изучения основ работы вычислительной техники? Конечно, есть. Рассмотрим следующий простой пример. Допустим, учитель попросил учеников проверить с помощью электронных таблиц Excel (кто возразит, что это программа не для пользователя?) тот факт, что $n! = (n - 1)! \cdot n$. Примерное решение задачи может выглядеть следующим образом.

	A	B	C	D
1	5		=ФАКТР (A1)	
2	=A1-1	=ФАКТР (A2)	=B2*A1	
3			=C1-C2	=ЕСЛИ (C1-C2=0; "да"; "нет")

Аккуратно внесем формулы в электронную таблицу, и Excel послушно подтвердит, что все правильно. А если взять другое число, допустим, 25? К нашему удивлению, появится следующий результат:

The screenshot shows the Microsoft Excel interface with the following data in the worksheet:

	A	B	C	D	E
1	25		1,55E+25		
2	24	6,2E+23	1,55E+25		
3			0	нет	

The formula bar shows the active cell D3 containing the formula: `=ЕСЛИ(C1-C2=0,"да","нет")`.

Вы, конечно, понимаете, что $25!$ достаточно большое число и в ходе вычислений происходит потеря точности. Но как это объяснить, не затрагивая принципов обработки чисел в ЭВМ?

И, наконец, последний пример — компьютерные вирусы. Каких только странных советов по предотвращению заражения не услышишь (вплоть до анекдотических — о необходимости мыть руки перед работой с компьютером)! Как слепо верят люди в могущество антивирусных программ, полагая, что их последние версии дают абсолютную гарантию обнаружения всех существующих вирусов! И в то же самое время многие пользователи не применяют даже такой простой и надежный метод сохранения дистрибутивных дискет, как их аппаратная защита от записи.

Кстати, именно на поверхностное знание компьютера порой и рассчитаны вирусы. Вспомните один из нашумевших прошлогодних почтовых вирусов, содержавший фотографию известной российской теннисистки Анны Курниковой [5]. На деле такая “фотография”, прилагаемая к письму, была исполняемым файлом с двойным расширением типа *AnnaKournikova.jpg.vbs*. Расчет был во многом сделан на то, что многие почтовые программы не отображают расширения приложенных файлов: в результате пользователи видели интригующую подпись *AnnaKournikova.jpg*, и многие из них, обученные “щелкать по всему яркому и движущемуся”, попадали впросак⁴.

Завершить этот раздел хочется известной цитатой из А. Конан Дойла, которая также призывает критически анализировать запоминаемую информацию:

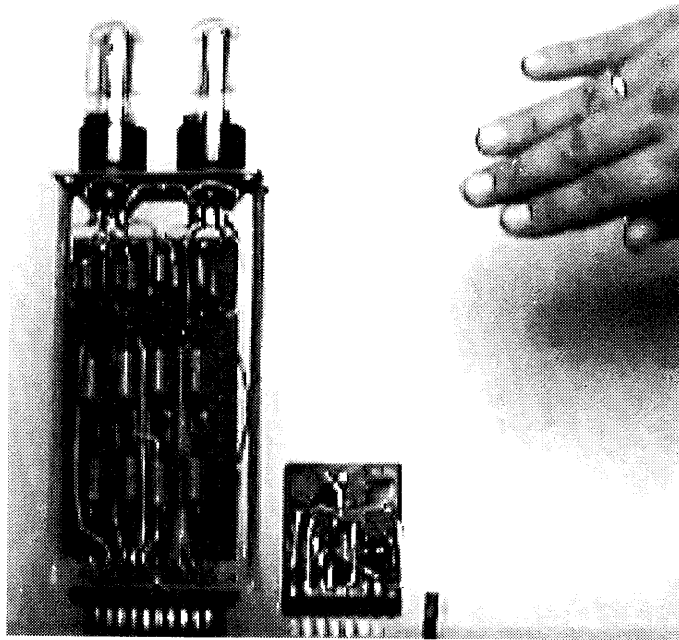
“Мне представляется, что человеческий мозг похож на маленький пустой чердак, который вы можете обставить, как хотите. Дурак натащит туда всякой рухляди, какая попадется под руку, и полезные, нужные вещи уже некуда будет всунуть, или в лучшем случае до них среди всякой завали и не докопаешься. А человек толковый тщательно отбирает то, что он поместит в свой мозговой чердак. Он возьмет лишь инструменты, которые понадобятся ему для работы, но зато их будет множество, и все он разложит в образцовом порядке. Напрасно люди думают, что у этой маленькой комнатки эластичные стены и их можно растягивать сколько угодно. Уверяю вас, придет время, когда, приобретая новое, вы будете забывать что-то из прежнего. Поэтому страшно важно, чтобы ненужные сведения не вытесняли собой нужных”.

1.2. Стремительное развитие ВТ и стабильность ее фундаментальных принципов

1.2.1. Темпы развития ЭВМ

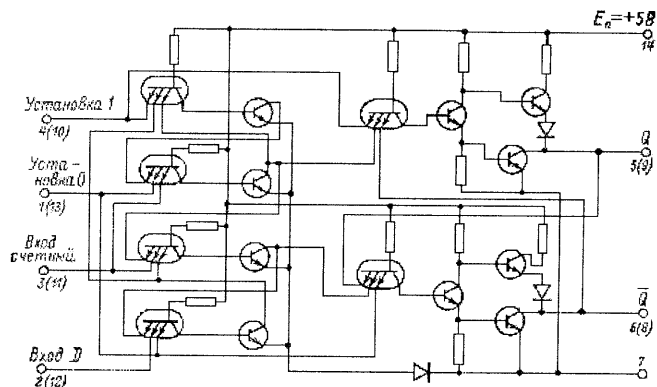
Чтобы показать огромную скорость развития в производстве вычислительной техники, обычно приводится масса разнообразных сравнений и фактов. Автору, чтобы не повторяться, хочется привести что-нибудь новое. Перед вами фотография ячеек ЭВМ трех разных поколений (см. слева направо): на лампах, на транзисторах и на микросхемах; надеюсь, качество фотографии позволяет вам различить последнюю маленькую ячейку в виде “многоногого жучка”. Подчеркнем, что с точки зрения выполняемых функций все ячейки абсолютно одинаковы — каж-

дая из них представляет собой два триггера (триггер — это электронная схема для хранения одной минимальной единицы информации, т.е. 1 бита).



Фотография фактически является наглядной диаграммой уменьшения размеров элементов вычислительной техники для первых трех поколений (кстати, рука сохранена специально для того, чтобы дать читателям возможность иметь некоторый характерный масштаб изображения). Понятно, что “выпилить” два триггера из современного микропроцессора четвертого поколения не представляется возможным: для этого надо из квадратного кристаллика со стороной менее 1 см выделить ничтожную его часть. Даже если бы это каким-то чудесным образом удалось, web-камера, с помощью которой сделана данная фотография, все равно не сумела бы отобразить объект таких ничтожных размеров.

Как уже было сказано, показанные на фотографии ячейки содержат триггеры. Поскольку триггер является важным цифровым устройством, способным хранить один бит информации, мы еще неоднократно встретимся с ним в нашем курсе. А пока в ознакомительных целях приведем схему некогда распространенной микросхемы К1ТК5522 (два D-триггера) [6], которая, кстати, и была сфотографирована. Левая часть схемы реализует определенную входную логику, а правая, состоящая из двух симметричных узлов, образует собственно триггер. На выходе Q отражается хранящаяся в триггере информация: 0 или 1.

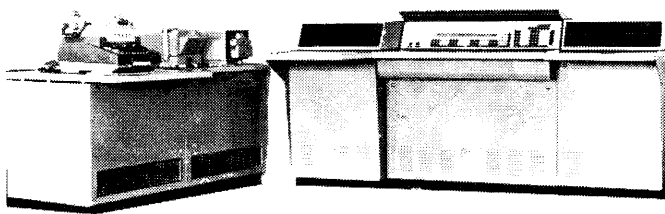


⁴ Между тем значки у файлов *vbs* и *jpg* разные, что наверняка можно было заметить.

Итак, темпы развития технологии производства компонентов вычислительной техники поистине впечатляющи. А насколько изменяются фундаментальные принципы построения ЭВМ? Сравним между собой две достаточно типичные вычислительные машины, с которыми автору приходилось лично иметь дело. Одна из них принадлежала ко второму поколению⁴ и уже лет десять как разобрана на составные части (кстати, из ее главного блока после небольшой переделки вышел замечательный стол огромных размеров). В качестве второго компьютера, с которым мы будем сравнивать, возьмем обычный IBM PC с процессором Intel 80486. Такие машины еще вполне можно увидеть, хотя, разумеется, это далеко не последняя модель.

1.2.2. Система команд типичной ЭВМ второго поколения

ЭВМ “Наири-2” [7] была разработана в 1968 году и серийно производилась в Ереване. Она могла работать с целыми и вещественными числами (вещественные числа обычной точности хранились в одной ячейке, а длинные — в двух последовательных ячейках).



В систему команд ЭВМ “Наири-2” входило 102 команды, из них 47 — собственно машинные операции (см. табл. 1), а остальные — так называемые “псевдооперации”, реализуемые при помощи специальных подпрограмм в долговременном запоминающем устройстве — ДЗУ⁵. Важное положение среди псевдоопераций занимали разнообразные действия над вещественными числами, которые мы для простоты изложения рассматривать не будем.

Приведем также примеры некоторых (далеко не всех!) псевдоопераций ЭВМ “Наири-2”, которые находились в ее ДЗУ из 16 384 ячеек. Это **ln** — вычисление логарифма, **sn** — вычисление синуса, **пп** — печать чисел с плавающей запятой, **пс** — печать содержимого (в двоичном виде), **пк** — печать команд, **сд**, **вд**, **уд**, **дд** — арифметические операции с длинными числами, **ск**, **вк**, **ук**, **дк** — действия над комплексными числами, **Га** — вычисление гамма-функции, **mn** и **mx** — нахождение минимума и максимума в массиве и многие другие.

1.2.3. Система команд типичной ЭВМ четвертого поколения

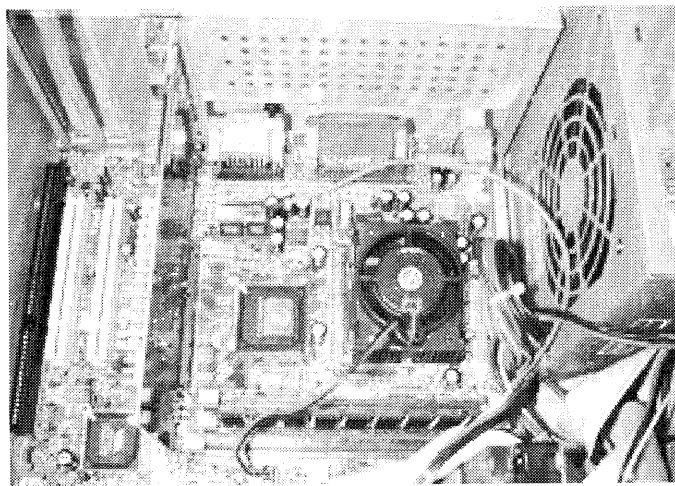
А теперь для сравнения приведем систему команд хорошо знакомого всем процессора Intel 80486, кото-

Таблица 1
Полная система машинных операций
ЭВМ “Наири-2” [7]. 1968 г.

Код команды	Название команды
п, p_i	передача числа
с, c_1, c_2, c_3	сложение
в, v_1, v_2, v_3	вычитание
у, u_1, u_2, u_3	умножение
а, a_1, a_2, a_3	деление
v_4	вычитание модулей
l, l_1	логическое сложение (OR)
$\wedge, \wedge_1$	логическое умножение (AND)
м, m_1	сложение по модулю 2 (XOR)
а, a_1	сдвиг вправо (арифметический)
$?, ?_1$	сдвиг влево (логический)
г, g_1	длинный сдвиг
н, n_1	нормализация
и, i_1, i_2, i_3	абсолютный переход с возвратом
е, e_1, e_2, e_3	относительный переход с возвратом
к, k_1	останов
ч	чтение
о, o_1	обращение (вывод)
х	холостая команда

Примечание. Большинство инструкций имеют несколько модификаций, которые обозначены цифровым индексом. Например, операция **п** копирует содержимое одной ячейки памяти в другую, а **п₁** сохраняет результат предыдущей операции в память.

рый и поныне все еще продолжает служить людям в некоторых не слишком новых компьютерах.



Вот эта таблица, построенная по данным монографии [8].

⁴ Мой первый сеанс общения с ЭВМ марки “Наири-2” состоялся в 1974 году.

⁵ ДЗУ было выполнено на ферритовых кольцах, причем информация в нем в прямом смысле слова “прошивалась” путем продергивания провода через сердечник при единице и мимо него — при нуле.

Таблица 2
Основные команды микропроцессора Intel 80486 [8]

Код команды	Название команды
MOV	пересылка данных
PUSH, PUSHA, POP, POPA	чтение/запись в стек
LEA, LDS, LES, LFS, LGS, LSS	загрузка адреса и селекторов
MOVSX, MOVZX	пересылка с расширением знака
XCHG	обмен содержимым
XLAT	преобразование кодов
BSWAP	перестановка байтов
ADD, ADC, XADD, INC, AAA, DAA	сложение
SUB, SBB, DEC, CMP, CMPXCHG, NEG, AAS, DAS	вычитание
MUL, IMUL, AAM	умножение
DIV, IDIV, AAD	деление
CBW, CWDE, CWD, CWDQ	преобразование разрядности данных
NOT	логическое НЕ
AND	И
OR	ИЛИ
XOR	исключающее ИЛИ
TEST	логическое сравнение операндов
SHL, SAL	логический и арифметический сдвиги влево
SHR	сдвиг вправо
SHLD, SHRD	двухоперандные сдвиги
ROL, ROR, RCL, RCR	циклические сдвиги
BT, BTS, BTR, BTC, BSF, BSR	операции с битами
LODS, STOS, INS, OUTS, MOVSB, CMPS, SCAS	операции со строками
JMP	безусловный переход
JA/JNBE, JAE/JNB/JNC, JB/JNAE/JC, JBE/JNA, JE/JZ и др.	условные относительные переходы
CALL, RET, INT, IRET	переход к подпрограммам и возврат
SETA/SETNBE, SETAE/SETNB и др.	условная установка байта
LOOP, LOOPE/LOOPZ, LOOPNE/LOOPNZ, JCXZ	организация циклов
IN	ввод из порта в аккумулятор
OUT	вывод из аккумулятора в порт
HLT	останов
NOP	отсутствие операции

Примечания

1. Для краткости в таблицу не включены некоторые группы команд, связанные с управлением процессором, организацией защиты памяти, поддержкой языков высокого уровня, и некоторые другие. Кстати, именно эти инструкции являются максимально “машинно-зависимыми”, т.е. существенно зависят от модели процессора. Иными словами, с точки зрения целей образования представляют гораздо меньший интерес.

2. Особо следует сказать об операциях с вещественными числами. В процессорах фирмы Intel для выполнения этих действий используется специальный блок — так называемый “сопроцессор”. Начиная с 486-й модели, сопроцессор изготавливается в том же самом кристалле, что и основной процессор. Команды сопроцессора в таблице также не отражены.

1.2.4. Сравнение систем команд

А теперь самая главная и интересная часть — сравнение между собой таблицы. Напомним читателю, что они разделены огромным по меркам развития вычислительной техники временем — более 20 лет. Серым цветом в таблицах выделены клетки, содержащие аналогичные операции. И, как ни удивительно, они охватывают большую часть анализируемых таблиц! Кроме того, не следует забывать, что процессоры фирмы Intel относятся к категории CISC-процессоров, т.е. процессоров с расширенной системой операций. А это значит, что для многих других процессоров результаты аналогичного сравнения будут еще более близкими⁶.

Итак, получаем важный и интересный вывод: ядро системы команд, образуемое такими группами операций, как перепись информации, арифметические и логические команды, сдвиги и переходы, за несколько поколений(!) ЭВМ *практически не изменилось*. Неужели вас, уважаемые читатели, это не удивляет?!

Еще одно наблюдение состоит в том, что дополнительные команды процессоров Intel (см. табл. 2) в основном расширяют возможности программиста, а не предоставляют принципиально новых возможностей. Скажем, наличие группы команд для организации циклов во многих случаях облегчает написание программы, но, с другой стороны, *любой* цикл можно реализовать и без них с помощью инструкций условных переходов. Или битовые операции, которые *всегда* можно выполнить на базе логических операций.

Пожалуй, наиболее важным прогрессом в развитии ЭВМ является переход к обработке значительно большего количества различных данных, который как раз не очень заметен из обсуждаемых таблиц. Хотя ЭВМ первых поколений и могли работать с несколькими видами информации (в основном числами) и даже имели “зачаточные” возможности по выводу отдельных символов, развитые средства обработки строковых данных там отсутствовали. Работа со звуком и видео в те времена, естественно, даже не обсуждалась. Основу памяти таких машин составляла ячейка под стандартное число или команду (в “Наири” ее длина равнялась 36 двоичным разрядам), поэтому данные другой длины не могли при такой организации рационально храниться⁷. Память современных компьютеров имеет значительно более гибкую байтовую структуру, о которой мы обязательно подробно поговорим в следующих лекциях.

И последнее замечание, которое касается обеих анализируемых таблиц. Обратите внимание на **простоту** команд, которые непосредственно может исполнять процессор даже современной ЭВМ. Данный факт инте-

ресен тем, что сложное программное обеспечение, содержащее тысячи таких простых команд, как сложение, сравнение и др., может вести себя настолько разумно, что как-то забывается, что в его основе лежат все те же несложные базовые инструкции процессора.

1.3. Некоторые выводы

Итак, подведем некоторые итоги того, что мы узнали из сегодняшней лекции.

- Понимание базовых принципов работы такого достаточно сложного в обращении и настройке устройства, как компьютер, полезно любому человеку, систематически его использующему.

- Необходимость в практическом применении фундаментальных основ ВТ возникает не всегда. Чем более необычна для пользователя ситуация, тем острее он нуждается в подобных знаниях и особенно в умении их анализировать.

- Несмотря на огромный технический прогресс, который увеличил на несколько порядков быстродействие ЭВМ и соответственно уменьшил их размеры, фундаментальные принципы построения ВТ в основном сохранились.

- Как это ни парадоксально звучит, в основе всех самых “умных” и изощренных программ лежит достаточно ограниченная система простых команд, которые непосредственно понимает процессор компьютера. Секрет состоит в универсальности этих команд и в возможности их объединять.

- На самом низком уровне инженеры стараются создать компьютер максимально просто. Для адаптации машины к сложным задачам, которые требуется решать на практике, используется сложное программное обеспечение. Очевидно, что переписать программное обеспечение значительно проще, чем изменять конструкцию компьютера.

1.4. Вопросы для самостоятельной работы

В разделе с таким названием в данной и в последующих лекциях мы будем предлагать вам подумать над проблемами, решения которых связаны с излагаемым материалом. Причем не всегда сформулированные вопросы имеют очевидные и однозначные ответы. Главная цель данного раздела — предложить читателям решить их тем или иным способом *для себя*. Автор полагает, что наличие собственной позиции по подобным вопросам поможет преподавателям более квалифицированно строить свои курсы.

1. Попробуйте четко ответить на вопрос: насколько, по-вашему, полезно знание фундаментальных основ работы вычислительной техники?

2. Постарайтесь проанализировать свой опыт общения с компьютером и припомнить ситуации, когда понимание логики работы компьютера вам помогло принять правильное решение. А может быть, вам вспоминаются и обратные примеры?

⁶ Читателям, которые усомнятся в описанном из-за того, что 486-й процессор уже устарел, советуем обратиться к любой книге по процессорам Pentium, например [9], и убедиться, что появившиеся в последних процессорах новые команды принципиально не изменяют ситуации.

⁷ Попутно обратите внимание на появление в табл. 2 новых по сравнению с табл. 1 групп команд преобразования данных.

3. Какие наиболее убедительные аргументы вы бы привели своему ученику в ответ на его традиционное “Кому все это нужно?” (автор убежден, что читатели найдут множество эффектных аргументов, которые не отражены в тексте лекции).

4. Сопоставьте эксплуатацию современного компьютера с другими устройствами, скажем, телевизором или автомобилем. Согласны ли вы с утверждением, что использование компьютера намного сложнее, чаще требует принятия решений и, следовательно, больше нуждается в понимании внутреннего устройства?

5. Проанализируйте реальную ситуацию, описанную в книге А.Гринзоу [10]:

“В офисе клиента, которого я консультировал, я однажды застал их директора по маркетингу за следующим занятием: он запускал Windows, программу-редактор, что-то выискивал, беспорядочно тыкая в разные кнопки на экране, после чего выключал компьютер из сети. Пока я наблюдал за ним, он повторил эту процедуру несколько раз. В конце концов я понял, что происходило: этот человек просто не знал, как удалить текст и начать новый документ. А включение/выключение компьютера просто оказалось ближайшим решением его проблемы. (Да, забыл упомянуть: клиентом являлась компьютерная компания.)”

Как вы думаете, каких минимальных знаний о работе с компьютером не хватало описанному пользователю?

6. Еще одна цитата (Майкл Крайтон, роман “Стрела времени”):

“Спросите себя... — каков преобладающий способ восприятия в конце двадцатого столетия? Какими люди видят вещи и какими они ожидают их увидеть? Ответ прост. В любой сфере деятельности, от бизнеса до политики, от маркетинга до образования, преобладает восприятие через развлечение.

...Сегодня каждый ожидает, что его будут развлекать, причем будут развлекать постоянно. Деловые встречи должны происходить молниеносно, иметь очень краткую программу и сопровождаться анимационной графикой, чтобы участники не скучали. Торговые центры и магазины должны быть привлекательными, чтобы развлекать нас в той же мере, в какой и предлагать товары. Политические деятели должны обладать телегеничными лицами и сообщать нам только то, что мы хотим услышать. Школы должны проявлять крайнюю осторожность, чтобы не скучали молодые умы, ожидаю-

щие от обучения динамичности и разнообразия, присущих телевизионным программам. Студентов нужно удивлять — всех нужно удивлять, или же они отвернутся; отвернутся от марки товара, от канала телевидения, от партии, откажутся от лояльности.

Такова реальность интеллектуального бытия западного общества в конце столетия”.

Насколько преподаватель должен идти навстречу обучаемому в деле превращения обучения в развлечение? Существует ли грань, которую нельзя переступать, не рискуя в угоду ученику превратить обучение в его видимость? Готовит ли такое “развлекающее” обучение к будущей реальной жизни? Оцените с точки зрения рассматриваемого вопроса построение данной лекции.

Литература

1. Информатика в понятиях и терминах / Под ред. В.А. Извозчикова. М.: Просвещение, 1991, 208 с.
2. Обязательный минимум содержания образования по информатике: Приложение к Приказу Министерства образования Российской Федерации № 56 от 30.06.99. Информатика и образование № 7, 1999. С. 2—3.
3. Кэй А. Идеям тоже нужна любовь. Компьютер в школе № 1, 1998. С. 11—15.
4. Кушниренко А.Г. Несколько замечаний о школьной информатике в России 1998 года. Информатика № 6, 1998 (статья доступна в Интернете по адресу <http://inf.1september.ru/old/table16/agk.htm>).
5. “Лаборатория Касперского” о вирусе “Анна Курникова”. http://www.rol.ru/news/it/news/01/02/13_028.htm
6. Справочник по полупроводниковым диодам, транзисторам и интегральным схемам. / Под ред. Н.Н. Горюнова. М.: Энергия, 1977, 744 с.
7. Малая универсальная вычислительная машина “Наири-2”. Техническое описание. Ч. 1. Краткое описание и система команд, 1968, 121 с.
8. Бродин В.Б., Шагури И.И. Микропроцессор i486. Архитектура, программирование, интерфейс. М.: Диалог-МИФИ, 1993, 240 с.
9. Гук М. Процессоры Intel: от 8086 до Pentium II. СПб.: Питер, 1997, 224 с.
10. Лу Гринзоу. “Философия программирования для Windows 95/NT”. СПб.: Символ-Плюс, 1997, 640 с.

С 1 СЕНТЯБРЯ ПРОИЗВОДИТСЯ ПОДПИСКА НА ПЕРВОЕ ПОЛУГОДИЕ 2003 ГОДА

Внимание! “Информатика” объявляет о проведении акции “Подарок”. Все, кто

а) подпишется на первое полугодие 2003 года (на 6 месяцев) до 15 октября;

б) пришлет в редакцию по обычной почте ксерокопию корешка подписного купона (именно ксерокопию; пожалуйста, не присылайте сами корешки купонов, они вам самим нужны!) или по электронной почте изображение корешка купона, получают подарок от “Информатики” — специальный подарочный номер.

Электронные письма должны поступить в редакцию не позднее 20 октября. Бумажные письма не позднее 20 октября должны быть отправлены. Подписные индексы, наши почтовый и электронный адреса указаны на последней странице каждого номера.

Компьютерное образование в Израиле.

Современное состояние (попытка очерка)

Е.Канел, З.Фрайман,
Израиль

История обучения компьютерным наукам в школах Израиля насчитывает примерно три десятилетия и сама по себе может являться темой отдельной статьи. Но в данном материале мы постараемся дать очерк того, в каком состоянии изучение этого предмета находится сегодня: какие направления приняты за “основные”, какие интересные новинки предлагаются для испытания (на учителях и учениках — а на ком же, если честно говорить, еще?), какие учебники, методические материалы и программные разработки имеются и какие результаты достигнуты.

Об общем — но вкратце

Расскажем кратко о системе образования в Израиле — точнее, о тех ее аспектах, к которым так или иначе придется обращаться в этой статье.

Во-первых, 12 лет обучения израильский школьник проводит как минимум в двух совершенно разных школах: первые шесть — в “базовой школе”, остальные — в средней школе, которая структурно составлена из двух ступеней, “промежуточной” (7—9-й классы) и “старшей” (10—12-й классы). При этом организационно последние две часто сводятся в одну общую школу с единым управлением, но с собственным директором и администрацией в каждой. Решающее слово при этом остается за директором “старшей” ступени. Надо сказать, что, как правило, такая структура функционирует практически бесконфликтно и вполне результативно.

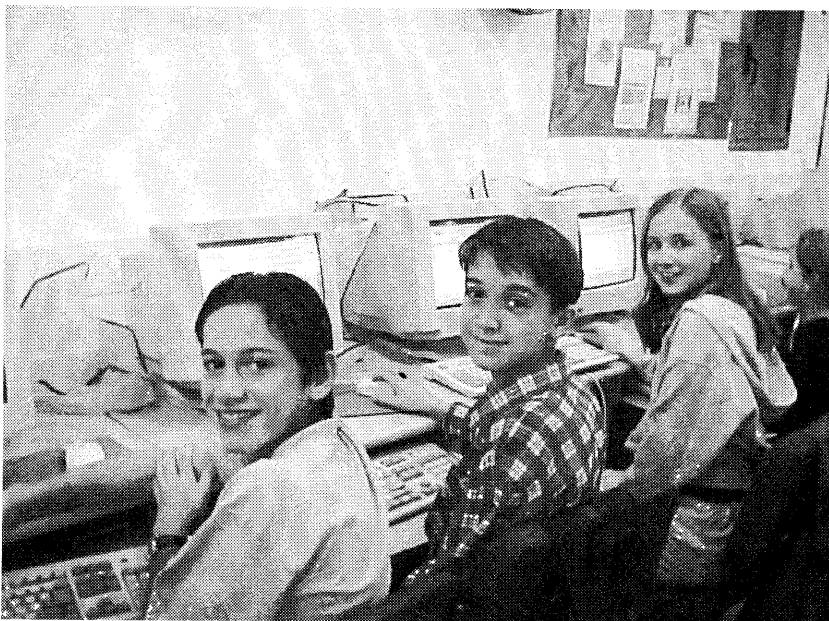
Во-вторых, текущие оценки выставляются по 100-балльной системе, при этом “водораздел” проходит по оценкам 55 (два с двумя плюсами) и 56 (три с двумя минусами); в некоторых случаях порог несколько более

приподнят и проходит по границе в 60 баллов. К системе этой привыкаешь довольно быстро (я имею в виду граждан бывшего СССР) и даже находишь в ней немалые преимущества: например, в контрольной с многими вопросами-заданиями-подпунктами принято проставлять максимальный балл за каждый из них, что позволяет ученику четче понять, сколько и за что им получено.

В-третьих, экзамены на аттестат зрелости начинаются с 10-го класса и распределены (иногда — в несколько этапов) на три года до конца школы; такой подход позволяет обучать определенному предмету (в том числе и тому, которому посвящена эта статья) на более высоком уровне, поскольку обучение растягивается минимум на два года (не включая одного-двух лет предварительного обучения предмету).

В-четвертых, каждый предмет может изучаться на аттестат зрелости на разных уровнях сложности (“учебных единицах”), градация которых установлена по 5-балльной шкале. Такой подход позволяет каждому ученику выбрать предмет для формирования аттестата на таком уровне, который он реально в состоянии осилить с максимально возможной оценкой. После окончания школы, при поступлении в колледжи и университеты, конкурентный уровень кандидата определяется по формуле, учитывающей и число “учебных единиц” по предметам, вошедшим в аттестат, и полученные по этим предметам оценки. Кстати, существует оговоренная законом практика повторной сдачи экзаменов для улучшения общего уровня аттестата — до 40 лет.

В-пятых, для получения полноценного аттестата зрелости ученик должен сдать экзамены по определенному набору обязательных предметов, к которому каждый добавляет набор предметов по выбору (из предлагаемого Министерством просвещения довольно широкого набора); компьютерные науки входят во вторую группу — “предметы по выбору”. В большинстве школ каждый ученик выбирает два-три таких предмета — после консультаций с учителями,



Урок — это радость

школьными психологами и т.д. Процесс выбора достаточно сложен и не всегда проходит безболезненно — впрочем, это тема для отдельного разговора.

И, наконец, в-шестых, подавляющее большинство экзаменов являются “едиными”, “государственными” и письменными. Это означает, что они проводятся одновременно по всей стране, письменные ответы учеников отсылаются в центр — и проверяются анонимными экзаменаторами, которые, в свою очередь, не знают, чьи работы они проверяют. Таким образом, достигается максимально возможная объективность оценки и практически исключается возможность “порадеть родному человеку”.

Компьютеры в “базовой” школе

На этой стадии приоритет отдан изучению прикладных программ, причем в “базовой” школе имеет место совершенно четкая ориентация на работу с прикладными программами — от выработки навыков работы с обучающими программами (по языку, географии, истории, математике и прочее) до основ офисных пакетов (по преимуществу Word и PowerPoint). Строго говоря, в этих классах все зависит от того, насколько школа богата соответствующим программным обеспечением и располагает толковым учителем — как таковые обязывающие программы Министерства образования отсутствуют, оставляя практически полностью выбор и координацию на усмотрение учителя и администрации. Надо сказать, что сегодня практически все “базовые” школы имеют компьютерные кабинеты и соответствующие пункты финансирования в смысле покупки программного обеспечения.

Кроме того, министерство (а нередко и спонсоры) финансирует подключение школ к Интернету, так что и эти знания в их базовом виде ученики получают — особенно если учесть, что пользователями Сети являются 25% израильтян.

Компьютеры в “промежуточной” школе

Здесь ситуация в значительной степени определяется тем, что организационно эта ступень слита в единую учебную структуру со старшими (10—12-ми) классами. И здесь изучаемый материал в значительной степени определяется самой школой, хотя в последние годы все активнее внедряется некий более или менее единый подход. Обычно речь идет об обучении навыкам работы (на максимально возможном уровне — что определяется в основном спецификой и уровнем ученического состава и квалификации учителя) с Word, PowerPoint, Excel и (кое-где) Access. Все активнее внедряются и обучение продвинутым формам работы с Интернетом, основы web-дизайна, работы с графическими программами — по преимуществу Paint Shop Pro. Сверх того, повторимся, все зависит от уровня учителя и учеников, в большинстве школ в классах одной параллели тот же Word изучается по-разному. Один из авторов этой статьи в своих физико-математических классах изучал и основы программиро-

вания на Visual Basic. Аналогичные удачные опыты известны во все большем числе израильских школ. Этому способствует и то, что сегодня практически все школы этих двух последних ступеней оборудованы 2—4 компьютерными классами (по 15—25 компьютеров в каждом).

В 9-м, последнем, классе “промежуточной” ступени сегодня в абсолютном большинстве школ начинают преподавать основы алгоритмики и программирования (в дословном переводе с иврита — “основы компьютерных наук”); конечно, речь идет о классах, в которых часть учеников намерены выбрать этот предмет как один из “аттестационных”, то есть станет изучать его с целью сдать соответствующий госэкзамен. В качестве языка программирования выбирается, как правило, Pascal, хотя сегодня все активнее начинают пользоваться Си. Выбор столь “древних” языков мотивируется министерством примерно так — “они не требуют отвлечения на дизайн, требуют минимума знаний для понимания и написания программ, позволяют сосредоточиться больше на идее, чем на ее оформлении”. Недостатка в желающих оспорить эту точку зрения нет (преподаватели компьютерных наук, похоже, по природе своей интернациональны в неукротимом стремлении прогрессировать, причем — с максимальным опережением...) — но она пока остается официальной.

Немного статистики

Согласно опросу, проведенному Общеизраильским Центром по преподаванию компьютерных наук в школе, 66% израильских учителей считают, что первым языком программирования для ученика должен быть Pascal. За то, чтобы начинать с языка Си, выступают 28% учителей.

Можно сказать, что год предварительного обучения подчинен двум сверхзадачам:

- освоить самые базовые понятия алгоритмики и основ программирования и сдать экзамен на аттестат зрелости по нашему предмету на минимальном уровне (без возможности продолжать освоение этого предмета);
- либо освоить все вышесказанное — но продолжить изучение предмета еще год и только тогда начать поэтапную сдачу экзаменов на аттестат зрелости.

Для этого подготовительного этапа несколько лет тому назад был написан учебник “Основы компьютерных наук”, для создания которого объединили усилия и школьные учителя, и работники “Института Вайцмана” (Академии наук Израиля): д-р Мордехай Бен-Ари, д-р Давид Гинат, Орна Гершкович, д-р Орна Лихтенштейн, Хана Махлев, Эла Цур, Ифат Коликент, Нурит Райх.

Разделы учебника более или менее соответствуют учебной программе и включают:

1. Введение (что есть компьютер, физические компоненты, программное обеспечение).
2. Алгоритм: основные понятия.
3. Основные компоненты алгоритма и этапы его разработки. Реализация алгоритма в языке Pascal, основные части программы, основные понятия.

4. Основные понятия алгоритмики: теория и практика. Написание простейших программ (с вводом-выводом) на языке Pascal.

5. Условный оператор (минимальная форма `if...then` и составной оператор) в алгоритме и программе. Простые и составные условия.

6. Проверка правильности алгоритма и программы: основные понятия. Таблица ручной прокрутки (трассировки).

7. Циклы в алгоритме и на языке Pascal (`for...do` и `while...do`).

8. Эффективность алгоритма и программы.

9. Вспомогательный алгоритм и функция (`function`) как его реализация в языке Pascal.

10. Массивы.

11. Структурированный алгоритм и подпрограммы.

Следует отметить, что разделы 9—11, строго говоря, не являются обязательными и решение учителя — изучать их или нет — во многом зависит от темпов продвижения класса.

В прошедшем учебном году было выпущено еще несколько учебников и учебных пособий для этого этапа. Однако основная нагрузка при изучении основ компьютерных наук (основ алгоритмики и программирования) приходится на 10—12-е классы, на “старшую” ступень — так как в этом случае речь идет уже об этапе, который должен закончиться успешной сдачей экзамена.

Основной курс

Учебные программы предусматривают возможность “набрать” по специальностям, так или иначе связанным с компьютерами, достаточно много учебных единиц — до 15; естественно, что при этом общее число курсов настолько велико, что обучение растягивается на три года. Для того чтобы сохранить при всем этом разнообразии некое “цементирующее ядро”, предусмотрено, что в основе всех этих вариантов лежит основной курс (“весом” 5 учебных единиц), носящий название “Компьютерные науки”.

Основной курс, в свою очередь, состоит из двух ступеней: *базовой* (3 учебных единицы) и *верхней* (2 учебных единицы); при успешном его завершении ученик имеет в аттестате курс в 5 учебных единиц. К прохождению верхней ступени допускается лишь тот, кто получил “проходную” оценку на базовой. Итоговая оценка (та, что записывается в аттестат зрелости) подсчитывается как средняя между оценками за базовую и верхнюю ступени.

Немного статистики

В 2000 году экзамен на базовую ступень держали почти 15 000 учеников, но продолжили обучение на верхней ступени и экзаменовались в 2001 году только 8350 из них. Средняя оценка экзамена на базовом уровне составила 84,1 (1999 год), 83,3 (2000) и 78,6 (2001). Оценки на верхней ступени несколько ниже — 73,5 (2000) и 72,8 (2001).

С точки зрения учебного материала сегодня существует достаточно большой выбор разделов, из которых можно сформировать учебные курсы базовой и верхней ступеней. Это сделано и для того, чтобы максимально учесть специфику знаний, предпочтений и возможностей учителя, и для того, чтобы охватить как можно более широкий круг сфер сегодняшнего мира программирования.

Структурно *базовая* ступень состоит из двух подкурсов:

1. Основы информатики и программирования (на языке Pascal или Си, еще несколько лет тому назад допускался и Basic, но сегодня он остался лишь в порядке исключения) с письменным экзаменом в конце.

2. Основы программирования в одном из дополнительных направлений — с созданием годового проекта и практическим экзаменом-защитой в конце года (по системе “экзаменуемый — экзаменатор”). Для этой части есть возможность выбрать между следующими направлениями:

- Устройство компьютера и программирование на ассемблере.

- Основы искусственного интеллекта и программирование на языке Prolog.

- Интернет и web-программирование (HTML, JavaScript, ASP).

- Основы компьютерной графики (опять же — программистский аспект).

- Введение в программирование информационных систем (SQL и Access).

- Функциональное программирование (курс полужестко-периментальный, проходит в этом году обкатку в нескольких школах страны).

Первый подкурс охватывает следующие разделы:

1. Модульный подход к разработке алгоритма и написанию программы на языках Pascal и Си (анализ задания для выявления этапов достижения цели, разбиение на модули-подпрограммы, функции и процедуры).

2. Типы данных (стандартные и описываемые, массив как тип данных).

3. Процедуры и массивы (поиск данных в массиве, изменение данных и их расположения в массиве, массив с счетчиков).

4. “Продвинутые” задачи: сортировка массива, слияние массивов, двоичный поиск в массиве. Данные строкового типа.

5. Рекурсия (рекурсивные алгоритмы, реализация рекурсии в языке Pascal).

6. Двухмерные массивы. Записи.

7. Прямая и косвенная рекурсия.

8. Задачи повышенной сложности на сортировку и слияние массивов.

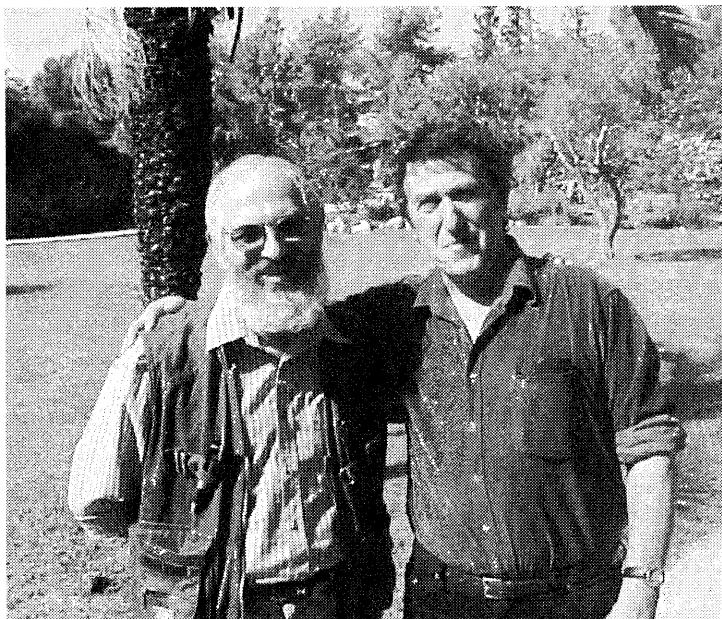
9. Задачи повышенной сложности для работы с данными строкового типа. Файлы.

Следует отметить, что разделы 5 и 7, а также часть раздела 9, посвященная файлам, практически перестали быть обязательными, хотя формально остаются включенными в программу. Видимо, в ближайшие годы нас ждет изменение учебных программ в отношении этих

тем. На этот подкурс выделяется 6 учебных часов в неделю, часть из которых отводится на изучение теории и разбор и решение примеров и задач в классе, а часть — на практические занятия в компьютерном кабинете; распределение времени — по усмотрению учителя. На практике большинство учителей львиную долю учебных часов посвящают теории, используя тот факт, что практически все ученики, выбравшие наш предмет, имеют дома компьютеры и соответствующее программное обеспечение и, кроме того, выбрали компьютерные науки сознательно, так что “давить на совесть” практически не приходится. Письменный экзамен по этому подкурсу длится 3 часа, и его успешная сдача дает ученику 2 учебных единицы из 3; для получения всех 3 единиц ученик обязан успешно справиться и со вторым подкурсом.

На второй подкурс выделяется 3 учебных часа — таким образом, в учебной неделе ученик, выбравший наш предмет, имеет 9 часов компьютерных предметов; это много и является причиной немалых сложностей при составлении учебного расписания — особенно если учесть, что далеко не все ученики класса выбирают этот предмет и ответственно за расписание приходится “запараллеливать” несколько учебных групп. Впрочем, имеет программное обеспечение, которое — увы, всего лишь частично — позволяет автоматизировать процесс. Отметим, к слову, что во многих школах ученики, не желающие изучать дополнительные к основному курсу аспекты “компьютерных наук” (т.е. планирующие получить только 5 учебных единиц), изучают материал, входящий в “базовую ступень”, в течение двух лет вместо года — с соответствующим уменьшением недельной нагрузки.

Второй подкурс завершается практическим экзаменом, который, естественно, включает в себя и теорию, но основу его составляет защита разработанного учеником проекта в виде собеседования. Как уже было сказано, каждый учитель выбирает то направление, в котором он чувствует себя наиболее уверенно — и, естественно, надеется наиболее результативно и эффективно подготовить своих учеников. По каждому из направлений оговорен круг тем, необходимый для изучения, и написаны учебники, а также, когда это представлялось необходимым, разработано и программное обеспечение. В частности, для изучения основ программирования на языке ассемблер в одном из израильских колледжей разработана прекрасная программная среда EasyCPU, которая, как и соответствующий курс, заслуживает отдельной статьи.



Автор статьи (слева) и поэт Игорь Губерман

После успешного завершения базовой ступени ученик может выбрать один из вариантов: либо удовлетвориться достигнутым результатом и получить в аттестат зрелости предмет “весом” в 3 учебных единицы, либо продолжить учебу и получить еще 2 учебных единицы (после успешной сдачи экзамена на *верхней* ступени), так что в итоге в аттестате зрелости у него будет записан курс “весом” в 5 учебных единиц (это весьма существенная разница, которая внимательно учитывается при поступлении в колледжи и университеты).

Верхняя ступень, как и *базовая*, состоит из двух подкурсов. Первый — модульное программирование и структуры данных — един для всех школ. Второй предполагает несколько вариантов, которые выбираются из всех тех же соображений, уже несколько раз упоминавшихся. Сегодня можно выбирать следующие разделы:

- Основы теории алгоритмизации и численных методов.
- Объектно-ориентированное программирование.
- Избранные главы основ общей теории программирования, формальные языки и алгоритмические модели.
- Основы параллельного программирования.
- Архитектура компьютера и язык ассемблер.
- Методы приближенных вычислений и теория ошибок.

На эту ступень выделяются 6 учебных часов в неделю. Курс “Модульное программирование и структуры данных” включает в себя следующие разделы:

1. Программирование “сверху вниз” и библиотечные модули.
2. Динамические переменные (работа с указателями).
3. Оценка эффективности исполнения (алгоритма или программы).
4. Абстрактные типы данных и их реализация в языке Pascal.
5. Стеки.
6. Списки.
7. Бинарное (двоичное) дерево.

Учебник для этого курса был написан группой авторов (учителей и научных сотрудников Еврейского Университета в Иерусалиме, руководитель группы — Офра Бернадес). В него, в частности, включены несколько интересных авторских разработок:

- интерфейсы для написания алгоритмов работы с динамическими типами переменных;
- соответствующие модули на языке Pascal.

Отметим, что данный учебник де-факто стал обязательным учебным пособием для изучения курса, хотя в

последние годы некоторыми учителями были разработаны дополнительные программные продукты и написаны вспомогательные учебные пособия. Во всяком случае, их использование для дополнительной, самостоятельной работы учениками только приветствуется.

Поскольку возможных вариантов для выбора второго подкурса достаточно много, мы приведем перечень тем только для одного из них — “Формальные языки и алгоритмические модели”; он пока еще не стал наиболее массовым, но интерес к нему растет с каждым годом.

Конечные автоматы.

- Формальное описание сложных систем и алгоритмизация решений логических задач.
- Конечный детерминистский автомат.
- Формальные языки.
- Дополнительные модели конечных автоматов.

Стековый автомат.

- Автомат-стек.
- Возможности и применение автомата-стека: достоинства и ограничения.

Машина Тьюринга.

Что еще?

Ученик, заинтересованный в углублении и расширении своих знаний в области компьютеров, может добавить к 5 учебным единицам, “заработанным” за основной курс, еще до 10 учебных единиц. Для этого школа должна получить от Министерства образования разрешение на открытие специализации “Информационные технологии”. Эта специализация включает в себя и основной курс “Компьютерные науки”, и дополнительные, в числе которых

- Разработка систем управления базами данных.
- Компьютерная графика.
- Операционные системы.
- Введение в теорию искусственного интеллекта и экспертные системы.

Несколько последних лет проходит апробацию новая учебная программа, которая включает в себя различные аспекты интернет-программирования — принципы построения и функционирования сети, серверного программирования, обеспечения целостности и данных и безопасности при эксплуатации сети и т.д.

Итогом трехлетнего изучения выбранной специализации (отметим, что каждый курс также построен по модульному принципу и позволяет набрать 1, 3 или 5 учебных единиц) является подготовка работающего реального проекта и защита его перед экзаменационной комиссией.

Несмотря на то, что учеба в рамках специализации (а это дело сугубо добровольное) требует от ученика значительных дополнительных затрат времени — от 4 учебных часов в неделю в 10-м классе до 12—15 часов в выпускном, 12-м, не считая домашней самостоятельной работы, — популярность специализации “Информационные технологии” достаточно высока. Этому есть много причин, назовем только некоторые из них:

- Возможность значительно увеличить количество учебных единиц в аттестате зрелости и тем самым повысить его конкурентоспособность.
- Выпускники специализации в дополнение к обычному аттестату зрелости получают от Министерства труда Израиля “удостоверение профессионала”, подтверждающее, что они владеют специальностью, что немаловажно в условиях свободной конкуренции на рынке рабочей силы.
- Большинство выпускников специализации “Информационные технологии” проходит военную службу в подразделениях, так или иначе связанных с компьютерными технологиями, — в качестве техников, операторов, программистов — Армия Оборона Израиля учитывает их знания и опыт.
- Программа обучения на факультетах и отделениях высоких технологий в израильских университетах и колледжах построена зачастую по принципу “нулевой отметки” — от студента-первокурсника не требуется, скажем, знание алгоритмов или умение программировать. Поэтому те студенты, которые обладают к моменту поступления определенным запасом знаний и навыков, получают значительную фору в учебе, а иногда и освобождение от прохождения базовых курсов.

Впрочем, для самих детей определяющим является скорее чувство приближенности к передовым рубежам технического прогресса и связанная с этим атмосфера “элитарности”, окружающая компьютерные науки.

Немного статистики

Согласно опросу, 19% израильских школьников выбирают предмет для углубленного изучения потому, что “этот предмет важен в дальнейшей жизни”, 14% — потому, что он “интересный”, 6% — потому, что “легкий”, около 10% основывают свой выбор на рекомендации родителей и почти 50% — потому, что об этом предмете “хорошо отзывались друзья”.

Об учителях, задачах и задачниках

Как уже было сказано, обучение компьютерным наукам в израильских школах началось примерно в середине 70-х годов. Постепенно были разработаны учебные программы, которые претерпевали значительные изменения как в результате объективного развития материально-технической базы школ и достижений педагогической науки, так и в результате изменения целевых установок — от компьютерного всеобуча до подготовки квалифицированных кадров для hi-tech-индустрии.

Преподавателей компьютерных дисциплин в израильских школах можно условно разделить на три категории: выпускники компьютерных отделений вузов и колледжей (в основном молодые женщины — мужчины ищут себя в программистских и компьютерных фирмах, где зарплата на порядок выше), переквалифицировавшиеся учителя математики, физики и иных естественных наук (по большей части имеющие большой опыт

преподавания в сочетании с нехваткой программистских навыков) и, в последние 5—6 лет, бывшие программисты, которых жесточайшая конкуренция на рынке труда выталкивает из сферы hi-tech (значительная часть из них — выходцы из бывшего Союза). Для израильской школы, которая, как и школа российская, страдает от “женского засилья”, велик процент учителей-мужчин — до половины преподавателей компьютерных дисциплин в старших классах относятся к “сильной половине” рода человеческого (кстати, среди учеников мальчики составляют примерно две трети).

Построена достаточно стройная система курсов повышения квалификации — практически по всем составляющим учебной программы проводятся 56-часовые семинары, участие в которых является неплохим способом самообразования. Эти семинары организуются как под эгидой Министерства образования — в этом случае учеба бесплатная для всех учителей, так и в рамках той или иной частной фирмы — с соответствующей оплатой. Последние несколько лет ужесточились требования к учителям, работающим в старших классах, — они обязаны иметь академическую степень (М.А., В.А., М.Еd.) по профильной специальности или учиться для получения такой степени.

За последние 5—6 лет в Израиле практически по всем учебным курсам, как основным, так и дополни-

тельным, написаны учебники, так что проблема теории в большей или меньшей степени решена. Гораздо более проблемной остается ситуация с задачами: для работы на уроке, для самостоятельной работы учеников, для дополнительной подготовки. Конечно, учитель со стажем имеет так или иначе в своей копилке набор наиболее типичных задач и в состоянии составить на основе этого набора собственные варианты. Кроме того, всегда есть возможность обложиться учебниками и “надергать” подходящий материал из них; одним из самых популярных остается путь взаимообмена с коллегами — правда, и он имеет свои минусы. Очень важным источником остается Интернет — на сайте Отдела преподавания информационных технологий и компьютерных наук Министерства образования наряду с официальной информацией размещены и методические пособия, экзаменационные задания последних лет, ряд экзаменов и on-line-курсов, разработанных теми или иными учителями, а также действуют несколько форумов по различным тематикам.

Идеальным был бы выпуск задачников — но вот как раз с этим проблема остается на уровне малоподнятой целины: задачников за последние 10 лет было выпущено фактически только два, да и те из-за весьма динамического изменения учебных программ быстро устаревают.

ВНИМАНИЕ: ПРОЕКТ “ЧИТАТЕЛЬСКИЙ ЗАКАЗ. ЖАРКОЕ ЛЕТО-2003”

Только начинается новый учебный год, а уже не за горами “Жаркое лето-2003”. Традиция формирования плана летних тематических выпусков требует, чтобы первый предварительный (но, как наши постоянные читатели знают, очень мало изменяющийся) план был опубликован в № 8/2003. В прошлые годы план летних номеров формировался нами на основе анализа результатов читательских анкет и встреч с подписчиками. Так оно будет и в этот раз: будут и встречи, и обсуждения. Но в этом году мы хотим предложить всем подписчикам принять более активное участие в формировании плана номеров “Жаркого лета-2003”.

Что для этого надо сделать?

На первом этапе представьте себя главным редактором “Информатики” с неограниченными возможностями (в отличие от реального главного редактора ☺). Представьте, что у вас нет никаких ограничений и вы можете заказать на лето любые номера, какие захотите. Составьте соответствующий план и пришлите его в редакцию. Учитывайте, что к названию номера, конечно, можно подходить творчески, но тематика его должна быть ясна, поэтому при необходимости уточняйте, пожалуйста, что вы имеете в виду. Включите в план столько номеров, сколько захотите, сколько придумаете.

Второй этап проекта начнется в октябре. Начиная с октября мы будем публиковать обновляемый список-рейтинг поступивших заявок. На этом этапе вы по-прежнему сможете добавлять свои заявки в список, но, кроме этого, бланк заявки можно будет использовать для голосования за номера, уже включенные в список.

“Выдвижение” номеров и голосование закончится в декабре. А в № 8/2003 мы по традиции опубликуем план номеров “Жаркого лета-2003”, сформированный на основе результатов голосования (но уже учитывающий и реальные возможности редакции).

А призы?

Все, кто примет участие в “выдвижении” номеров и голосовании, получат подарок от редакции “Информатики” — специальный подарочный номер (поэтому, пожалуйста, аккуратно указывайте свой полный почтовый адрес). Подписчики, принявшие наиболее активное участие в акции, получат специальные призы, среди которых подписка на второе полугодие 2003 года и приглашение на День учителя информатики, который состоится в апреле 2003 года в рамках Второго московского педагогического марафона.

Я предлагаю включить в серию “Жаркое лето-2003” следующие номера:

название номера _____

название номера _____

название номера _____

название номера _____

Мой почтовый адрес: _____

Мой электронный адрес: _____

Мой телефон: _____

Электронный адрес и телефон указывать не обязательно, но желательно. Это позволит при необходимости связаться с вами оперативно.



Каждый выбирает свой ПИК!

В.Н. Пинаев,
г. Рыбинск

От автора

В этой статье приведены условия нескольких задач, которые были предложены для решения на шестом чемпионате "ПИК-2002". Для педагогов эта подборка может быть полезна с методической точки зрения, так как дает представление о задачах, решения которых оцениваются не в абсолютном, а относительном выражении. В отличие от типовых олимпиад, где решение, как правило, признается либо верным, либо ошибочным, у нас практически любой ответ (программа, алгоритм, файл и пр.) является допустимым и участвует в розыгрыше. Конкретные оценки зависят от относительных результатов, показанных в сравнении решений друг с другом в компьютерных турнирах. Примеры заданий прошлых лет вы можете найти на сайте <http://pic200x.chat.ru> или в моей прошлой статье [3].

Вы не найдете здесь полных решений, а познакомитесь лишь с условиями заданий и комментариями к ним. Отсутствие полных решений объясняется тем, что я сам не пытался конкурировать с участниками. А, во-вторых, как уже подчеркивалось выше, сила любого решения относительна, и все зависит, как и в реальной жизни, от соперников.

Я буду считать свою задачу выполненной, если вы найдете здесь для себя что-то новое, необычное и интересное; если эта статья приведет вас к своей оригинальной задаче; если вы проведете у себя в вузе, школе, классе или кружке свой компьютерный турнир.

Критические замечания, пожалуйста, направляйте по адресу: pinaev@rgata.adm.yar.ru.

Каждый из нас лишь выиграет, создавая время от времени "игрушечные" программы с заданными искусственными ограничениями, заставляющими нас до предела напрягать свои способности...

Искусство решения мини-задач на пределе своих возможностей оттачивает наше умение для реальных задач.

Д.Кнут¹

Вниманию читателей предлагаются условия и краткий разбор заданий чемпионата по информатике и программированию "ПИК-2002". О самом чемпионате "ПИК" уже рассказывалось в 2000 году на страницах газеты [3].

Кратко упомянем отличительные особенности нашего состязания.

Чемпионат изначально был задуман как альтернатива олимпиадам, где от участника требуется прежде всего скорость решения, возможно, в ущерб даже качеству. Мы предлагаем наши задачи для домашнего решения на 2 недели. Безусловно, за это время можно и почитать книги, и пообщаться с товарищами и учителями, и перепробовать различные варианты. Эти обстоятельства, с одной стороны, повышают качество решений, с другой — не позволяют жюри гарантированно определить автора решения. Поэтому участникам-абитуриентам мы не предоставляем традиционные для предметных олимпиад льготы при поступлении в вуз.

Все участники чемпионата делятся по уровню образования на три лиги: учащиеся (лига U), студенты (лига S), профессионалы (лига P). К последней лиге мы относим выпускников, аспирантов, студентов старших курсов, победителей лиги S прошлых лет, а также всех желающих.

Большинство заданий чемпионата формулируются таким образом, чтобы была возможность сравнить решения по ряду параметров (время, число шагов, размер текста и т.п.). Это позволяет проводить компьютерные турниры, когда программы состязаются между собой в присутствии самих авторов. Именно эта особенность делает наши чемпионаты зрелищными и запоминающимися.

Вместе с условиями заданий участники получают вспомогательное программное обеспечение, которое разрабатывается студентами Рыбинской государственной авиационной технологической академии имени П.А. Соловьева.

Опыт проведения чемпионата позволяет утверждать, что такая форма состязания интересна, поучительна и нужна учащейся и студенческой молодежи.

В приведенной ниже таблице перечислены задания чемпионата с распределением по лигам. Знаком "+" обозначены задания, решения по которым должны были сдать участники указанной лиги. Разрешалось сдавать решения не всех заданий.

Задания	Лига U, учащиеся	Лига S, студенты	Лига P, профессионалы
Задание А. "Энтропия"		+	+
Задание В. "Сканер"	+	+	+
Задание С. "Ханойские башни"		+	+
Задание D. "Шифровка"	+		
Задание E. "Таблица"	+		

¹ Текст эпиграфа взят из речи Д.Кнута "Программирование как искусство" при вручении ему премии Тьюринга в 1974 году [2].

Задание А. “Энтропия”

Вашему вниманию предлагается программа², моделирующая работу некоего “черного ящика”. Для этого устройства определена размерность n , которая задается пользователем. Кроме того, имеются два входа ($In1$, $In2$) и один выход (Out). Внутри “черного ящика” также хранится секретный ключ key (натуральное число) и его состояние $sost$ (булево значение), зная которые можно точно определить выход “черного ящика” как функцию от его входов: $F(In1, In2, key, sost) = Out$.

На входы $In1$ и $In2$ должны подаваться непересекающиеся множества целых чисел от 1 до n , где $3 \leq n \leq 666$. Выход “черного ящика” — это один из символов “<”, “>” или “=”.

Ваша задача состоит в том, чтобы разобраться, как работает “черный ящик”. В конечном итоге необходимо, подавая информацию на входы ящика и получая значение выхода, по возможности быстрее определить значение ключа.

Модель “черного ящика” представлена программой *box.exe*³. Данная программа в процессе работы создает текстовый файл *protocol.box*, в котором протоколируются все запросы и ответы “черного ящика”. В качестве дополнительной информации “черный ящик” выводит на экран некоторую количественную оценку входного набора данных, которая должна помочь (или помешать?) в формировании запросов.

Входы “черного ящика” можно формировать либо вручную, либо вызывая внешнюю программу (программу участника). Программа участника анализирует файл *protocol.box* и формирует запрос к “черному ящику”, создавая файл входов *in.box*, или “сообщает” ключ ящика, создавая файл *key.box*. Ответ на свой запрос программа сможет узнать только при следующем вызове, анализируя файл *protocol.box*.

Победителем будет считаться тот участник, чья программа узнает ключ за меньшее число своих вызовов по совокупности тестов.

Формат файла *protocol.box*

Первая строка — размерность “черного ящика”. Эта строка всегда присутствует в файле.

Далее на каждый выполненный запрос перечисляются:

- количества чисел, поданных на первый и второй входы;
- числа, подаваемые на первый и второй входы;
- выход “черного ящика” (один символ).

Все числа отделяются друг от друга пробелами и могут размещаться на нескольких строках.

Пример файла *protocol.box*

```
13
2 2 1 7 3 4
>
```

Здесь указана размерность “черного ящика” (13), в первом запросе на вход $In1$ подаются два числа: 1 и 7, а на вход $In2$ — числа 3 и 4. Результат запроса (выход) — “>”.

Формат файла *in.box*

Сначала указываются размеры наборов чисел, подаваемых на первый и второй входы. Далее перечисляются сами числа, подаваемые на первый и второй входы. Числа отделяются друг от друга пробелами и могут размещаться на нескольких строках.

Пример файла *in.box*

На вход $In1$ подаются 1 и 7, а на вход $In2$ — 3 и 4

```
2 2 1 7 3 4
```

Формат файла *key.box*

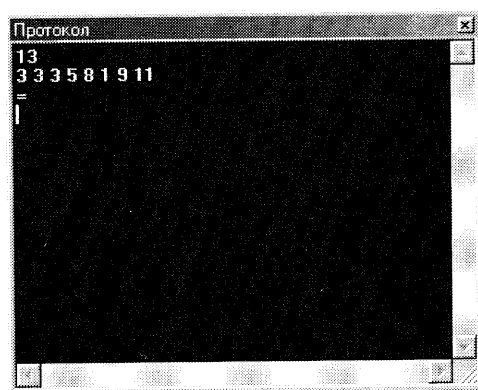
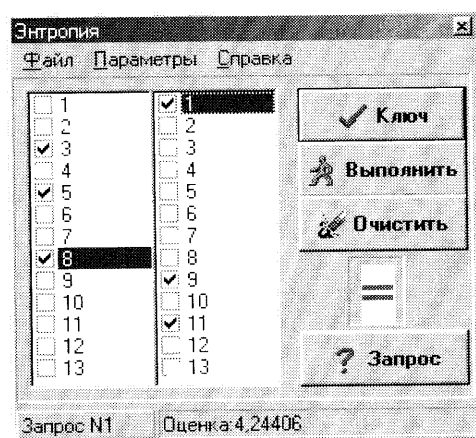
Файл содержит одно число — ключ “черного ящика”.

Ограничения для программ

Программа не должна изменять и создавать никаких файлов, кроме *in.box* и *key.box*, и может использовать файл *protocol.box* только для чтения.

Время работы программы не должно превышать 3 секунд на компьютере типа Pentium III (500 МГц).

Пример работы программы *box.exe* и пример файла протокола приведены ниже.



Комментарий к заданию

Анализ решений показал, что далеко не все участники поняли назначение “черного ящика”. Ниже приводятся несколько присланных объяснений, которые иллюстрируют диапазон толкований.

² Эту программу и все программное обеспечение чемпионата можно взять на сайте <http://pic200x.chat.ru>.

³ Автор программы — Артем Кудрявов.

“Черный ящик” — это весы с двумя чашами.

Числа от 1 до n ($n = 3..666$) — это номера монет, одна из которых является фальшивой (легче или тяжелее других). Ключ — это номер фальшивой монеты.

Исходное условие задачи можно сформулировать так: определить номер фальшивой монеты при помощи весов (“черного ящика”) за минимальное число взвешиваний.

Ключ — загаданное число.

При равном количестве элементов последовательностей $In1$, $In2$ и если ни одна из них не содержит ключ, то выход “=”; если одна из последовательностей содержит ключ, то выход “<” или “>”.

На мой взгляд, “черный ящик” работает следующим образом.

Если количество элементов первого множества меньше количества элементов второго множества, то выходом будет “<”. Если количество элементов первого множества больше количества элементов второго множества, то выходом будет “>”.

Если количество элементов во входных множествах одинаково, то:

- если ключ входит в первое множество, то выход “<” или “>” (“<” или “>” определяется тем самым неким состоянием ключа);
- если ключ входит во второе множество, то выход “>” или “<”;
- если ключ не входит ни в одно из входных множеств, то выход “=”.

Еще я посмотрел в словаре значение слова “энтропия”. Думал в нем подсказку найти. Оказалось, что это термин из физики. Кинулся к справочнику по физике, оказалось, это функция, описывающая термодинамическое состояние систем и связанная с вероятностью нахождения системы в конкретном (стабильном) состоянии. Сначала я подумал, что ключ — это и есть стабильное состояние, а символы “<” и “>” “показывают” направление к этой стабильности. Но при чем тут тогда количество элементов в множестве (а они на выход влияют)? И почему тогда знак то “<”, то “>” указывает, где это состояние? В результате я решил, что есть другие значения слова “энтропия”, которых я не нашел в словаре, и на этом успокоился :).

Авторская идея “черного ящика” совпадает с первым приведенным мнением. Действительно, программа *box.exe* моделирует работу обычных чашечных весов без гирь, которыми пользуются для того, чтобы найти одну фальшивую монету среди n имеющихся. Что касается названия задания, то вот как определяется это понятие в толковом словаре [7]: “Энтропия (*entropy*) — мера количества информации, вырабатываемой источником, попадающей к получателю в пересчете на символ (секунду и т.п.). По-

нятие энтропии в теории информации введено К.Э. Шенноном в 1948 году и позднее развито другими исследователями. ...Энтропия (в зависимости от выбранного основания логарифма, используемого в основной формуле) измеряется в битах, натах или Хартли. Термин “энтропия” взят по аналогии с энтропией в термодинамике, где она определяется выражением, имеющим ту же форму (с точностью до множителя) и знака”.

Существует простой способ выявления фальшивой монеты, если известно, легче она или тяжелее настоящей. В этом случае необходимо делить исходное множество на три по возможности равных подмножества (здесь и далее см. [9], гл. III). Если же мы не знаем, как отличается вес фальшивой монеты от веса настоящей, то мы можем потратить одно лишнее взвешивание для установления этого факта. Отсюда следует, что при правильной стратегии, которая может быть устроена и по-другому, мы можем выиграть не более одного взвешивания. Именно за это взвешивание на каждом тесте фактически и шла борьба. Например, для тринадцати монет мы гарантированно можем найти фальшивую за 3 взвешивания⁴. Формула энтропии, предложенная Шенноном, позволяет проверять, является ли проводимое взвешивание максимально информативным.

Задание В. “Сканер”

Рассмотрим особый сканер, который предназначен для распознавания печатного текста. Этот сканер имеет матрицу из 7 чувствительных элементов (нумерация элементов сверху вниз, начиная с 0). Он передвигается по тексту слева направо.

За один такт сканируется один столбец изображения символа, причем пользователь должен выборочно включить показ чувствительных элементов.

Пример. На рис. 1 серым фоном выделены включенные элементы (0, 1 и 4) второго столбца.

*						
	*					
	*	*	*			
					*	
	*				*	
	*	*	*			

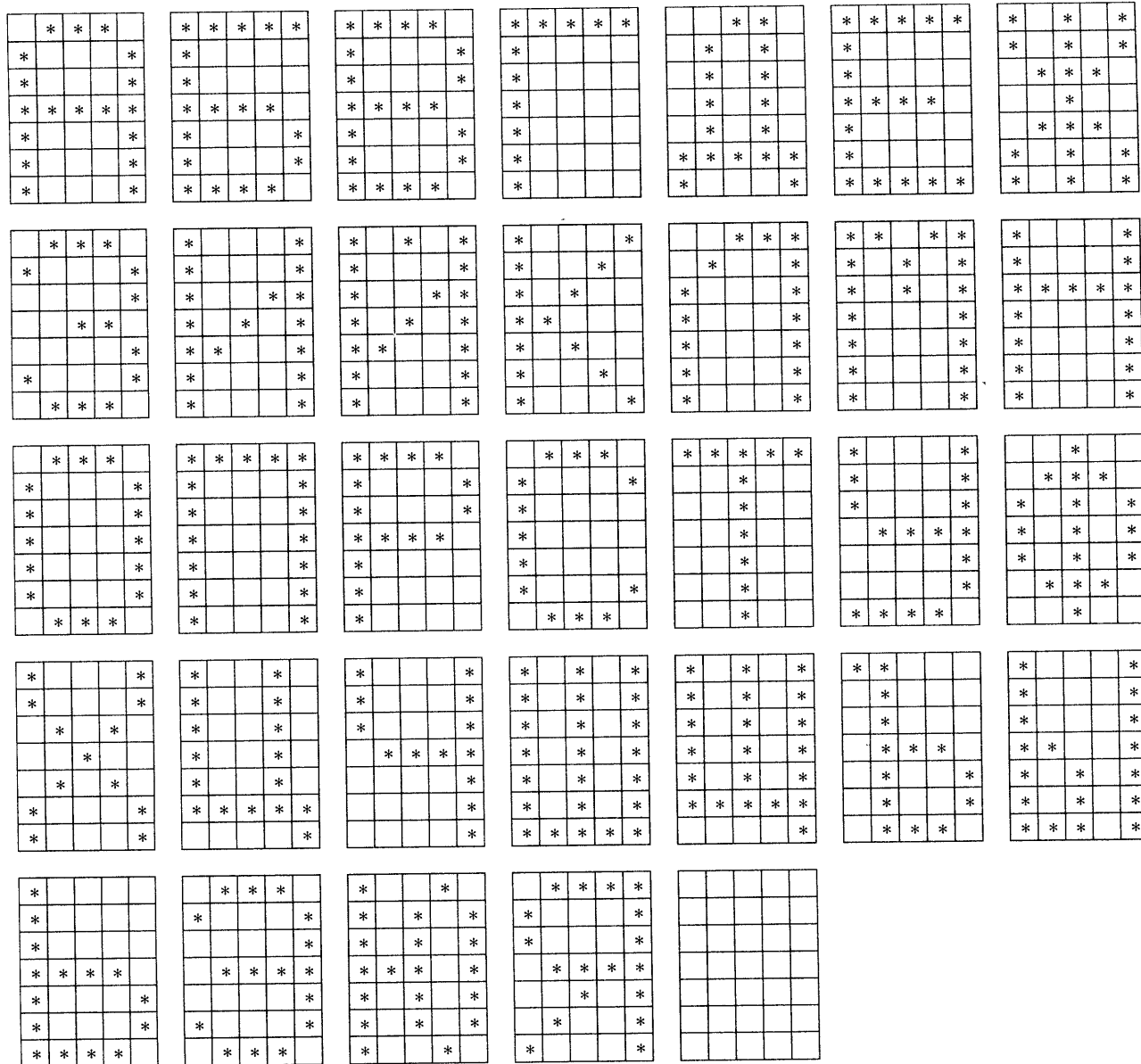
Рис. 1

На с. 21 приводится таблица изображений всех символов (включая пробел).

Участникам предстоит распознать текст, включая как можно меньше чувствительных элементов сканера.

Каждый участник после регистрации первоначально имеет на своем счете 1000 очков. За каждый включенный элемент сканера с участника списывается определенное число очков. Для участников лиги U стоимость просмотра элемента равна 1. Для участников лиг S и P стоимость включения элемента равна номеру столбца, в котором он находится.

⁴ Для любителей таких задач могу предложить следующий вариант условия с подвохом. Банкир знает, что у него в мешке с монетами есть одна фальшивая, отличающаяся от настоящей своим весом. Используя чашечные весы без гирь, он сравнивает пары монет между собой и складывает проверенные монеты в сторону. Спустя некоторое время к банкиру подходит математик, который, оценив ситуацию, заявляет: “Я мог бы сейчас найти фальшивую монету за 3 взвешивания!”. Оцените минимальное и максимальное количество еще не проверенных монет. (Подсказка: максимальное количество еще не проверенных монет больше 13.)



Например, при сканировании 2-го столбца целиком участник лиги U теряет 7 очков, а участники лиг S и P теряют $7 \cdot 2 = 14$ очков. Так как каждый символ представлен таблицей из 7 строк и 5 столбцов, то полное сканирование символа обойдется участникам разных лиг в 35 и 105 очков соответственно.

Если участник досрочно сообщает очередную распознанную букву, матрица сканера перемещается на первый столбец следующего символа.

За каждую верно угаданную букву участнику начисляется премия в 105 очков. Если на счете участника будет недостаточно очков, то будет невозможно включать чувствительные элементы и придется угадывать буквы вслепую.

Итак, любой символ сканируемого текста представляется таблицей 7×5 . В тексте⁵ могут использоваться 32 заглавные русские буквы и пробел. Пробел кодируется полностью пустой таблицей. Изображения всех сим-

волов доступны участникам через соответствующий массив (описание типов данных и переменных см. ниже).

Решение должно быть оформлено в виде программы на одном из следующих языков: стандартный язык Pascal, Object Pascal (без каких-либо модулей, кроме указанного ниже) или C++⁶. Служебные функции содержатся в модулях *letters.cpp* и *letters.pas* и описаны ниже. Участникам запрещается использовать любые вспомогательные файлы.

Решение может быть дисквалифицировано за нечестную игру (попытка изменения счета, чтение символов напрямую из файла или служебного массива и т.п.).

Программа участника продолжает играть до тех пор, пока ей не будет передан весь текст (служебные модули считывают этот текст из файла *letters.txt*). Длина текста предположительно будет составлять несколько тысяч символов. Общее время работы программы не должно превышать 5 секунд на компьютере с процессором, аналогичным Pentium III (500 МГц).

⁵ Распознаваемый текст является осмысленным текстом на русском языке. Знаки препинания и другие символы будут пропущены сканером.

⁶ Для компиляции решений использовались компиляторы Borland C++ 5.5 for Win 32 и Borland Delphi 5.0 (version 13.0).

С++ Интерфейс (модуль letters.cpp) для задачи “Сканер”

Число символов алфавита (включая пробел): **const int** max_letters = 33;

Число строк в изображении буквы: **const int** max_line = 7;

Число столбцов в изображении буквы: **const int** max_col = 5;

Счет участника (только для чтения!): **long int** score;

Изображения символов: **const int** Letters[max_letters][max_line][max_col];

Функция инициализации: **void** registration(**char** *ID, **char*** Name, **int** &messageLength)

ID — идентификатор (регистрационный номер) участника,

Name — имя участника (команды), под которым он будет выступать во время игры;

messageLength — количество символов в сообщении.

Функция отправки распознанного символа: **int** sayLetter(**char** letter)

letter — порядковый номер символа (символ 0 — это буква “А”, символ 32 — это пробел).

Функция возвращает ненулевой результат (**true**), если символ распознан правильно.

Функция также перемещает матрицу сканера на следующий символ.

Функция сканирования следующего столбца: **char*** getNextColumn(**char** lines[max_line])

Эта функция возвращает указатель на очередной столбец изображения символа. Через параметр lines передаются номера элементов сканера, которые должны быть включены (программа передает массив из 7 символов, в котором символ “1” означает, что элемент сканера включен). Если все столбцы одного символа были переданы, то функция возвращает указатель на массив {0, 0, 0, 0, 0, 0, 0} (для перехода к следующему символу нужно использовать функцию sayLetter).

Функция завершения игры: **void** done(**void**)

Pascal Интерфейс (модуль letters.pas) для задачи “Сканер”

Число символов алфавита (включая пробел): **const** max_letters = 33;

Число строк в изображении буквы: **const** max_line = 7;

Число столбцов в изображении буквы: **const** max_col = 5;

Счет участника (только для чтения!): **var** score: **longint**;

Изображения символов: **const** Images: **array**[0..max_letters-1, 0..max_line-1, 0..max_col-1] **of** **byte**;
{непустой символ клетки таблицы кодируется ненулевым значением}

Процедура инициализации: **procedure** registration(ID, Name: **string**; **var** messageLength: **integer**);

ID — идентификатор (регистрационный номер) участника,

Name — имя участника (команды), под которым он будет выступать во время игры;

messageLength — количество символов в сообщении.

Функция отправки распознанного символа: **function** sayLetter(letter: **byte**): **boolean**;

letter — порядковый номер символа (символ 0 — это буква “А”, символ 32 — это пробел).

Функция возвращает результат **true**, если символ распознан правильно.

Вызов функции одновременно перемещает матрицу сканера на следующий символ.

Тип TColumn (один столбец изображения символа): **type** TColumn = **array**[0..max_line-1] **of** **byte**;

Функция сканирования следующего столбца: **procedure** getNextColumn(scanner_in: TColumn;
var scanner_out: Tcolumn);

Через параметр scanner_in процедура передает модулю номера элементов сканера, которые должны быть включены (массив из 7 байт, в котором 1 означает, что элемент сканера включен). Через параметр scanner_out процедура получает очередной столбец изображения символа.

Примечание. Значения невключенных элементов сканера в массиве scanner_out не изменяются. Если все столбцы одного символа были переданы, то переменная scanner_out не изменяется. Для перехода к следующему символу нужно использовать функцию sayLetter.

Процедура завершения игры: **procedure** done;

Комментарий к заданию

Эта задача вызвала у участников наибольший интерес. Лучшие решения были основаны на анализе буквенных сочетаний, что позволяло отбрасывать многие варианты как маловероятные. В конечном итоге это приводило к существенной экономии потраченных на “вскрытие” очков.

Приводимые ниже таблицы участников лиги U и лиги S дают представление о разбросе набранных очков. Для участников лиги U был выбран текст из 7612 символов, для участников лиги S — текст из 8733 символов.

Место	Угаданные буквы	Очки	Место	Угаданные буквы	Очки
1	7612 из 7612	754 843	1	8733 из 8733	831 218
2	7612 из 7612	749 617	2	8732 из 8733	829 230
3	7597 из 7612	742 868	3	8733 из 8733	828 564
4	7612 из 7612	740 261	4	8733 из 8733	827 463
5	7430 из 7612	739 984	5	8733 из 8733	827 215
6	7612 из 7612	734 840	6	8733 из 8733	824 977
7	7592 из 7612	704 362	7	8733 из 8733	822 542
8	7460 из 7612	700 193	8	8733 из 8733	815 350
9	6677 из 7612	657 534	9	8427 из 8733	807 562
10	7612 из 7612	587 124	10	8728 из 8733	800 835
11	7612 из 7612	541 452	11	8733 из 8733	739 893
12	5351 из 7612	518 937	12	8190 из 8733	660 091
13	4202 из 7612	413 300	13	5952 из 8692	253 719
14	945 из 7612	98 702			

Задание С. “Ханойские башни”

В теории программирования хорошо известна так называемая машина Тьюринга, предложенная для исследования понятия вычислимости. Кратко опишем эту абстрактную конструкцию.

Машина Тьюринга — это автомат, который обрабатывает потенциально бесконечную ленту, разбитую на ячейки-клетки, одна из которых является обозреваемой (текущей). Блок управления машины Тьюринга в каждый момент времени хранит некоторое текущее состояние. За один шаг блок управления по паре <текущий символ на ленте, текущее состояние> может сменить текущий символ и текущее состояние и переместиться по ленте налево (символ “<”) или направо (символ “>”) на одну клетку или остаться на месте (символ “=”).

Программа для машины Тьюринга — это таблица из пяти столбцов (см. пример ниже).

Будем считать, что первоначально машина Тьюринга находится в состоянии 1. На ленте размещены исходные данные, ограниченные слева и справа символами “#”. Блок управления в начальный момент обозревает первый слева символ входных данных. В одну ячейку ленты разрешено запи-

сывать один символ из алфавита {“А”..“Z”, “#”}. В таблице не должно быть двух и более строк, задающих одинаковые пары <текущий символ, текущее состояние>. Состояние машины Тьюринга задается целым числом без знака.

Машина Тьюринга прекращает работу, если не предусмотрено действие в программе-таблице для текущей пары <символ, состояние>.

Задание. Составьте для машины Тьюринга программу, при выполнении которой с исходного стержня А на любой другой (В или С) перекадываются все диски игры-головоломки “Ханойские башни”.

Решение представляется в файле <регистрационный номер>.s.txt.

Описание игры-головоломки “Ханойские башни”. Суть игры состоит в следующем. Имеются три стержня: А, В, С. На стержне А размещено в убывающем снизу вверх порядке диаметров некоторое количество дисков. За один ход разрешается переложить верхний диск с одного стержня на один из двух других стержней при соблюдении условия: недопустимо класть диск большего диаметра на диск меньшего диаметра.

Цель игры. Переместить все диски со стержня А на любой другой стержень.

Входные данные. Каждому диску соответствует своя ячейка ленты. Первоначально все диски находятся на стержне А, поэтому все ячейки ленты содержат символ А. Слева и справа от этих ячеек размещены символы “#”. Блок управления первоначально указывает на самую левую ячейку с символом “А”, соответствующую диску минимального диаметра.

В процессе игры ячейки ленты машины Тьюринга, соответствующие дискам, должны изменяться строго по правилам игры. Работа машины Тьюринга прекращается, если какой-либо диск (то есть соответствующая ему ячейка) перемещается либо с нарушением правил игры, либо переносится на несуществующий стержень. Остальные ячейки ленты могут принимать любые значения из определенного алфавита.

Пример. Ниже приведен пример таблицы-программы, руководствуясь которой машина Тьюринга выполнит перекадывание трех дисков. Входные данные: “...#AAA#...”.

Текущий символ на ленте	Текущее состояние	Новый символ на ленте	Новое состояние	Движение
A	1	B	2	>
A	2	C	3	<
B	3	C	4	>
C	4	C	5	>
A	5	B	6	<
C	6	C	7	<
C	7	A	8	>
C	8	B	9	<
A	9	B	0	=

Работа приведенной программы проиллюстрирована на с. 24, где приведена лента после выполнения каждого шага. Обозреваемый символ помечен серым.

* Безусловно, участники проделали здесь громадную работу, на которую жюри не решилось, поэтому у жюри, увы, нет своего решения.

Номер шага	Лента машины Тьюринга (после выполнения шага)	Текущее состояние	Комментарий
0	... # A A A # ...	1	Все диски на стержне А
1	... # B A A # ...	2	Первый диск перемещен на В
2	... # B C A # ...	3	Второй диск перемещен на С
3	... # C C A # ...	4	Первый диск перемещен на С
4	... # C C A # ...	5	Перемещение к третьему диску
5	... # C C B # ...	6	Третий диск перемещен на В
6	... # C C B # ...	7	Перемещение к первому диску
7	... # A C B # ...	8	Первый диск перемещен на А
8	... # A B B # ...	9	Второй диск перемещен на В
9	... # B B B # ...	0	Первый диск перемещен на В

Лучшим будет признано решение, которое решает головоломку “Ханойские башни” за наименьшее число шагов машины Тьюринга по совокупности всех тестов.

Число дисков головоломки будет изменяться от 1 до 15.

Ограничения. Количество строк таблицы не должно превышать 2000. Количество используемых состояний не должно превышать 100. Количество используемых ячеек ленты не должно превышать 1000. Количество шагов машины Тьюринга не должно превышать разумного предела⁸.

Для отладки программы жюри предоставляет желающим участникам тестовую программу *turing.pas* (имя файла с таблицей управления передается параметром в командной строке).

Формат файла с таблицей управления.

В файле перечисляются строки таблицы управления. Все элементы отделяются друг от друга пробелом. Текст от знака “%” до конца строки считается комментарием и не учитывается. Комментарий либо занимает всю строку (тогда знак процента записывается первым в строке), либо следует за строкой управления.

В первой строке файла обязательно указываются регистрационный номер и Ф.И.О. участника. Далее следуют строки таблицы управления и возможные комментарии.

Пример файла с решением головоломки.

% P000 Пинаев Владимир

A 1 B 2 > % переместили первый диск на стержень В

A 2 C 3 <

B 3 C 4 >

C 4 C 5 >

A 5 B 6 <

C 6 C 7 <

C 7 A 8 >

C 8 B 9 <

A 9 B 0 =

Комментарий к заданию

Машина Тьюринга — это хорошо известная формальная алгоритмическая модель. Читатели могут познакомиться подробнее с машиной Тьюринга по публикации Б.Ф. Трахтенброта [8]. В нашей интерпретации машина Тьюринга останавливается, если в таблице управления не предусмотрена строка для пары <текущий символ, текущее состояние>. В изложении Трахтенброта требуется, чтобы машина перешла в особое заключительное состояние (стоп-состояние). Понятно, что можно договориться считать любую непредусмотренную ситуацию стоп-состоянием.

С этой задачей связана следующая интересная история. Десять лет назад студент второго курса РГАТА Иван Басов, ныне живущий и работающий в США, предложил запрограммировать головоломку “Ханойские башни” на машине Тьюринга. Тогда же Иван предложил удобный способ описания размещения дисков по стержням, благодаря чему и он, и я написали каждый свою программу для машины Тьюринга. В основе наших решений лежал известный простой рекурсивный алгоритм перекладывания дисков [1].

Основная идея этого алгоритма⁹ звучит так.

1. Каждый нечетный ход — это циклический ход первого диска: либо в направлении $A \rightarrow B \rightarrow C \rightarrow A \rightarrow B \rightarrow C \rightarrow \dots$ (в случае четного количества дисков), либо в направлении $A \rightarrow C \rightarrow B \rightarrow A \rightarrow C \rightarrow B \rightarrow \dots$ (в случае нечетного количества дисков).

2. Каждый четный ход определяется однозначно (!): находим наименьший отличный от первого диск и перемещаем его на тот стержень, где нет первого диска.

В нашем случае задача упрощается, так как по условию разрешено перемещать башню на любой стержень, и поэтому нам не потребуется определять четность количества дисков.

Окончательно имеем следующий алгоритм:

1. Перемещаем на один шаг по кругу $A \rightarrow B \rightarrow C \rightarrow A$ первый диск;

2. Перемещаемся по дискам вправо до тех пор, пока идентификатор стержня совпадает с идентификатором стержня, на котором лежит первый диск;

3. Если обнаружен знак конца дисков “#”, то работа закончена;

4. Если найден диск, лежащий на стержне, отличном от стержня первого диска, то перемещаем найденный диск на допустимый стержень (это третий оставшийся стержень, если исключить стержни первого и текущего дисков);

5. Возвращаемся к пункту 1.

На с. 25 приведена соответствующая описанному алгоритму таблица управления.

⁸ В тестовом модуле, который был доступен всем участникам, стояло ограничение: $10 \cdot 2^n$.

⁹ Кстати, здесь возникает интересный вопрос обоснования правильности этого алгоритма. Пример доказательства по индукции приводится в книге А.А. Брудно и А.И. Каплана [1].

№ строки	Текущий символ на ленте	Текущее состояние	Новый символ на ленте	Новое состояние	Движение	Комментарий
1	A	1	B	2	>	Ход первым диском с A на B (запоминаем этот ход через состояние 2)
2	B	1	C	3	>	Ход первым диском с B на C (запоминаем этот ход через состояние 3)
3	C	1	A	4	>	Ход первым диском с C на A (запоминаем этот ход через состояние 4)
4	B	2	B	2	>	Пропускаем диски на стержне B (состояние 2 означает, что первый диск находится на стержне B)
5	A	2	C	5	<	Если нашли диск на стержне A, то переместим его на стержень C и начинаем подъем к первому диску
6	C	2	A	5	<	Если нашли диск на стержне C, то переместим его на стержень A и начинаем подъем к первому диску
7	C	3	C	3	>	Пропускаем диски на стержне C
8	B	3	A	5	<	Если нашли диск на стержне B, то переместим его на стержень A
9	A	3	B	5	<	Если нашли диск на стержне A, то переместим его на стержень B
10	A	4	A	4	>	Пропускаем диски на стержне A
11	B	4	C	5	<	Если нашли диск на стержне B, то переместим его на стержень C и начинаем подъем к первому диску
12	C	4	B	5	<	Если нашли диск на стержне C, то переместим его на стержень B и начинаем подъем к первому диску
13	A	5	A	5	<	Подъем к первому диску
14	B	5	B	5	<	Подъем к первому диску
15	C	5	C	5	<	Подъем к первому диску
16	#	5	#	1	>	Переходим в начальное состояние

Как видите, решение оказалось не таким уж и сложным.

Однако в нашем задании нужно было найти не любое решение, а то, в котором число шагов машины Тьюринга будет по возможности минимальным. Экономия здесь может быть достигнута за счет подбора состояний таким образом, чтобы не выходить за пределы дисков и не обнаруживать символ “#”.

Последнее было возможным, так как заранее были известны ограничения на количество дисков. Решения победителей, хотя и выигрывают по числу шагов, уже недостаточно компактны, чтобы быть напечатанными на страницах газеты.

Вернемся к истории с Иваном Басовым. Десять лет назад Иван, будучи студентом второго курса, написал исследование, которое было представлено на конкурс студенческих научных работ. Исследуя машину Тьюринга и головоломку “Ханойские башни”, он провел анализ нескольких алгоритмов, оценил число шагов машины Тьюринга, подробно описал программную реализацию и в том числе привел таблицу управления, подобную вышеприведенной.

Пришедший из Новосибирского государственного университета отзыв доктора физико-математических наук и уважаемого профессора обескуражил нас своей категоричностью: “Работа вызывает, с одной стороны, уважение своей

законченностью, аккуратностью изложения, тщательностью проработки, искусством программирования. С другой стороны, работа вызывает недоумение — проделана весьма тяжелая и квалифицированная работа — зачем? Что принципиально новое для других открыл автор, “подковав” эту программистскую “блоху”?”.

Готовя эту рукопись, я снова и снова задавался вопросом: оправданны ли такие задачи? Разве погружение в предмет не способствует в конечном итоге более глубокому пониманию существа дела, а учащиеся и студенты, решающие эту и подобные ей задачи, не получают удовлетворения? Пример решения головоломки на машине Тьюринга, пожалуй, более содержателен, чем операции над унарными числами, которыми изобилуют примеры в публикациях по этой тематике (см., например, [6]). Именно поэтому я и вынес слова Кнута в эпиграф.

К слову сказать, в 1996 году на первом в России полуфинале командного чемпионата мира по программированию среди студентов была предложена задача: составить для машины Тьюринга программу умножения двух двоично-десятичных чисел. Тогда эту задачу никто даже не попытался решать¹⁰! Действительно, та-

¹⁰ Позже по этой задаче было проведено отдельное заочное соревнование, где студенты нашей академии заняли второе и третье места.

кие задачи не предназначены для решения за ограниченное время, а требуют спокойного домашнего обдумывания.

Отмечу, что у меня есть свое личное решение¹¹, отличающееся тем, что оно короче (по числу строк таблицы управления) любого присланного участниками, включая то, которое было приведено выше. Более того, мое решение обеспечивает переключивание дисков не на произвольный стержень, а всегда на стержень В! Кроме построения машины Тьюринга, мне удалось построить конечный автомат, который решает эту же задачу. Особенностью автомата является то, что его входная и выходная ленты соединены и представляют собой кольцо. Автомат просматривает входную ленту, модифицируя ее в нужных местах. После чего эта модифицированная лента вновь подается на вход автомата. Для полноты картины осталось запрограммировать “Ханойские башни” марковским алгоритмом! Впрочем, последнее не представляется невозможным.

В заключение упомяну еще один интересный вариант головоломки, названный мною “Перемешанные Ханойские башни” [4], когда в исходном положении диски заранее распределены по стержням произвольным образом. При этом допустим любой порядок следования дисков на стержнях. Сама игра выполняется по стандартным правилам. Конечная цель — собрать в правильном порядке на любом из стержней.

Задание D. “Шифровка”

Человечеству всегда было свойственно скрывать секреты. Вот и мы скрываем от вас важную информацию. С другой стороны, нам бы хотелось, чтобы вы узнали то, что мы скрываем от вас, поэтому ниже мы публикуем зашифрованный текст¹².

Ваша задача — предоставить нам свой файл. Что именно должно быть в этом файле, вы узнаете, прочитав наше сообщение.

ГДОХСМЕЯФ ГДССЙЖЦХФДИЧХТЗ ГДССТМЪДМЖД ГДОХЙЫМЗТПТСЦЙЦ
ГМРЙИДОД (ДЦДЗФ) — СМИТ ЛМ МРАХТЖ ЦЯССТМЪДМЖД ЖТЛЧЖ ММХХТФ.
БДС ЛЧЖ — ЙТССЙЖЦХСМИЙ Ж ЙИТФТЗ ЙОХСМЕЯФ ЙТССЙЖЦХФДИЧХТЗ
ЙЙБХЯЖ ЙТСЕЙЫЧ ЙМСЙИЙЖДЛ. ГХРЙИДС, ТЦЫ ЯЖ ТСАПЙЦАЛГЕТ
ЙЦМЦТЩДЛ АЦМЦЙХТУ БДС ИТФТЗ, АЦДЖАЕТУ Ж ЩДСЙЦХ ДЦДЗФ ТСЫМП.
ЯР ДИЗЙХЖ ЯИДФ РГЦХТЗ.

...

Д ЯЕТЦЫ ГХАЦМИЙЕЧ, ТЦЫ ЯЖ МПЗТРХ АЦДЦМЫТФУ ЦТЦБ ЦХОЙЦ,
РМХТФУ АЦДХМУДС ТЗТИКДО ЛМ ХДЖ, ОДО СТ ЦЙКТР ГХАЦДФЕТИ
ТИ НЙБДС ММРЙИДОД.

ЦЙЖЦТ СЙКПТИ АЦЯЕ СДЖТФМИТОДА РМОДЦ ЙК РТРЦМФТЗПА,
РЯФТЦТО АХМПДЖТААПТУ М ЯР.

ЙБДЖ ЙМСЙЙФ ЦЙИЧЕ ТДСЛМФУ РЯСФЙЖ,

МПХЙ ДБДС ГДЭВЧФМИТОЙИ ДРРДФЗТФУ ЦЙЧЛДФЕТЙФУ ТЗЙ Ж

“НЯСАПИЕДЦМЫ ИМЖ”, М НЯССЛАДОЧ МРДЖ ЦЧФБФДР ЦЙИЧЕ РЯСАПМЖДФУ.

Комментарий к заданию

Эта задача не представлялась нам сложной, хотя не все участники смогли с ней справиться. В сообщении был зашифрован рекламный текст для абитуриентов, поступающих на программистские специальности. В конце шифровки предлагалось ответить на вопрос: “Как можно добраться до рыбинской академии, где обучают в том числе и этим специальностям?” Ответ следовало зашифровать по тому же принципу, что и само сообщение.

Алгоритм шифрования был следующим: символы исходного текста “сдвигались” внутри алфавита на 4 позиции (так называемый “код Цезаря”), и каждое слово “переворачивалось”. Например, слово “РЫБИНСК” последовательно превращалось в слово ОХСМЕЯФ: РЫБИНСК → ФЯЕМСХО → ОХСМЕЯФ.

Задание Е. “Таблица”

Мы предлагаем вам сыграть в следующую игру.

Пусть нам известна таблица 2×2 (например, такая, как указано ниже). Игра заключается в том, что два участника заранее называют по n номеров строк и по n номеров столбцов. Числа, оказавшиеся на пересечении названных строк и столбцов, добавляются к счету участника, играющего за строки, и одновременно минусуются со счета участника, играющего за столбцы. Побеждает игрок, чей счет окажется больше.

		Столбцы	
		0	1
Строки	0	+5	−2
	1	−1	+1

Пример. Ниже приведены ходы игроков. В этом примере первые три хода первый игрок играет за строки, следующие три хода он играет за столбцы. Второй игрок сначала играет за столбцы, затем — за строки. В таблице показано, как меняется счет каждого из игроков. В этом примере побеждает первый игрок.

¹¹ Я сознательно не публикую свое решение, чтобы каждый мог попытаться сам подковать эту блоху. На тот случай, если мне потребуется подтвердить авторские права на текст решения, я послал его по нескольким адресам в апреле 2002 года :).

¹² Этот же зашифрованный текст приводился в отдельном текстовом файле.

Номер хода	Ход первого игрока		Ход второго игрока		Выбранное число	Счет первого игрока	Счет второго игрока
1	0	строки	0	столбцы	+5	+5	—5
2	1		0		—1	+4	—4
3	1		1		+1	+5	—5
4	0	столбцы	0	строки	+5	0	0
5	1		1		+1	—1	+1
6	1		0		—2	+1	—1

Ваша задача — написать программу, которая по таблице, заданной в файле *input.txt* текущего каталога, формирует файл *output.txt*, содержащий ходы.

В первой строке файла *output.txt* должен быть записан ваш регистрационный номер и ваши Ф.И.О. Начиная со следующей строки, последующие 2000 символов — это ваши ходы: 1000 номеров строк и 1000 номеров столбцов (сначала перечисляются все ходы, когда вы играете за строки, затем перечисляются ходы за столбцы). Символы номеров строк и столбцов могут никак не разделяться. Все символы в файле (кроме первой строки), отличные от 0 и 1, игнорируются.

Решение представляется в виде исходного текста программы в файле с именем

<регистрационный_номер>.e.<расширение — язык программирования>

Формат файла *input.txt*

В файле задана одна строка, содержащая четыре целых числа, разделенные пробелами. Эти числа соответствуют элементам таблицы, перечисленным по строкам. Каждое из чисел не превосходит по абсолютной величине числа 99.

Пример файла *input.txt*

```
5 -2 -1 1
```

Подведение итогов

Жюри организует турнир решений по принципу “каждый с каждым” для нескольких тестовых таблиц. За победу в отдельной партии начисляется 2 балла, за ничью — 1 балл, за поражение — 0 баллов.

По окончании турнира участники ранжируются по количеству набранных баллов. При равенстве баллов учитываются набранные очки.

Комментарий к заданию

Эта задача достаточно хорошо известна в теории игр. Здесь мы можем сослаться, например, на фантазию 18 из книги Слойера “Играйте оптимально” [5].

Идея этой стратегии заключается в подборе коэффициента, который задает частоту выбора нулевой и первой строк (или нулевого и первого столбцов) для случая, когда знаки элементов одной диагонали отличаются от знаков элементов второй диагонали.

Пусть заданы вероятности выбора нулевой строки (P_0), первой строки ($1 - P_0$), нулевого столбца (Q_0) и первого столбца ($1 - Q_0$), то ожидаемый выигрыш в пересчете на один ход определяется по формуле:

$$V = A_{00}P_0Q_0 + A_{01}P_0(1 - Q_0) + A_{10}(1 - P_0)Q_0 + A_{11}(1 - P_0)(1 - Q_0).$$

После несложных преобразований можно выразить это значение в виде:

$$V = C_1(P_0 - R)(Q_0 - S) + C_2,$$

где C_1 , C_2 , R , S — некоторые константы, однозначно определяемые элементами таблицы.

Отсюда следует, что каждый из игроков может “нейтрализовать” соперника, если выберет $P_0 = R$ или $Q_0 = S$, так как в этом случае первое слагаемое обращается в ноль, и математическое ожидание выигрыша равно C_2 .

Но ситуация осложняется тем, что в нашем задании игрок не может анализировать стратегию соперника и соответственно реагировать на нее.

Кроме того, необходимо аккуратно разобрать случаи, когда элементы таблицы имеют другие сочетания знаков.

Справедливости ради я должен признаться, что не взялся бы предсказать результаты турнира, даже зная, кто какую выбрал стратегию, так как многое зависит от набора тестов.

В нашем чемпионате 1994 года мы предлагали аналогичную задачу, но тогда участники сдавали в качестве решения тексты “играющих” процедур на языке Паскаль. Процедура на каждом своем вызове через набор параметров получала информацию о предыдущем ходе соперника и могла сохранять до следующего вызова свои данные через специальный массив. Нынешний отказ от решений-процедур был связан с желанием не ограничивать участников каким-либо одним языком программирования.

Литература

- Брудно А.А., Каплан А.И. Московские олимпиады по программированию. М.: Наука. Гл. ред. физ.-мат. лит., 1990.
- Лекции лауреатов премии Тьюринга. /Пер. с англ. под ред. Р.Эшенхёрста. М.: Мир, 1993.
- Пинаев В.Н. Опыт организации и проведения молодежных чемпионатов по информатике и программированию. Информатика № 12/2000. С. 29—31.
- Пинаев В.Н. Ханойские перемешанные башни. Информатика и образование № 3—4/1992. С. 97—98.
- Слойер К. Математические фантазии. М.: Мир, 1993.
- Соколинский Ю.А. Задачи по теме “Машина Тьюринга”. Информатика № 42/1997. С. 1—8.
- Толковый словарь по вычислительным системам. Под ред. В.Иллингворта и др. М.: Машиностроение, 1990.
- Трахтенброт Б.А. Машина Тьюринга. Информатика № 41/1997. С. 1—8.
- Яглом А.М., Яглом И.М. Вероятность и информация. М.: Гос. изд. физ.-мат. лит., 1960.

ПРОЕКТ ФЕДЕРАЛЬНОГО КОМПОНЕНТА ГОСУДАРСТВЕННОГО ОБРАЗОВАТЕЛЬНОГО СТАНДАРТА ОБЩЕГО ОБРАЗОВАНИЯ ПО ИНФОРМАТИКЕ И ИНФОРМАЦИОННЫМ ТЕХНОЛОГИЯМ

Руководители коллектива по разработке проектов стандартов: академики РАО Э.Д. Днепров, В.Д. Шадриков.
Предметная группа по информатике: А.А. Кузнецов, А.Л. Семенов (руководители), С.А. Бешенков, К.К. Колин,
А.А. Муранов, Е.А. Ракитина.

В соответствии с письмом заместителя министра образования РФ Е.Е. Чепурных, замечания и предложения по этому документу должны направляться до 20 сентября в Департамент общего образования Министерства образования Российской Федерации (с пометкой — "Образовательный стандарт").

Почтовый адрес: 101990, Москва, Чистопрудный бульвар, д. 6/19 • Контактный телефон (095) 923-36-19 • E-mail: avb@miem.edu.ru

НАЧАЛЬНАЯ ШКОЛА

ОБЯЗАТЕЛЬНЫЙ МИНИМУМ СОДЕРЖАНИЯ ОСНОВНЫХ ОБРАЗОВАТЕЛЬНЫХ ПРОГРАММ

ИНФОРМАТИКА И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Изучение информатики и информационных технологий в начальной школе направлено на достижение следующих целей:

- **развитие** умений ориентироваться в информационных потоках окружающего мира;
- **овладение** практическими способами работы с информацией: поиск, анализ, преобразование, передача, хранение информации, ее использование в учебной деятельности и повседневной жизни;
- **формирование** начальной компьютерной грамотности и элементов информационной культуры;
- **развитие** умений, позволяющих обмениваться информацией, осуществлять коммуникации с помощью имеющихся технических средств (телефон, магнитофон, компьютер, телевизор и др.).

Реализация указанных целей достигается в результате освоения следующего содержания образования.

Первичное освоение информации

Информация и ее виды (текстовая, графическая, звуковая), информационные объекты (тексты, изображения, звуковые и видеозаписи), ссылки. Источники информации (книги, телевидение, Интернет, устное общение). Работа с информацией: поиск, отбор, преобразование, хранение, представление, использование.

Средства информационных технологий (телефон, телевизор, магнитофон, компьютер и др.). Управление по инструкции кнопками устройствами (телефон с цифровым набором и т.д.).

Организация и поиск информации: оглавление, указатели. Библиотеки, каталоги, справочники, Интернет, записные книжки.

Работа с компьютером

Правила поведения и техника безопасности. Простейшие операции (создание, сохранение, поиск файлов, запуск программы). Работа с простыми широко используемыми программами: работа с текстом, изображениями и др. Выступление (сообщение) с использованием компьютера и проектора. Игры с правилами.

ТРЕБОВАНИЯ К УРОВНЮ ПОДГОТОВКИ ОКАНЧИВАЮЩИХ НАЧАЛЬНУЮ ШКОЛУ

ИНФОРМАТИКА И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Знать и понимать:

- основные информационные объекты, источники информации (в рамках изученного);
- назначение различных средств информации;
- правила техники безопасности при работе с компьютером.

Уметь (владеть способами познавательной деятельности):

- осуществлять поиск информации, пользуясь алфавитом и каталогом, оглавлением, ключевыми словами, ссылками;
- создавать простые информационные объекты и использовать их в личной деятельности и при выполнении учебных задач (схемы, таблицы, расписания, дневник наблюдений и т.п.).

Использовать приобретенные знания и умения в практической деятельности и повседневной жизни:

- пользоваться основными средствами информационных технологий (телефон, телевизор, магнитофон) для передачи и получения информации;
- выполнять простейшие операции по обработке информационных объектов на компьютере: работа с файлами, текстовой и графический редакторы (при наличии условий).

ОСНОВНАЯ ШКОЛА

ОБЯЗАТЕЛЬНЫЙ МИНИМУМ СОДЕРЖАНИЯ ОСНОВНЫХ ОБРАЗОВАТЕЛЬНЫХ ПРОГРАММ

Изучение информатики и информационных технологий в основной школе направлено на достижение следующих целей:

- **приобретение** компьютерной грамотности и начальной компетентности в использовании информационных и коммуникационных технологий, простейших компьютерных моделей при решении учебных и практических задач в школе и вне ее; получение необходимой подготовки для использования методов информатики и средств информационных технологий при изучении учебных дисциплин основной школы и образовательных программ последующего этапа обучения, а также для освоения профессиональной деятельности, востребованной на рынке труда;
- **овладение** навыками работы с различными видами информации с помощью компьютера и других средств информа-

ционных технологий, **умением** применять эти навыки: искать, отбирать, критически оценивать, организовывать, представлять и передавать информацию, планировать и организовывать собственную информационную деятельность и ее результаты;

- **приобретение опыта** выполнения индивидуальных и коллективных проектов, относящихся к различным учебным дисциплинам, в том числе — издания школьных журналов, создания школьных страниц в Интернете, виртуальных краеведческих музеев и т.д. с применением информационных и коммуникационных технологий: использования информации, имеющейся в Интернете и на различных носителях;
- **освоение системы знаний**, относящихся к информационной картине мира, в том числе: базовых понятий, необходимых для формирования конкретных представлений об информационных процессах, системах и технологиях; представлений об общности и закономерностях информационных процессов в различных социальных и технологических системах, о механизмах восприятия и обработки информа-

ции человеком, технологическими и социальными системами, о современной информационной цивилизации;

- **ознакомление** с использованием информационных и коммуникационных технологий как методов познания природы и общества, наблюдения и регистрации природных и социальных явлений, представления их результатов в виде информационных объектов;
- **развитие познавательных интересов, интеллектуальных и творческих способностей** в информационной деятельности;
- **воспитание** необходимых норм поведения и деятельности в соответствии с требованиями информационного общества как закономерного этапа развития цивилизации.

Реализация указанных целей достигается в результате освоения следующего содержания образования¹.

ОБЩЕСТВЕННЫЕ УМЕНИЯ, НАВЫКИ, СПОСОБЫ ДЕЯТЕЛЬНОСТИ

- Оперирование с информационными объектами в различных их представлениях: в мысленном, изобразительном (на бумаге, экране), звуковом представлении.
- Приобретение опыта и навыков собственной информационной деятельности, применения средств информационно-коммуникационных технологий (ИКТ) в изучении школьных предметов и другой образовательной деятельности, организации собственного информационного пространства (набора личных папок), охватывающего (на простейших примерах) наиболее массовые сферы применения информационных и коммуникационных технологий.
- Решение задач построения простейших наглядных информационных моделей объектов и процессов реального мира.

БАЗОВЫЕ ПОНЯТИЯ ИНФОРМАЦИОННЫХ ПРОЦЕССОВ

Базовые понятия информатики: информационный объект, процесс, алгоритм, основные виды алгоритмов и способы их представления, управление, обратная связь. Процесс передачи информации, источник и приемник информации, сигнал, кодирование и декодирование, *искажение информации при передаче*, скорость передачи информации.

Основные компоненты компьютера и их функции: процессор, память, монитор, интерфейсы, клавиатура и мышь; программный принцип работы компьютера. Классификация основных видов общепользовательских и профессиональных информационных ресурсов. Динамичные изменения качественных характеристик средств ИКТ, их учет при принятии решения о выборе средств ИКТ, в том числе для индивидуального пользования.

Создание информационных моделей реальных объектов и процессов, *адекватность модели объекту и целям моделирования*. Язык как способ представления и передачи информации: естественные и формальные языки.

Особенности восприятия, запоминания и обработки информации человеком.

Роль информации в современном обществе, в его экономической, социальной, образовательной сферах. Средства массовой информации, информационная этика и право, информационная безопасность. Личная информация, информационные права личности. Компьютерные игры, их позитивные и негативные стороны.

ПРАКТИКА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Основные устройства ИКТ

Соединение блоков и устройств средств ИКТ, в том числе — компьютера, простейшие операции по управлению ими (включение и выключение, понимание сигналов о готовности и неполадке и т.д.), использование различных носителей информации: загрузка электронных и бумажных носителей, расходных материалов. Безопасность, гигиена, эргономика, ресурсосбережение; технические условия эксплуатации.

¹ Достижение указанных целей в полном объеме возможно, если в рамках образовательного процесса обеспечен достаточный доступ к средствам информационных и коммуникационных технологий (компьютерам, устройствам и инструментам, подсоединяемым к компьютерам, и бескомпьютерным информационным ресурсам).

Образовательные области приоритетного освоения: информатика и информационные технологии.

Компьютер

Оперирование с информационными объектами в наглядно-графическом взаимодействии (интерфейсе). Создание, сохранение, именование объектов и их семейств, выбор имен и организация индивидуальных папок, разархивирование, удаление. Защита от компьютерных вирусов. Выбор компьютера в соответствии с задачами использования и финансовыми возможностями.

Оценка числовых параметров информационных объектов и процессов: объем памяти, необходимый для хранения объектов, скорость передачи и обработки объектов, *стоимость продуктов и услуг*.

Образовательные области приоритетного освоения: информатика и информационные технологии.

Создание и обработка информационных объектов

Тексты. Ввод текста (беглый или вслепую) с помощью компьютерной клавиатуры. Экранные представления текста, шрифты, использование шаблонов. Переменные фрагментов. Структурирование: шрифтовые и цветовые выделения, абзацы и их форматирование, номера страниц, списки, заголовки разного уровня в заданном стиле, ссылки, указатели, оглавления, чтение примечаний, колонтитулы, адресация, обращение, подпись. Распечатка. Использование возможностей проверки орфографии и грамматики, *тезаурусов*. Создание и включение в текст таблиц. Включение в текст изображений.

Образовательные области приоритетного освоения: информатика и информационные технологии, языки.

Динамические (электронные) таблицы, другие математические инструменты ИКТ. Ввод данных в готовую таблицу, изменение данных, переход к графическому представлению. *Ввод математических формул, представление зависимости на графике*.

Образовательные области приоритетного освоения: математика, естествознание.

Базы данных. Поиск выбором из списка и заданием значений полей. Создание записей в базах данных (типа заполнения анкет), в том числе относящихся к информации об учащемся.

Образовательные области приоритетного освоения: информатика и информационные технологии, обществознание, физическая культура, внеклассная работа.

Рисунки и фотографии. Ввод с помощью графической панели и сканера. Геометрические, стиливые преобразования.

Образовательные области приоритетного освоения: изобразительное искусство, краеведение, внеклассная деятельность.

Чертежи. Выделение, объединение, перемещение и геометрические преобразования фрагментов и компонентов. *Трехмерное изображение на экране компьютера и инструменты работы с ним. КATALOGI объектов*. Простейшие чертежи. Специальные виды чертежей: диаграммы, карты и др. Построение чертежа данного предмета. Конструирование (сборка) предмета по чертежу.

Образовательные области приоритетного освоения: технология (в частности, черчение).

Звук и видеоизображение, в том числе музыки и анимации. Инструменты работы (редактирования, нелинейного монтажа) в экранном наглядном представлении.

Образовательные области приоритетного освоения: языки, музыка, краеведение, проектная деятельность в различных областях.

Фиксация информации об объектах и процессах окружающего мира

Запись объектов и процессов:

- природных, культурно-исторических, школьной жизни, индивидуальной и семейной истории и др., в том числе запись изображений (с использованием цифровых фотоаппаратов, микроскопов, различных видов сканеров и др.);
- текстов, полученных распознаванием текстового изображения и устной речи;
- видео- и аудиозаписей (в частности, с использованием цифровых записывающих устройств);
- музыки (в частности, с использованием музыкальной клавиатуры);
- таблиц результатов измерений (в частности, с использованием присоединяемых к компьютеру датчиков);
- результатов наблюдений, опросов и т.д.

Образовательные области приоритетного освоения: обществознание, естествознание.

Поиск и использование информации

Компьютерные энциклопедии и справочники; информация в базах данных, Интернете, некомпьютерных источниках информации. Формулирование запросов. Каталоги. Поисковые машины. Критическая оценка и организация информации. Формирование индивидуального информационного пространства с применением файловой системы, ссылок на файлы и источники в Интернете. *Правила ссылок и цитирования источников информации.*

Образовательные области приоритетного освоения: обществознание.

Организация, представление и передача информации

Представление информации в виде печатной публикации, в Интернете, в выступлении учащегося. Организация собственного информационного пространства, включающего работы учащегося, различные их версии, ссылки на соответствующие задания, проекты, работы других учащихся, собранные массивы информации, используемые источники. Групповое обсуждение, фиксация его хода и результатов в текстовой форме и в сочетании с видеозаписью. Телефон, пейджер, электронная почта: правила переписки, приложения к письмам, отправка, получение. Сохранение для индивидуального использования объектов из Интернета и ссылок на них. Телеконференции, видеоконференции. Использование цифровой телекоммуникации в коллективной учебной деятельности.

Образовательные области приоритетного освоения: информатика и информационные технологии, языки, обществознание.

ТРЕБОВАНИЯ К УРОВНЮ ПОДГОТОВКИ ВЫПУСКНИКОВ ОСНОВНОЙ ШКОЛЫ

В результате изучения информатики и информационных технологий ученик должен

Знать и понимать основные факты о:

- главных устройствах средств информационных и коммуникационных технологий и их функциях, в частности: о центральном процессоре, устройствах памяти, периферийных устройствах, интерфейсах, программном принципе работы компьютера; используемых в деятельности учащегося носителях информации;
- источнике и приемнике информации, сигнале, кодировании и декодировании, единицах измерения количества и скорости передачи информации, скорости обработки информации;
- восприятии, запоминании и обработке информации человеком;
- информационной деятельности человека, языке как способе представления и передачи информации; естественных и формальных языках;
- простейших примерах информационных моделей реальных объектов;
- алгоритмах, основных алгоритмических конструкциях: следовании, ветвлении, цикле; вспомогательных алгоритмах;
- роли информации и информационных технологий в современном обществе, в его экономической, социальной, образовательной сфере;
- средствах массовой информации, информационной этике и праве, информационной безопасности; компьютерных играх, их позитивных и негативных аспектах; динамичном изменении качественных характеристик средств ИКТ, их учете при принятии решения о выборе средств ИКТ, в том числе для индивидуального пользования.

Уметь (владеть способами познавательной деятельности):

- оперировать с информационными объектами в наглядно-графическом взаимодействии (интерфейсе), используя имеющиеся знания об устройстве компьютера, в том числе
 - создавать, именовать, сохранять объекты, создавать удобные для использования индивидуальные каталоги, разархивировать информацию;
 - пользоваться меню и окнами;
 - пользоваться экранной справочной системой;
 - соблюдать права собственности на информацию;
 - предпринимать меры антивирусной безопасности;
- оценивать числовые параметры информационных объектов и процессов, в частности, при рассмотрении выполнимости действия
 - объем памяти, необходимый для хранения информации;
 - скорость передачи информации;
- создавать информационные объекты с уровнем сложности, отвечающим потребностям образовательного процесса, в том числе

- создавать тексты (школьные сочинения и рефераты по различным предметам и т.д.) посредством квалифицированного клавиатурного письма (беглого или вслепую), используя базовые средства текстовых редакторов для выражения содержания и восприятия произносимой или записанной устной речи;
- структурировать текст (нумерация страниц, использование списков, ссылок, оглавлений, колонтитулов, указателей, адресов, обращений, подписей и т.п.); проводить проверку орфографии и грамматики; использовать в тексте таблицы, изображения;
- фиксировать информацию об объектах и процессах окружающего мира (природных, культурно-исторических, школьной жизни, индивидуальной и семейной истории и др.) в форме: изображений (в том числе с использованием цифровых фотоаппаратов, микроскопов, различных видов сканеров); текстов, полученных распознаванием текстового изображения и расшифровкой устной речи; видео- и аудиозаписей (в частности, с использованием цифровых записывающих устройств, музыкальных клавиатур); таблиц (в том числе с использованием присоединяемых к компьютеру датчиков); записывать простейшие музыкальные фрагменты;
- создавать и использовать различные формы представления информации: математические формулы, графики, диаграммы, таблицы (в частности, в практических задачах домашней экономики); переходить от одного представления данных к другому;
- создавать рисунки, чертежи (в том числе с использованием графического планшета), графические представления реального объекта, в частности, в процессе проектирования и дизайна (элементы машин, архитектурные объекты, одежда, электрические и электронные схемы, планы местности, географические и исторические карты) с использованием простейших графических редакторов, учебных систем автоматизированного проектирования; осуществлять конструирование по чертежу, простейшую обработку цифровых изображений;
- создавать записи в базе данных (типа заполнения анкет), в частности, относящиеся к информации об учащемся, объединяемые в индивидуальные записные книжки; создавать комплексные объекты, включая связанные между собой информационные объекты всех видов (мультимедиа и гипермедиа-сочинения), экранные объекты со ссылками (типа страниц Интернета), в том числе индивидуальные страницы учащегося, класса, школы, страницы проектов.

Использовать приобретенные знания и умения в практической деятельности и повседневной жизни:

- выполнять простейшие операции, связанные с использованием современных средств ИКТ, соблюдая при этом требования техники безопасности, гигиены, эргономики, ресурсосбережения и условий эксплуатации по отношению к: соединению различных устройств; управлению ими (включению и выключению; пониманию простейших сигналов, в том числе о неполадке); загрузке электронных и бумажных носителей (помещению бумаги в лоток принтера, иллюстрации в сканер, загрузке дискеты, компакт-диска, видеокассеты и т.д.); чередованию различных форм деятельности и гигиенических упражнений;
- передавать информацию по телекоммуникационным каналам, соблюдая соответствующие нормы и этикет, пользуясь: телефоном, электронной почтой (применяя вложения, ответные письма, следуя этикету электронной почты);
- искать информацию с применением конструкций и правил поиска (в том числе — построения запросов): в базах данных, в Интернете, в некомпьютерных источниках информации: справочниках и словарях, каталогах, библиотеках (периодика, литература и др.) при выполнении заданий и проектов по различным дисциплинам;
- оценивать и организовывать информацию, в том числе получаемую из средств массовой информации, свидетельства очевидцев, интервью: использовать ссылки и цитирование источников информации; анализировать и сопоставлять различные источники;
- представлять информацию в виде мультимедиа-объекта с системой ссылок (например, для размещения в Интернете); готовить и проводить выступления, включающие сформированную заранее систему изображений на проекционном экране и звукового сопровождения; использовать в устном выступлении микрофон, проекционный экран; использовать экран в групповом обсуждении, фиксировать его ход и результаты.



роботландский университет

начало информатика конструирование программирование интернет английский

учителя

школьники

учебники

программы

лаборатории

теория

практика

турниры

олимпиады

общение

конференции

творчество

помощь

РОБОТЛАНДСКИЙ УНИВЕРСИТЕТ ОТКРЫВАЕТ НАБОР СТУДЕНТОВ НА 2002/2003 УЧЕБНЫЙ ГОД

В университет принимаются "индивидуальные студенты" и "коллективные студенты".

"Коллективный студент" — это группа детей, работающая под руководством одного или нескольких наставников (наставник, как правило, — школьный учитель).

"Индивидуальный студент" — это учитель, желающий пройти обучение индивидуально, без группы детей.

В конце годичного курса при успешном обучении школьный учитель получает удостоверение от негосударственного образовательного учреждения "Роботландия+" и удостоверение государственного образца от Российской академии повышения квалификации и переподготовки работников образования.

Занятия в университете платные (с ценами можно ознакомиться на сайте www.botik.ru/~robot).

Они начинаются 15 октября текущего года и продолжаются в течение двух семестров до мая следующего года. Для подписчиков "Информатики" предусмотрена 10%-ная скидка за обучение.

Для получения более подробной информации можно написать по адресу

kurs@robotland.botik.ru

(администрация университета) или заглянуть на сайт

www.botik.ru/~robot

(здесь удобно отправить заявку, заполнив электронную форму).

КРАТКОЕ ОПИСАНИЕ КУРСОВ

10. Введение в информатику

Курс только для "индивидуальных студентов" — учителей, преподающих или собирающихся преподавать информатику в младших классах на базе программно-методических систем "Роботландия" и "Хиты Роботландии".

11. Зимние вечера

Курс для младших школьников (3—4-е классы). Знакомство с алгоритмами и исполнителями на базе программно-методической системы "Роботландия"; электронная почта; детские конкурсы.

12. Азы информатики-1

Первая часть курса для младших школьников (3—5-е классы). Обучение проводится на основе двух интерактивных гипертекстовых книг-лабораторий "Знакомство с компьютером" и "В мире информации".

14. Азы информатики-2

Вторая часть курса для младших школьников (3—5-е классы). Обучение проводится на основе интерактивных гипертекстовых книг-лабораторий "Пишем на компьютере", "Рисуем на компьютере", "Выходим в Интернет".

21. Тропинка конструктора

Курс для детей 4—7-х классов, изучивших основы информатики. Создание компьютерных проектов, имеющих практическое применение на школьных уроках, на базе роботландских программ-конструкторов.

31. Азы программирования

Для детей 6—8-х классов, изучивших основы информатики (исполнитель, алгоритм, программа, первичные инструментальные навыки работы с компьютером). Этот курс для детей, которые нашли себя в программировании и желают повысить уровень своего мастерства, составляя головоломные алгоритмы для исполнителей Кукарача и Корректора.

32. Буки программирования

Курс наиболее эффективен для старшеклассников. Классические задачи и методы их решения. Составление и анализ алгоритмов не "по наитию", а "по науке". Олимпиадные задачи по программированию.

41. Введение в Интернет

Для детей 5—8-х классов. Введение в сетевые коммуникации на базе электронной почты. Коммуникативные проекты и конкурсы, работа с сетевыми исполнителями.

42. Web-конструирование

Для старшеклассников. Создание HTML-документов, в том числе с использованием специально созданной для курса библиотеки на JavaScript и CSS. Основы web-дизайна и проектирования. Создание гипертекстовых страниц и приложений, пригодных для использования на школьных уроках. Особое внимание уделяется вопросу разработки школьного сайта.

43. Web-программирование

Для старшеклассников, имеющих опыт программирования и интерес к этой деятельности. Изучение JavaScript, CSS. Построение динамических гипертекстовых приложений.

50. Английский с компьютером

Методические вопросы преподавания языка с использованием компьютера рассматриваются автором апробированного многолетней учительской практикой курса английского языка — Magic Land. Проектирование и реализация компьютерных упражнений к урокам.

ОБРАЗЕЦ ЗАЯВКИ

ЗАЯВКА НА ОБУЧЕНИЕ В РОБОТЛАНДСКОМ УНИВЕРСИТЕТЕ

Предполагаемый курс

(отделение):

Форма обучения:

Руководитель (обучаемый):

Профессия:

Стаж работы в школе:

Директор школы:

Электронный адрес:

Название учреждения,

почтовый адрес

Почтовый адрес для посылок
Предполагаемое число детей
в группе и их возраст
(для "коллективного студента"):

Первый раз в Роботландском университете

Руководитель (обучаемый) —

подписчик газеты "Информатика"

Откуда получена информация об университете:

Заявки принимаются по адресу:

43. Web-программирование

"Коллективный студент"

Иванов Петр Николаевич

Учитель информатики

средней шк.

3 года

Петров Семен Сергеевич

ivanov@sch62.nsk.ru

средняя школа № 62

630090, г. Новосибирск,

ул. Жемчужная, 6,

тот же

20 школьников,

10—11-е классы

да

да

газета "Информатика"

kurs@robotland.botik.ru

до 6 октября.

1996/1997

седьмой
учебный
год
в наших
сетях

2002/2003



Внимание! На страницах 9 и 15 опубликованы, важные (очень!) объявления

Страницы повышения квалификации..... 3-9

Е.А. Еремин. ОСНОВЫ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ. ЛЕКЦИЯ № 1

На "Страницах повышения квалификации" начинается новый учебный год. Сегодня мы представляем первую лекцию курса, посвященного базовым принципам устройства и функционирования ЭВМ. Эта тема курса информатики имеет свою особую судьбу. Свой звездный час она пережила в первые годы появления информатики в школе. Тогда даже сам предмет назывался "Основы информатики и вычислительной техники". Позже начались "гонения": и название курса изменилось, и из учебников "выгнали", и в учебных программах места не находилось. И все же, преимущественно явочным порядком, исключительно на энтузиазме учителей, этот раздел курса уцелел. Но в последнее время все отчетливее становится видна другая проблема: старое содержание темы "Основы вычислительной техники" устарело, а систематизацией нового никто специально не занимался. Поэтому мы и заказали соответствующий курс Евгению Александровичу Еремину, одному из энтузиастов изучения основ вычислительной техники. Надеемся, курс будет не просто интересен (что не вызывает сомнений), но и полезен всем творчески работающим учителям.

Круглый стол..... 10-15

З.Фрайман. КОМПЬЮТЕРНОЕ ОБРАЗОВАНИЕ В ИЗРАИЛЕ. СОВРЕМЕННОЕ СОСТОЯНИЕ (попытка очерка)

Почему именно в Израиле? Честно говоря, причина простая — наш коллега, Зев Фрайман, который преподает информатику (в Израиле предмет называется "компьютерные науки") в одной из израильских школ, по собственной инициативе прислал нам статью. Мы прочитали ее с огромным интересом и решили, что, несмотря на то, что цикла статей о преподавании информатики в разных странах нам в ближайшее время скорее всего не осилить (хотя мечта такая есть), даже одна эта статья может быть интересна сама по себе. Кстати, это не единственный материал, который прислал Зев Фрайман. В следующих номерах мы опубликуем задания выпускного экзамена по информатике в Израиле. На редакторов "Информатики" эти задания произвели очень сильное впечатление. Нам есть о чем задуматься.

На стенд в кабинете информатики 16-17

СКАНЕРЫ

Один из редакторов "Информатики" (было это уже давно) имел в своей педагогической практике вопиющий случай: его любознательные ученики, вооружившись обычной отверткой, наглядно изучили устройство жесткого диска. Этот заслуженный винчестер до сих пор используется в качестве наглядного пособия, но тогда скандал был страшный. В течение прошлого года мы систематически уберегали коллег от повторения подобного, публикуя в рубрике "На стенд" подробные рассказы об устройстве различных узлов ЭВМ. Сегодня мы спасаем наши сканеры...

Олимпиады 18-27

В.Н. Пинаев. КАЖДЫЙ ВЫБИРАЕТ СВОЙ ПИК!

Наши постоянные читатели, конечно, помнят материалы, посвященные рыбинским чемпионатам по программированию. Задачи этих чемпионатов не просто хорошие, они особенные. Они не являются олимпиадными в традиционном понимании, хотя и используются на соревнованиях. Мы считаем, что нашим читателям будут интересны не только сами задачи, но и методика проведения чемпионатов (подробнее о ней можно прочитать в предыдущих статьях, ссылки на которые имеются в этом номере).

Официальные документы 28-30

ПРОЕКТ ФЕДЕРАЛЬНОГО КОМПОНЕНТА ГОСУДАРСТВЕННОГО ОБРАЗОВАТЕЛЬНОГО СТАНДАРТА ПО ИНФОРМАТИКЕ И ИНФОРМАЦИОННЫМ ТЕХНОЛОГИЯМ

Об этом документе каждый, конечно, составит свое мнение сам. Скажем лишь, что читать его немного обидно. И к содержанию документа это чувство отношения не имеет, речь идет о более прозаичных вещах: ну почему в этом документе, который должен скоро стать для нас практически законом, встречается так много нелепых, иногда просто несогласованных фраз, непонятных слов, требующих специального толкования, наконец, просто ошибок? Ну почему?!

Информация 31

Наши хорошие друзья из Роботландского университета объявляют о наборе на новый учебный год. Мы рады в очередной раз порекомендовать вам лучшие дистанционные курсы для учителей информатики (имеется также курс для учителей английского языка) в России. И не забудьте, что наши подписчики имеют право на скидку 10% от стоимости обучения.

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 6500 экз. Срок подписания в печать по графику 28.08.2001. Номер подписания 28.08.2001. Заказ № 15400 Цена свободная
	ИНДЕКС ПОДПИСКИ для индивидуальных подписчиков 32291 комплекта изданий 32744		Internet: inf@1september.ru WWW: http://www.1september.ru
	Тел.: (095)249-31-38, 249-33-86. Факс (095)249-31-84		

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

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