

НАУЧНО-МЕТОДИЧЕСКИЕ ОСНОВЫ ШКОЛЬНОГО КУРСА ИНФОРМАТИКИ

В конце 1984 г. в связи с развитием положений школьной реформы, а также выполнением государственной задачи – достижением всеобщей компьютерной грамотности учащейся молодежи – Министерство просвещения СССР приняло решение о введении в советской школе с 1985/86 учебного года нового предмета "Основы информатики и вычислительной техники".

К этому времени в Сибирском отделении Академии наук СССР уже восьмой год велись исследования способов применения ЭВМ в школьном учебном процессе. Начавшись как инициативная поисковая работа, это исследование впоследствии стало заданием ГКНТ СССР, темой международного сотрудничества и предметом совместной работы с преподавателями и специалистами Академии педагогических наук СССР. Существенными чертами нашего исследования были поиск решений, приемлемых для массовой школы, и ориентация на микропроцессорную вычислительную технику, на персональные компьютеры. Этот научный задел позволил конструктивно подойти к задаче, выдвинутой Министерством просвещения, и без больших промедлений сформулировать определенную концепцию школьного курса информатики, которая и является темой данного научного сообщения.

Хотелось бы отметить большую важность того обстоятельства, что этот вопрос обсуждается в столь авторитетном научном собрании. Трудно назвать более воодушевляющую и ответственную задачу для ученого, нежели критический пересмотр всей суммы знаний, накопленных в его профессии, выделение в ней зерен "вечной истины" и их внедрение в почву общей культуры в виде школьного предмета и учебника по этому предмету.

Представьте, что вам предстоит написать текст, который меньше чем через год должны прочесть миллионы молодых людей, который должен стать опорой для нескольких десятков тысяч учителей. Вы ощущаете себя незванным гостем в стане великих предшественников, мучаетесь над каждой строкой в поисках точного и ясного выражения, освобожденного от колебаний и заблуждений. Поистине Лютерова работа – опыт, одновременно и вознаграждающий, и пугающий.

Вступление информатики в массовую школу – это событие, задевающее каждого специалиста по вычислительному делу. Даже если он сам не преподает, он столкнется с этой проблемой в качестве шефа базового предприятия, родителя своих детей, руководителя молодого пополнения в НИИ или на заводе.

Как известно, недавно вышедшее постановление ЦК КПСС и Совета Министров СССР предусматривает повсеместное начало преподавания основ информатики и вычислительной техники в новом учебном году. Хотя оно задает беспрецедентные темпы и объемы поступления вычислительной техники в школы, тем не менее пройдет несколько лет, прежде чем районный вычислительный кабинет станет доступен для каждой школы хотя бы в режиме посещения. Поэтому начальный вариант программы курса должен учитывать относительную недоступность вычислительных средств и опираться на традиционные формы проведения урока.

Некоторые рассматривают эту ситуацию трагически, считая, что она скомпрометирует саму идею введения информатики в школьное образование.

Хотелось бы, однако, подчеркнуть ряд положительных моментов в такого рода начале, которые, по моему мнению, облегчат становление курса информатики в школе.

Продуктивное решение задач на ЭВМ, вообще говоря, требует двух типов умений: условно назовем их "теоретическим" и "операционным". Теоретическое умение означает способность понять и точно сформулировать условие задачи, придумать или выбрать алгоритм и выразить его в виде программы на подходящем языке программирования. Операционное умение означает способность правильно организовать и выполнить все манипуляции, связанные с решением задачи на ЭВМ. Частично операционное умение требуется на стадии алгоритмизации и программирования, но главным образом оно проявляется во всей конкретике действий, связанных с физическим присутствием машины.

В практике повседневного, в особенности профессионального, применения ЭВМ оба эти умения сплетаются в непрерывную деятельность, однако никто не доказал, что они не могут приобретаться относительно раздельно. Мало того, есть немало наблюдений, показывающих, что неограниченный доступ к ЭВМ создает ситуацию, когда "руки работают впереди головы", а это может пагубно сказываться на эффективности применения ЭВМ.

Второе обстоятельство касается учителей. Есть основания думать, что для большей части преподавателей курса информатики предшествующее освоение ЭВМ на логическом уровне в виде абстрактного устройства будет проходить легче, нежели внезапный подход к ней как к "железному устройству" со стороны дисплея и клавиатуры.

Заклучая это общее обоснование, можно сказать, что "безмашинное" срочное начало курса информатики, чреватое очевидными издержками, тем не менее содействует вычленению теоретической части курса, способствует выработке взгляда на алгоритмизацию и программирование как на новый вид математической практики, дает шанс учителю растянуть во времени процесс освоения вычислительной техники и привыкания к ней, позволяет использовать традиционные формы ведения урока, создавая в то же время в школе обстановку нетерпеливого ожидания компьютера.

Именно эти положения легли в основу выработки программы и пробного учебного пособия по курсу "Основы информатики и вычислительной техники". Укажем сначала основные разделы курса и их объем в количестве уроков.

9-й класс (34 часа):

введение в информатику и первые сведения об ЭВМ (2 часа);
алгоритм и алгоритмический язык (6 часов);
алгоритмы работы с величинами (10 часов);
построение алгоритмов (4 часа);
решение задач с помощью алгоритмов – из курсов математики, физики и химии (12 часов).

10-й класс (36 часов):

знакомство с ЭВМ (12 часов);
программирование для ЭВМ (16 часов);
ЭВМ в обществе (2 часа);
экскурсии в вычислительный центр (6 часов).

К общему количеству – 70 часам – добавляется минимум 34 часа работы в вычислительном кабинете, оснащенном персональными ЭВМ или терминалами, обеспечивающими индивидуальную работу (там, где это доступно).

Наиболее нетривиальной и методологически немотивированной частью является программа 9-го класса, так что основная часть изложения будет посвящена ей.

ВВЕДЕНИЕ В ИНФОРМАТИКУ

Мы трактуем информатику как науку о законах и методах накопления, передачи и обработки информации. Укажем, что благодаря использованию несколько расплывчатого термина "накопление" трактовка информатики здесь скорее расширительная, нежели ограничительная. Это, пожалуй, не столько выношенное убеждение, сколько дань времени, отражающая период становления новой науки и естественную для этого периода экспансию.

Несколько отвлекаясь, заметим, что определение разграничительных линий между информатикой и науками, изучающими процессы познания, а также между информатикой и естественными науками должно стать предметом глубоких исследований, причем естественно допустить, что эти границы будут постепенно меняться по мере развития каждой из этих групп наук.

Затем мы формируем представление об информации как о любом знании, которое получает человек, читая текст, разглядывая изображение, воспринимая внешний мир. При этом мы исходим из того, что сама устойчивая и наблюдаемая структура внешнего мира формирует своего рода знаковую и сигнальную системы, которые обычно связываются с понятием информации. Такой подход подразумевает содержательное и относителное восприятие информации, когда информативность связывается с объемом перестройки тезауруса наблюдателя, а теоретико-информационное количество информации становится минимально-предельной характеристикой, достаточной для различения информационного прообраза среди других возможных прообразов, образующих информационный ресурс наблюдателя.

Здесь уместно сказать еще об одной общей методологической основе курса. В преподавании его начала мы исходим из того, что учащийся отождествляет себя с источником и накопителем информации, с исполнителем алгоритмов. Такой антропоцентристский подход, как нам кажется, активизирует учащегося, позволяет взывать к его личному опыту, побуждает его, знакомясь с правилами информатики, лучше осознавать свои действия.

Содержательно усваивая эти правила, признавая их сначала как правила своеобразной игры, он постепенно осознает их устойчивость, объективность, формальность и тем самым лучше постигает основную сверхзадачу курса – понять, как человек может значительную часть своей умственной деятельности передать машине.

Раскрывая содержание информатики, мы апеллируем к повседневному опыту ребенка, связанному с чтением, разглядыванием картинок, разговорами по телефону, просмотрами телевизионных передач, писанием писем, постоянным решением задач (переход от данного знания – условия задачи к новому знанию – ее решению) и, наконец, любой целеустремленной деятельностью, совершаемой по правилам (переход от слова к делу).

Далее, мы показываем на многочисленных примерах, что предмет информатики стар, как сама жизнь. Появление информатики как науки связывается с появлением ЭВМ как устройства для автоматической обработки информации. Необходимость передать машине знания человека о способах закономерного решения задач побуждает понять природу этих способов, открыть правила и законы обработки информации, дать в руки человеку практические методы использования ЭВМ.

Затем мы раскрываем два главных секрета могущества ЭВМ: ее скорость и способность к автоматическому выполнению программы, под которой понимается специальное выражение правила решения задачи, составляемого в виде алгоритма. Так появляются три главных понятия информатики: информация, ЭВМ, алгоритм. При этом ЭВМ, в соответствии с тезисом А.Чёрча, актуализирует и выражает единство законов обработки информации в живых и машинных системах. Понимание и использование этого единства – крупнейшее научное достижение после установления единства материальной (вещественной) и энергетической картины мира.

Мы показываем на примерах, как богат мир алгоритмов и программ. Большая часть нашей собственной деятельности выполняется по алгоритмам, встроенным в нас природой, данным нам обучением и тренировкой, выработанным собственным опытом. Мы говорим, что информатика позволит найти черты общего во всем этом разнообразии.

В заключение мы отмечаем, что алгоритмизация, программирование и решение задач с помощью ЭВМ являются теми главными видами человеческой деятельности, которые требуют знания информатики и ее методов. Основу этой деятельности составляет передача человеком машине своего знания в виде точной информации, строгих правил, правильных алгоритмов и эффективных программ. Вот почему алгоритмическое мышление, способность построить план действий и предвидеть их результат, знакомство с ЭВМ и понимание ее роли в обществе становятся частью общего образования, элементом человеческой культуры и школьным предметом.

АЛГОРИТМ И АЛГОРИТМИЧЕСКИЙ ЯЗЫК

Мы даем следующее определение алгоритма. Алгоритм – это данное исполнителю точное и понятное предписание выполнить последовательность действий, направленных на достижение указанной цели или решение поставленной задачи.

Конечно, эта весьма тщательно построенная формулировка является, так сказать, квазиопределением. Оно не сводит новую концепцию к комбинации известных вещей, скорее просто вводит некоторую осмысленную комбинацию важных слов, которые в дальнейшем будут интенсивно поясняться. Эти слова: предписание, исполнитель, последовательность действий, точность, понятность, результативность и сам алгоритм.

Затем мы поясняем с разной степенью детальности разные свойства алгоритмов.

Дискретность означает, что запись алгоритма состоит из отдельных команд, каждая из которых предписывает исполнителю выполнить одно законченное действие. Команды выполняются поочередно, формируя последовательность действий, ведущих к решению задачи.

Понятность означает, что каждая команда алгоритма может быть выполнена исполнителем. Разъяснение этого термина одновременно и тонкое и важное место изложения. Мы пытаемся избежать сомнительных дискуссий по поводу слова "понимать", допуская его метафорический характер. Тем не менее мы стремимся показать, что степень необходимого понимания алгоритма его автором может и фактически является совсем другой, нежели степень необходимого понимания алгоритма его исполнителем. Явно или неявно, но мы подменяем во втором случае концепцию "понимания" алгоритма концепцией способности выполнить его и тем самым решить задачу. Для этой цели, однако, исполнителю достаточно строго следовать каждой отдельной команде алгоритма.

Мы вводим алгоритмический язык как средство повседневной человеческой практики, предшествующее использованию языков программирования, трактуемых нами как специальные алгоритмические языки, при использовании которых исполнителем является ЭВМ. Это простое предположение имеет на самом деле неисчерпаемые педагогические и методологические последствия, большая часть которых полезна и продуктивна.

АЛГОРИТМ РАБОТЫ С ВЕЛИЧИНАМИ

Мы рассматриваем постоянные и переменные величины как пары, образованные именем и значением. Когда мы напоминаем, что алгоритмы работы с величинами хорошо знакомы нам благодаря решению задач по математике, физике или химии или что они лучше всего отражают концепцию памяти ЭВМ, мы тем не менее даем понять, что интуитивно они составляют только часть возможных алгоритмов. Неформально говоря, наиболее очевидным дополнением к алгоритмам работы с величинами являются алгоритмы работы во внешней обстановке, где исполнитель прокладывает траектории своей деятельности (алгоритмы движения в лабиринте, алгоритмы построения геометрических фигур и т.п.). Естественно, что на определенном уровне общности использование глобальных структурных величин позволяет выразить алгоритмы работы в обстановке в виде работы с величинами, однако в начальной стадии объединение этих понятий не кажется продуктивным.

Вкратце, при введении алгоритмов над величинами алгоритмический язык конкретизируется такими понятиями, как аргументы и результаты алгоритма, выражения, присваивание, линейные и прямоугольные массивы (называемые таблицами), отношения в качестве условий в ветвлениях и циклах. Мы не требуем линейной записи выражений для того, чтобы сохранить приобретенную в курсе математики свободу манипулирования алгоритмическими выражениями. Неявно мы вводим типизированный язык (по крайней мере по отношению к аргументам, результатам и другим осмысленным величинам), возможно, допуская неуказанный "универсальный" тип для рабочих величин.

Приведем пример записи на алгоритмическом языке:

алг КВАДРАТНОЕ УРАВНЕНИЕ (**вещ** a, b, c, x1, x2, **лит** R);

арг a, b, c; **рез** x1, x2, R;

нач **вещ** Δ;

$$\Delta := \sqrt{b^2 - 4ac}$$

если $\Delta \geq 0$

то R := "решение";

$$\Delta := \sqrt{\Delta};$$

$$x1 := (\Delta - b)/2a$$

$$x2 := -(\Delta + b)/2a$$

иначе R := "нет решения"

все

кон

ПОСТРОЕНИЕ АЛГОРИТМОВ

Мы подчеркиваем с самого начала, что алгоритмизация – творческий процесс, поддерживаемый, однако, эффективной и надежной эвристикой. Здесь мы опираемся на три главных и мощных метода, поддерживаемых непосредственным опытом: расположение команд в серии, пошаговое уточнение, рекурсивное описание.

В целом мы стараемся убедить учащихся, что, не пренебрегая догадками, нужно все же стремиться извлекать алгоритм из условия задачи или из свойств требуемого решения.

Мы пытаемся показать, что правильное расположение команд в серии всегда логически мотивировано и отражает либо направление движения информации, либо причинно-следственные связи. Мы также показываем, что ветвления и повторения обычно явно подсказываются условием задачи.

Метод пошагового уточнения вводится путем следующей эвристики. Имея условие задачи, нужно придумать такие команды исполнителю, чтобы он мог прийти к решению с помощью простого и прямо записываемого алгоритма. Построив такой алгоритм первого яруса, нужно проанализировать его команды, уточнить, к каким действиям они должны вести. Если в алгоритме встречаются команды, непонятные исполнителю, следует поставить осуществление этого действия как отдельную задачу и искать алгоритм для нее, который потом будет вызываться как вспомогательный алгоритм, используемый в алгоритме первого яруса. Нужно поступать так до тех пор, пока все вспомогательные алгоритмы не станут непосредственно понятны исполнителю. Так мы

приходим к понятию процедур, модулей и подзадач и органически используем все полезные стороны этих конструкций.

Рекурсия трактуется как частный случай пошагового уточнения, когда мы редуцируем исходную задачу к такой же, но в чем-то более простой. При такой редукции шаг упрощения в сочетании с решением простейшей задачи (обычно тривиальным) дает достоверный алгоритм решения задачи в общем случае.

Мы не стесняемся вводить понятие рекурсии на таких простейших примерах, как следующий:

```
алг ЕШЬ КАШУ
  нач если тарелка пуста
    то иди из-за стола
  иначе съешь ложку каши;
    ЕШЬ КАШУ
  все
кон
```

Кстати говоря, как показывает опыт применения языка ЛОГО, подтверждаемый и нашими наблюдениями, рекурсивная организация повторяющихся действий иногда лучше осваивается детьми, нежели явные циклы.

РЕШЕНИЕ ЗАДАЧ С ПОМОЩЬЮ АЛГОРИТМОВ

Эта часть курса содержит интенсивный практикум по алгоритмизации и решению задач. Проводя его, учитель должен придерживаться следующей схемы решения задачи:

- содержательная формулировка задачи;
- выбор или построение математической модели;
- формальная постановка задачи;
- построение алгоритма;
- программирование и выполнение алгоритма;
- анализ результатов.

Естественно, что эта развернутая схема не может педантично использоваться при решении любой задачи, однако правильный подбор примеров должен демонстрировать необходимость формализации постановки задачи и точность в этапах построения алгоритма, обеспечивающую соблюдение всех его нужных свойств по отношению к выбранному исполнителю.

В том случае, когда ЭВМ недоступна, учитель может полагаться на микрокалькуляторы или даже на "ручное исполнение" в практической работе.

Возвращаясь еще раз к объяснению исходных предпосылок, хотелось бы подчеркнуть, что даже при полном отсутствии ЭВМ курс информатики для 9-х классов может быть вполне благополучно проведен в стиле традиционного курса алгебры с той лишь разницей, что материал для выбора задач становится существенно богаче.

В заключение скажу несколько слов о программе 10-го класса. Эта часть курса по своему характеру гораздо ближе к уже достаточно традиционным факультативам по основам ЭВМ и программирования, которые неоднократно преподавались в ряде физико-математических школ.

Есть одно важное соображение, касающееся связи информатики-10 с информатикой-9. Безотносительно к тому, какой язык программирования будет выбран для работы с учениками, программирование на нем должно начинаться со спецификации алгоритма на алгоритмическом языке из информатики-9 (может быть, с необходимыми расширениями). Такая прививка структурного программирования в сочетании его с методом пошагового уточнения позволит ученикам "выжить" даже при получении больших доз Бейсика или аналогичного языка.

В настоящем сообщении была изложена научно-методологическая основа начального курса школьной информатики в условиях ограниченного доступа к машине и преподаваемого учителями, для которых этот предмет в существенной степени будет новым.

Думается, что положенные в основу учебного пособия принципы должны быть конкретизированы в методическом пособии и внедрены в практику преподавания довольно регулярными средствами нормативного характера. Это позволит более четко контролировать ход введения этого курса в массовой школе.

В то же время во все большем количестве школ будут создаваться современные вычислительные кабинеты и появляться учителя информатики, прошедшие достаточно длительную и специальную подготовку. Сохраняя единство программы и общий объем знаний и умений, работа в этих классах должна тем не менее носить в основном экспериментальный и, по возможности, вариантный характер для того, чтобы как можно быстрее приобрести опыт в преподавании этого нового курса, и в частности для безошибочного решения еще не решенной задачи – полноценного объединения теоретической и прикладной подготовки при достижении компьютерной грамотности.