

А.И. Бочкин, Д.Р. Кузьмичев

О выборе второго языка программирования при изучении информатики

В связи с переносом информатики в 8–9 класс и ниже возникла проблема: что же делать дальше, в 10–11 классе? Многие учителя понимали, что освоение только готовых программных средств наподобие Word, Excel лишь обедняет информатику как учебный предмет. Своеобразной реакцией педагогов на концепцию пользовательского курса стало повсеместное изучение языка Паскаль в 10–11 классах. Характерно, что изучается собственно язык Паскаль как объект и цель (а в Паскале есть что изучать!), а не язык как средство для решения задач. Но язык как объект изучения ведет к изоляции информатики от других дисциплин. До решения прикладных задач дело доходит редко; больше учебных, абстрактных задач на цикл, ветвление, процедуру. (Например, задания на команды ветвления выглядят нередко несколько устрашающе: «гора» из корней и синусов). Что касается изучения структур данных, то «очередь» едва ли ассоциируется с магазином, а множество – с реальным набором вещей. Как говорил один из героев «Алисы в стране чудес», множество из ничего – просто множество! Итак, ситуация с выбором языка требует методической оценки.

Для этого следует определиться с целями. Если ставить целью углубление в профессиональную информатику, начальную подготовку программистов, то Паскаль в школе практически вне конкуренции. Но сомнение вызывает уже сама эта цель даже для лицеев, не говоря о массовой школе. Нам представляется более полезной ориентация школьной информатики на взаимодействие с другими дисциплинами, решение прикладных задач [1]. Вне всякого сомнения это требует профессионального выбора средств, в частности языка. И в зону внимания попадает... Бейсик (конечно же, его новые и новые версии (Power, Turbo, Visual) без обязательной нумерации строк).

Мы не намерены снова открывать старую дискуссию на тему: какой язык лучше? Чтобы правильно выбрать язык как инструмент, нужно идти от практики, от задач. Причем задачи полезнее брать не из информатики, а из других дисциплин. Рассмотрим некоторые типично школьные задачи и темы. Понятно, что любая из них реализуется и на Паскале, но за счет усилий, дополнительных и внешних, по отношению к изучаемой теме или задаче. Внимание учащегося отвлекается от задачи на особенности ее реализации на довольно «болтливом» языке Паскаль. Но, как говорил Г.А. Кушниренко, «хорошая программа подобна стеклу на письменном столе: бумаги под стеклом видны, а само стекло – нет!» Итак, объект изучения – темы и задачи из других дисциплин.

Быстрое вычисление формулы или выражения, скажем для параллельного сопротивления, на Бейсике займет одну строчку: $?R1*R2/(R1+R2)$. На Паскале надо написать минимум 5 строк: `program ... var ... begin ... writeln ... end`. И это относится ко всему языку: на Бейсике можно простые вычисления писать сразу же, в начале курса, без изучения вводной темы «Общая структура программы».

Построение графика функции по точкам. Доступ к графическому режиму гораздо проще из Бейсика: SCREEN 12, и уже можно рисовать. Для перебора значений аргумента функции нужен цикл типа FOR, но лучше с дробным шагом, и в

Бейсике он есть. Кроме того, вещественные координаты точек графика сами, автоматически преобразуются Бейсиком в целочисленные экранные координаты. Важный частный случай: изображение траектории тела, брошенного под углом к горизонту: удобны вещественные координаты и дробный шаг по времени.

Моделирование биологических процессов во времени: размножение и гибель. Снова нужен цикл с дробным шагом ΔT по времени и простой доступ к графике.

Показательная функция A^B – одна из важнейших в школьной математике. В Паскале реализуется только через $\text{EXP}(B \cdot \text{LN}(A))$, что требует от школьника дополнительных знаний. Запись A^2 выглядит для ученика естественнее, чем паскалевское $\text{SQR}(A)$. Внутрипредметная связь: в терминах Бейсика корень – это, с одной стороны, $\text{SQR}(A)$, а с другой – $A^{(1/2)}$.

Отсчет реального времени при моделировании. Бейсик-функция `TIMER` возвращает «готовое» вещественное время в секундах с сотыми долями; Паскаль-процедура `TIME` возвращает целую структуру из суток, часов, минут, секунд и сотых секунды; прошедшее время здесь придется вычислять, пользуясь одновременно 60-ричной, 24-ричной и 10-ной системами счисления(!), причем надо перевести все данные к одному из вещественных типов.

Вычисление площади под кривой [2]: снова нужен цикл с дробным шагом по аргументу.

Можно продолжить ряд примеров в связи с другими дисциплинами.

Ситуация с изучением Паскаля требует объяснения. Причин массового перехода именно на Паскаль несколько.

Во-первых, у учителей-практиков возникло множество проблем со старым, «нумерованным» Бейсиком на маломощных школьных компьютерах – ни тебе процедур, ни циклов, ни структур данных. С появлением новых ЭВМ в школе они постарались забыть этот Бейсик. То, что данный язык развился до версий типа VB6, VBA, осталось незамеченным.

Во-вторых, заниматься внутренними проблемами Паскаля проще, чем межпредметными задачами. Да и выглядит солидно, не то что какой-то Бейсик. Стоит, однако, отметить, что определенная часть абитуриентов на экзамене дальше слов `PROGRAM`, `USES CRT` и `BEGIN` и `END` так и не продвигается. А дальше лежит уже не синтаксис языка, а решение задачи, **содержание** алгоритма, а не форма его записи.

В-третьих, информатика, как и многое другое, подвержена инерции. Она прошла от алгоритмизации через точку равновесия – межпредметные связи и уклонилась в решение собственных, внутренних задач на Паскале.

Наконец, по синтаксису Кумир и особенно Интал весьма близки именно к Паскалю. Впрочем, составные команды новых версий Бейсика по мощности не уступают паскалевским.

В случае перехода на Бейсик нужно что-то отвечать на вопрос ученика: «А зачем мы всегда описывали переменные в УАЯ или Интале, ведь Бейсик и так все понимает?». Ответ учителя в случае малых задач не очевиден.

Де-факто язык олимпиад – это Паскаль, а там – тесты с ограничением по времени. Но наши измерения показали, что на целочисленных циклах версия, например, `POWER-Basic` не уступает Паскалю в быстродействии.

Возможность не описывать переменные (что нередко понимается как серьезный недостаток старого Бейсика) отменяется при желании опцией `Explicit`. На маленьких алгоритмах, характерных для учебных задач на моделирование, в этом нет необходимости.

Можно предположить, что постепенно, по мере освоения учителями межпредметных задач и макросов в средах типа `WORD`, `EXCEL` возобновится и их интерес к Бейсику. Но уже к *новому* Бейсику как лаконичному и практично-

му языку, удобному при моделировании и решении межпредметных задач. Впрочем, для ряда задач как будто не нужен язык вообще, подойдут электронные таблицы, скорее всего Excel. (А в них может понадобиться составить функцию пользователя: а это – снова Бейсик, но уже версия VBA).

Итак, по нашему мнению, переход к новым версиям Бейсика вместо рывка в Паскаль был бы более плавным, эволюционным движением для школьной информатики. Частично сохранились бы методические разработки учителей, сделанные в связи с Бейсиком, особенно конкретные задачи. К сожалению, так не получилось.

В заключение отметим, что среды типа DELPHI (да и VBA) позволяют настроить окна, меню, кнопки и прочий сервис и замаскировать тем самым отсутствие собственно алгоритма решения задачи и даже неточность ее понимания и постановки.

ЛИТЕРАТУРА

1. **Ершов, А.П.** Основы информатики и вычислительной техники / А.П. Ершов [и др.]. – М.: Просвещение, 1985.
2. **Кушниренко, А.Г.** Основы информатики и вычислительной техники / А.Г. Кушниренко, Г.В. Лебедев, Р.А. Сворень. – М.: Просвещение, 1993.

S U M M A R Y

This article fully reveals a possibility of making strong relations between such subjects as informatics, physics and mathematics. On the whole it offers alternative solutions to the problems of school informatics.

Поступила в редакцию 27.01.2008