ПРЕДМЕТНОЕ ОБУЧЕНИЕ

Степанов Алексей Владимирович

СИСТЕМА КОМПЬЮТЕРНОЙ ГЕНЕРАЦИИ ЗАДАНИЙ ПО МАТЕМАТИКЕ

Представьте себе, что Вам надо составить контрольную работу по теме, например, «Квадратные уравнения». Вы хотите, чтобы в каждом варианте было по пять квадратных уравнений вида ax^2+bx+c, где a, b и c – целые числа, а корни уравнений – рациональные. Несмотря на то, что требуется только подставить числа по простому алгоритму, это отнимает довольно много времени, возникают ошибки и опечатки. Чем может помочь компьютер? Тут мы должны сразу разочаровать читателя: компьютер не может придумывать тексты задач – это делает преподаватель. Однако компьютер может достаточно много.

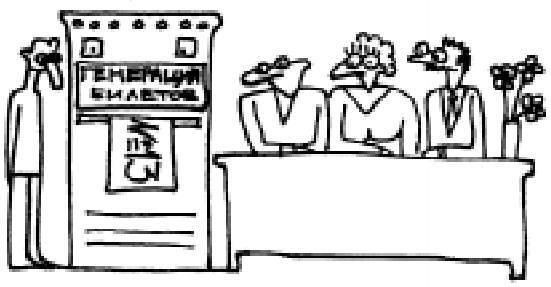
Во-первых, он может *создавать данные* для задачи. В рассмотренном выше примере, компьютер может выбрать два рациональных числа, x_1 и x_2, записать их в ответ и на основании этих чисел по теореме Виета создать коэффициенты a, b и c. При этом соответствующая программа очень проста, она может быть написана на любом языке программирования. В дальнейшем мы обсудим, какие программные средства лучше использовать для генерации данных.

Во-вторых, компьютер, а точнее, принтер, умеет *печатать* быстрее, чем Вы пишете от руки, если, конечно, «сказать» ему, что печатать. Выдавать учащимся листочек, на котором написаны только числовые данные, а условие задачи писать на доске – не лучший вариант при современных возможностях компьютера. Так что вторая проблема – найти подходящий способ вывода информации на печать. Обсудим по очереди каждую из этих двух проблем.

У нас есть несколько возможностей для создания данных. Во-первых, для этого может быть написана программа на обычном алгоритмическом языке. Вторая возможность – использование диалоговых средств, например, **MathCad**. Третья возможность – пользоваться универсальными математическими пакетами, которые позволяют программировать на более высоком уровне – **Maple** или **Mathematica**.

Какие достоинства и недостатки у каждого из перечисленных способов? В диалоговом режиме вы нечто вводите и нечто получаете от компьютера. Но идеал, который состоит в том, чтобы нажать одну клавишу и получить готовый текст, здесь недостижим. Поэтому диалоговый режим вычеркиваем сразу.

Преимущества программ класса Mathematica в том, что они способны, в



Чем может помочь компьютер?

частности, более или менее красиво выводить материал на печать, производить комплексные вычисления без их программирования – например, умножение многочленов или матриц делается с помощью одного оператора. К недостаткам таких программ можно отнести, во-первых, их собственный «диалект» (который надо еще выучить), в результате чего возникают трудности обмена данными с другими программами (скажем, Mathematica не создает исполняемого файла; для того чтобы выполнить программу, надо загрузить среду Mathematica, загрузить в нее программу, и только потом она ее выполнит; как видите, одним нажатием клавиши не обойтись). Второй недостаток в том, что они занимают много места на винчестере и требуют хорошего быстродействия компьютера. Несмотря на указанные недостатки, использование мощных математических пакетов при генерации данных является одной из возможных стратегий.

Система, о которой пойдет речь далее, построена на использовании обычных алгоритмических языков типа Pascal.

Теперь о том, как выводить информацию. И здесь есть несколько вариантов. Например, вывод данных в виде текста. Это не очень удобно, поскольку уже при выводе квадратных уравнений возникнут проблемы с печатью x^2 (при текстовом выводе это в лучшем случае можно получить, расположив x и 2 в разных строках). Еще больше трудностей будет с системами линейных уравнений (печать большой фигурной скобки) или заданиями, содержащими символ интеграла.

Другая возможность – использование мощных издательских систем, например, **Word**. Однако при использовании в наших целях любого текстового процессора типа **wysiwyg** (*what you see is what you get* – что видите, то и получите) возникнет одна и та же проблема. Все такие редакторы «замкнуты в себе». В каком формате программа должна вводить внешний файл, чтобы напечатать его из Word? Только во внутреннем формате Word. Для этого вам надо было бы изучить внутренний формат Word, что мы считаем совершенно нереальным.

Всех этих недостатков лишены издательские системы *компилирующего* типа, среди которых в первую очередь следует рассказать про **TEX**. Вы создаете файл с расширением **tex**, в котором находятся текст и ключевые слова (макрокоманды). Каждое ключевое слово начинается с символа «\». Например, если вы хотите начать новый абзац, вы пишете **\par**. Если вы хотите написать центрированный текст, вы набираете **\centerline{текст}**. Если вы хотите написать x^2, вы пишете **$x ^2$** (знаки **$** окружают математические формулы). После обработки программа TEX создает файл с расширением **dvi** (*device independent* – независимый от устройства). При наличии шрифтов такой файл можно вывести на печать независимо даже от операционной системы. Если такой файл создан, например, операционной системой **UNIX**, которая часто используется на Западе, то его можно распечатать и из **Windows** и из **DOS**. Это и означает «независимый от устройства» (совместимость – еще одно преимущество TEX над Word). Наличие макрокоманд, которые пользователь системы TEX может сам определять, а также возможности TEX работать с несколькими файлами одновременно, позволяют правильно организовать вывод на печать и взаимодействие с программой, создающей данные.

Поясним сказанное на том же примере о квадратных уравнениях. В основном файле мы можем написать:

```
Решите квадратное уравнение:
$\a x^2+\b x+\c=0$
```

в то время как программа, создающая данные, выводит их в другой файл в виде

```
\a=2 \b=5 \c=2
```

(справедливости ради, надо отметить, что в реальной жизни все немного сложнее, о чем еще будет сказано ниже).

Итак, мы выбрали TEX. Кстати, одно из преимуществ среды Mathematica, которое нам пока не уда-

лось до конца использовать, состоит в том, что она умеет создавать при выводе **tex**-файл.

Далее есть две стратегии. Одна состоит в том, что исполняемый файл после создания данных обрабатывает текстовый файл с условием задач (написанный в специальном, но довольно простом формате) и выдает на выходе **tex**-файл, содержащий нужное количество вариантов с вставленными числовыми данными.

Другая стратегия, которую пропагандирует автор, состоит с том, чтобы за каждое действие – генерацию данных, текст задания, вставку данных в текст, формат вывода (шрифт, размеры страницы) – отвечал бы свой файл. Далее мы расскажем немного подробнее о том, что же, собственно, сделано с помощью данной стратегии, но сначала два слова о том, как можно этим пользоваться.

Есть три уровня использования созданного программного продукта. Вы можете пользоваться только теми заданиями, которые уже подготовлены (а это – около 20 заданий по различным темам первого-второго курсов технического вуза). В этом случае вы запускаете **bat**-файл, после чего программа спрашивает количество вариантов, номер серии (любое число от 1 до 2^{32}) и затем выдает текст. Это первый уровень использования (вскоре, видимо, будет возможность сделать это через Интернет). Второй уровень: в готовых заданиях вы можете исправить текст и формат вывода. И третий уровень – создание своих индивидуальных заданий на основе уже построенной системы. Это требует некоторых навыков программирования и базовых знаний о языке системы TEX.

Перейдем к описанию системы **IDZ** (индивидуальные домашние задания). Система имеет следующую структуру (мы приводим упрощенную схематичную версию). В основном каталоге **IDZ** находятся директории с именами заданий, а также файлы **idz.bat** и **idz.tex**. Файл **idz.tex** отвечает за вставку данных в текст (он общий для всех заданий). Для того чтобы понять, как он работает, надо свободно владеть языком TEX. К счастью, понимать это нет необходимости ни при каком уровне пользования системой. Файл **idz.bat** предназначен для последовательного вызова всех задействованных компонент. В каждой директории с именем задания (назовем ее **XXX**) находятся файлы **formula.tex**, **solut.tex**, **idz.exe**, **format.tex** и **macros.tex**.

- **idz.exe** – это исполняемый файл, создающий числовые данные (а также случайные числа – номера задач, если вы хотите варьировать текст задачи);
- **formula.tex** и **solut.tex** – тексты условий задач и ответов, соответственно;
- **format.tex** – это, как следует из его названия, – задание формата вывода, то есть шрифт, размеры страницы, верхнего и левого отступа от края страницы, количество вариантов в строке (удобнее, чтобы листочки, выдаваемые учащимся, были квадратными) и т.п.;
- **macros.tex** – здесь записаны определения макрокоманд, необходимых для данного задания.

В процессе работы система создает временные файлы: с количеством вариантов (запрашивается у пользователя в начале работы), с номером серии (база генератора случайных чисел, чтобы вы могли воссоздать точно те же самые условия и ответы; это нужно, если учащийся потеряет свое условие или вы потеряете ответы), с числовыми данными и некоторые другие, которые нужны только для отладки. Опытный пользователь может попросить не стирать временные файлы, но обычно в них нет необходимости.

На втором уровне пользования системой вы имеете дело только с файлами **XXX\formula.tex**, **XXX\solut.tex** и **XXX\format.tex**. В последнем надо менять только числа, причем названия параметров, например, **\NumVarInRow**, говорят сами за себя. В файле **XXX\formula.tex** очень просто поменять текст. Например, вместо «Решите квад-

ратное уравнение», написать «Найдите точки пересечения параболы с осью абсцисс».

В заключение скажем несколько слов о том, как правильно писать формулы в файлах **XXX\formula.tex**, и **XXX\solut.tex**.

Рассмотрим пример. Представьте себе, что в формуле для квадратного уравнения вы (как мы советовали выше(!)) написали: **$\a x^2+\b x+\c=0\$**. Но если a = 1, b = –2, c = 0, то на печати мы получим: $1x^2 + -2x + 0 = 0$.

Согласитесь, что прочитать это можно, но очень неудобно. В программе это предусмотрено – например, вместо «+» пишут **\m+**, где макрокоманда **\m** определена следующим образом: если коэффициент равен 1, он не печатается, а печатается только переменная *x*, если равен 0 – не печатается ничего, если равен -1, то печатается «–», в остальных случаях печатается сам коэффициент. Макрокоманда зависит от трех параметров: знак («+», «–» или «.» для первого одночлена суммы), коэффициент и переменная (например, *x* или x^2). Эта макрокоманда, как и несколько других, определены в стандартном файле **macros.tex**, который обычно (но не всегда) просто копируется в **XXX\macros.tex**.

Еще одно замечание состоит в том, что иногда хочется варьировать текст внутри варианта, то есть выдавать разным студентам задачи, отличающиеся не только числовыми данными, например, одному – тригонометрический предел, другому – показательный и т.д. В этом случае: первое число в строке параметров для данного варианта (напомним, что эта строка создается программой **XXX\idz.exe**) указывает номер файла, из которого надо считывать текст; в файле **XXX\formula.tex** записано только: **\input XXX/f\a**, а сами тексты находятся в файлах **XXX\f1.tex**, **XXX\f2.tex** и т.д.

Степанов Алексей Владимирович, доцент кафедры ВМ-2 СПбГЭТУ (ЛЭТИ).