

М.Ф. Молочко, С.В. Тимченко

## ГЕНЕРАТОРЫ ПО ИНФОРМАТИКЕ В КОМПЬЮТЕРНЫХ ЭКЗАМЕНАХ

Опыт создания и использования компьютерных контрольных работ и экзаменаторов, накопленный в ТМЦДО, показал, что студенты быстро приспосабливаются к небольшому (100 вопросов) экзамену, заготавливают шпаргалки; происходит простое механическое запоминание ответов на данный вопрос, поскольку в ответ нужно ввести конкретное число или выбрать конкретный вариант. Таким образом, уменьшается эффективность проведения экзаменационных сессий и контрольных точек. Для устранения указанных недостатков предлагается несколько вариантов действий: расширить количество вопросов в компьютерных экзаменационных и контрольных работах, используя имеющуюся технологию; создать новую технологию, базирующуюся на идее использования «генераторов». Первый вариант решения проблемы имеет ряд существенных недостатков: преподаватель-методист должен записать новые вопросы, это всегда связано с большими затратами; количество вопросов существенно не увеличится, через достаточно небольшой период времени вопросы необходимо будет переписывать заново. Второй вариант решения проблем основывается на идее создания и использования генератора тестовых заданий и вопросов.

Генератор в компьютерной контрольной работе — это программа, генерирующая набор вопросов по какому-либо алгоритму, шаблону. Под шаблоном обычно понимают заготовку текста, в котором некоторые элементы можно изменять в соответствии с заданным алгоритмом. Шаблоны широко используются в программировании, например, шаблоны в C++ — мощный и развитый механизм, на основе которого была реализована идея генерирующего программирования. Под шаблоном задачи будем понимать ее описание, в котором исходные данные и/или часть задачи могут меняться. В тестовых системах наряду с формулировкой конкретной задачи необходимо иметь ее правильное решение или правиль-

ный ответ. Поэтому к шаблону нужно приложить программу решения задачи по сгенерированным параметрам.

Для шаблона задачи необходимо: выбрать некоторую задачу и выделить множество параметров, которое будет генерироваться; записать алгоритм решения; для каждого параметра записать множество изменения (это может быть список значений, интервалы или список интервалов); для каждого параметра записать алгоритм генерации значения; записать варианты формулировок задач. В некоторых случаях формулировка задачи может измениться в зависимости от значений параметров, поэтому необходимо записать алгоритм формулировки задачи.

Рассмотрим пример шаблона:

«Дан целочисленный массив длиной 8. Массив необходимо отсортировать по возрастанию. Массив дан после  $n$  шагов пузырьковой сортировки. Введите через пробел элементы массива после  $n+1$  шага,  $n$  изменяется от 1 до 5, первые  $n$  элементов уже отсортированы. Элементы массива случайным образом задаются в диапазоне от 1 до 99».

Ниже приведен алгоритм сортировки на языке Паскаль.

```
{сортировка пузырьковым методом}
procedure Bubble;
var
  i, j: index;
  x: item;
begin
  for i := 2 to n do
  begin
    for j := n downto i do
      if a[j-1].key > a[j].key then
        begin
          x := a[j-1];
          a[j-1] := a[j];
          a[j] := x;
        end;
    end;
  end; {конец сортировки пузырьковым методом}
```

В приведенном примере шаблона меняются следующие параметры: входной вектор и номер итерации. Количество конкретных тестовых заданий будет вычисляться по формуле  $v = 99^8 \cdot 5$ .

Это позволяет практически каждому студенту получить индивидуальное задание.

Разработка и внедрение генераторов является перспективным направлением совершенствования программного обеспечения автоматизированного обучения, которое нацелено на повышение качества образовательного процесса.

