

ГЕНЕРАЦИЯ ТЕСТОВЫХ ЗАДАНИЙ С ИСПОЛЬЗОВАНИЕМ СИСТЕМЫ КОМПЬЮТЕРНОЙ АЛГЕБРЫ

Паркалова О. В.

Кафедра компьютерных технологий и систем, Белорусский государственный университет
Минск, Республика Беларусь
E-mail: gorvtzova.olya@yandex.ru

Рассмотрен алгоритм для генератора тестовых заданий с использованием системы компьютерной алгебры Wolfram Mathematica. Сгенерированные задания используются в системе компьютерного тестирования.

ВВЕДЕНИЕ

Компьютерное тестирование как средство контроля знаний имеет большое практическое значение в сфере образования. Этим обусловлены многочисленные исследования по компьютерному тестированию специалистами из различных областей знаний. Актуальность этих исследований заключается в том, что совершенствование методического обеспечения способствует повышению качества образования. Один из аспектов, требующих детального рассмотрения, связан с процессом создания тестовых заданий. Отдельного рассмотрения заслуживает процесс генерации тестовых заданий.

ГЕНЕРАЦИЯ ТЕСТОВЫХ ЗАДАНИЙ

Основная идея генерации тестовых заданий состоит в автоматизации процесса составления большого количества неповторяющихся тестовых заданий.

Стоит отметить, что не существует универсального генератора тестовых заданий. Это связано с тем, что каждая область знаний и учебная дисциплина имеет свои особенности, а значит и отличия в формулировке вопросов и заданий. Поэтому необходимо наличие отдельных алгоритмов генерации условий для каждого вида заданий. Задача генерации тестовых заданий очень трудоемкая и требует наличия специалистов в различных областях знаний: в предметной области, по которой проводится тестирование, в программировании, в методике преподавания.

Формально принцип генерации заданий может быть представлен в виде шаблона задания, представляющего собой заготовку текста, в котором могут меняться фрагменты. То есть вместо определенных частей текста в шаблоне используются параметры, значения которых генерируются алгоритмами.

Помимо формулировки тестового задания требуется наличие правильного решения или ответа. Поэтому наряду с алгоритмом для генерации условия должен существовать алгоритм, решающий задачу и не допускающий формирования условий, не имеющих решения. В случае, если задача с какими-то сгенерированными па-

раметрами не имеет корректного решения, процесс генерации задания возобновляется с новыми сформированными параметрами, то есть имеет циклический характер [1].

Циклический характер имеет также сам алгоритм генерации заданий, в результате чего получается набор однотипных неповторяющихся заданий.

Рассмотрим частный случай генерации тестовых заданий, имеющих точный однозначный ответ, для дисциплин с математической нотацией. Средством генерации тестовых заданий выбрана система компьютерной алгебры Wolfram Mathematica. Функциональные возможности системы позволяют использовать ее для этих целей благодаря возможности аналитических преобразований и численных расчетов, работы со строками, наличию средств программирования. Язык программирования Wolfram поддерживает различные парадигмы программирования, поэтому есть возможность программирования каждой задачи в естественной для нее парадигме [2].

Шаблон задания генерируется с помощью функции StringJoin. Строка формулировки задания состоит из неизменяемых участков текста и сгенерированных по определенным правилам и специальными алгоритмами изменяемых частей.

Система Wolfram Mathematica поддерживает широко распространенные форматы данных и обеспечивает средства для экспорта и импорта данных [3].

Приведем пример шаблона генерации одного задания (см. рис. 1). В данной отрывке кода изменяемыми параметрами являются переменные a , b , f , инициализация которых с помощью формул, использующих случайные величины, происходит ранее. Условие задачи в этом примере сохраняется с помощью функции Export в PDF-файл.

```

task =
StringJoin[
"Найти предел последовательности  $x_n$  в
метрическом пространстве  $C$ ", ToString[a],
",", ToString[b], "], если он существует.\n",
"x_n=", ToString[f, TraditionalForm], "\n"];

Export[StringJoin[ NotebookDirectory[],
"//tasks_C//conditions//", ToString[j],
"_cond.pdf"], task];

```

Рис. 1 – Формат генерации задания

Задания и ответы, сгенерированные в системе, сохраняются в виде файлов в различных форматах, а затем используются в тестовой оболочке. Выбор формата, в который экспортируются сгенерированные задания, зависит от потребностей составителя тестовых заданий и возможностей системы компьютерного тестирования.

Отдельного внимания заслуживает формат сохранения ответов сгенерированных заданий в текстовом файле с расширением .txt (см. рис. 2). Это обусловлено тем, что ответ на задание вводится в поле ответа в клиентской части системы компьютерного тестирования и в таком виде может быть передан на сервер и обработан. А правильный ответ, используемый в качестве эталона для проверки, также должен быть представлен в строковом виде. Но здесь возникает задача формализации правил ввода ответов. Проблема заключается в преобразовании ответов к унифицированному виду, который можно однозначно интерпретировать как правильный ответ.

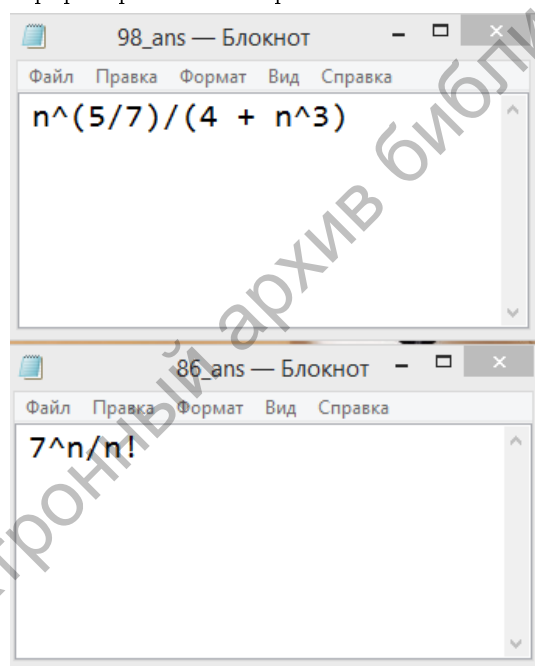


Рис. 2 – Формат ответов

Использование разработанных для отдельных видов заданий генераторов освобождает от рутинного процесса составления и формулирования каждого задания отдельно. И число сгенерированных тестовых заданий может быть очень велико. Это зависит от конкретного алгоритма генерации и области возможных значений параметров. Помимо этого, генерация тестовых заданий позволяет решить некоторые практические проблемы, возникающие в процессе проведения тестирования. Если существует база тестовых заданий, материалы из которой используются повторно (например, при проведении промежуточного и итогового контроля), то существует риск заготовки шпаргалок и механического запоминания ответов учащимися. При одновременной сдаче теста у нескольких испытуемых могут быть одинаковые задания, что также повышает вероятность списывания. А сгенерированные задания отличаются уникальностью, так как есть возможность создать отдельное задание для каждого студента [1].

ЗАКЛЮЧЕНИЕ

Генерация тестовых заданий является трудоемким процессом, но имеет практическую значимость в преподавательской деятельности для проведения контроля знаний. Созданные генератором задания удобно использовать в системах компьютерного тестирования и тем самым повысить эффективность работы с системой компьютерного тестирования и совершенствовать процесс обучения.

1. Кручинин, В. В. Генераторы в компьютерных учебных программах / В. В. Кручинин // Томск: Издательство Том. ун-та, 2003. – 200 с.
2. Таранчук, В. Б. Введение в язык Wolfram: учебные материалы для студентов факультета прикладной математики и информатики специальности 1-31 03 04 «Информатика» / В. Б. Таранчук // Минск: БГУ, 2015. – 51 с.
3. Таранчук, В. Б. Основы программирования на языке Wolfram: учебные материалы для студентов факультета прикладной математики и информатики специальности 1-31 03 04 «Информатика» / В. Б. Таранчук // Минск: БГУ, 2015. – 49 с.