

# Разработка современных систем педагогического тестирования

Кривиченко И.А.; Никульшин Б.В.

ИПиЭ, ФКП

УО «Белорусский государственный университет информатики и радиоэлектроники»

г. Минск, Республика Беларусь

e-mail: iarifina@mail.ru

**Аннотация** – В данной статье рассмотрены основные проблемы, возникающие при разработке систем тестирования без привязки конкретной предметной области, а также принципы контроля выполнения заданий.

**Ключевые слова:** педагогическое тестирование, генерация задания, пошаговый контроль, представление тестового задания

## I. ГЕНЕРАЦИЯ ЗАДАНИЯ

При разработке системы тестирования практических заданий, на сегодняшний день, главной проблемой является ограниченное количество вариантов тестовых заданий и как следствие необходимость постоянного обновления тестовых заданий.

На сегодняшний день не разработана ни одна система тестирования, в которой была бы решена проблема ограниченной вариативности. В данной статье предлагается следующий алгоритм решения этой проблемы: ввод варианта, генерация первого числа и выбор шаблона задания из базы данных, генерация массива чисел и заполнение ими шаблона задания, выполнение задания тестируемым. Остановимся на каждом этапе более подробно.

Сначала тестируемому предлагается ввести вариант, выданный ему преподавателем. Следует отметить, что номер варианта может принимать любые натуральные значения в пределах от 1 до  $2^{32}$  (4 294 967 296) для ПК с 32-разрядной ОС и до  $2^{64}$  (8 589 934 592) для ПК с 64-разрядной ОС.

Далее, используя алгоритм генерации случайных чисел, генерируется первое число, при этом введенный номер варианта используется как один из входных параметров используемого алгоритма генерации случайных чисел. Сгенерированное число умножается на количество шаблонов в БД и округляется до целого. В результате будет получен номер шаблона задания, который следует использовать в качестве основы тестового задания.

На следующем шаге можно генерировать массив чисел и, используя их, заполнять шаблон задания значениями числовых величин. После этого уже можно выводить задание на устройство ввода вывода. Тестируемый может приступить к выполнению задания.

Данный алгоритм может быть применен к любой практической задаче. Желательно, сформулировать шаблоны задания таким образом, чтобы решения задания во многом зависело от числовых величин.

В результате количество вариантов заданий будет зависеть от двух факторов: используемого алгоритма генерации случайных чисел и его входных параметров и разрядности ОС.

Был рассмотрен ряд алгоритмов генерации случайных равномерно распределенных чисел. В качестве наиболее важных характеристик рассматривались равномерность распределения первого элемента последовательности при разных

значениях входного параметра и равномерность распределения 20 последовательных элементов при разных значениях входного параметра. Равномерность распределения первого элемента последовательности при разных значениях входного параметра, т.е. варианта, обеспечивает равномерность из пользования БД. Равномерность распределения 20 последовательных элементов при разных значениях входного параметра обеспечивает различие двух вариантов заданий с одним шаблоном.

Табл. 1. Алгоритмы генерации случайных равномерно распределенных чисел

Название алгоритма генерации псевдослучайных последовательностей чисел	Односторонняя вероятность распределения хи-квадрат	
	для первого элемента последовательности	для 20 последовательных элементов
Алгоритм Лемера [1]	1,000	1,000
Алгоритм Блюма-Блюма-Шуба [3]	0,000	0,000
Алгоритм Л'Экюера [2]	0,297	0,639
Алгоритм Парка-Миллера [2]	0,000	0,000
Использование таблицы случайных чисел [4]	1,000	1,000

Наиболее оптимальным является алгоритм Лемера [1]. Рекуррентная формула в алгоритме Лемера выглядит следующим образом:

$$x_n = (ax_{n-1} + c) \bmod m. \quad (1)$$

На параметры  $a$ ,  $c$ ,  $m$  накладываются следующие ограничения: во-первых,  $c$  и  $m$  взаимно просты, во-вторых ( $a - 1$ ) кратно всем простым делителям  $m$ .

Значения первого члена  $x_0$  следует взять равным введенному номеру варианта (рис. 1).

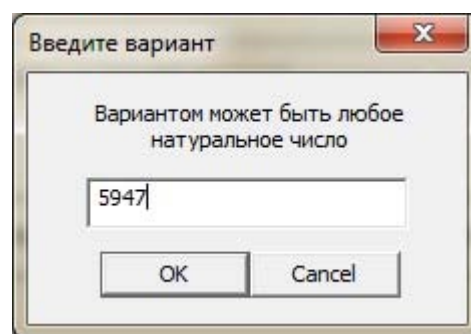


Рис. 1. Пример окна выбора варианта

## II. РАЗДЕЛЕНИЕ АТТЕСТАЦИОННОГО РЕЖИМА И РЕЖИМА ОБУЧЕНИЯ

Целесообразно разделять режимы обучения и аттестации, т.к. данные режимы предназначены для разных целей.

В режиме обучения следует организовать просмотр подробного описания совершаемых ошибок, учитывая нюансы алгоритмов решения.

Аттестационный режим предназначен для проверки знаний тестируемого. В этом режиме необходимо организовать отправку отчета о выполнении тестируемым задания преподавателю.

Рекомендуется изменить генерируемые задания в обоих режимах для того, чтобы не совпадали решения вариантов в аттестационном режиме и режиме обучения. Этого можно достичь, присвоив другие значения параметрам  $a$ ,  $c$ ,  $m$  при проектировании системы.

### III. ПОШАГОВЫЙ КОНТРОЛЬ ТЕСТОВОГО ЗАДАНИЯ

Если тестовое задание представляет собой сложную задачу с решением в несколько шагов, то проверку выполнения задания следует производить с пошаговым контролем правильности решения. Причем возможны две формы представления окна системы тестирования: с визуализацией всех шагов, но доступным заполнением только текущего шага и с визуализацией, только текущего и предыдущих шагов.

Первую форму представления следует применять, если алгоритм гибок и зависит от формулировки конкретного задания, это позволит тестируемому ориентироваться в ходе решения теста и видеть требуемое от него конечное решение задания (рис. 2).

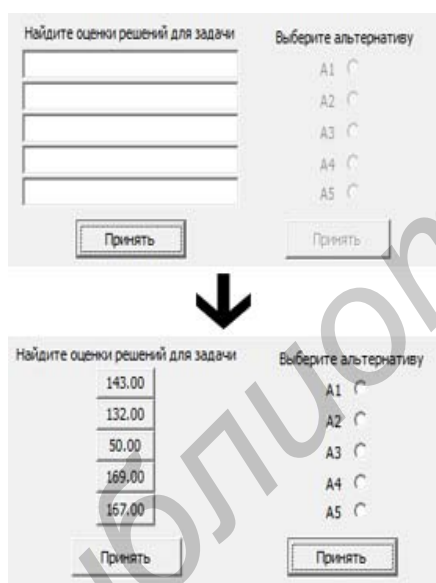


Рис. 2. Форма представления с визуализацией всех шагов

Вторую форму следует применять, если алгоритм решения задан четко и последующий шаг дает полный либо частичный ответ на предыдущий (рис. 3).

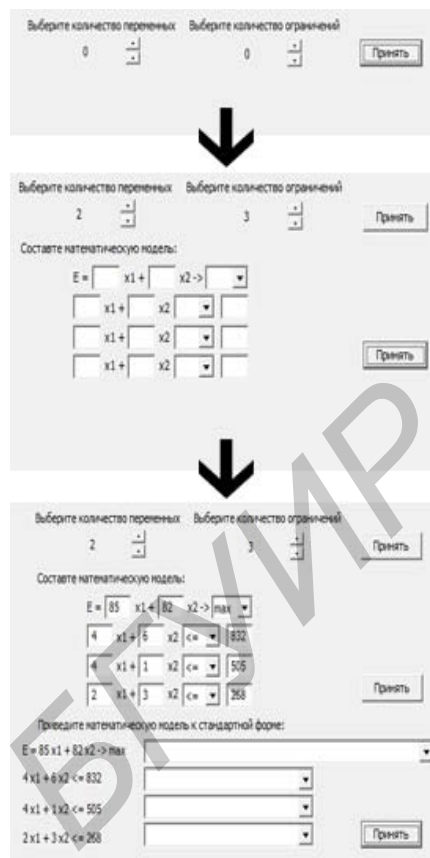


Рис. 3. Форма представления с визуализацией текущего и предыдущих шагов

### IV. ЗАКЛЮЧЕНИЕ

Таким образом, применяя современные методы генерации вариантов тестовых заданий и пошаговый контроль правильности решения, улучшается система педагогического тестирования.

[1]. С.С. Смородинский, Н.В. Батин. Оптимизация решений на основе компьютерных имитационных методов и моделей. Учебное пособие по курсу “Моделирование систем” для студентов специальности АСОИ дневной и дистанционной форм обучения. В 2-х частях. Часть 1. Мн: БГУИР, 2004, с.80  
 [2]. Генерация случайных чисел [Электронный ресурс]. – Электронные данные.– Режим доступа: <http://www.html.by/threads/13214-Generacija-sluchajnogo-chisla>  
 [3]. Алгоритм Блюма — Блюма — Шуба [Электронный ресурс]. – Электронные данные.– Режим доступа: [http://ru.wikipedia.org/wiki/Алгоритм\\_Блюма\\_—\\_Блюма\\_—\\_Шуба](http://ru.wikipedia.org/wiki/Алгоритм_Блюма_—_Блюма_—_Шуба)  
 [4]. Случайные числа [Электронный ресурс]. – Электронные данные.– Режим доступа: [http://www.math\\_algoritm.ru/read/1/5153/5162/](http://www.math_algoritm.ru/read/1/5153/5162/)