

## **ИСПОЛЬЗОВАНИЕ ТЕСТОВОГО КОНТРОЛЯ ЗНАНИЙ В КУРСЕ «ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ И МЕТОДЫ АЛГОРИТМИЗАЦИИ»**

Мороз Л. А., Затман Т. С. (Учреждение образования «Могилевский государственный университет имени А. А. Кулешова», кафедра математики и информатики)

Аннотация. Выделяются возможности и описывается назначение использования тестового контроля знаний студентов в курсе «Технологии программирования и методы алгоритмизации», приводится классификация групп используемых авторами тестов.

Эффективность обучения программированию можно оценить путем систематического качественного контроля знаний. Применяются различные способы диагностирования успеваемости студентов: опросы, самостоятельные работы, контрольные работы, курсовые работы. Обычная практика проведения таких занятий предполагает личное общение, где может иметь место небеспристрастность и списывание.

Очень важно иметь объективные результаты об усвоении материала, о недостаточности знаний или затруднениях в усвоении определенных тем, чтобы вовремя принять необходимые меры.

Более рационально организовать процесс обучения позволяют современные компьютерные системы тестирования. Использование тестов для контроля знаний студентов значительно сокращает время проверки для преподавателя. Кроме того, имеется ряд следующих преимуществ:

- тестовые задания генерируются динамически, что ставит всех в равные условия;
- тестируемый студент получает результаты сразу по завершении теста;
- тестируемый студент часто имеет возможность просмотреть неверные ответы;
- студенту необходимо не только знать определенный материал, но и уметь применить его для решения конкретной задачи;
- стимулируется познавательная активность студентов;
- низкая вероятность списать, подсмотреть, угадать правильный ответ;
- объективный результат теста снижает риск конфликта с преподавателем по неадекватно выставленной оценке.

Тестовые задания, используемые в изучении программирования можно разделить на группы по форме предполагаемых ответов:

1. задания закрытой формы с единственным правильным ответом, который выбирается из нескольких представленных;
2. задания открытой формы, где ответ вводится самостоятельно;
3. задания на установление соответствия, в которых элементам одного множества требуется поставить в соответствие элементы другого множества;
4. задания, в которых требуется установить правильную последовательность каких-то действий, шагов, операций и т. п.

Первая группа – это простейший вид задания, в котором правильный ответ уже имеется, и задача состоит в его узнавании. Примером такого задания является следующее.

Какое из следующих утверждений является правильным выходом для C#.NET фрагмента кода, приведенного ниже?

```
Console.WriteLine(13 / 2 + " " + 13 % 2);
```

1. 6.5 1;
2. 6.5 0;

3. 6 0;

4. 6 1.

Ко второй группе относятся задания, которые конструируются в виде утверждения, рядом с которыми готовые ответы с выбором не приводятся. Испытуемый сам дописывает в отведенном для этого месте свой ответ так, чтобы в результате получилось истинное высказывание. Эта форма задания сводит возможность догадки к минимуму. Примером такого задания является следующее.

Каким будет вывод в C#.NET фрагменте кода, приведенном ниже?

```
int a = 10, b = 20, c = 30;
```

```
int res = a < b ? a < c ? c : a : b;
```

```
Console.WriteLine(res);
```

В третьей группе заданий на каждый элемент слева должен найтись, по крайней мере, один элемент справа, а каждому элементу справа должен соответствовать только один элемент слева. Число элементов в правом столбце, как правило, больше числа элементов в левом. Примером такого задания является следующее.

Укажите соответствие типов данных в языке C#			
1	Целочисленные типы	short	A
		int	B
2	Типы с плавающей точкой	float	C
		byte	D
		double	E

К четвертой группе относятся задания на установление правильной последовательности. Они используются для проверки знаний хода процесса, цепочки событий, действий и операций, а также определений и понятий. Такие тесты помогают формировать у учащихся алгоритмическое мышление. Задания этой формы полезны как в качестве средства контроля знаний, так и в качестве средства обучения. Примером такого задания является следующее.

Для построения алгоритма вычисления среднего арифметического значений элементов целочисленного массива  $a[n]$  запишите блоки в нужной последовательности.

A	for (k = 0;	B	for (i = n;
C	k < n; k++)	D	i >= 0; i--)
E	if (0 == a[k])	F	s += a[i];
G	s = a[k];	H	s = s + 1;
I	s = s / N	J	s = 0;

Использование тестов позволяет не только выявить уровень знаний студентов по программированию, оценить качество усвоения ими учебного материала, но и мотивировать на более глубокое изучение дисциплины. Тесты способствуют развитию умственной деятельности, благотворно влияют на развитие интуиции и логического мышления и позволяют провести более широкий и более глубокий контроль за усвоением материала.

Нами разработаны тестовые задания по различным разделам курса «Технологии программирования и методы алгоритмизации», которые активно используются для промежуточного контроля знаний по предмету. Кроме того, имеется возможность адаптировать тесты для различного уровня подготовки студентов, постепенно повышать сложность заданий, тем самым добиваться достаточного качества знаний.