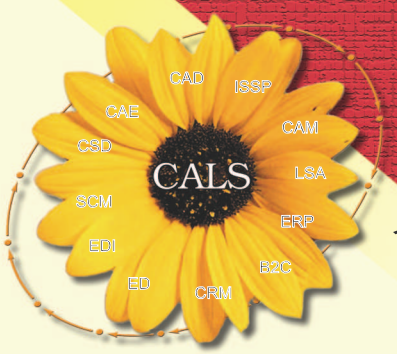
# КАЧЕСТВО ИННОВАЦИИ ОБРАЗОВАНИЕ

ISSN 1999-513X

№4
2013




журнал в журнале

КАЧЕСТВО и ИПИ (CALS)-технологии

www.quality-journal.ru

ЕЖЕМЕСЯЧНЫЙ НАУЧНО-ПРАКТИЧЕСКИЙ ЖУРНАЛ

# КАЧЕСТВО ИННОВАЦИИ ОБРАЗОВАНИЕ

№ 4 (95)
апрель 2013



АДРЕС РЕДАКЦИИ И ИЗДАТЕЛЯ
105118, Москва, ул. Буракова, д.8.
Тел.: +7 (495) 916-28-07,
+7 (495) 916-89-29,
факс: + 7 (495) 917-81-54
E-mail: quality@eqc.org.ru (для статей),
hg@eqc.org.ru (по общим вопросам)
www.quality-journal.ru; www.quality21.ru

ИЗДАТЕЛЬ
Европейский центр по качеству

НАУЧНЫЙ РЕДАКТОР
**Соболевский А.А.**
as@eqc.org.ru

ХУДОЖЕСТВЕННЫЙ РЕДАКТОР
**Каленова К.В.**

ЛИТЕРАТУРНЫЙ РЕДАКТОР
**Савин Е.С.**

ОТВЕТСТВЕННЫЙ СЕКРЕТАРЬ
**Нарыжная Е.С.**
ne@eqc.org.ru

ЖУРНАЛ ЗАРЕГИСТРИРОВАН
в Министерстве РФ по делам печати, телерадиовещания и средств массовых коммуникаций. Свидетельство о регистрации ПИ №77-9092.

ПОДПИСНОЙ ИНДЕКС
в каталоге агентства «Роспечать» 80620, 80621;
в каталоге "Пресса России" 14490.

ОТПЕЧАТАНО
Полиграфическая компания «КВМ-дизайн».
Москва, ул.Волжский б-р, д.29. www.kvm-d.ru



Журнал входит в перечень ВАК РФ

*Статьи рецензируются*

## СОДЕРЖАНИЕ

**ПРОБЛЕМЫ ПОДГОТОВКИ СПЕЦИАЛИСТОВ**

В.В. ПОДБЕЛЬСКИЙ, О.В. МАКСИМЕНКОВА
Особенности формулировок тестовых заданий по программированию . . . . . 3

А.С. ГУЗЕНКОВА
Подготовка специалистов в области экологического менеджмента . . . . . 10

А.С. БЕССОНОВ, Ю.Ю. КОЛБАС, Т.И. СОЛОВЬЕВА
Лабораторный практикум на основе аппаратно-программного комплекса для функциональной диагностики микроциркуляции крови . . . . . 13

**ИННОВАЦИОННЫЙ МЕНЕДЖМЕНТ**

С.А. ОГУДИН, И.Б. ТЕСЛЕНКО
Инновационные подходы в развитии сферы туризма региона . . . . . 19

М.Ю. СЕМУШКИН
Кластер как фактор модернизации металлургического производства в регионе (на примере Уральского Федерального Округа) . . . . . 23

Е.А. ГОПТА
Механизм генерирования инноваций: автоматизация процесса сетевого синтеза физического принципа действия . . . . . 28

**КАЧЕСТВО И ИПИ(CALS)-ТЕХНОЛОГИИ**

**ПРИБОРЫ, МЕТОДЫ И ТЕХНОЛОГИИ**

А.В. АНДРЕЙЧИКОВ, О.Н. АНДРЕЙЧИКОВА, Е.В. ТАБУНОВ, Ю.А. ФИРСОВ
Концептуальное проектирование технических инноваций с использованием интеллектуальной системы «IT QFD & AHP» . . . . . 32

Р.В. СКВОРЦОВ
Анализ технологических процессов изготовления печатных плат, используемых в радиоэлектронных приборах . . . . . 42

Ю.И. ГУДКОВ, А.Л. ТУВ
Перспективы использования активных инфракрасных датчиков в инструментальных средствах контроля защищенных зон . . . . . 48

А.Д. КАЛУЖСКИЙ
Некоторые вопросы эффективности работы эргатических и потребительских систем . . . . . 52

**ЭКОНОМИКА И УПРАВЛЕНИЕ**

А.В. ВИШНЕКОВ, Е.М. ИВАНОВА
Сравнительная оценка вычислительных систем по критериям пользователя . . . . . 63

А.А. ЛЯБИН
Модернизация управления бизнес-процессами и ресурсами в организации (на примере ООО «Эквант») . . . . . 69

А.М. КАРЯКИН, А.А. КАЛИНИН
Модель оценки Гудвилла компании . . . . . 78

Сведения о членах редколлегии и об авторах статей можно найти на сайте www.quality-journal.ru

В.В. Подбельский, О.В. Максименкова

# ОСОБЕННОСТИ ФОРМУЛИРОВОК ТЕСТОВЫХ ЗАДАНИЙ ПО ПРОГРАММИРОВАНИЮ

Рассматривается применение общих рекомендаций по работе над формулировками тестовых заданий в контексте дисциплины "Программирование". Приводятся примеры некорректных заданий и варианты их исправления. Формулируются дополнительные рекомендации для тестовых заданий, связанных с программированием.

**Ключевые слова: программирование, тестовые задания, показатели качества**

Применение тестов для контроля знаний по различным предметным областям требует от преподавателя работы не только по созданию новых, но и по изменению (модификации) уже существующих тестовых заданий. Необходимость модификации формулировок готовых тестовых заданий возникает в случаях, когда, например, задание обладает низкими показателями качества. Напомним общие теоретические положения, позволяющие количественно описать эти показатели. Для оценки качества тестовых заданий вычисляют такие числовые характеристики, как индекс (коэффициент) дискриминативности и коэффициент трудности (решаемости) тестового задания.

При дихотомической оценке тестовых заданий (когда за решенное задание начисляется 1 балл, а за нерешенное 0 баллов) в соответствии со статистическими методами классической теории тестирования [1, 2, 4] коэффициент трудности вычисляется как доля испытуемых, справившихся с тестовым заданием:

$$k = \frac{n}{N},$$

где $n$ - количество испытуемых, верно решивших задание, $N$ - общее количество испытуемых. Традиционно задания с коэффициентом, не превосходящим 0,2, относят к трудным, а задания с коэффициентом не меньшим 0,8 - к легким.

На основе метода контрастных групп коэффициент дискриминативности тестового задания вычисляется по формуле: $D = \frac{n_b}{N_b} - \frac{n_w}{N_w},$

где $n_b$ - количество испытуемых в группе лучших, набравших 1 балл за задание; $n_w$ - количество испытуемых в группе худших, набравших 1 балл за задание; $N_b$ - общее количество испытуемых в группе лучших; $N_w$ - общее количество испытуемых в группе худших.

Принято [1, 2, 11] подвергать пересмотру и отбраковке задания, имеющие показатель дискриминативности меньший 0,3.

Говоря о структуре тестовых заданий, будем придерживаться следующей терминологии:

-стем - содержательная постановка задачи;

-опции - варианты ответов на тестовое задание;

-ключ (ответ) - правильный ответ на тестовое задание;

-дистрактор- неправильный ответ на тестовое задание, внешне схожий с верным ответом.

В российской и, в особенности, зарубежной литературе встречаются рекомендации по составлению и исправлению содержания тестовых заданий, которые можно обобщить следующим образом [1, 2, 7, 11, 8]:

-в тексте задания должна отсутствовать двусмысленность и неясность формулировок;

-следует избегать слов-подсказок и общих грамматических подсказок, таких как род, число или падеж, позволяющих "вычислить" верный ответ;

-стем (основная часть) задания должен быть сформулирован в утвердительной форме;

-предложения, составляющие стем, следует формулировать максимально полно, оставляя в опциях (ответах и дистракторах) как можно меньше слов;

-следует избегать частично верных дистракторов;

- дистракторы должны быть разумны, правдоподобны и привлекательны для испытуемых;

-опции, выраженные числами, следует упорядочивать по убыванию или возрастанию;

-опции, выраженные словами, следует упорядочивать по алфавиту;

-желательно избегать повторяющихся слов в начале опций, повторения должны быть вынесены в стем;

-и др.

Кроме того, при составлении тестовых заданий необходимо учитывать такие индивидуальные особенности студентов, как, например, слабое зрение или отсутствие определенных навыков [11].

**О тестовых заданиях по программированию**

Несмотря на то, что эти рекомендации являются достаточно общими, в публикациях [1, 4, 8] они сопровождаются, в основном, примерами тестовых заданий по гуманитарным и естественно-научным дисциплинам. Отдельные источники [7] опираются на абстрактные и искусственные примеры заданий [7]. Разбор заданий, связанных с программированием встречается редко и относится, преимущественно, к процедурным учебным языкам программирования (школьный алгоритмический, Бейсик и Паскаль).

В высших учебных заведениях в дисциплинах, связанных с объектно-ориентированным програм-

мированием (обычно это курс "Алгоритмические языки и программирование"), в качестве учебных используются современные языки программирования (C#, C++, Java) и интегрированные среды разработки (MS Visual Studio, NetBeans IDE и проч). Нарушения перечисленных выше рекомендаций для тестовых заданий по таким дисциплинам редко анализируются в публикациях и не всегда очевидны.

В данной работе рассмотрены особенности некорректных тестовых заданий, ориентированных на язык программирования C#. Задания с низкими показателями качества отобраны из тестов промежуточного контроля по дисциплине "Программирование", преподаваемой (изучаемой) на первом курсе бакалавриата Отделения программной инженерии Национального исследовательского университета Высшая школа экономики.

**Формы тестовых заданий по программированию**

В классификациях тестовых заданий, приводимых в специальной литературе [1, 6, 11], принято деление тестовых заданий на задания закрытого типа и открытого типа на дополнение. Задания закрытого типа и с одним верным ответом, и с несколькими верными ответами в общем случае состоят из содержательной постановки задачи (стема) и набора вариантов ответов (опций), среди которых присутствует один или несколько верных ответов и набор неверных ответов (дистракторов). Задания открытого типа на дополнение состоят из инструкции к заданию и его формулировки, в теле или после которой присутствуют пропуски для записи ответа.

С учетом указанного деления и в связи со спецификой предмета "Программирование" тестовые задания по содержанию могут быть отнесены к двум типам.

**Тип Т**: Тестовое задание для проверки теоретических знаний по синтаксису и семантике языка. Оформляется как задание закрытого типа с одним или несколькими правильными ответами, без полного исходного кода программ или функционально-законченных блоков кода.

**Тип П**: Тестовое задание прикладного характера для проверки практических навыков применения синтаксиса и семантики языка к анализу функциональности программ и разработке программ с заданной функциональностью. Предусматривает включение в формулировку задания кода на изучаемом языке программирования. Оформляется как задание закрытого типа с одним или несколькими верными ответами (**тип Пз**), либо как задание открытого типа с кратким ответом (**тип По**), либо как комбинированное задание, состоящее из задания открытого типа с вариантами закрытых ответов, позволяющих оценить умение анализировать синтаксис и семантику кода на предмет его корректности (**тип Пк**) [3, 12].

Задания, содержащие вопрос "Какое сообщение выдаст компилятор или исполняющая система в результате обработки данной программы?" или подобные, требующие точного воспроизведения формулировок сообщений инструментальных средств разработки и/или исполнения программ, недопустимы.

**Спецификация теста по программированию**

Каждый тест, применяемый в курсе "Программирование" на отделении программной инженерии НИУ ВШЭ, включает 30-40 вопросов и подготавливается в виде отдельного документа.

Задание закрытой формы включает пять вариантов ответов, из которых один или несколько (в том числе и все) могут быть правильными (ключи).

Текст каждого задания не превышает 24 строк, так как вопросы большей длины иногда не умещаются на экране и плохо воспринимаются тестируемыми.

Тестирование проводится как на компьютере с применением тестирующих программ, так и на бланках. В обоих случаях (и при использовании бланков, и при компьютерном тестировании) на выполнение теста (из 30-40 вопросов) отводится фиксированное время (от 40 до 60 минут). Результаты оцениваются по бинарной системе (дихотомическая система оценок) [3, 10, 12].

Отметим, что указанные тесты в целом имеют удовлетворительные показатели надежности и валидности [10]. Появление некачественных заданий в этих тестах является достаточно редким событием, что видно из рисунков 1 и 2, демонстрирующих количество испытуемых, решивших отдельные задания некоторых конкретных сеансов тестирований. Исходя из приведенных на рисунках гистограмм, можно обратить внимание на рис. 1 на задания с номерами 17 (один верный ответ) и 25 (два верных ответа), а также на задание 29 на рис. 2 (шесть верных ответов).

Покажем на конкретных примерах, как могут быть сформулированы такие "провальные" задания и каким образом можно их модифицировать с целью повышения их качества. Еще раз подчеркиваем, что работа посвящена формулировкам заданий тестов по дисциплине "Программирование".

Исправление некорректных тестовых заданий проведем на основе перечисленных выше рекомендаций.


Рис. 1. Результаты выполнения теста 1


Рис. 2. Результаты выполнения теста 2

**Примеры модификации заданий**

*Задание 1 закрытого типа с несколькими верными ответами*

| **Верно ли, что:** |
|---|
| +1) При объявлении локальной переменной, её тип может определяться типом инициализирующего значения.<br>2) Локальная переменная, объявленная во вложенном блоке, может иметь то же имя, что и переменные охватывающего блока.<br>3) Локальная переменная, объявленная во вложенном блоке, доступна во всех операторах охватывающего блока.<br>4) Локальная переменная, объявленная во вложенном блоке, доступна в операторах охватывающего блока, размещенных вслед за вложенным блоком.<br>+5) Локальная переменная доступна для всех операторов вложенных блоков |

Здесь и далее верные варианты ответов тестового задания отмечены знаком "+".

Существует несколько способов повышения качества приведенного задания.

Вопросительная формулировка стема неявно подразумевает ответы "верно"/"не верно", что соответствует форме задания с двумя ответами.

*Первый способ*. Создание пяти заданий, в каждом из которых стем формулируется в виде утвердительного предложения. Варианты заданий:

| **При объявлении локальной переменной её тип может определяться типом инициализирующего значения** |
|---|
| +1) да |
| **Локальная переменная, объявленная во вложенном блоке, может иметь то же имя, что и переменные охватывающего блока** |
| 1) да<br>+2) нет |
| **Локальная переменная, объявленная во вложенном блоке, доступна во всех операторах охватывающего блока** |
| 1) да<br>+2) нет |
| **Локальная переменная, объявленная во вложенном блоке, доступна в операторах охватывающего блока, размещенных вслед за вложенным блоком** |
| 1) да<br>+2) нет |
| **Локальная переменная доступна для всех операторов вложенных блоков** |
| +1) да<br>2) нет |

При выполнении задания с двумя ответами испытуемым затрачивается меньше времени [11], чем при выполнении задания с несколькими верными ответами. Поэтому увеличение числа заданий теста слабо скажется на времени его прохождения испытуемыми. К недостаткам заданий с двумя ответами относят высокую вероятность угадывания верного ответа, поэтому при такой замене исходного задания пятью для снижения вероятности угадывания общее количество заданий в тесте увеличивают.

*Второй способ*. Предложенные варианты ответов и дистракторы содержат избыточные слова, которые могут быть вынесены в стем. Пример того, как это можно выполнить в данном задании, очевиден из предыдущего изменения задания, при котором сформулированы пять заданий.

*Задание 2 открытого типа с кратким ответом*

| **В результате выполнения следующего фрагмента кода:** |
|---|
| int x = 14;<br>Console.Write("{0,3:f4}", x*1000);<br>**на экране появится представление числа, содержащее X нулей.**<br>**Укажите значение X:** ____ |
| *Ответ: 7* |

Во фрагменте кода использована переменная с именем x, инициализированная числом 14, в формулировке задания требуется указать некоторое значение X, под которым подразумевается количество отобразившихся на экране нулей. Апробация данного тестового задания показала, что некоторые студенты в качестве ответа указывают не 7, а 14 (значение переменной x). Одинаковые имена переменных и неизвестных вносят в задание двусмысленность, которую легко ликвидировать, заменив одно из имен, например, так:

**В результате выполнения следующего фрагмента кода:**

```
int a = 14;
Console.Write("{0,3:f4}", a*1000);
```

**на экране появится представление числа, содержащее X нулей.**
**Укажите значение X: ____**

*Ответ: 7*

*Задание 3 закрытого типа с одним верным ответом*

**Полная форма условного оператора имеет формат:**

```
if (<выражение_условие>)
        <оператор_1>
else
        <оператор_2>
```

**В сокращенной форме условного оператора отсутствует:**

1) (<выражение_условие>)
2) <выражение_условие>
3) <оператор_1> и <оператор_2>
4) else
+ 5) else и <оператор_2>

Низкие показатели качества данного задания связаны с тем, что дистрактор № 4 является частично верным. Студенты ошибочно выбирают его вместо верного ответа (ключа) № 5. Это во многом связано с шаблонной формулировкой, встречающейся в учебной литературе: "В сокращенной форме условного оператора отсутствует else". Также из формулировки задания не очевидно, что речь идет об анализе структуры оператора, приведенной в стеме.

Для исправления предлагается в качестве вариантов ответа к заданию дать только структурные составляющие условного оператора, преобразовать задание к форме с множественным выбором (с несколькими ключами) и внести уточнения в стем. Преобразованное задание примет такой вид:

**Структура полной формы условного оператора имеет формат:**

```
if (<выражение_условие>)
        <оператор_1>
else
        <оператор_2>
```

**В структуре сокращенной формы условного оператора отсутствует:**

1) if
2) <выражение_условие>
3) <оператор_1>
+ 4) else
+ 5) <оператор_2>

*Задание 4 открытого типа с кратким ответом*

**Что будет выведено на экран в результате выполнения следующего фрагмента кода:**

```
static void Main()  {
        int i = 7;
        while (i > 0)  {
                do
                        if (i > 3) Console.Write(i);
                while (i-- > 2);
                Console.WriteLine(i);
                break;
        }
}
```

**Если Вы считаете, что код содержит ошибки – укажите в качестве ответа *****

*Ответ: 76541*

Очевидным расхождением с рекомендациями по составлению тестовых заданий является, во-первых, вопросительная форма формулировки текстовой части стема. Второе затруднение связано с особенностью средств языка программирования C#. В приведенном в задании коде использован оператор **Console.WriteLine()**, выводящий в консольное окно строку и переводящий курсор на новую строку в консольном окне. Применение такого оператора вывода с переводом строки нежелательно для заданий с открытым ответом, так как вызывает дополнительные сложности при вводе ответа как в случае компьютерного тестирования, так и при записи ответа в случае бланкового тестирования. Дополнительное уточнение вопроса в виде фразы: "Если Вы считаете, что код содержит ошибки - укажите в качестве ответа *****", - описывает только случай наличия ошибки в коде. Из приведенной формулировки задания не ясно, например, что именно требуется ввести (внести в бланк) в качестве ответа в случае, если программа не выведет на экран ничего, но в коде нет ошибки.

Исправить задание предлагается следующим образом:

**В результате выполнения следующего фрагмента кода:**

```
static void Main()  {
        int i = 7;
        while (i > 0)  {
                do
                        if (i > 3) Console.Write(i);
                while (i-- > 2);
                Console.Write(i);
                break;
        }
}
```

**на экран будет выведено: ____**
*Примечание:*
*Если возникнет ошибка компиляции, введите:* ***
*Если ошибок и исключений нет, но на экран не выведется ничего, введите:* ---
*Если возникнет ошибка исполнения или исключ ение, введите:* +++

*Ответ: 76541*

Заметим, что только что рассмотренное задание представляет собой комбинацию заданий отрытого и закрытого типов и относится к типу Пк.

Мы только что привели пример задания типа Пк, в котором тестируемому предлагалось не только "вычислить" верный ответ, но и проанализировать код на предмет выявления синтаксических и семантических ошибок. Вариант следующего задания концентрирует внимание тестируемого только на логике кода, и этот код в стеме не должен содержать ошибок.

*Задание 5 открытого типа с кратким ответом*

| **В результате выполнения следующего фрагмента программы:** |
|---|
| char s, beg = 'S', end = (char)(beg+5);<br>for ( s = beg; s < end; s++)<br>  if (s % beg == 0) Console.Write(s);<br>**на экран будет выведено: ___** |
| *Ответ: S* |

В качестве значения переменной beg выбран символ S, имеющий в некоторых случаях плохо различимые очертания. Во-первых, буква S имеет одинаковую форму как в строчном, так и заглавном начертании. Кроме того, в некоторых случаях при выводе на дисплей символ S можно спутать с цифрой 5. Имя другой переменной - s, выводимой на экран, с точностью до регистра совпадает с верным ответом. Разбор кода займет больше времени у студентов со слабым зрением.

Второй дефект разбираемого задания связан с тем, что студенты, забывшие последовательность букв в латинским алфавите, могут испытать существенные затруднения при выполнении задания 5, так как в формулировке отсутствует подсказка в виде латинского алфавита. Хотя знание алфавита не требуется в данном задании, тестируемый может решить, что только алфавита ему не хватает.

Устранить перечисленные трудности можно, так переформулировав задание:

| **В результате выполнения следующего фрагмента программы:** |
|---|
| char s, beg = 'G', end = (char)(beg+5);<br>for ( s = beg; s < end; s++)<br>  if (s % beg == 0) Console.Write(s);<br>**на экран будет выведено:_ _**<br>*Примечание:*<br>*ABCDEFGHIJKLMNOPQRSTUVWXYZ* |
| *Ответ: G* |

*Задание 6 закрытого типа с одним верным ответом*

| **В результате выполнения следующего фрагмента программы:** |
|---|
| int dif;<br>char ch1 = 'A';<br>char ch2 = 'c';<br>dif = Char.ToUpper(ch2).CompareTo(ch1);<br>Console.WriteLine(dif);<br>**на экран будет выведено:** |
| 1) -1<br>2) 1<br>+ 3) 2<br>4) 34<br>5) 0 |

Для понимания предлагаемого кода нужно знать назначение статического метода **ToUpper()** и метода **CompareTo()**. Метод **ToUpper()** преобразует код символа-аргумента к коду того же символа, но в верхнем регистре. В нашем случае будет получен из символа 'c' символ 'C'. В результате выполнения метода **CompareTo()** с символами 'C' и 'A' будет вычислено "расстояние" между ними в алфавите (разность значений числовых кодов символов). Все достаточно просто, но в задании использованы символы 'A' и 'C', имеющие одинаковые начертания в русском и латинском алфавитах. Догадаться, что используется латинский алфавит можно из анализа дистракторов, но это повышает время выполнения задания. Кроме того, дистрактор № 4 заведомо неправдоподобен. Напомним, что в случае, когда в качестве вариантов ответа приводят числовые значения, рекомендуется упорядочивать их, что в данном задании не выполнено.

Важным изменением в задании является замена букв A и C истинно латинскими буквами. Кроме того, аналогично изменениям задания № 5, в формулировку следует добавить латинский алфавит, заменить дистрактор № 4 на более правдоподобный и упорядочить варианты ответов по возрастанию (убыванию). С учетом сказанного, преобразованное задание может принять следующую формулировку:

| **В результате выполнения следующего фрагмента программы:** |
|---|
| int dif;<br>char ch1 = 'L';<br>char ch2 = 'n';<br>dif = Char.ToUpper(ch2).CompareTo(ch1);<br>Console.WriteLine(dif);<br>**на экран будет выведено:**<br>*Примечание:*<br>*ABCDEFGHIJKLMNOPQRSTUVWXYZ* |
| 1) 0<br>2) -1<br>3) 1<br>+ 4) 2<br>5) 3 |

*Задание 7 комбинированного типа*

**В результате выполнения следующей программы:**

```
using System;
namespace Тест{
   class A   {
      public int Field1;
   }
   class B : A   {
      public int Field2;
   }
   class Program   {
      static void Main()  {
         A myVar1 = new A();
         B myVar2 = (B)myVar1;
      }
   }
}
```

**на экран будет выведено: ___**

*Примечание:*
*Если возникнет ошибка компиляции, введите: ****
*Если ошибок и исключений нет, но на экран не выведется ничего, введите: ---*
*Если возникнет ошибка исполнения или исключение, введите: +++*

*Ответ: +++*

В формулировке задания предлагается определить результат вывода при выполнении фрагмента кода. Отметим, что в коде отсутствуют обращения к методам вывода на экран. В коде присутствует ошибка, приводящая к возникновению исключения. Отсутствие выводимого результата и наличие синтаксической ошибки делают выбор ответа неоднозначным.

При исправлении задания следует уточнить формулировку (код программы), добавив, по крайней мере, оператор вывода на экран. Задание примет такой вид:

**В результате выполнения следующей программы:**

```
using System;
namespace Тест{
   class A   {
      public int Field1;
   }
   class B : A   {
      public int Field2;
   }
   class Program   {
      static void Main()  {
         A myVar1 = new A();
         B myVar2 = (B)myVar1;
                  Console.Write(myVar2.Field1);
      }
   }
}
```

**на экран будет выведено: ___**

*Примечание:*
*Если возникнет ошибка комп иляции, введите: ****
*Если ошибок и исключений нет, но на экран не выведется ничего, введите: ---*
*Если возникнет ошибка исполнения или исключение, введите: +++*

*Ответ: +++*

*Задание 8 закрытого типа с несколькими верными ответами (ключами)*

**Верно, что декларация интерфейса может быть снабжена модификаторами:**

+ 1) new
+ 2) public
+ 3) protected
+ 4) internal
+ 5) private

Все приведенные варианты ответов являются верными. Отсутствие дистракторов снижает вероятность правильного ответа на задания и повышает время его выполнения, так как испытуемые (в особенности группа сильнейших) подозревают скрытые в задании хитрости.

Рекомендуется заменить некоторые верные ответы на дистракторы, или создать несколько новых заданий. Задание можно переформулировать следующим образом:

**Верно, что декларация интерфейса может быть снабжена модификаторами:**

1) static
+ 2) public
+ 3) protected
+ 4) internal
+ 5) private

**Заключение**

Рассмотренные примеры демонстрируют источники возникновения неточностей и ошибок, допускаемых при составлении тестовых заданий по дисциплине "Алгоритмические языки и программирование". Кроме того, из детального разбора каждого задания видно, что по составлению тестовых заданий по программированию могут быть предложены следующие рекомендации:

-Не использовать в кодах операторы вывода на экран, переводящие после вывода курсор на новую строку (пример 4).

-В формулировке заданий, содержащих код, следует предусматривать все возможные ситуации, такие как возникновение ошибок компиляции, ошибки исполнения или отсутствие вывода (отсутствие результата).

-В заданиях, содержащих в кодах символы и строки, приводить (при необходимости) в качестве справки алфавит того языка, символы которого используются в задании.

-В заданиях, содержащих в кодах символы и строки, применять символы, различаемые в строчном и прописном начертании, и не совпадающие по начертанию в алфавитах разных культур (например, русском и латинском алфавитах).

Сформулированные в работе рекомендации по составлению тестовых заданий, связанных с языками программирования, применяются авторами на протяжении нескольких лет. Тесты по программированию студенты проходят регулярно. Например, на первом курсе бакалавриата Отделения программ-

ной инженерии НИУ ВШЭ тестирование проводится шесть раз на разных этапах изучения дисциплины "Программирование". Анализу результатов авторы уже посветили работы [9, 10, 12].

**Литература:**

1.Майоров А.Н. Теория и практика создания тестов для системы образования. (Как выбирать, создавать и использовать тесты для целей образования). - М.: "Интеллект-центр", 2001. 296 с.

2. Звонников В.И., Челышкова М.Б. Современные средства оценивания результатов обучения. - М.: Издательский центр "Академия", 2009. 224 с.

3.Подбельский В.В., Максименкова О.В. Разработка тестов по программированию для тестирования в компьютерной форме // Информатизация образования - 2011: материалы Международной научно-практической конференции, Елец: ЕГУ им. И.А. Бунина, 2011. С. 192-195.

4. Бурлачук Л.Ф., Морозов С.М. Словарь-справочник по психодиагностике. - СПб.: Питер, 2008. 688 с.

5. Карданова Е.Ю. Моделирование и параметризация тестов: основы теории и приложения. - М.: ФГУ "Федеральный центр тестирования", 2008. 303 с.

6. McKenna C., Bull J. "Designing effective objective test questions: an introductory workshop", CAA Centre, June 1999 [Электронный ресурс] http://www.caacentre.ac.uk/dldocs/otghdout.pdf (дата обращения: 02.02.2011)

7.Anderson P., Morgan G. "National assessments of educational achievement, V.2: Developing tests and questionnaires for a national assessment of educational achievement, World Bank Publications, 2008. 288 p.

8. Osterfind S.J. Constructing Test Items : Multiple-Choice, Constructed-Response, Performance and Other Formats. / Kluwer academic publishers, 1997. - 352 p.

9. Podbelskiy V.V., Maksimenkova O.V. Programming as a part of the Software Engineering education // Proceedings of the 4-th Spring/Summer Young Researchers' Colloquium on Software Engineering (SYRCoSE 2010), 2010. PP.165 - 168.

10 .Podbelskiy V.V., Maksimenkova O.V. Educational tests in "Programming" academic subject development // SYRCoSE 2011. Proceeding of the 5-th Spring/Summer Young Researchers' Colloquium on Software Engineering, 2011. PP. 88-93.

11 .Alabama professional development modules [Электронный ресурс] http://www.alabamapepe.com/profdevmodule/index.htm (дата обращения: 15.05.2012)

12.Максименкова О.В., Подбельский В.В. Опыт разработки тестов и анализ результатов тестирования по программированию // Преподавание информационных технологий в Российской Федерации: материалы Десятой открытой Всероссийской конференции (16-18 мая 2012 года). - М.: МГУ им. М.В. Ломоносова, 2012. С. 367-368.

***Подбельский Вадим Валериевич,***
*д-р техн. наук, профессор НИУ ВШЭ.*
*e-mail: vpodbelskiy@hse.ru,*

***Максименкова Ольга Вениаминовна,***
*ст. преподаватель НИУ ВШЭ.*
*-mail: omaksimenkova@hse.ru*

V.V. Podbelskiy, O.V. Maksimenkova

## FEATURES OF THE LANGUAGE OF TESTS ON PROGRAMMING

The paper deals with the application of general recommendations concerning design of educational test's questions in the "Programming" (educational) course. Examples of incorrect test tasks and ways to improve them are provided. Extra recommendations on programming-based test's questions are given.

**Keywords: educational test, test questions, test quality, test question quality**

**References:**

1.Majorov A.N. (2001) Teoria i praktika sozdania testov dlya sistemy obrazovania. (Kak vibirat, sozdavat i ispolzovat testy dlya celey obrazovania). Moscow: Intelekt-center.

2.Zvonnikov V.I., Chelishkova M.B. (2009) Sovremennie sredstva ocenivania resultatov testirovania. Moscow: Izdatelskiy center "Academia".

3.Podbelskiy V.V., Maksimenkova O.V. (2011) Razrabotka testov po programmirovaniu dlya testirovania v komputernoy forme. Informatizacia obrazovania 2011 - Materialy mejdunarodnoy nauchno-prakticheskoy konferencii. ESU im. Bunina, 192-195.

4.Burlachuk L.F., Morozov S.M. (2008) Slovar-sparvochnik po psichodiagnostike. Sankt-Peterburg: Piter.

5.Kardanova E.U. (2008) Modelirovanie i parametrizacia testov: osnovi teorii i prilojenia. Moscow: FSU "Federalniy center testirovania".

6.C. McKenna, J. Bull. (1999). "Designing effective objective test questions: an introductory workshop". Retrieved February, 2, 2011 from CAA Centre web-site: http://www.caacentre.ac.uk/dldocs/otghdout.pdf

7.P. Anderson, G. Morgan. "National assessments of educational achievement, volume 2: Developing tests and questionnaires for a national assessment of edu-

cational achievement, World Bank Publications, 2008. - 288 p.

8.S.J. Osterfind, "Constructing Test Items : Multiple-Choice, Constructed-Response, Performance and Other Formats", Kluwer academic publishers, 1997. - 352 p.

9.V.V. Podbelskiy, O.V. Maksimenkova. "Programming as a part of the Software Engineering education" // Proceedings of the 4-th Spring/Summer Young Researchers' Colloquium on Software Engineering (SYRCoSE 2010), 2010. 165 - 168

10.V.V. Podbelskiy, O.V. Maksimenkova. "Educational tests in "Programming" academic subject development" // SYRCoSE 2011. Proceeding of the 5-th Spring/Summer Young Reseachers' Colloquium on Software Engineering, 2011. 88-93

11.Alabama professional development modules. Retrieved May, 15, 2012 from alabamapepe.com web-site: http://www.alabamapepe.com/profdevmodule/index.htm

12.Maksimenkova O.V., Podbelskiy V.V. (2012). Opit razrabotky testov i analiz rezultatov testirovania po programmirovaniu. // Materialy X otkritoy vserossiyskoy konferencii. Moscow: MSU im. M.V. Lomonosova, 367-368

***Podbelskiy Vadim Valeryevich**,*
*Doctor of Sciences, professor.*
*NRU HSE, Software Management Department.*
*vpodbelskiy@hse.ru*

***Maksimenkova Olga Veniamynovna**,*
*Senior Lecturer, NRU HSE,*
*Software Management Department.*
*e-mail: omaksimenkova@hse.ru*

А.С. Гузенкова

# ПОДГОТОВКА СПЕЦИАЛИСТОВ В ОБЛАСТИ ЭКОЛОГИЧЕСКОГО МЕНЕДЖМЕНТА

В статье рассматриваются вопросы подготовки специалистов в области экологического менеджмента. С одной стороны, цель подготовки специалистов в этой области - систематизировать подходы к предотвращению возникновения экологических проблем, с другой стороны, сократить издержки, вызванные нерациональным использованием энергии, природных ресурсов, сырья и материалов. Востребованными становятся специалисты, получившие подготовку в области менеджмента и в сфере технологии (как технологии производства, так и промышленной экологии). В статье рассмотрена концепция устойчивого развития и стандарты серии 14000, позволяющие преобразовать глобальную тенденцию устойчивого развития в практические действия.

**Ключевые слова: устойчивое развитие, подготовка специалистов, экопроектирование, экологический менеджмент, стандарты ИСО**

В июне 2012 года прошла конференция Рио+20 по устойчивому развитию. Концепция устойчивого развития предполагает экономический рост при соблюдении условий сохранения экологического равновесия и достижения социальных целей, и является отражением принципа системности управления на всех уровнях.

Математическое выражение IPAT (*Influence - Population - Afluence - Technology*) связывает экологическое воздействие человеческой деятельности с численностью населения и валовым внутренним продуктом (ВВП) - мерой промышленной и экономической активности:

$$\begin{bmatrix}\text{Экологическое}\\ \text{воздействие}\end{bmatrix} = \begin{bmatrix}\text{Численность}\\ \text{населения}\end{bmatrix} \bullet \begin{bmatrix}\text{ВВП}\\ \text{на душу}\\ \text{населения}\end{bmatrix} \bullet \begin{bmatrix}\text{Экологическое}\\ \text{воздействие}\\ \hline \text{единица}\\ \text{ВВП}\end{bmatrix}.$$

Так как цель устойчивого развития - ограничение воздействия человечества на окружающую среду его современными рамками, то необходимо рассмотреть возможные тенденции изменения всех трех членов уравнения во времени.

Первый член - рост численности населения - определяется главным образом не технологически ми, а социальными аспектами. И хотя различные страны и культуры подходят к решению этой проблемы по-разному, тенденция к его росту явно выражена. Второй член - ВВП на душу населения - существенно различается по разным странам и регионам в соответствии с экономическими условиями, стадиями исторического и уровнями технологического развития. Общая тенденция его изменения в целом положительна. Третий член - степень воздействия на окружающую среду на единицу ВВП - представляет технологическую величину, однако социальные и экономические аспекты ограничивают его существенные изменения, но именно этот член уравнения дает наибольшие перспективы перехода к устойчивому развитию, изменения его находятся среди центральных принципов промышленной экологии, а также внедряемых на предприятиях систем экологического менеджмента.

В итоговом документе третьей конференции по устойчивому развитию под названием "Будущее, которого мы хотим", подчеркивается, что "обеспечение устойчивого развития будет зависеть от активного вовлечения как государственного, так и частного сектора".