

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ  
«САМАРСКИЙ ГОСУДАРСТВЕННЫЙ АЭРОКОСМИЧЕСКИЙ  
УНИВЕРСИТЕТ имени академика С.П.КОРОЛЕВА  
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)»

**Языки и методы программирования**

*Электронные тесты  
промежуточного контроля знаний*

2011

Составители: МЯСНИКОВ Евгений Валерьевич

Электронные тесты промежуточного контроля знаний предназначены для студентов факультета информатики, направление 010400 – Прикладная математика и информатика, бакалавриат (010400.62)/магистратура (010400.68, магистерская программа – Технологии параллельного программирования и суперкомпьютинг).

## Тема 1. Основы программирования на C++

Вопрос: Укажите ошибку в коде

```
double max( double* a, int cnt )
{
    double m = a[0];
    for ( ; cnt; cnt--)
        if ( a[cnt-1] > m ) m = a[cnt-1];
    return m;
}
```

- 1) В цикле отсутствует секция инициализации
- 2) Неверное условие цикла
- 3) Счетчик цикла уменьшается, а не увеличивается
- 4) Вместо a[cnt-1] должно быть указано a[cnt]
- 5) Нет ошибки

Вопрос: Укажите ошибку в коде

```
int indmax( double* a, int cnt )
{
    int m;
    for ( m = 0; cnt; cnt--)
        if ( a[cnt-1] > a[m] ) m = cnt-1;
    return m;
}
```

- 1) Переменная m должна быть проинициализирована при объявлении
- 2) Неверно выбран тип возвращаемого значения
- 3) Неверно выбран тип переменной m
- 4) Счетчик цикла не проинициализирован, вместо него в цикле инициализируется другая переменная
- 5) Счетчик цикла уменьшается, а не увеличивается
- 6) Вместо выражения cnt-1 должно быть указано cnt
- 7) Нет ошибки

Вопрос: Укажите ошибку в коде

```
double avg( double* a, int cnt )
```

```

{
    double s = 0;
    for (int i = 0; i != cnt; i++) s += a[i];
    return s/cnt;
}

```

- 1) Неверное условие цикла. Должно быть  $i \leq cnt$
- 2) В цикле вместо переменной  $i$  нужно изменять  $cnt$
- 3) Необходимо привести  $cnt$  к типу `double`:  $s/(double)cnt$
- 4) Результат суммирования нужно делить на  $(cnt-1)$ :  $s/(cnt-1)$
- 5) Нет ошибки

**Вопрос: Укажите ошибку в коде**

```

int find( double* a, int cnt, double val )
{
    for (cnt--; cnt != -1; cnt--)
        if ( a[cnt] == val ) return cnt;
    return -1;
}

```

- 1) Неверно выбран тип возвращаемого значения
- 2) Ошибка в заголовке цикла – дважды указано выражение `cnt--`
- 3) Ошибка в условии цикла: `cnt != -1`
- 4) Неверное условие в операторе `if`
- 5) Оператор `return` вызывается в цикле
- 6) Возвращаемое в результате значение `-1` неверно
- 7) Нет ошибки

**Вопрос: Вычислите значение выражения**

**$(2\&1)?(2>1?1:2):(2|1?3:4)$**

- 1) Ввести верное значение

**Вопрос: Что будет выведено на экран в результате вызова функции со значением параметра, равным 3**

```

int f( int a )
{
    int r = (a > 1) ? a * f(a-1) : 1;
    printf("%d",r);
}

```

```
return r;  
}
```

1) Ввести верное значение

**Вопрос: Что будет выведено на экран в результате вызова функции со значением параметра, равным 3**

```
void g( int a )  
{  
    printf( "%d", a );  
    if (a < 8) g( a + 2 );  
}
```

1) Ввести верное значение

**Вопрос: Что будет выведено на экран в результате выполнения функции**

```
int main_()  
{  
    static s = 0;  
    for (int i = 1; i <= 3; i++) {  
        static int j = 1;  
        for (; j <= i; j++) s+=j;  
    }  
    printf("%d", s);  
    return 0;  
}
```

1) Ввести верное значение

**Вопрос: Выберите корректные объявления массива**

- 1) char ar[];
- 2) char [] ar;
- 3) char ar[5];
- 4) char [5] ar;

**Вопрос: Выберите корректные объявления массива**

- 1) char ar[]={};
- 2) char [] ar="";
- 3) char ar[]="345";
- 4) char [] ar="345";
- 5) char ar[]={3,4,5};

- 6) `char [] ar[]={3,4,5};`
- 7) `char ar[]={ '3','4','5'};`
- 8) `char []ar={ '3','4','5'};`

**Вопрос: Выберите корректные способы объявления и создания массива**

- 1) `int ar[] = new [] int;`
- 2) `int ar[] = new int[];`
- 3) `int ar[] = new [10] int;`
- 4) `int ar[] = new int[10];`
- 5) `int *ar = new [] int;`
- 6) `int *ar = new int[];`
- 7) `int *ar = new [10] int;`
- 8) `int *ar = new int[10];`

**Вопрос: Пусть сделаны следующие объявления**

`struct st1{ int a; int b[3]; } s1, s2;`

`struct st2{ int a; int b[3]; } s3;`

`struct st3{ int a; int b[2]; } s4;`

**Какие из следующих выражений корректны?**

- 1) `s1=s2;`
- 2) `s1=s3;`
- 3) `s1=s4;`
- 4) `s4=s1;`

**Вопрос: Пусть функция f объявлена следующим образом:**

`void f(int &x);`

**Укажите правильный способ передачи в нее переменной**

- 1) `int i;`
- 2) `f(i);`
- 3) `f(&i);`
- 4) `f(*i);`
- 5) `f(@i);`
- 6) `f(i*);`
- 7) `f(i&);`
- 8) `f(i@);`

**Вопрос: Пусть функция f объявлена следующим образом:**

`void f(int *x);`

**Укажите правильный способ передачи в нее переменной**

- 1) int i;
- 2) f(i);
- 3) f(&i);
- 4) f(\*i);
- 5) f(@i);
- 6) f(i\*);
- 7) f(i&);
- 8) f(i@);

**Вопрос: Что будет выведено в результате работы функции? Если в программе есть ошибки, укажите значение ноль (0).**

```
void f() {  
    int a=1,b=2,c=3;  
    int &d=b;  
    int *e=&b;  
    int *f=new int(b);  
    a=b=c;  
    printf("%d%d%d%d%d%d",a,b,c,d,*e,*f);  
    delete f;  
}
```

- 1) Ввести верное значение

**Вопрос: Выберите корректные вещественные типы**

- 1) float
- 2) double
- 3) long double
- 4) int
- 5) short
- 6) real
- 7) long real
- 8) short double

**Вопрос: Выберите корректные целые типы**

- 1) integer
- 2) unsigned
- 3) long
- 4) signed int
- 5) signed long

- 6) short
- 7) long integer
- 8) long double

**Вопрос: Выберите правильный вариант объявления переменных**

- 1) int i=10; bool b=true; char c="x";
- 2) int i(10); bool b; char c='\t';
- 3) i :int; b :bool; c :char;
- 4) int i, bool b, char c;

## **Тема 2. Классы и наследование**

**Вопрос: Пусть сделаны следующие определения:**

```
class C1 {
    public:
        C1() { printf("c;"); }
        ~C1() { printf("d;"); }
        C1(const C1& x) { printf("cc;"); }
};

void a1(C1 x)
{
    printf("a1;");
}

void f()
{
    C1 x;
    a1(x);
}
```

**Что будет выведено после выполнения функции f()?**

- 1) Ввести верное значение

**Вопрос: Пусть сделаны следующие определения:**

```
class A2 {
    public:
        void f() { printf("a;"); }
};

class B2 : public A2 {
    public:
```



```
void f() { printf("b;"); }  
};
```

**Что будет выведено после выполнения функции f()?**

```
void f()  
{  
    A2 x;  
    B2 y;  
    A2 *px = &x, *py = &y;  
    x.f();  
    y.f();  
    px->f();  
    py->f();  
}
```

1) Ввести верное значение

**Вопрос: Пусть сделаны следующие определения:**

```
class A3 {  
    public:  
        virtual void f() { printf("a;"); }  
};  
class B3 : public A3 {  
    public:  
        void f() { printf("b;"); }  
};
```

**Что будет выведено после выполнения функции f()?**

```
void f()  
{  
    A3 x; B3 y;  
    A3 *px = &y;  
    B3 *py = &y;  
  
    px->f();  
    py->f();  
    px->A3::f();  
    py->A3::f();  
}
```

- 1) Ввести верное значение

**Вопрос: Выберите верные утверждения:**

- 1) В каждом классе программистом должен быть определен конструктор
- 2) Если конструктор не определен, он будет создан автоматически
- 3) В каждом классе программистом должен быть определен конструктор копирования
- 4) Если конструктор копирования не определен, он будет создан автоматически
- 5) В каждом классе программистом должен быть определен конструктор по умолчанию
- 6) Если конструктор по умолчанию не определен, он будет создан автоматически

**Вопрос: Конструктор по умолчанию – это**

- 1) конструктор, создаваемый автоматически
- 2) конструктор без параметров
- 3) конструктор, определенный со спецификатором default
- 4) ничего из вышеперечисленного

**Вопрос: Конструктор может быть**

- 1) скрытым (private)
- 2) виртуальным
- 3) чисто виртуальным
- 4) статическим
- 5) дружественным
- 6) константным
- 7) перегруженным

**Вопрос: Деструктор может быть**

- 1) скрытым (private)
- 2) виртуальным
- 3) чисто виртуальным
- 4) статическим
- 5) дружественным
- 6) константным
- 7) перегруженным

**Вопрос: Пусть сделаны следующие определения:**

**class A**

**{**

**public:**

**A() { printf("ca;"); }**

**~A() { printf("da;"); }**

```
};  
class B : public A {  
    public:  
    B() { printf("cb;"); }  
    ~B() { printf("db;"); }  
};
```

**Что будет выведено после выполнения функции f()?**

```
void f() {  
    A *p = new B();  
    A a;  
    delete p;  
}
```

- 1) Ввести верное значение

**Вопрос: Укажите строки, в которых есть ошибки**

```
class A {  
    static int x;  
protected:  
    int y;  
public:  
    static void SetX( int v ) { x = v; } // 1  
    static void SetY( int v ) { y = v; } //2  
public: // 3  
    int GetX() { return x; } // 4  
    int GetY() { return y; } // 5  
};
```

- 1) 1
- 2) 2
- 3) 3
- 4) 4

### **Тема 3. Перегрузка операций и исключения**

**Вопрос: Перегружаемая в классе операция может быть объявлена виртуальной?**

- 1) Да
- 2) Нет

**Вопрос: Перегружаемая в классе операция может быть объявлена чисто виртуальной?**

- 1) Да
- 2) Нет

**Вопрос: Пусть в программе определен класс C. Укажите, каким образом может быть выполнена перегрузка операции сложения вне класса C?**

- 1) `int operator +(const C& x, const C& y);`
- 2) `int operator +(const C& x, int y);`
- 3) `int operator +(int x, const C& y);`
- 4) `int operator +(int x, int y);`
- 5) `C operator +(const C& x, const C& y);`
- 6) `C operator +(const C& x, int y);`
- 7) `C operator +(int x, const C& y);`
- 8) `C operator +(int x, int y);`

**Вопрос: Пусть в программе определен класс C. Укажите, каким образом может быть выполнена перегрузка операции сложения с использованием нестатической функции-члена класса C?**

- 1) `int operator +(const C& x, int y);`
- 2) `int operator +(const C& x);`
- 3) `int operator +(int x);`
- 4) `C operator +(const C& x, int y);`
- 5) `C operator +(const C& x);`
- 6) `C operator +(int x);`

**Вопрос: Какие из перечисленных ниже операций могут быть перегружены только как нестатические функции-члены классов**

- 1) операция простого присваивания
- 2) составные операции присваивания
- 3) обращение по индексу
- 4) вызова функции
- 5) операции сравнения
- 6) преобразования типа (функция преобразования)
- 7) постфиксного инкремента
- 8) префиксного инкремента

**Вопрос: Сколько параметров имеет функция перегрузки постфиксной операции декремента, реализованная вне класса**

1) Ввести верное значение

**Вопрос: Что будет напечатано в результате выполнения функции?**

```
void f()
{
    try {
        cout << "1";
        try {
            cout << "2";
            throw "2";
            cout << "3";
        }
        catch ( int ) {
            cout << "4";
        }
        catch ( char* ) {
            cout << "5";
            throw;
        }
        catch ( ... ) {
            cout << "6";
            throw;
        }
    }
    catch ( int ) {
        cout << "7";
    }
    catch ( char* ) {
        cout << "8";
    }
    catch ( ... ) {
        cout << "9";
    }
}
```

}

- 1) Ввести верное значение

**Вопрос: Какие из приведенных ниже операций могут быть перегружены для класса автоматически?**

- 1) операция простого присваивания для объектов класса
- 2) операция вызова функции для вызова методов класса
- 3) операция обращения по индексу, если в составе класса объявлен массив
- 4) операция sizeof для определения размера
- 5) операция сравнения для объектов класса
- 6) операции потокового ввода-вывода

**Вопрос: Возвращает ли операция присваивания значение?**

- 1) Да
- 2) Нет

**Вопрос: Можно ли перегружать операции для встроенных типов данных?**

- 1) Да
- 2) Нет

**Вопрос: Можно ли изменить приоритет перегружаемой операции?**

- 1) Да
- 2) Нет

**Вопрос: Можно ли определить новую операцию путем перегрузки?**

- 1) Да
- 2) Нет

**Вопрос: Сколько операндов у операции обращения по индексу «[]»**

- 1) один
- 2) два
- 3) один или более
- 4) два или более
- 5) ни один из вышеперечисленных

**Вопрос: Сколько операндов у операции вызова функции «()»**

- 1) один

- 2) два
- 3) один или более
- 4) два или более
- 5) ни один из вышеперечисленных

## **Тема 4. Шаблоны**

**Вопрос: Пусть определен шаблон функции**

```
template <class T>
```

```
void fun(T a, T b)
```

**Каким образом можно вызвать его, передав в качестве параметров целочисленные переменные i и j (int i, j)?**

- 1) fun(i, j);
- 2) fun<class T>(i,j);
- 3) fun<int>(i,j);
- 4) fun<T>(i,j);
- 5) fun((int)i,(int)j);
- 6) fun((T)i,(T)j);
- 7) fun(<int>i,<int>j);
- 8) fun(<T>i,<T>j);

**Вопрос: Можно ли параметрам шаблона присваивать значения по умолчанию?**

- 1) Да
- 2) Нет

**Вопрос: Может ли в качестве параметра шаблона класса выступать другой шаблон?**

- 1) Да
- 2) Нет

**Вопрос: Специализация может быть выполнена**

- 1) для метода класса
- 2) для класса целиком
- 3) как для метода, так и класса целиком
- 4) Ни один из перечисленных выше вариантов

**Вопрос: Задать особое поведение шаблонной функции для определенных типов данных можно**

- 1) Выполнив специализацию шаблона
- 2) Выполнив перегрузку функции
- 3) Выполнив специализацию шаблона или перегрузку функции
- 4) Ни один из перечисленных выше вариантов

**Вопрос: Выберите верные утверждения:**

- 1) Шаблон класса может быть наследником обычного класса
- 2) Шаблон класса может быть наследником шаблона класса
- 3) Обычный класс может быть наследником шаблона класса

**Вопрос: Пусть определен шаблон класса**

```
template <class T1, class T2> class TClass { /*...*/ };
```

**Каким образом можно создать объект такого шаблона класса?**

- 1) TClass x;
- 2) TClass<int,double> x;
- 3) TClass <T1,T2>x;
- 4) TClass x<int,double>;
- 5) TClass x<T1,T2>;
- 6) TClass x(int,double);
- 7) TClass x(T1,T2);
- 8) int i; double d; TClass x(i,d);