

Ответы на вопросы экзамена по курсу «Языки программирования» 19.01.2015

В ответах курсивом выделены необязательные пояснения, которые можно опустить (особенно на экзамене)

Вариант 1

Задача 1-1

Что означает операция «>>>>» в языке Java? Почему её нет в языках C/C++, C#?

Ответ

Операция «>>>>» в языке Java означает побитовый логический сдвиг влево. В языках C, C++, C# есть только операция «>>», которая в зависимости от типа левого операнда означает либо арифметический сдвиг (знаковый операнд), либо логический (беззнаковый операнд). Поскольку в Java нет беззнаковых типов (кроме byte), то операция «>>» может означать только арифметический сдвиг. *В то же время операция логического сдвига необходима для эффективного выполнения ряда критически важных действий, поэтому ее добавили в язык со специальным обозначением «>>>>».*

Задача 1-2

Напишите спецификацию шаблона `template <int N> struct Fun {...};` на языке C++ так, чтобы значение `Fun<N>::result` было равно сумме квадратов целых чисел от 1 до N для любого неотрицательного целого константного значения N.

Ответ

Поскольку речь идет о шаблоне, то подразумевается, что вычисления производятся статически (во время трансляции). Текст программы (в ответе достаточно было описания шаблонов):

```
#include <iostream>
template <int N> struct Fun
{
    enum {
        result = Fun<N-1>::result + N*N
        // используем конкретизацию этого же шаблона
    };
};

// полная специализация для значения N=1
template <> struct Fun<1>
{
    enum {
        result = 1
    };
};

int main()
{
    std::cout << Fun<10>::result << std::endl;
    return 0;
}
```

Задача 1-3

В каких из перечисленных ниже языков можно объявлять приватные абстрактные функции? Если можно, то для каждого языка приведите пример.

Си++, Java, С#, Оберон, Оберон-2

Ответ

Оберон, Оберон-2 не подходят, так как в них нет такой языковой конструкции, как абстрактный класс. Заметим, что в Си++, Java, С# абстрактные функции — виртуальны (имеют динамическое связывание) и должны быть замещены в потомке. Однако в С# любые виртуальные функции (не только абстрактные) не могут быть приватными.

Что касается Java, то в этом языке приватные члены являются **НЕВИДИМЫМИ** в производных классах (то есть работают правила видимости, а не доступа). Поэтому приватная функция не может быть замещена в производном классе, отсюда приватные абстрактные функции невозможны, и компилятор языка выдает ошибку в этом случае.

Остается только С++. В этом языке можно объявлять приватные виртуальные функции, которые можно замещать в производных классах. Они отличаются от защищенных виртуальных тем, что приватные функции нельзя вызывать в производных классах (но можно — в самом абстрактном классе). *Необходимость в таком механизме достаточно редка, но он допустим.* Так что приватные виртуальные функции разрешены только в С++ (из перечисленных языков). *Вот где они практически бессмысленны, так это в интерфейсах (где нет конкретных функций).*

Задача 1-4

Напишите определение класса Fib на языке С# так, чтобы следующий фрагмент программы выдавал первые 10 чисел Фибоначчи ($Fib_1 = Fib_2 = 1$, $Fib_n = Fib_{n-1} + Fib_{n-2}$):

```
foreach (var f in new Fib(100)) System.Console.WriteLine(f);
```

В оригинале вместо 10 было 100 (без комментариев)...

Ответ

Текст программы (в ответе достаточно было только описания класса Fib):

```
using System.Collections;

namespace Test
{
    class Fib : IEnumerable
    {
        int _n;
        public Fib(int n)
        {
            _n = n;
        }
        public IEnumerator GetEnumerator()
        {
            int a = 1; int b = 1;
            for (int i = 1; i <= _n; i++)
            {
                yield return a;
                int c = a + b;
                a = b;
                b = c;
            }
        }
    }
}
```

```

    }
}
class Program
{
    static void Main(string[] args)
    {
        foreach (var x in new Fib(10))
            System.Console.WriteLine(x);
    }
}
}

```

Задача 1-5

Дайте общее определение операции индексирования в языке программирования. Напишите пример пользовательского переопределения операции индексирования в языке C++ (определение операции и ее вызов). Есть ли аналогичные средства в языке C#? Если есть, то приведите пример на этом языке, аналогичный приведенному для C++.

Ответ

Операция индексирования – это функция:

```

[]: (ContainerType, IndexType) => ElementRefType

```

Применима к контейнерным типам ContainerType (прежде всего - массивы), по индексу типа IndexType (обычно целочисленный, начиная с 0), возвращает тип ссылки на элемент контейнера.

Переопределение в C++ (все, что нежирным шрифтом — в ответе необязательно):

```

class Vector {
    int *body;
    int size;

public:
    Vector (int sz)
    {
        body = new int[size = sz];
    }
    Vector (const Vector & v)
    {
        body = new int [size = v.size];
        for (int i = 0; i < size; i++)
            body[i] = v.body[i];
    }
    Vector &operator = (const Vector& v)
    {
        if (this != &v) {
            delete [] body;
            body = new int [S size = v.size];
        }
    }
}

```

```

        for (int i = 0; i < size; i++)
            body[i] = v.body[i];
    }
    return *this;
}
~Vector() { delete [] body; }
int& operator[] (int index) { return body[index]; }
};

```

Пример вызова:

```

Vector v(20);
for (int i = 0; i < 20; i++)
    v[i] = i*i;

```

В языке C# операции, не являющиеся арифметико-логическими, переопределять нельзя, однако специально для возможности переопределения семантики индексирования туда ввели специальную языковую конструкцию — индексатор. Пример индексатора:

```

class ContainerType
{
    private Dictionary<string, int> _dic = new Dictionary<string, int>();
    public int this[string k]
    {
        get { return _dic[k.ToLower()]; }
        set { _dic.Add(k.ToLower(), value); }
    }
}

```

Задача 1-6

Дайте определение и пример использования анонимной (лямбда-функции) в языке Python. В каких языках из перечисленных ниже есть аналогичное понятие? Для каждого такого языка приведите пример.

JavaScript, Ада, Си++, Оберон, Оберон-2

Ответ

Лямбда-функция — это безымянная функция, тело которой — выражение, которое возвращается как результат функции:

```
lambda arguments : expression
```

Выражение может включать в себя не только аргументы, но и доступные переменные из окружающих областей видимости, которые «захватываются» лямбда-функцией и составляют так называемое «замыкание».

Пример:

```
r = [1, 2, 3, 4, 5]
```

применяем лямбда-функцию к списку r

```
r1 = map(lambda x: x*x, r)
for x in r1:
    print(x)
```

Вывод (список квадратов):

```
1
4
9
16
25
```

Лямбда-функции есть также (из перечисленных в условии языков) в JavaScript, Си++.

JavaScript:

```
f = function(x) { return x*x; }
```

Си++:

```
// определим лямбда-функцию и немедленно применим ее к 5
auto twentyfive = [](int x) {return x*x; }(5);
```

Задача 1-7

Дайте определение понятия «защищенный член класса» в языке Java. Перечислите его отличия от аналогичного понятия в языке C#.

Ответ

«Защищенный член класса» в языке Java — это член, помеченный атрибутом доступа **protected**.

Защищенный член класса X:

1. Доступен из всех функций этого же класса
2. Доступен из всех функций производных классов через ссылку `this`
3. Доступен из функций-членов классов пакета, где находится класс X

В языке C# защищенность означает то же, что и 1. и 2., только без ограничения на ссылку. *Объяснять, что значит «доступность через ссылку this не надо (так как долго и не в формате этого экзамена). Желающие могут прочитать в конспектах лекций этого года (if any), либо, например, в книге Арнольда, Гослинга, Холмса (3-е издание) в п.3.5 «Что в действительности означает protected»*

Задача 1-8

Задача №8 имела 2 варианта. Первый:

Что будет напечатано в результате работы следующей программы на C#?

```
using System;
namespace test
{
    class Program {
        static void Main() {
```

```

        B a = new B(), b = new B(); P(a, b);
        Console.WriteLine("-----");
        C c = new C(), d = new C(); P(c, d);
    }
    static public void P(A a, B b) {
        a.f(); a.g();
        b.f(); b.g();
    }
}
class A {
    public void f() { Console.WriteLine("A.f"); g(); }
    public virtual void g() { Console.WriteLine("A.g"); }
}
class B : A {
    public virtual new void f() { Console.WriteLine("B.f"); }
    public override void g() { Console.WriteLine("B.g"); f(); }
}
class C : B {
    public override void f() { Console.WriteLine("C.f"); }
    public override void g() { Console.WriteLine("C.g"); f(); }
}
}

```

Ответ

A.f
 B.g
 B.f
 B.g
 B.f
 B.f
 B.g
 B.f

A.f
 C.g
 C.f
 C.g
 C.f
 C.f
 C.g
 C.f

Второй вариант задачи №8:

Что будет напечатано в результате работы следующей программы на C#?

```

using System;
namespace test
{
    class Program {
        static void Main() {
            B a = new B(), b = new B(); P(a, b);
            Console.WriteLine("-----");
            C c = new C(), d = new C(); P(c, d);
        }
        static public void P(A a, B b) {
            a.f(); a.g();
            b.f(); b.g();
        }
    }
}

```

```

    }
}
class A {
    public void f() { Console.WriteLine("A.f"); g(); }
    public virtual void g() { Console.WriteLine("A.g"); }
}
class B : A {
    public void f() { Console.WriteLine("B.f"); }
    public override void g() { Console.WriteLine("B.g"); f(); }
}
class C : B {
    public void f() { Console.WriteLine("C.f"); }
    public override void g() { Console.WriteLine("C.g"); f(); }
}
}

```

Ombem

A.f
 B.g
 B.f
 B.g
 B.f
 B.f
 B.g
 B.f

A.f
 C.g
 C.f
 C.g
 C.f
 B.f
 C.g
 C.f