

Ответы на вопросы экзамена по курсу «Языки программирования» 10.01.2018

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

Вариант 1

Задача 1-1

Замените знаки вопроса (????) в приведенной ниже программе на языке C# так, чтобы в стандартный вывод выдавалась сумма квадратов элементов списка l. Никакой код куда-либо еще добавлять нельзя. Примечание: метод `ForEach` списка имеет прототип `List<int>.ForEach(Action<int> action)` и выполняет действие `action` для каждого элемента списка.

```
List<int> l = new List<int>() { 1, 2, 3, 5, 7, 9 };
int s = 0;
l.ForEach(????);
System.Console.WriteLine(s);
```

Ответ

Нужна либо лямбда-функция `x => s += x*x`, либо анонимный делегат `void delegate(int x) { s += x*x; }`

Задача 1-2

Пусть `s` описана как переменная строкового типа (`string`), и в программе есть присваивание `s = "кафе"` (все буквы - кириллические). Какова будет длина этой строки в языках C++ (`s.size()`), Java (`s.length`), Go (`len(s)`), Swift (`s.count`)? Для каждого языка дайте ответ и объясните различия (если они есть).

Ответ

Для правильного ответа на внешне простой вопрос нужно понимать следующие моменты:

1. В каких единицах (какой тип?) измеряется длина в языке (точнее — в соответствующих функциях и свойствах).

C++ - в данном случае — `char`, так как имеется в виду `string` (`basic_string<char>`). Если бы был тип `wstring` (`basic_string<wchar_t>`) и литерал `L"кафе"`, тогда — `wchar_t`.

Java – `char`

Go – `byte` (“строка Go – это последовательность байтов”) - там типа `char` вовсе нет!

Swift – `Character`

2. В какой кодировке хранятся строки.

C++ - не определено языком. Последовательность байтов-символов в литерале “кафе” определяется исключительно кодировкой исходного текста программы. Если вы готовили текст в однобайтовой кодировке, то длина текста — 4. Если же кодировка текста — UTF-8 – то длина строки — 8, так как каждая кириллическая буква представлена 2 байтами UTF-8. У каждого транслятора есть нестандартизированные опции и прагмы,

задающие кодировку исходного текста. *Я как-то работал с программой в однобайтовой кодировке русских букв, в которой кодировка нескольких файлов случайно и незаметно изменилась на UTF-8. Прежде чем я разобрался, почему программа «полетела» прошло немало времени...*

Java (и C#) - кодировка UTF-16, каждый символ — это двухбайтный код, но свойство `s.length` считает СИМВОЛЫ, а не байты, поэтому длина — 4.

Swift — как хранятся символы — определяется реализацией. Каждый символ представляет собой `extended grapheme cluster` из Unicode-стандарта, то есть означает ровно одно знакоместо для изображения Юникодовского символа. Если программиста интересует представление строки в кодировке UTF-8/16/32, то есть соответствующие свойства строки (Views), которые возвращают массив `UByte8/UInt16/UInt32`. А длина считается именно в символах и НИКАК не зависит от кодировки. Поэтому длина — 4.

Go — кодировка ТОЛЬКО UTF-8, поэтому длина — 4.

Короткий ответ: C++ - не определено и зависит от кодировки исходного текста, Java, C#,Swift – 4, Go – 8.

Задача 1-3

Объясните, допустимо ли присваивание `aobj = afoo` в следующем фрагменте программы на Java. Если допустимо, то какие проблемы с `aobj` и `afoo` могут возникнуть в дальнейшем при выполнении программы?

```
class Foo { public void foo () {} }
class Bar { public void bar () {} }
Object [] aobj; Foo [] afoo = new Foo[10];
aobj = afoo;
```

Ответ

Язык Java допускает такое присваивание. То, что после него нельзя вызывать `aobj.foo()` - это не проблема, а свойство любого ОО-языка. Проблема в том, что `aobj` нельзя модифицировать так, как это следовало бы из его объявления. Ведь если забыть об этом присваивании, то можно в `aobj` положить любой объект. Это ведь массив из просто объектов. Но тут надо вспомнить об объектно-референциальной модели языка Java. После присваивания `aobj` ссылается на объект-массив, который является ОДНОРОДНЫМ и состоит только из объектов типа `Foo` или производных от него, поэтому попытка положить в него что-то другое (`aobj[0] = new Object()` или `aobj[0] = new Bar()`) приводит к «аварии» - возбуждению исключения `ArrayStoreException`.

Заметим, что для динамических массивов `ArrayList` принята логичная точка зрения, что списки из разных (неважно, ковариантных или контравариантных) типов всегда инвариантны.

Задача 1-4

Написать на языке C++ шаблонную функцию `int MakeWord()`, которая возвращает целое число, являющееся побитовой суммой (операция суммирования — побитовое «или» `|`) аргументов шаблона. Аргументы шаблона должны быть целочисленными.

Например `std::cout << MakeWord<1,2>();` должно выдать 3, а `std::cout << MakeWord<2,4,9>();` должно выдать 15.

Ответ

Тут **ОБЯЗАТЕЛЕН** вариативный шаблон.

C++17:

```
template <int ... Bits> int MakeWord()
{
    return (Bits | ...); // скобки обязательны!!!
}
```

Задача 1-5

Являются ли реализации обобщенных классов на языке Java более эффективными, чем их необобщенные аналоги. Например, будет ли реализация обобщенного класса `Stack<Integer>` более эффективной, чем реализация класса `Stack` из `Object`?

Ответ

Нет. Поскольку в Java происходит «стирание информации о типе» (type erasure), то код, который генерируется (причем только один раз!) - так называемый raw-код — не сильно отличается от кода непараметрических классов, использующих `Object`, включая проверки типа, упаковку и распаковку и т. п. Поэтому увеличения эффективности, как в случае `C#` и особенно `C++`, не происходит.

Задача 1-6

Напишите какой-нибудь пример генератора (но не сопрограммы) на языке Python. Чем сопрограммы языка Python отличаются от простых генераторов в этом языке?

Ответ

```
def NN(n):
    for i in range(1, n+1):
        yield i*i
```

Сопрограммы Python (**не путать с сопрограммами Кнута!!!**) не просто генераторы, а генераторы, которые ПОЛУЧАЮТ информацию от вызывающей программы с помощью `yield`-выражения (в сопрограмме) и метода `send` (в вызывающей программе):

```
x = (yield)
```

Не путайте его с `yield`-оператором из примера, который ОТДАЕТ информацию (вместе с управлением) в вызываемую программу. Сопрограммы Python однопоточны (*желающие могут почитать про `Global Interpreter Lock`*). Есть и асинхронные расширения современного Питона, но мы их на лекциях не рассматривали.

Замечание: конструкции типа $l = [x \text{ for } x \text{ in } \text{range}(0, N)]$ иногда называют генераторами списка, но это неточно. По-английски эта конструкция называется “list comprehension”, а не «generator».

Задача 1-7

Дайте определение вырезки из массива. Приведите примеры вырезок в языке Python и Go. Чем они отличаются?

Ответ

Вырезка (другое название — срез) — это конструкция для работы с подмножеством элементов массива. Конкретнее сказать нельзя, поскольку языки различаются.

Python:

```
array = [1,2,3,4,5,6,7] # массив  
slc = array[1:3] # срез  
и т.д.
```

Go:

```
primes := [6]int{2, 3, 5, 7, 11, 13} // массив  
var s []int = primes[1:4] // срез
```

Главное отличие — срез Python создает новый массив, копируя элементы из базового, а срез Go – это «окно» со ссылками на базовый массив. Нового массива не создается.

Задача 1-8

В приведенной ниже программе на языке Go есть ошибка, которая проявляется во время выполнения. В чем она заключается, и как ее исправить (указание — надо переставить одну строку в другое место)?

Что будет выдано после исправления?

```
package main; import "fmt"  
func fibonacci(c, quit chan int) {  
    x, y := 0, 1  
    for {  
        select {  
        case c <- x:  
            x, y = y, x+y  
        case <-quit:  
            fmt.Println("quit")  
            return  
        }  
    }  
}  
  
func main() {  
    c := make(chan int)  
    quit := make(chan int)  
    go func() {  
        for i := 0; i < 10; i++ {  
            fmt.Print(<-c);fmt.Print(" ")  
        }  
    }()  
    quit <- 0  
    fibonacci(c, quit)  
}
```

Ответ

Ошибка состоит в том, образуется «тупик». Анонимная го-программа пытается считать из канала с 10 чисел, но зависает на первом чтении из канала. Главная программа после запуска анонимной го-программы пишет в quit нуль, но так как из канала никто не читает, а его емкость — 0, то главная программа зависает на записи. А функция fibonacci вообще не запускается. Типичный тупик. Надо, чтобы го-программа писала в канал сразу по завершении, то есть поднять quit <- 0 на строчку вверх. После этого будут выданы 10 чисел Фибоначчи: 0 1 1 2 3 5 8 13 21 34 quit

Замечание. Можно скопировать текст программы и запустить ее на сайте play.golang.org. Там можно проверить другие варианты ответов (и убедиться в их неправильности).