

# Java Basics

## Questionnaires

Olga Smolyakova

1. Укажите имя файла, в которых могут размещаться следующие классы. Объясните свой выбор.

a) **public class X{}** X.java - потому что имя файла должно совпадать с именем класса

b) **class Y{}** Y.java - потому что имя файла должно совпадать с именем класса

2. Впишите в класс MyClass метод main и объявите, что данный класс лежит в пакете by.epam.tr.check

```
package by.epam.tr.check;

public class MyClass{
    public static void main (String[] args) {
    }

}
```

3. Напишите недостающий код в пример, решающий задачу нахождения наименьшего из двух чисел.

```
int x;
int y;
int min;
```

```
Scanner sc = new Scanner(System.in);
```

```
System.out.print("Введите x: ");
x = sc.nextInt();
```

```
System.out.print("Введите y: ");
y = sc.nextInt();
```

```
if (x>y) {
    min=y;
} else {
    min=x;
}
```

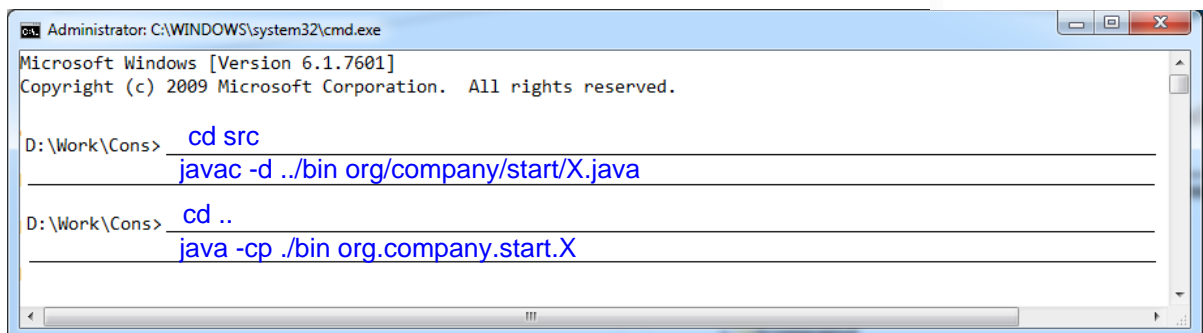
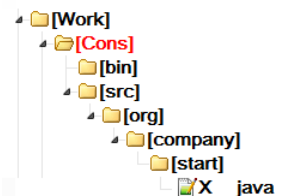
```
System.out.println("min = " + min);
```

4. Напишите компилируемый класс, выводящий на консоль числа от 1 до 10.

```
public class Task {  
    public static void main(String[] args) {  
        int i;  
        for (i=1; i<11; i++) {  
            System.out.println(i);  
        }  
    }  
}
```

5. Дан public класс X, лежащий в пакете org.company.start

Запишите команды компиляции и запуска этого класса из-под командной строки.



6. Исправьте ошибки в коде так, чтобы программа могла быть скомпилирована и запущена.

```
public class Calc {  
    public static void main(String[] args) {  
        double x = 3.4;  
        double y = 5.6;  
        double z = 7.8;  
  
        double result;  
  
        result = func(x, y, z);  
  
        System.out.println("result = " + result);  
    }  
  
    (double x, double y, double z)  
    public static double func(int a, int b, int c) {  
        return a+b+c;  
        x+y+z;  
    }  
}
```

7. Запишите результат выполнения кода, данного ниже. Объясните результат.

```
String str1 = "string1";  
String str2 = new String("string1");  
String str3 = "string1";  
  
if(str1 == str2){  
    System.out.print("a");  
}else{  
    System.out.print("b");  
}  
  
if(str1 == str3){  
    System.out.print("c");  
}else{  
    System.out.print("d");  
}
```

Результат: bc

Объяснение: (str1 == str2) - здесь сравниваются ссылка на объект и аргумент-ссылка, поэтому по условию if на консоль выводится "b"

(str1 == str3) - здесь сравниваются две аргумент-ссылки, поэтому по условию if на консоль выводится "c"

---

---

---

8. В классе Calc дан метод extent. Напишите для этого метода junit-тест, проверяющий его работу при корректных параметрах.

```
public class Calc {  
    public static double extent(double x, double y){  
        double result = 1;  
  
        if (y == 0){  
            return 1;  
        }  
        for(int i=0; i<Math.abs(y); i++){  
            result = result*x;  
        }  
        if(y<0){  
            result = 1 / result;  
        }  
        return result;  
    }  
}
```

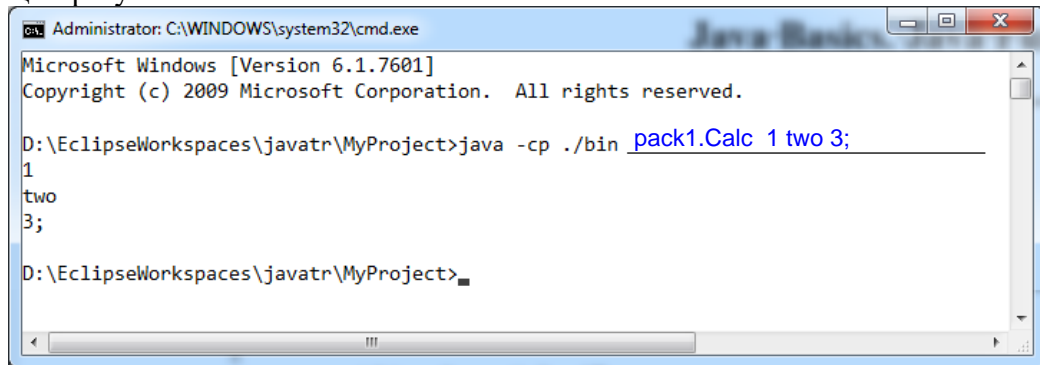
```
import org.junit.Assert;  
import org.junit.Test;  
  
public class CalcTest {  
    @Test  
    public void extent001() {  
        double x = 3;  
        double y = 2;  
        double realResult;  
        double expectedResult = 9.0;  
        realResult = Calc.extent(x,y);  
        Assert.assertEquals(expectedResult, realResult, 0.00000001);  
    }  
}
```

9. Дан код.

```
package pack1;

public class Calc {
    public static void main(String[] args) {
        for (int i = 0; i < args.length; i++) {
            System.out.println(args[i]);
        }
    }
}
```

Напишите команду запуска этого класса из-под консоли так, чтобы получился следующий результат.



10. Какой параметр нужно передать в метод `valueOf`, чтобы при запуске на консоль вывелось сообщение "blue"?

```
public class Main {

    public static void main(String[] args) {
        Color c;
        c = Color.valueOf(_____"BLUE"_____);
        switch(c){
            case RED:
                System.out.println("red");
                break;
            case GREEN:
                System.out.println("green");
                break;
            case BLUE:
                System.out.println("blue");
                break;
            case YELLOW:
                System.out.println("yellow");
                break;
        }
    }
}
```

11.Допиште код, позволяющий найти сумму элементов в массиве.

```
public class Main {  
  
    public static void main(String[] args) {  
        int[] mas = new int[]{23,54,12,76,45,98,12,56};  
        int sum;  
        sum = 0;  
        for (int i = 0; i < mas.length; i++) {  
            sum +=mas[i];  
        }  
  
  
  
  
        System.out.println("sum = " + sum);  
    }  
}
```

12. В чем разница при передаче параметров для методов  $x$  и  $y$ ? Дайте развернутое объяснение.

```
public class Main {
    public void x(int x){}
    public void y(Date d){}
}
```

public void x(int x) - тут передаётся текущее значение переменной с примитивным типом данных, т.е. копию текущего значения x

public void y(Date d) - тут передаётся значение ссылки на объект, т.е. копию ссылки на объект

13. Напишите код, создающий и инициализирующий массив следующего вида.


```
int array2D[][] = {  
    {1,2},  
    {3,4,5},  
    {5,6},  
    {7,8,9,1,2},  
};
```

14. Создайте класс так, чтобы его объекты можно было создать приведенными ниже способами:

```
Book b1 = new Book("Java", "Blinov", 2015);  
Book b2 = new Book("Java", "Blinov");  
Book b3 = new Book(20000.0);
```

```
public class Book {  
    private String title;  
    private String author;  
    private int price;  
    private int yearPublished;  
  
    public Book(String title, String author, int yearPublished) {  
        this.title = title;  
        this.author = author;  
        this.yearPublished = yearPublished;  
    }  
  
    public Book(String title, String author) {  
        this.title = title;  
        this.author = author;  
    }  
  
    public Book(int price) {  
        this.price = price;  
    }  
}
```



15. Дан класс.

```
public class Main {  
}
```

Создайте объект этого класса и объясните свое решение.

```
Main maintest = new Main();
```

Создан объект типа Main с именем maintest класса мейн, который дан выше.  
Если в классе не определить ни одного конструктора, то компилятор добавляет в класс конструктор по умолчанию (без параметров и с пустым телом).

16. Перечислите модификаторы довтупа в java и укажите их характеристики.

private-package (friendly), public, private и protected

private-package (friendly) - устанавливается по умолчанию, доступ будет разрешён только в рамках пакета

public - доступ будет разрешён внутри класса и вне класса, в том числе и в производных классах

private - доступ будет только внутри класса и запрещён вне класса

protected - доступ будет разрешён из любого подкласса и запрещён извне класса

17. В чем разница при передаче параметров для методов *x* и *y*? Дайте развернутое объяснение.

```
public class Main {  
    public void x(int x){}  
    public void y(Date d){}  
}
```

`public void x(int x)` - тут передаётся текущее значение переменной с примитивным типом данных, т.е. копию текущего значения *x*

`public void y(Date d)` - тут передаётся значение ссылки на объект, т.е. копию ссылки на объект

18. Дан класс. Исправьте в нем ошибки так, чтобы код класса мог быть скомпилирован. Объясните свои исправления.

```
public class Note {  
    private String note;  
    private Data data;  
  
    public Note(){  
        this("", null);  
    }  
  
    public Note(String note, Data data){  
        this.note = note;  
        this.data = data;  
    }  
}  
  
class Data{  
}
```

19. Дайте определение понятиям класс и объект.

класс - тип данных который сожержит описание состояния и поведения создаваемых от него объектов, это как идея или описание некой сущности

объект - экземпляр или некая сущность которую можно создать на основании класса

20. Реализуйте для класса A шаблон Singleton. Объясните, почему при применении шаблона Singleton можно создать только единственный объект класса A.

```
public class A {  
    private static final A instance = new A();  
    private A() {}  
    public static A getInstance() {  
        return instance;  
    }  
}
```

Из-за модификатора private - доступ будет только внутри класса и запрещён вне класса. При попытке повторно создать объект A возвращается ссылка на уже существующий объект A.

21. Дан класс Print. Перегрузите в классе Print методы так, чтобы был доступен вызов следующих методов через объект класса Print.


```
Print p = new Print();  
p.print(234);  
p.print("string");  
p.print(456, "string");
```

```
public class Print {  
  
    public void print(int x) {  
        System.out.println(x);  
    }  
    public void print(String y) {  
        System.out.println(y);  
    }  
    public void print(int x, String y) {  
        System.out.println(x + " " + y);  
    }  
  
}
```

22. Переопределите для класса Main методы equals() и hashCode().

```
import java.util.Objects;
public class Main {
    private int x;
    private String y;

    @Override
    public boolean equals(Object obj){
        if (this == obj)
            return true;
        if (obj == null)
            return false;
        if (getClass() != obj.getClass())
            return false;
        Main other = (Main) obj;
        if (x != other.x)
            return false;
        if (y == null) {
            if (other.y != null)
                return false;
        } else if (!y.equals(other.y))
            return false;
        return true;
    }
}
```



```
@Override
public int hashCode() {
    return Objects.hash(x, y);
}
```

23. Даны классы A и B. Напишите в классе B такой конструктор, чтобы объект класса B можно было создать следующим образом

```
A a = new B(123, "bstub", 'd');
```

Объясните свое решение.

(возможно несколько решений, приведите одно)

```
class A{
    A(int x){                this("stub");}
    A(String str){           super();    }
}

class B extends A{
    char y;
    B(int x, String str, char y) {        //наследуем конструктор из класса A, параметр char
        super (x, str)                   определяем и добавляем в конструктор с помощью this
        this.y = y;
    }
}
```

24. Даны два класса.

```
class A{}
class B extends A{}
```

Объясните, что “делают” следующие два оператора.

1) A a = new B(); Ссылка a типа A начинает ссылаться на объект класса B, который унаследовал дефолтный конструктор класса A	2) B b = new A(); Ссылка b типа B начинает ссылаться на объект класса A, который создан по дефолтному конструктору класса Object
--	---

25. Дан код.

```
public class V {  
    public static void main(){  
        A a = new B();  
        a.f(); //5  
    }  
}  
  
class A{  
    public void f(){}  
}  
  
class B extends A{  
    public void f(){}  
}
```

Метод какого класса вызовется в строке 5 и почему?

Создан объект класса "B", с помощью ссылки "a" мы вызываем метод "f" класса B, а так как класс B унаследовал метод "f" класса "A" то метод "f" класса "A" выполниться в пятой строке, т.к он базовый  
но это не точно =)

26. Дан код

```
public class V {  
    public static String m(String n) {  
        try {  
            if (n.length() > 0) {  
                System.out.println(n);  
            }  
            System.out.println(n + "2");  
        } catch (NullPointerException e) {  
            e.printStackTrace();  
        }  
        return n;  
    }  
}
```

Метод m вызывается как `V.m("1");`.

Объясните, что будет в этом случае происходить при выполнении кода и почему.

На консоль выведется

1

1,2

27. Дан код.

```
public static void m() {  
    try {  
        System.out.println("a");  
        return;  
    } finally{  
        System.out.println("b");  
    }  
}
```

Что выведется на консоль при вызове этого метода и почему?

На консоль выведет: a  
b

Пытаемся выполнить вывод на консоль "a" - успешно, return - идём выходить из try, но так как дальше есть блок finally (а он выполняется всегда) заходим в него и выполняем вывод на консоль "b"

28. Дана иерархия классов

```
abstract class A{}  
class B extends A{}  
class C extends A{}  
interface X{}  
class D extends B implements X{}  
class E extends C implements X{}  

```

Укажите, какие из операторов являются верными, а какие нет. Объясните причины своего решения.

A a = new A(); нет - т.к. A - абстрактный класс, а из него нельзя создать класс

B b = new A(); нет - т.к. B - не суперкласс для класса объекта A

D d = new B(); нет - т.к. D - не суперкласс для класса объекта B

X x1 = new X(); нет - т.к. нельзя из интерфейса X создать объект X

X x2 = new B(); нет - т.к. нету имплементации (implements)

C c = new B(); нет - т.к. C - не суперкласс для класса объекта B

E e = new D(); нет - т.к. E - не суперкласс для класса объекта D

X x3 = new E(); да - т.к. создастся объект E - в который унаследуются интерфейс X

X x4 = new D(); да - т.к. создастся объект D - в который унаследуются интерфейс X

x3 = x4; да - одинаковые переменные



29. Дан интерфейс X, лежащий в пакете xxx. Реализуйте его для класса Y, лежащего в пакете ууу.

```
package xxx;

public interface X {
    void a();
    double b(double c);
}
```

```
package yyy;
import xxx.X;
@Override
public class Y implements X{
    public void a(){
    }
    @Override
    public double b(double c){
    }
}
```

30. Дан класс Student.

```
public class Student {
    private String name;
    private String surname;
    private int idFaculty;
}
```

- 1) Перепишите класс, реализовав для него интерфейс Comparable, сортирующий студентов по имени.

```
public class Student implements Comparable<Student>{
    public class Student {
        private String name;
        private String surname;
        private int idFaculty;
    }
    @Override
    public int compareTo (Student student) {
        return name.compareTo(student.name);
    }
}
```

- 2) Напишите для объектов класса Student компаратор, сортирующий студентов по имени и фамилии

```
Comparator<Student> cmp Comparator.comparing ((Student student -> student.name))  
.then Comparing (student -> student.surname);
```