

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

a) `public class X{}` X.java

b) `class Y{}` может быть любое имя

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();
```

```
min = Math.min(x, y);
```

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

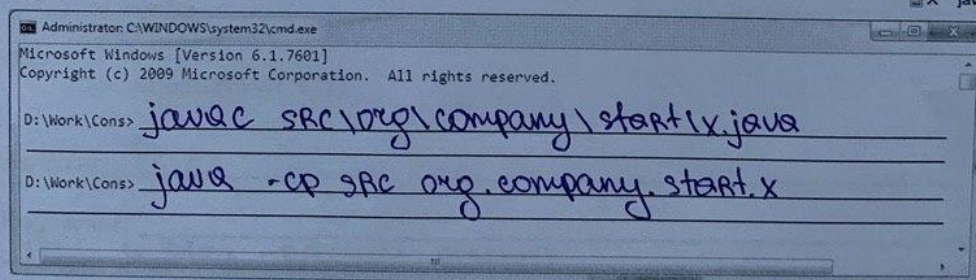
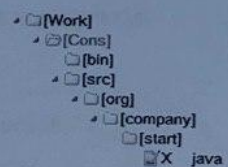
4. Напишите компилируемый класс, выводящий на консоль числа от 1 до 10.

```
public static void main (String args[]) {  
    for (int i=1; i<=10; i++) {  
        System.out.println(i);  
    }  
}
```

5

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 {  
    Static public 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);  
    }  
  
    public static double func(double int a, double int b, double int c) {  
        return a+b+c;  
    }  
}
```

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: b c

Объяснение:

str1 и str2 - разные объекты

str1 и str3 - ссылки на одну строковую const.

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;  
    }  
}
```

```
class CalcTest {  
    @Test  
    public void extend() {  
double x = 10;  
double y = 2;  
double realRes;  
        double expectedRes = 1000;  
  
        realRes = Calc.extent(x, y);  
        Assert.assertEquals(expectedRes, realRes, 0.0001);  
    }  
}
```

```
realRes = Calc.extent(x, y);  
Assert.assertEquals(expectedRes, realRes, 0.0001);
```

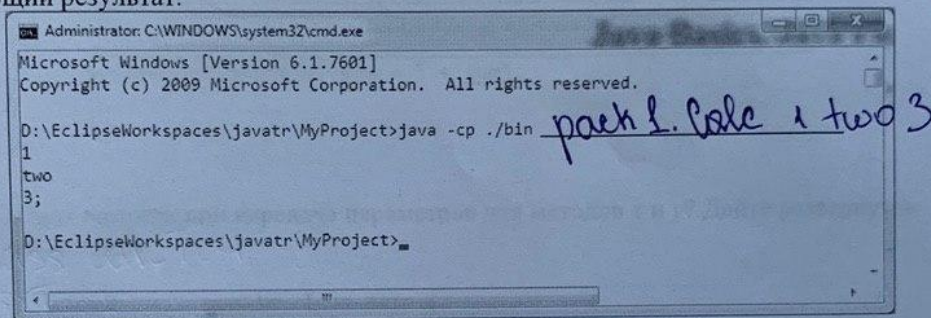
```
}
```


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.



```
Administrator: C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

D:\EclipseWorkspaces\javatr\MyProject>java -cp ./bin pack1.Calc 1 two 3
1
two
3;

D:\EclipseWorkspaces\javatr\MyProject>
```

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. Допишите 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 = 0;  
        for (int i = 0; i < mas.length - 1; i++) {  
            sum = sum + mas[i];  
        }  
        System.out.println("sum = " + sum);  
    }  
}
```

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

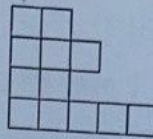
```
public class Main {  
    public void x(int x){}  
    public void y(Date d){}  
}
```

передается значение примитива

передается ссылка на объект

13

Напишите код, создающий и инициализирующий массив следующего вида.



~~int array = new int[4][4];~~
~~array[0][0] = 1;~~
~~array[0][1] = 2;~~
~~array[0][2] = 3;~~
~~array[0][3] = 4;~~
~~array[1][0] = 1;~~
~~array[1][1] = 2;~~
~~array[1][2] = 3;~~
~~array[1][3] = 4;~~
~~array[2][0] = 1;~~
~~array[2][1] = 2;~~
~~array[2][2] = 3;~~
~~array[2][3] = 4;~~
~~array[3][0] = 1;~~
~~array[3][1] = 2;~~
~~array[3][2] = 3;~~

```
int array = new int[4][4];
array[0][0] = new int[] {1, 2};
array[0][1] = new int[] {1, 2, 3};
array[0][2] = new int[] {1, 2};
array[0][3] = new int[] {1, 2, 3, 4, 5};
```

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 year;
    private int price;
```

```
    public Book(String title; String author; int year) {}
    public Book(String title; String author) {}
    public Book(int price) {}
```

15. Дан класс.

```
public class Main {  
}
```

Создайте объект этого класса и объясните свое решение.

`Main main = new Main();`

объект main класса Main
и допустим что там ~~нет~~ конструктор. конструктор

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

`public` - доступен всем

`private` - доступен внутри класса

`protected` - доступен внутри пакета + производным от
класса ~~или~~ если они в другом пакете

`friendly`
«по умолчанию» - внутри пакета

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){}  
}
```

→ примитивный тип

↓
ссылка 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.

~~Объект~~ класс - общее описание чего-то конкретного
т.е. это шаблон для ~~какого-то~~ объекта

объект - экземпляр того класса, созданный
на основе его описания

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

конструктор с модификатором private не
дает создать еще один экземпляр.
поэтому, чтоб получить объект надо
вызвать static A getInstance();

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 i) { ...  
    }  
  
    public void print (String s) { ...  
    }  
  
    public void print (int i ; String s) { ...  
    }  
  
}
```

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

```
public class Main {  
    private int x;  
    private String y;
```

```
    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 != other.y) {  
            return false;
```

```
        }  
        return true;
```

```
    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{
```

```
    B(int x; String str; char c){
        super(str);
```

```
}
```

24. Даны два класса.

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

Объясните, что "делают" следующие два оператора.

1) `A a = new B();`

присваивает A ссылке
на B

2) `B b = new A();`

ошибку

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?

метод класса B/B - ^{наследник} расширения
класса A и наследует его
public метод)

~~сам вызов не выполняется~~

он такой же, как в А или
но не имени, но вызов
метод у B

26. Дан код

```
public class V {
    public static String m(String n) {
        try {
            if (n.length() > 0) {
                System.out.println(n); ← 1
            }
            System.out.println(n + "2"); ← 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.

результат вызова строки не
используется

27. Дан код.

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

аи всего выполнится

Что выведется на консоль при вызове этого метода и почему? *последовательно а, в*

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(); *false / нельзя создать абстрактного класса*

B b = new A(); *false (B не экв. суперклассу A)*

D d = new B(); *false (D не экв. суперклассу B)*

X x1 = new X(); *false / у интерф. нельзя создать класс*

X x2 = new B(); *false / нарушение контракта*

C c = new B(); *false (C не экв. суперклассу B)*

E e = new D(); *false / E не суперкласс D*

X x3 = new E(); *true / создали объект ~~класса~~ интерфейса, который является объектом класса E*

X x4 = new D(); *true / созд. обь. интерф. кот. экв. обьек. класса D*

x3 = x4; *true / ~~они~~ один сложный тип, переопределение*

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

```
package xxx;
```

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

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

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>{
```

```
    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<StudentPerson> cmp =  
    = Comparator.comparing((Student s) -> s.name)  
    .thenComparing(students) -> s.surname);
```