

- I. Define abstract class *Singer*. Each singer has a name and start number assigned automatically when creating the object. Class *Singer* should have the constructor *Singer(String name)* and the following methods::

- **abstract**: *abstract String sing()* - returns the text sung by the singer in the competition.
- *public String toString()* - returns information about the singer.
- **static**: ... *loudest(...)* takes an array of objects/singers and returning singer object whose sung text has the most capital letters (based on the method *sing()*).

Create the *Singer* class in such a way that the following method *main* from the class *Main*:

```
public class Main {
    public static void main(String[] args)
    {
        Singer s1 = new Singer("Eminem"){
            /*<- code */
        };

        Singer s2 = new Singer("Eagles"){
            /*<- code */
        };

        Singer s3 = new Singer("Simon & Garfunkel"){
            /*<- code */
        };

        Singer sp[] = {s1, s2, s3};

        for (Singer s : sp)
            System.out.println(s);

        System.out.println("\n" + Singer.loudest(sp));
    }
}
```

gave feedback as below:

- (1) Eminem: You own it, you better never let it go
  - (2) Eagles: Hotel California
  - (3) Simon & Garfunkel: Hello darkness, my old friend. I've come to talk with you again
- (2) Eagles: Hotel California

- II. We consider the world of 2D geometric figures. The figure always has a surface area and a circumference that can be calculated. We can also assume that the figures may have colors. The figures include squares, rectangles, circles, triangles, hexagons, etc. Each figure has a different formula for calculating the area and perimeter. What's more, each of them has completely different values defining them, e.g. for a square it will be the length, for a circle its radius, while the rectangle has already 2 sides defining it.

Once you know the theory, try to implement the above world of figures using the **abstract** and normal classes. Use **inheritance** and **polymorphism** to avoid redundancy in code. Then create a few figures from each class and place them on the array of figures. Display information about them using *toString* method.

- III. Create *Square* class with fields *int length*, *int number* and *static int counter*. Add a constructor that receives *int length* as it's argument and initializes the fields accordingly. Also write the *getArea()* method and overwrite the *toString()* method so that it returns *"(" + number + "):" + getArea()*.

In the *Square* class declaration, implement the *Comparable* interface parameterized for data type *Square* and implement the required *compareTo(...)* method.

In the *main* method, create 5 objects of the *Square* class and add them to the array, and then print them on the console. Sort the list using the *sort* method from the *Arrays* class and print it again on the console.

- IV. Create *Figure* interface with field *int max* initialized with value *6*. Declare the following methods:

*public int getArea()*

*public int getPerimeter()*

In the *Square* class declaration, implement the *Figure* interface. In the constructor of the *Square* class, check if the variable *length* given as an argument is greater than the field *max* from the interface *Figure*. If the value is higher, raise the exception *TooBigSquareException* with message *"Maximum length is" + max*.

- V. (Additional task):

There are various types of mobile electronic devices. however, they share some common features. Each of them has the name *manufacturer* and *price*. The devices have various functionalities such as the ability to: *calling, writing messages, surfing the Internet, playing games*.

Depending on the device, you can use it to perform some or all of the above operations.

Available devices:

- *Landline phone* (only offers the option of calling)
- *Mobile phone* (offers the ability to call and send SMS, but only if it has a sim card)
- *Game console* (offers the ability to play games and surf the Internet - only if it has an Internet connection)
- *Smartphone* (you can do everything on it and more, if you have the right conditions)

Implement the above assumptions using classes, inheritance, abstract classes and interfaces. Create several objects of these devices and sort them into appropriate categories (categories are functionalities, e.g. calling devices). Test each of them to see if the functionality is working properly.

Finally, collect all devices in one place (array) and sort all devices by price using the *Arrays.sort()* method using your own comparator defined in the anonymous class. Remember about good programming practice of overwriting *toString* method so that it returns understandable device information.