

Университет ИТМО, кафедра ВТ

**Лабораторная работа №3 по
“Программированию Интернет-Приложений”
Вариант 717**

Работу выполнил
студент группы Р3200
Рогов Я. С.

Преподаватель:
Харитонов А.Е.

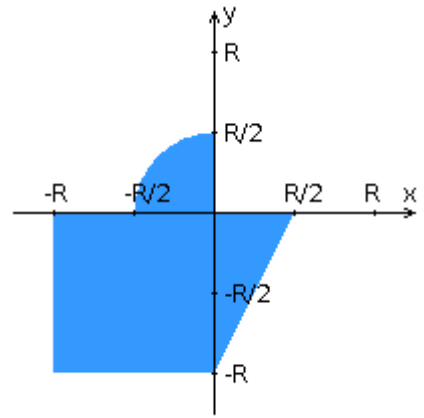
Санкт-Петербург, 2016

Задание:

На языке Java написать консольную программу, которая определяет, какие точки из множества $\{\{3, 3\}, \{-3, 3\}, \{-2, -1\}, \{-4, 3\}, \{-3, 3\}, \{0, 0\}, \{-3, 2\}, \{5, -4\}, \{-5, -5\}\}$ входят в заданную область S.

Приложение должно содержать следующие классы:

1. Класс `Vertice`, представляющий точку с координатами X и Y типа `double`.
 2. Класс `Kontur`, представляющий область с заданным параметром `R` типа `float`, в котором должен быть реализован метод, возвращающий для заданной точки значение 1, если точка входит в область, и 0, если не входит. Попадание на границу области не считается попаданием в область.
 3. Основной класс `Program3`, в котором необходимо реализовать получение значения `R` в качестве аргумента командной строки. Преобразование из строки в число реализовать с помощью метода `Float.valueOf()`.
- Точки хранятся в виде коллекции непараметризованного типа `Vector`.
 - Обход коллекции реализовать с помощью цикла `for` с итератором.
 - Приложение должно выводить на экран список только попадающих в область точек.



Код программы:

```
import java.util.Vector;

public class Program3{

    final static float defaultR = 1.0f;

    public static void main(String[] args){

        float R = defaultR;

        if (args.length== 0){
            System.out.println(
                "No arguments has been passed.\nUsing default R value: " +
                defaultR);
        }
        else{
            try{
                R = Float.valueOf(args[0]);
            }
            catch(NumberFormatException e){
                System.out.println(
                    "Argument isn't a number. Using default R value: " +
                    defaultR);
            }
        }

        Contour contour717 = new Contour(R, (float r, Vertice v) ->
            v.x > 0 ?
                v.y<=0 && v.y > 2*v.x-r :
                (v.y <= 0?
                    v.x >= -r && v.y >= -r:
                    v.x * v.x + v.y * v.y <= r*r/4)
            );

        double[][] pairs = {
            {3, 3}, {-3, 3}, {-2, -1},
            {-4, 3}, {-3, 3}, {0, 0},
            {-3, 2}, {5, -4}, {-5, -5}};

        Vector verticeVec = new Vector(pairs.length);
        for (int i=0; i< pairs.length; i++){
            verticeVec.add((Object) new Vertice(pairs[i]));
        }

        for (Object o : verticeVec){
            Vertice vert = (Vertice) o;
            if (contour717.checkVertice(vert))
```

```

        System.out.println(vert);
    }
}

class Vertice{
    public double x;
    public double y;

    public Vertice(){
        super();
    }

    public Vertice(double x, double y){
        this.x = x;
        this.y = y;
    }

    public Vertice(double[] vert){
        x = vert[0];
        y = vert[1];
    }

    @Override
    public String toString(){
        return String.format("%.3f; %.3f", x, y);
    }
}

interface BoundsLambda{
    boolean check(float R, Vertice v);}

class Contour{
    private float R;
    private BoundsLambda bounds;

    public Contour(float R, BoundsLambda bounds){
        this.R = R;
        this.bounds = bounds;
    }

    public boolean checkVertice(Vertice v){
        return bounds.check(R, v);
    }
}

```

Вывод

В ходе выполнения данной лабораторной работы мной были изучены основные пакеты JDK версии 1.8, основы обработки исключений в Java, а также классы-коллекции, входящие в состав JDK.