САНКТ-ПЕТЕРБУРЖСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

*Кафедра Вычислительной техники*

**Лабораторная работа №3**

Выполнил:

студент II курса группы 2125

Припадчев Артём

Проверил:

Харитонова А.Е.

Санкт-Петербург

2013

**Задание:** На языке Java написать консольную программу, которая определяет, какие точки из набора A входят в заданную область S.

**Приложение должно содержать следующие классы:**

Класс Mark, представляющий точку с координатами X и Y типа float.

Класс Contour, представляющий область с заданным параметром R, в котором должен быть реализован метод, возвращающий для заданной точки значение true, если точка входит в область, и false, если не входит. Попадание на границу области не считается попаданием в область.

Класс Lab2, который получает параметр R типа int в качестве аргумента командной строки. Получение числа из строки реализовать c помощью метода Scanner.nextInt().

Точки хранятся в виде коллекции параметризованного типа LinkedList.

Обход коллекции реализовать с помощью цикла do.

Приложение должно выводить на экран список всех точек с указанием попадания в область.



# Код программы

import java.util.InputMismatchException;

import java.util.LinkedList;

import java.util.ListIterator;

import java.util.Scanner;

public class Lab3 {

 public static void main(String[] args)

 {

 int R = 0;

 try

 {

 Scanner sc = new Scanner(args[0]);

 R = sc.nextInt();

 }

 catch (InputMismatchException ex)

 {

 System.out.println("Invalid input format");

 return;

 }

 catch (ArrayIndexOutOfBoundsException ex)

 {

 System.out.println("Error! No argument.");

 return;

 }

 if(R>0)

 {

 Contour contour = new Contour(R);

 LinkedList<Mark> ListOfMark = new LinkedList<Mark>();

 Mark mark = new Mark(3,4);

 ListOfMark.add(mark);

 mark = new Mark(1,-2);

 ListOfMark.add(mark);

 mark = new Mark(3,-4);

 ListOfMark.add(mark);

 mark = new Mark(1,0);

 ListOfMark.add(mark);

 mark = new Mark(-3,-3);

 ListOfMark.add(mark);

 mark = new Mark(2,-1);

 ListOfMark.add(mark);

 mark = new Mark(-4,-3);

 ListOfMark.add(mark);

 ListIterator<Mark> Iter = ListOfMark.listIterator();

 do {

 Mark nextMark = Iter.next();

 if(contour.hitInArea(nextMark))

 {

 System.out.println("{" + nextMark.getX() + ";" + nextMark.getY() + "}" + "---->" + "Ok");

 }

 else

 {

 System.out.println("{" + nextMark.getX() + ";" + nextMark.getY() + "}" + "---->" + "No");

 }

 }

 while (Iter.hasNext());

 }

 else

 {

 System.out.println("Invalid format of R");

 }

 }

}

public class Contour {

 private int R;

 public Contour(int R) {

 this.R = R;

 }

 public boolean hitInArea(Mark mark) {

 if (((mark.getX() >= 0) & (mark.getY() < 0) & (mark.getX() < R) & (mark.getY() > -R)) ||

 ((mark.getX() <= 0) & (mark.getY() <= 0) & ((Math.pow(mark.getX(), 2) + Math.pow(mark.getY(), 2)) < Math.pow(R, 2))) ||

 ((mark.getX() < 0) & (mark.getY() >= 0) & (mark.getY() < (mark.getX() / 2 + R / 2)))) {

 return true;

 } else {

 return false;

 }

 }

}

class Mark {

 private float x;

 private float y;

 public float getX()

 {

 return x;

 }

 public float getY()

 {

 return y;

 }

 public Mark(float x, float y)

 {

 this.x = x;

 this.y = y;

 }

}

**Результат программы:**

R=2

{3.0;4.0}---->No

{1.0;-2.0}---->No

{3.0;-4.0}---->No

{1.0;0.0}---->No

{-3.0;-3.0}---->No

{2.0;-1.0}---->No

{-4.0;-3.0}---->No

R=3

{3.0;4.0}---->No

{1.0;-2.0}---->Ok

{3.0;-4.0}---->No

{1.0;0.0}---->No

{-3.0;-3.0}---->No

{2.0;-1.0}---->Ok

{-4.0;-3.0}---->No

R=7

{3.0;4.0}---->No

{1.0;-2.0}---->Ok

{3.0;-4.0}---->Ok

{1.0;0.0}---->No

{-3.0;-3.0}---->Ok

{2.0;-1.0}---->Ok

{-4.0;-3.0}---->Ok

**Вывод:** В рамках выполнения работы были изучены следующие аспекты: коллекции параметризованного типа, цикл с постусловием, условные операторы, метод nextInt() класса Scanner.