**Санкт-Петербургский национальный исследовательский университет**

**информационных технологий, механики и оптики**

**Кафедра информатики и прикладной математики**

Программирование интернет-приложений

Лабораторная работа №3

Выполнил Кудряшов А.А.

Группа 2121

2012 г.

## 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 double.

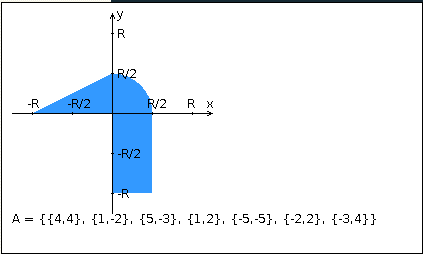
Класс Outline, представляющий область с заданным параметром R, в котором должен быть реализован метод, возвращающий для заданной точки значение 1, если точка входит в область, и 0, если не входит. Попадание на границу области не считается попаданием в область.

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

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.



## Текст программы

//var. 105

**import** java.io.\*;

**import** java.util.\*;

**public** **class** Lab3 {

**public** **static** **void** main(String [] args){

BufferedReader b = **new** BufferedReader(**new** InputStreamReader(System.*in*));

**int** R;

**try**{

R=Integer.*parseInt*(args[0],10);

}

**catch**(Exception e){

System.*out*.print("Неверный формат данных");

**return**;

}

OutLine Target =**new** OutLine(R);

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

Mark m = **new** Mark(4,4);

ListOfMark.add(m);

m =**new** Mark(1,-2);

ListOfMark.add(m);

m =**new** Mark(5,-3);

ListOfMark.add(m);

m =**new** Mark(1,2);

ListOfMark.add(m);

m =**new** Mark(-5,-5);

ListOfMark.add(m);

m =**new** Mark(-2,2);

ListOfMark.add(m);

m =**new** Mark(-3,4);

ListOfMark.add(m);

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

**do**{

Mark mGo = Iter.next();

**if**(Target.Shot(mGo)==1){

System.*out*.print("{"+mGo.x+";"+mGo.y+"} ---> Ok\n");}

**else**{

System.*out*.print("{"+mGo.x+";"+mGo.y+"} ---> No\n");

}

}**while**(Iter.hasNext());

}

}

**class** Mark { //Класс точек (выстрелов)

**double** x;

**double** y;

**public** Mark(**double** x,**double** y) //Конструктор класса

{

**this**.x = x;

**this**.y = y;

}

}

**class** OutLine{ //Класс описывающий мишени

**int** R;

**public** OutLine(**int** R){ //Конструктор класса

**this**.R = R;

}

**public** **short** Shot(Mark m ){ //Проверка попадания

**if**(Math.*pow*(m.x, 2)+Math.*pow*(m.y, 2)<Math.*pow*(R/2, 2)&&m.x>0&&m.y>0||m.x>0&&m.x<R/2&&m.y<0&&m.y>-R||m.y>0&&m.x<0&&m.y<m.x/2+R/2)

**return** 1;

**else**{

**return** 0;}

}

}

## Результаты выполнения

1)

R = 1

{4.0;4.0} ---> No

{1.0;-2.0} ---> No

{5.0;-3.0} ---> No

{1.0;2.0} ---> No

{-5.0;-5.0} ---> No

{-2.0;2.0} ---> No

{-3.0;4.0} ---> No

2)

R = 7

{4.0;4.0} ---> No

{1.0;-2.0} ---> Ok

{5.0;-3.0} ---> No

{1.0;2.0} ---> Ok

{-5.0;-5.0} ---> No

{-2.0;2.0} ---> No

{-3.0;4.0} ---> No

3)

R = 1000

{4.0;4.0} ---> Ok

{1.0;-2.0} ---> Ok

{5.0;-3.0} ---> Ok

{1.0;2.0} ---> Ok

{-5.0;-5.0} ---> No

{-2.0;2.0} ---> Ok

{-3.0;4.0} ---> Ok

## Выводы по работе

В рамках выполнения работы были изучены следующие аспекты: коллекции параметризованного типа, цикл с постусловием, условные операторы, методы изменения типа данных.