СПбНИУ ИТМО

Отчет

По лабораторной работе №3

«Программирование интернет приложений»

Выполнил: Ратников В.И

Группа 3105

Преподаватель: Гаврилов А.В.

**Исходный код программы**

**File : lab3.java**

//var 333

import java.io.IOException;

import java.util.Iterator;

import java.util.LinkedHashSet;

public class Lab3

{

public static void main(String[] args)

{

//variables declaration

LinkedHashSet<Mark> lhs = new LinkedHashSet<Mark>();

float argRadius;

String unparsedRadius;

byte[] buffer = new byte[128];

Contour ctr;

Mark m\_tmp;

Iterator<Mark> iterator;

System.out.println("enter radius");

try

{

//LinkedHashSet filling

lhs.add(new Mark(5f, 3f));

lhs.add(new Mark(2f, 2f));

lhs.add(new Mark(4f, -5f));

lhs.add(new Mark(-1f, 0f));

lhs.add(new Mark(-3f, -4f));

lhs.add(new Mark(1f, 1f));

lhs.add(new Mark(-5f, 4f));

iterator = lhs.iterator();

//reading and parsing [String] to [Float]

System.in.read(buffer);

unparsedRadius = new String(buffer);

argRadius = Float.parseFloat(unparsedRadius);

//creating contour with parsed radius

ctr = new Contour(argRadius);

//collection iteration

do

{

m\_tmp = iterator.next();

if(ctr.is\_hit(m\_tmp))

{

System.out.println( m\_tmp + " hit the target" );

}

}

while( iterator.hasNext() );

}

catch(NumberFormatException NFE)

{

System.err.println(NFE + NFE.toString());

System.exit(-1);

}

catch(IOException IOE)

{

System.err.println(IOE + IOE.toString());

System.exit(-1);

}

catch(Exception E)

{

System.err.println(E + E.toString());

System.exit(-1);

}

}

}

**File : Mark.java**

public class Mark

{

float \_x,\_y;

/\*\* constructor \*/

public Mark(float x,float y)

{

this.\_x=x;

this.\_y=y;

}

@Override

public String toString()

{

StringBuffer sb = new StringBuffer();

sb.append("<");

sb.append(this.\_x);

sb.append(",");

sb.append(this.\_y);

sb.append(">");

return sb.toString();

}

}

**File : Contour.java**

public class Contour

{

//radius

float R;

//constructor

public Contour(float \_r)

{

this.R = (\_r==0.0f) ? 0 : Math.abs(\_r);

}

/\*\* returns true if Mark m hit the target, else returns false \*/

boolean is\_hit(Mark m)

{

//first quarter

if(m.\_x>0 && m.\_y>0)

{

if( m.\_y < (this.R-m.\_x) ) return true;

}

//third quarter

else if(m.\_x<0 && m.\_y<0)

{

if( Math.abs(m.\_x)<(this.R/2) && Math.abs(m.\_y)<(this.R) ) return true;

}

//fourth quarter

else if(m.\_x>0 && m.\_y<0)

{

if( (Math.pow(Math.abs(m.\_x),2)+Math.pow(Math.abs(m.\_y),2)) < Math.pow(this.R/2, 2) ) return true;

}

return false;

}

}