**Санкт-Петербургский национальный исследовательский университет**

**информационных технологий, механики и оптики**

**Кафедра информатики и прикладной математики**

Программирование Интернет-Приложений

Лабораторная работа №3

Вариант 3111

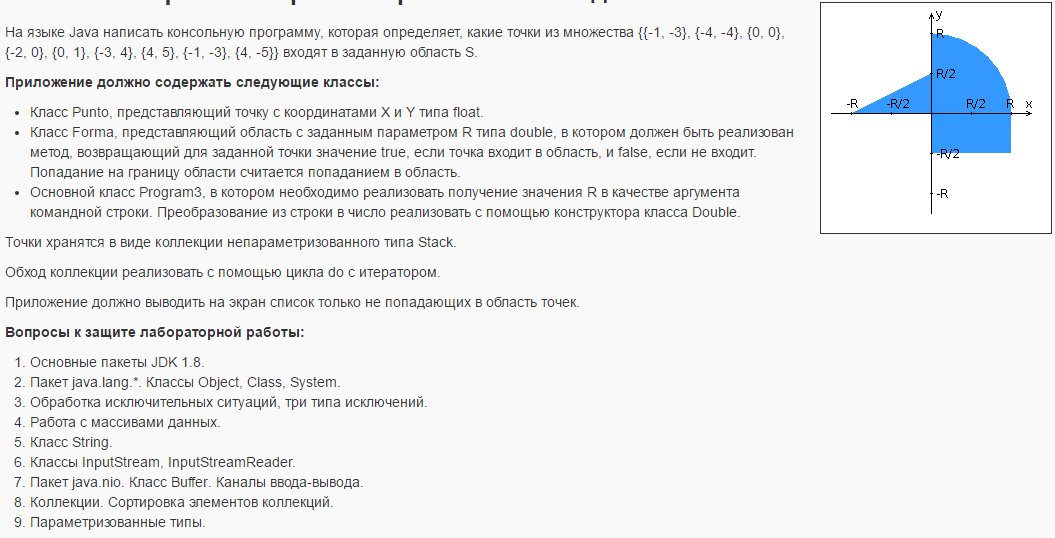
Выполнил: Гхази Даниэль

Группа P3218

Преподаватель: Гаврилов А.В.

2016 г.

**Текст задания**



**Текст программы**

**import** java.util.ArrayList;  
**import** java.util.Stack;  
**import** java.util.regex.Pattern;  
  
**public class** Program3  
{  
 **public static void** main(String[] args)  
 {  
 Forma chart = **new** Forma();  
 Stack dotsOnChart = **new** Stack();  
 *dotsInput*(dotsOnChart);  
 ArrayList<Punto> notInZoneDots = **new** ArrayList<Punto>();  
  
 Double getterOfR;  
 **if** (args.**length** != 0)  
 {  
 **switch** (*checkWithRegExp*(args[0]))  
 {  
 **case** 0:  
 getterOfR = **new** Double(args[0]);  
 chart.**R** = getterOfR;  
 System.***out***.println(**"R = "** + chart.**R**);  
 **break**;  
  
 **case** 1:  
 System.***out***.print(**"Use '.',not ','."**);  
 **return**;  
  
 **case** 2:  
 System.***out***.print(**"R can't be negative."**);  
 **return**;  
  
 **case** 3:  
 System.***out***.print(**"R is too big."**);  
 **return**;  
  
 **case** 4:  
 System.***out***.print(**"R is not a number."**);  
 **return**;  
 }  
 }  
 **else** {  
 System.***out***.print(**"R not given."**);  
 **return**;  
 }  
  
 **int** i = 0;  
 **do** {  
 **if** (!chart.IsDotInZone((Punto)dotsOnChart.peek(), chart.**R**))  
 notInZoneDots.add((Punto)dotsOnChart.pop());  
 **else** dotsOnChart.pop();  
 i++;  
 }  
 **while** (i < 9);  
  
 System.***out***.print(**"Dots that are out of chart zone: "**);  
 **for** (Punto notInZoneDot: notInZoneDots)  
 System.***out***.print(**"{"** + notInZoneDot.**X** + **", "** + notInZoneDot.**Y** + **"} "**);  
  
  
 }  
  
 **private static void** dotsInput(Stack dotsOnChart)  
 {  
 Punto tmpDot = **new** Punto();  
 tmpDot.**X** = -1;  
 tmpDot.**Y** = -3;  
 dotsOnChart.push(tmpDot);  
  
 tmpDot = **new** Punto();  
 tmpDot.**X** = -4;  
 tmpDot.**Y** = -4;  
 dotsOnChart.push(tmpDot);  
  
 tmpDot = **new** Punto();  
 tmpDot.**X** = 0;  
 tmpDot.**Y** = 0;  
 dotsOnChart.push(tmpDot);  
  
 tmpDot = **new** Punto();  
 tmpDot.**X** = -2;  
 tmpDot.**Y** = 0;  
 dotsOnChart.push(tmpDot);  
  
 tmpDot = **new** Punto();  
 tmpDot.**X** = 0;  
 tmpDot.**Y** = 1;  
 dotsOnChart.push(tmpDot);  
  
 tmpDot = **new** Punto();  
 tmpDot.**X** = -3;  
 tmpDot.**Y** = 4;  
 dotsOnChart.push(tmpDot);  
  
 tmpDot = **new** Punto();  
 tmpDot.**X** = 4;  
 tmpDot.**Y** = 5;  
 dotsOnChart.push(tmpDot);  
  
 tmpDot = **new** Punto();  
 tmpDot.**X** = -1;  
 tmpDot.**Y** = -3;  
 dotsOnChart.push(tmpDot);  
  
 tmpDot = **new** Punto();  
 tmpDot.**X** = 4;  
 tmpDot.**Y** = -5;  
 dotsOnChart.push(tmpDot);  
 }  
  
 **private static int** checkWithRegExp(String R)  
 {  
 **if** (Pattern.*matches*(**"^\\d{1,16}$|^\\d{1,16}\\.\\d{1,16}$"**, R))  
 **return** 0;  
  
 **if** (Pattern.*matches*(**"^\\d{1,16},\\d{1,16}$"**, R))  
 **return** 1;  
  
 **if** (Pattern.*matches*(**"^-\\d+$|^-\\d+\\.\\d+$"**, R))  
 **return** 2;  
  
 **if** (Pattern.*matches*(**"^-?\\d+$|^-?\\d+\\.\\d+$"**, R))  
 **return** 3;  
  
 **return** 4;  
 }  
}  
  
  
**class** Punto  
{  
 **float X**;  
 **float Y**;  
}  
  
  
**class** Forma  
{  
 **double R**;  
  
 **boolean** IsDotInZone(Punto curDot, **double** R)  
 {  
 **if** (curDot.**X** < 0)  
 {  
 **if** ((curDot.**X** >= -R) && (curDot.**Y** >= 0) && (curDot.**Y** <= 0.5 \* (curDot.**X** + R)))  
 **return true**;  
 }  
 **else if** ((curDot.**X** <= R) && (curDot.**Y** >= -0.5 \* R) && (curDot.**Y** <= Math.*sqrt*(Math.*pow*(R, 2) - Math.*pow*(curDot.**X**, 2))))  
 **return true**;  
  
  
  
 **return false**;  
 }  
}

**Вывод:**

В результате выполнения лабораторной работы я ознакомился с преобразованием типов в Java, научился работать с коллекциями, узнал о работе с регулярными выражениями и стеком. Полученные знания можно использоваться для написание более сложных программ на Java.