Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по программе «**Название практического занятия**»

Выполнил: Шмелев Максим

Группа: ПР-22

Преподаватель: Мирошниченко Г.В

2024

**Задание:**

****

**Входные**

**X, Y, Z, L**

**выходные данные**

**L**

**Блок-схема**

**Листинг программы (если есть)**

fun main() {  
 try {  
  
 *println*("Введите значени x y z")  
 var x = *readLine*()!!.*toInt*()  
 var y = *readLine*()!!.*toInt*()  
 var z = *readLine*()!!.*toInt*()  
  
 var l = 2 \* *maxOf*(x, z) - 3 \* *minOf*(x, y, z)  
 *println*(l)  
  
  
 } catch (e: Exception) {  
 *println*("не верно")  
 }  
}

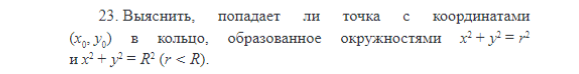
**Тестовые ситуации**

****

**Вывод**

****

**Задание:**

****

**Входные**

**Y, x, R,r**

**выходные данные**

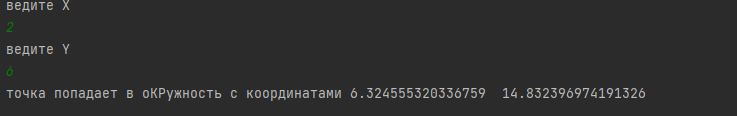
точка попадает в коужность с координатами

**Блок-схема**

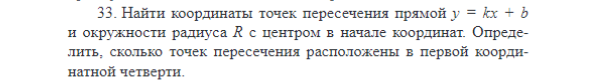
**Листинг программы (если есть)**

import java.lang.Exception  
import kotlin.math.pow  
import kotlin.math.sqrt  
  
fun main(){  
 try {  
 *println*("ведите Х")  
 var x = *readLine*()!!.*toDouble*()  
 *println*("ведите Y")  
 var y = *readLine*()!!.*toDouble*()  
  
 var r = *sqrt*(x.*pow*(2) + y.*pow*(2))  
 var R = *sqrt*(x.*pow*(2) + y.*pow*(2))  
  
 when{  
 (r < R)-> *println*("точка попадает в коужность с координатами $r $R")  
 (r > R)-> *println*("точка попадает в коужность с координатами $r $R")  
 }  
 }catch (e:Exception){  
 *println*("не верю")  
 }  
}

**Тестовые ситуации И Вывод**

****

**Задание:**

****

**Входные**

a, b1, k, R, c, d, b, x1,x2, y1,y2

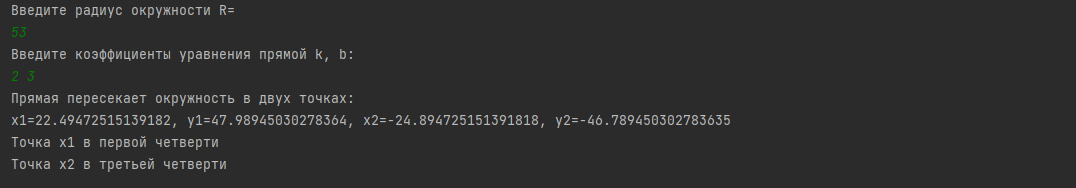
**выходные данные**

**Блок-схема**

**Листинг программы (если есть)**

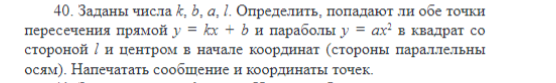
import java.lang.Exception  
import kotlin.math.sqrt  
  
fun main(){  
 try {  
  
 *println*("Введите радиус окружности R=")  
 val R = *readLine*()!!.*toDouble*()  
  
 *println*("Введите коэффициенты уравнения прямой k, b:")  
 val (k, b) = *readLine*()!!.*split*(" ").*map* **{ it**.*toDouble*() **}** val a = k \* k + 1  
 val b1 = 2 \* k \* b  
 val c = b \* b - R \* R  
 val d = b1 \* b1 - 4 \* a \* c  
  
 if (d < 0) {  
 *println*("Точек пересечения нет.")  
 } else if (d == 0.0) {  
 val x1 = -b1 / (2 \* a)  
 val y1 = k \* x1 + b  
 *println*("Прямая касается окружности в точке x=$x1, y=$y1")  
 } else if (d > 0) {  
 val x1 = (-b1 + *sqrt*(d)) / (2 \* a)  
 val y1 = k \* x1 + b  
 val x2 = (-b1 - *sqrt*(d)) / (2 \* a)  
 val y2 = k \* x2 + b  
 *println*("Прямая пересекает окружность в двух точках:")  
 *println*("x1=$x1, y1=$y1, x2=$x2, y2=$y2")  
  
 if (x1 > 0 && y1 > 0) {  
 *println*("Точка x1 в первой четверти")  
 } else if (x1 < 0 && y1 > 0) {  
 *println*("Точка x1 во второй четверти")  
 } else if (x1 < 0 && y1 < 0) {  
 *println*("Точка x1 в третьей четверти")  
 } else if (x1 > 0 && y1 < 0) {  
 *println*("Точка x1 в четвертой четверти")  
 }  
  
 if (x2 > 0 && y2 > 0) {  
 *println*("Точка x2 в первой четверти")  
 } else if (x2 < 0 && y2 > 0) {  
 *println*("Точка x2 во второй четверти")  
 } else if (x2 < 0 && y2 < 0) {  
 *println*("Точка x2 в третьей четверти")  
 } else if (x2 > 0 && y2 < 0) {  
 *println*("Точка x2 в четвертой четверти")  
 }  
 }  
  
  
 }catch (e:Exception){  
 *println*("Не верно")  
 }  
}

**Тестовые ситуации И Вывод**





**Задание:**

****

**Входные**

**Disq, a,b,k,x1,x2,y1,y2**

**выходные данные**

**x1,y1,x2,y2**

**Блок-схема**

**Листинг программы (если есть)**

import java.lang.Exception  
import kotlin.math.sqrt  
  
fun main() {  
  
 try {  
  
  
 val k = *readLine*()!!.*toDouble*()  
 val b = *readLine*()!!.*toDouble*()  
 val a = *readLine*()!!.*toDouble*()  
  
 val disq = b \* b - 4 \* a \* k \* k  
  
 if (disq >= 0) {  
 val x1 = (-b + *sqrt*(disq)) / (2 \* a \* k)  
 val x2 = (-b - *sqrt*(disq)) / (2 \* a \* k)  
  
 val y1 = k \* x1 + b  
 val y2 = k \* x2 + b  
  
 val dlin = 1.0  
 val poldln = dlin / 2  
  
 if (x1 >= -poldln && x1 <= poldln && y1 >= -poldln && y1 <= poldln &&  
 x2 >= -poldln && x2 <= poldln && y2 >= -poldln && y2 <= poldln  
 ) {  
 *println*("Обе точки пересечения находятся внутри квадрата")  
 *println*("Точка 1: ($x1, $y1)")  
 *println*("Точка 2: ($x2, $y2)")  
 } else {  
 *println*("Хотя бы одна точка пересечения находится вне квадрата")  
 }  
 } else {  
 *println*("Нет точек пересечения.")  
 }  
 }catch (e:Exception){  
 *println*("не верно")  
 }  
}

**Тестовые ситуации И Вывод**

