Министерство образования и молодежной политики Свердловской области

ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по программе «Практическое занятие 7 (задание 3)»

Выполнил: Зыкова Софья Александровна

Группа: ПР-21

Преподаватель: Мирошниченко Г.В

2023

**Задание 3.1:**

****

**Входные данные:**

**С -** длина окружности, вещественный тип

**выходные данные:**

**S\_krug -** площадь круга, вещественный тип

**Листинг программы (если есть)**

**fun main()**

**{**

**try{**

***println*("введите длину окружности: ")**

**var C = *readLine*()!!.*toDouble*()**

**while(C <= 0) {**

***println*("длина окружности не может быть меньше или равно нулю, введите еще раз")**

**C = *readLine*()!!.*toDouble*()**

**}**

***println*("площадь круга = ")**

**var S\_krug = (C \* C)/(4 \* Math.*PI*)**

***print*(S\_krug)**

**}**

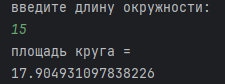
**catch (e: Exception){**

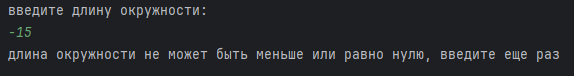
***println*("неверный формат данных")**

**}**

**}**

**Тестовые ситуации**

****



**Задание 13.1:**

****

**Входные данные:**

**x1, y1** - координаты первой вершины, вещественный тип

**x2, y2** - координаты второй вершины, вещественный тип

**x3, y3** - координаты третьей вершины, вещественный тип

**выходные данные:**

P - периметр треугольника, вещественный тип

S - площадь треугольника, вещественный тип

**Листинг программы (если есть)**

**fun main()**

**{**

**try{**

***println*("введите координаты первой вершины треугольника: ")**

***print*("x1 = ")**

**var x1 = *readln*()!!.*toDouble*()**

***print*("y1 = ")**

**var y1 = *readln*()!!.*toDouble*()**

***println*("введите координаты второй вершины треугольника: ")**

***print*("x2 = ")**

**var x2 = *readln*()!!.*toDouble*()**

***print*("y2 = ")**

**var y2 = *readln*()!!.*toDouble*()**

***println*("введите координаты третьей вершины треугольника: ")**

***print*("x3 = ")**

**var x3 = *readln*()!!.*toDouble*()**

***print*("y3 = ")**

**var y3 = *readln*()!!.*toDouble*()**

**val ab = Math.sqrt(Math.pow((x2 - x1), (2.0)) + Math.pow((y2 - y1), (2.0)))**

**val bc = Math.sqrt(Math.pow((x3 - x2), (2.0)) + Math.pow((y3 - y2), (2.0)))**

**val ca = Math.sqrt(Math.pow((x1 - x3), (2.0)) + Math.pow((y1 - y3), (2.0)))**

**var P = ab + bc + ca**

**var S = 0.5 \* Math.abs(x1 \* (y2 - y3) + x2 \* (y3 - y1) + x3 \* (y1 - y2))**

***println*("периметр треугольника = $P")**

***println*("площадь треугольника = $S")**

**}**

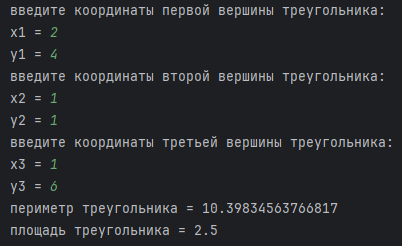
**catch (e: Exception){**

***println*("неверный формат данных")**

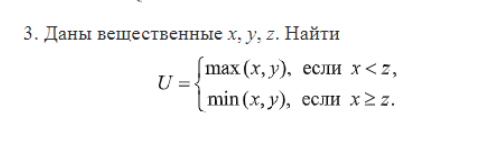
**}**

**}**

**Тестовые ситуации**

****

**Задание 3.2:**

****

**Входные данные:**

**x -** вещественный тип

**y -** вещественный тип

**z -** вещественный тип

**выходные данные:**

**U -** вещественный тип

**Листинг программы (если есть)**

**fun main(){**

**try{**

***print*("x = ")**

**var x = *readLine*()!!.*toDouble*()**

***print*("y = ")**

**var y = *readLine*()!!.*toDouble*()**

***print*("z = ")**

**var z = *readLine*()!!.*toDouble*()**

**var U: Double = 0.0**

**when{**

**x < z -> U = Math.max(x,y)**

**x >= z -> U = Math.min(x,y)**

**else -> {**

***println*("не верный ввод")**

**}**

**}**

***println*(U)**

**}**

**catch (e: Exception){**

***println*("неверный формат данных")**

**}**

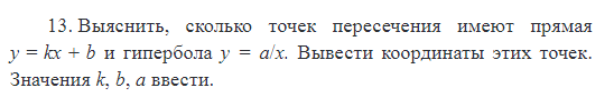
**}**

**Тестовые ситуации**





**Задание 13.2:**

****

**Входные данные:**

**k -** вещественный тип

**b -** вещественный тип

**a -** вещественный тип

**выходные данные:**

**x, y -** точки пересечения, вещественный тип

**Листинг программы (если есть)**

import kotlin.math.sqrt //импорт функции sqrt из стандартной библиотеки котлин

fun main(){

try{

*print*("k = ")

var k = *readLine*()!!.*toDouble*()

*print*("b = ")

var b = *readLine*()!!.*toDouble*()

*print*("a = ")

var a = *readLine*()!!.*toDouble*()

// Вычисление дискриминанта

val discriminant = b \* b - 4 \* k \* (-a)

when {

discriminant > 0 -> {

// 2 точки пересечения

val x1 = (-b + *sqrt*(discriminant)) / (2 \* k)

val x2 = (-b - *sqrt*(discriminant)) / (2 \* k)

val y1 = k \* x1 + b

val y2 = k \* x2 + b

*println*("Две точки пересечения:")

*println*("Точка 1: ($x1, $y1)")

*println*("Точка 2: ($x2, $y2)")

}

discriminant == 0.0 -> {

// 1 точка пересечения

val x = -b / (2 \* k)

val y = k \* x + b

*println*("Одна точка пересечения:")

*println*("Точка: ($x, $y)")

}

else -> {

// Нет точек пересечения

*println*("Нет точек пересечения.")

}

}

}

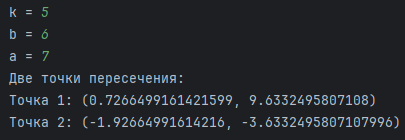
catch (e: Exception){

*println*("неверный формат данных")

}

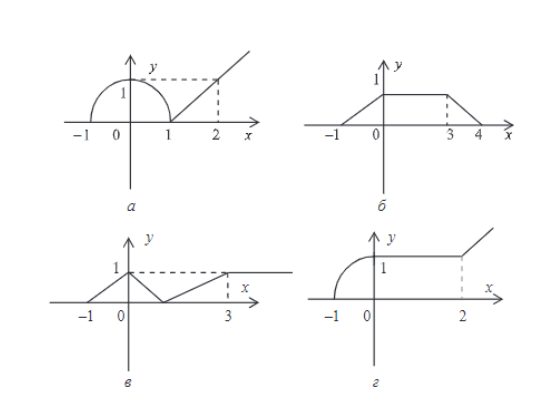
}

**Тестовые ситуации**



**Задание 20:**

****

****

**Входные данные:**

**x -** вещественный тип

**выходные данные:**

**y -** точки пересечения, вещественный тип

**Листинг программы (если есть)**

fun main() {

var x:Double = *readln*().*toDouble*()

var y:Double = 0.0

//a

when {

x <= 0 -> {

y = x+1

}

x <= 3 -> {

y = 1.0

}

x <= 4 -> {

y = 4-x

}

}

*println*(y)

//б

when {

x <= 1 -> {

y = (Math.*PI*/2) \* (x+1)

}

x > 1 -> {

y = x - 1

}

}

*println*(y)

//в

when {

x <= 0 -> {

y = x+1

}

x <= 1 -> {

y = 1 - x

}

x <= 3 -> {

y = 1 - (x/3)

}

x > 3 -> {

y = 1.0

}

}

*println*(y)

//г

when {

x <= 0 -> {

y = (Math.*PI*/2) \* (x+1)

}

x <= 2 -> {

y = 1.0

}

x > 2 -> {

y = x+1

}

}

*println*(y)

}

**Тестовые ситуации**

