Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по программе «**Практическое занятие 7.2**»

Выполнил: Лешуков Денис Юрьевич

Группа: ПР-21

Преподаватель: Мирошниченко Г.В

2025

**Задание**: Вариант №10

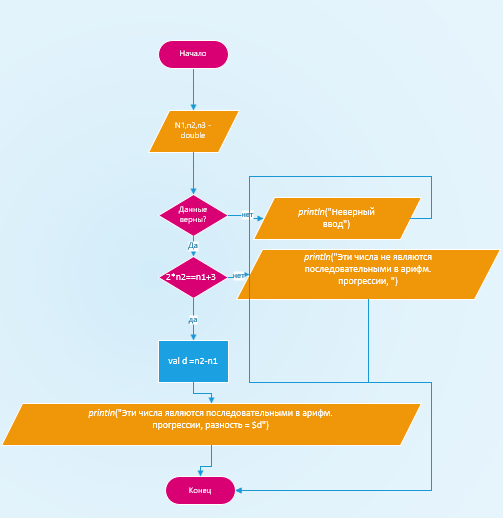
**Входные данные:**

n1, n2, n3– первые три числа арифм. прогрессии, double

**Выходные данные:**

d – разность, double

**Блок-схема:**



**Листинг программы:**

import kotlin.math.sqrt

fun main()

{

try {

println("Введите первое число")

var n1 = readln().toDouble()

println("Введите второе")

var n2 = readln().toDouble()

println("Введите третье")

var n3 = readln().toDouble()

when

{

2\*n2 == n1+n3 -> {

val d =n2-n1

println("Эти числа являются последовательными в арифм. прогрессии, разность = $d")

}

else -> {

println("Эти числа не являются последовательными в арифм. прогрессии, ")

}

}

}

catch (e:Exception){

println("Неверный ввод")

}

}

**Тестовые ситуации:**

**Задание**: Вариант №20

**Входные данные:**

x– переменная для ф-ии, double

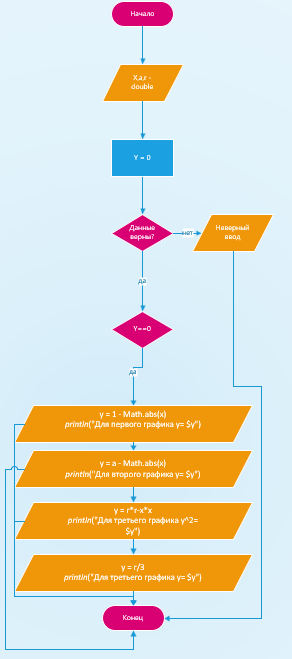
a-переменная для ф-ии, double

r- радиус, double

**Выходные данные:**

y – значение ф-ии, double

**Блок-схема:**



**Листинг программы:**

fun main()

{

try {

println("Введите x")

var x = readln().toDouble()

println("Введите a")

var a = readln().toDouble()

println("Введите R")

var r = readln().toDouble()

var y: Any = 0

when

{

y == 0 -> {

y = 1 - Math.abs(x)

println("Для первого графика y= $y")

y = a - Math.abs(x)

println("Для второго графика y= $y")

y = r\*r-x\*x

println("Для третьего графика y^2= $y")

y = r/3

println("Для третьего графика y= $y")

}

}

}

catch (e:Exception){

println("Неверный ввод")

}

}

**Тестовые ситуации:**

**Задание**: Вариант №30

**Входные данные:**

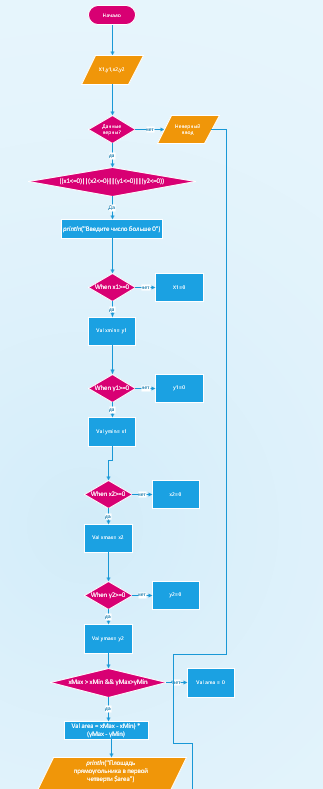
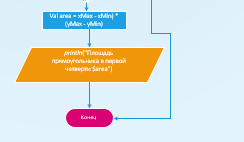
x1, y1–координаты вершин, double

x2, y2–координаты вершин, double

**Выходные данные:**

area – площадь прямоугольника в первой координатной четверти , double

**Блок-схема:**

**Листинг программы:**

fun main()

{

try {

println("Введите x1")

var x1 = readln().toDouble()

println("Введите y1")

var y1 = readln().toDouble()

println("Введите x2")

var x2 = readln().toDouble()

println("Введите y2")

var y2 = readln().toDouble()

when

{

((x1<=0)||(x2<=0)||(y1<=0)||(y2<=0)) -> {

println("Введите число больше 0")

}

}

val xMin = when

{

x1>=0->x1

else-> 0.0

}

val yMin = when

{

y1>=0->y1

else-> 0.0

}

val xMax = when

{

x2>=0->x2

else-> 0.0

}

val yMax = when

{

y2>=0->y2

else-> 0.0

}

val area = when{

xMax > xMin && yMax>yMin -> (xMax - xMin) \* (yMax - yMin)

else-> 0.0

}

println("Площадь прямоугольника в первой четверти $area")

}

catch (e:Exception){

println("Неверный ввод")

}

}

**Тестовые ситуации:**

**Задание**: Вариант №40

**Входные данные:**

k–коэф. числа, double

a – коэф. числа, double

b - коэф. числа, double

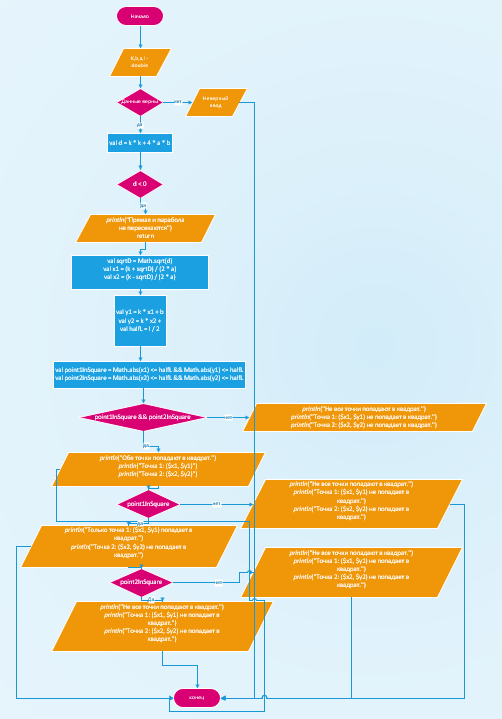
l – длина прямой,double

**Выходные данные:**

String.Строка с информацией о:

x1,y1,x2,y2 – позиция точки, попадает она в квадрат или нет.,duoble

**Блок-схема:**



**Листинг программы:**

fun main() {

try{

println("Введите k")

var k = readln().toDouble()

println("Введите b")

var b = readln().toDouble()

println("Введите a")

var a = readln().toDouble()

println("Введите l")

var l = readln().toDouble()

val d = k \* k + 4 \* a \* b

when{

(d < 0) ->{

println("Прямая и парабола не пересекаются")

return

}

}

val sqrtD = Math.sqrt(d)

val x1 = (k + sqrtD) / (2 \* a)

val x2 = (k - sqrtD) / (2 \* a)

val y1 = k \* x1 + b

val y2 = k \* x2 + b

val halfL = l / 2

val point1InSquare = Math.abs(x1) <= halfL && Math.abs(y1) <= halfL

val point2InSquare = Math.abs(x2) <= halfL && Math.abs(y2) <= halfL

when {

point1InSquare && point2InSquare -> {

println("Обе точки попадают в квадрат.")

println("Точка 1: ($x1, $y1)")

println("Точка 2: ($x2, $y2)")

}

point1InSquare -> {

println("Только точка 1: ($x1, $y1) попадает в квадрат.")

println("Точка 2: ($x2, $y2) не попадает в квадрат.")

}

point2InSquare -> {

println("Только точка 2: ($x2, $y2) попадает в квадрат.")

println("Точка 1: ($x1, $y1) не попадает в квадрат.")

}

else -> {

println("Не все точки попадают в квадрат.")

println("Точка 1: ($x1, $y1) не попадает в квадрат.")

println("Точка 2: ($x2, $y2) не попадает в квадрат.")

}

}

}

catch (e:Exception)

{

println("Неверный ввод")

}

**Тестовые ситуации:**

**Задание**: Вариант №43

**Входные данные:**

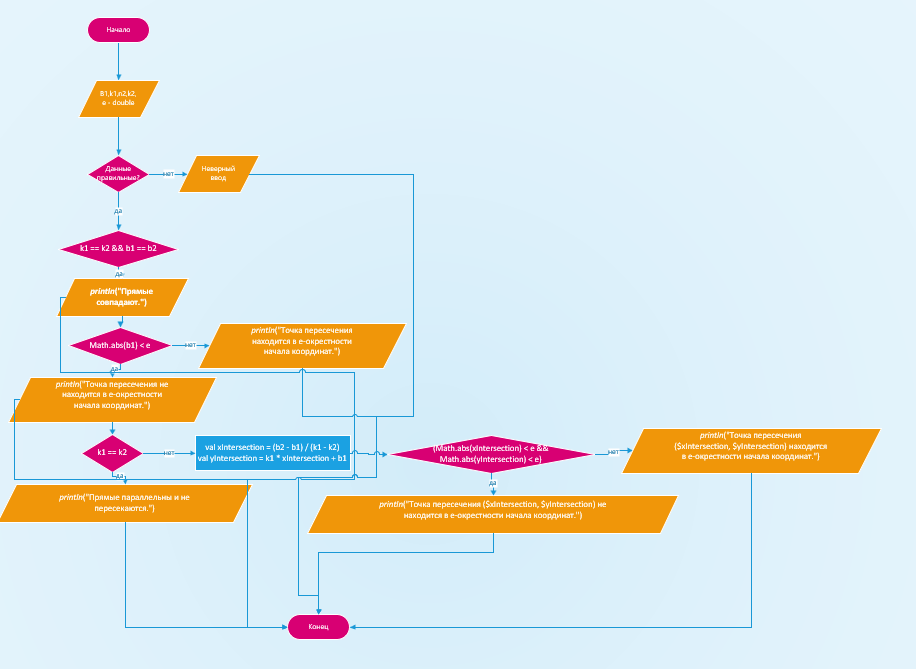
k1, b2, k2, b2–координаты прямых, double

e - эпсилон, double

**Выходные данные:**

xIntersection, yIntersection – x и y координаты, о том находятся ли они в окрестности эпсилона, double

**Блок-схема:**



**Листинг программы:**

fun main() {

try{

println("Введите k1")

var k1 = readln().toDouble()

println("Введите b1")

var b1 = readln().toDouble()

println("Введите k2")

var k2 = readln().toDouble()

println("Введите b2")

var b2 = readln().toDouble()

println("Введите e")

var e = readln().toDouble()

when {

k1 == k2 && b1 == b2 -> {

println("Прямые совпадают.")

when {

(Math.abs(b1) < e)-> {

println("Точка пересечения находится в e-окрестности начала координат.")

}

else -> {

println("Точка пересечения не находится в e-окрестности начала координат.")

}

}

}

k1 == k2 -> {

println("Прямые параллельны и не пересекаются.")

}

else -> {

val xIntersection = (b2 - b1) / (k1 - k2)

val yIntersection = k1 \* xIntersection + b1

when{

(Math.abs(xIntersection) < e && Math.abs(yIntersection) < e) -> {

println("Точка пересечения ($xIntersection, $yIntersection) находится в e-окрестности начала координат.")

}

else ->{

println("Точка пересечения ($xIntersection, $yIntersection) не находится в e-окрестности начала координат.")

}

}

}

}

}

catch (e:Exception)

{

println("Неверный ввод")

}

}

**Тестовые ситуации:**

**Вывод:**

Я научился работать с оператором when.