Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по программе «**Практическое занятие 7\_1 7\_2 7\_3**»

Выполнил: Максимов Кирилл Эдуардович

Группа: ПР-21

Преподаватель: Мирошниченко Г.В

2023

**Задание:**

**Входные и выходные данные**

**Блок-схема**

**Листинг программы (если есть)**

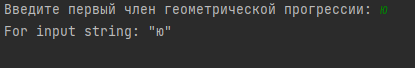
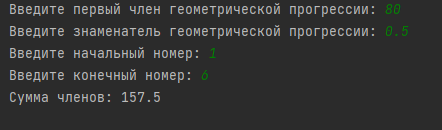
**Тестовые ситуации**

**Вывод**

3. Заданы первый член и знаменатель геометрической прогрессии. Найти сумму членов с номерами от заданного номера К до заданного номера Р (считать К < P).

Входные данные: b, q, k, p – вещественные числа (double)  
Выходные данные: sum – вещественное число (double)

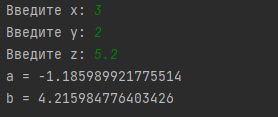
fun main()  
{  
 try  
 {  
 *print*("Введите первый член геометрической прогрессии: ")  
 var b = *readLine*()!!.*toDouble*()  
 *print*("Введите знаменатель геометрической прогрессии: ")  
 var q = *readLine*()!!.*toDouble*()  
 *print*("Введите начальный номер: ")  
 var k = *readLine*()!!.*toDouble*()  
 *print*("Введите конечный номер: ")  
 var p = *readLine*()!!.*toDouble*()  
 if (k > p) throw Exception ("K должно быть меньше P")  
 var n = p - k + 1  
 var sum = b \* (Math.pow(q, n) - 1) / (q - 1)  
 *println*("Сумма членов: $sum")  
 }  
 catch (ex: Exception)  
 {  
 *println*(ex.message)  
 }  
}



7(3). Заданы x, y, z. Вычислить a, b если:  
Входные данные: x, y, z – вещественные числа (double)

Выходные данные: a, b – вещественные числа (double)

import kotlin.math.*PI*import kotlin.math.cos  
import kotlin.math.pow  
import kotlin.math.sin  
  
fun main()  
{  
 try  
 {  
 *print*("Введите x: ")  
 val x = *readLine*()!!.*toDouble*()  
 *print*("Введите y: ")  
 val y = *readLine*()!!.*toDouble*()  
 *print*("Введите z: ")  
 val z = *readLine*()!!.*toDouble*()  
 val a = 2 \* *cos*(x - *PI*/6) / (0.5 + *sin*(y).*pow*(2))  
 val b = 1 + z.*pow*(2) / (3 + (z.*pow*(2) / 5))  
 *println*("a = $a \nb = $b")  
 }  
 catch (ex: Exception)  
 {  
 *println*(ex.message)  
 }  
}

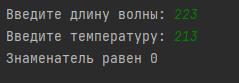
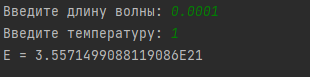


24. Написать программу, вычисляющую энергию Е, излучаемую черным телом на волне длиной при температуре Т, по формуле

Входные данные: lymda, temperature – вещественные числа (double)

Выходные данные: energy – вещественное числе (double)

import kotlin.math.*E*import kotlin.math.*PI*import kotlin.math.pow  
  
fun main()  
{  
 try  
 {  
 val c = 2.997924 \* 10.0.*pow*(8.0)  
 val h = 6.6252 \* 10.0.*pow*(-34.0)  
 val b = 5.6687 \* 10.0.*pow*(-8.0)  
 *print*("Введите длину волны: ")  
 var lyamda = *readLine*()!!.*toDouble*()  
 *print*("Введите температуру: ")  
 var temperature = *readLine*()!!.*toDouble*()  
 var energy = (2 \* *PI* \* c \* h \* lyamda.*pow*(-8)) / (*E*.*pow*((c \* h) / (b \* lyamda \* temperature)) - 1.0)  
 if (b \* lyamda \* temperature == 0.0 || *E*.*pow*((c \* h) / (b \* lyamda \* temperature)) - 1.0 == 0.0)  
 throw Exception("Знаменатель равен 0")  
 *println*("E = $energy")  
 }  
 catch (ex: Exception)  
 {  
 *println*(ex.message)  
 }  
  
}

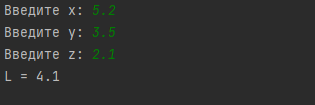


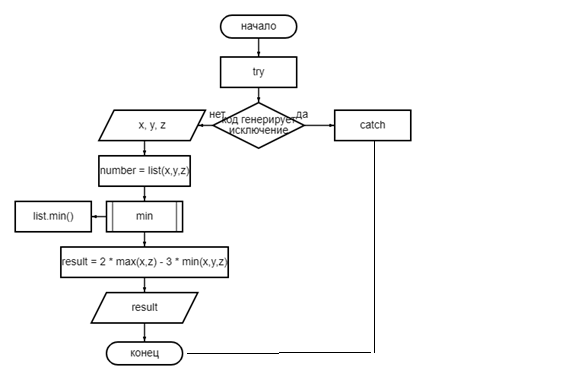
3. Даны вещественные числа x, y, z. Получить:

Входные данные: x, y, z – вещественные числа (double)

Выходные данные: result – вещественное число (double)

import kotlin.math.\*  
  
fun Min(list: List<Double>): Double? {  
 return list.*min*()  
}  
  
fun main()  
{  
 try  
 {  
 *print*("Введите x: ")  
 val x = *readLine*()!!.*toDouble*()  
 *print*("Введите y: ")  
 val y = *readLine*()!!.*toDouble*()  
 *print*("Введите z: ")  
 val z = *readLine*()!!.*toDouble*()  
 val numbers = *listOf*(x, y, z)  
 var min = *Min*(numbers)  
 val result : Double = 2 \* *max*(x, z) - 3 \* min!!.toDouble()  
 *println*("L = $result")  
 }  
 catch (ex: Exception)  
 {  
 *println*(ex.message)  
 }  
}

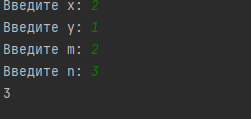


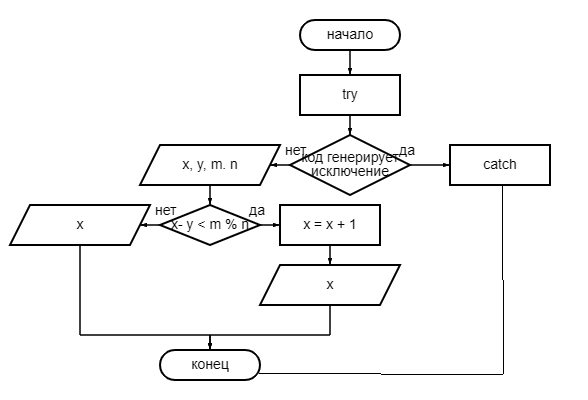


14. Заданы целые числа, у, т, п. Если разность (х - у) меньше остатка от деления на п, увеличить х на 1.

Входные данные: x, y, m, n – целые числа (int)  
Выходные данные: x – целое число (int)

fun main()  
{  
 try  
 {  
 *print*("Введите x: ")  
 var x = *readLine*()!!.*toInt*()  
 *print*("Введите y: ")  
 var y = *readLine*()!!.*toInt*()  
 *print*("Введите m: ")  
 var m = *readLine*()!!.*toInt*()  
 *print*("Введите n: ")  
 var n = *readLine*()!!.*toInt*()  
  
 if((x - y) < (m % n)) x++  
 *println*("$x")  
 }  
 catch (ex: Exception)  
 {  
 *println*(ex.message)  
 }  
  
}

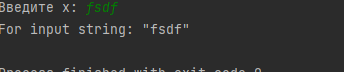
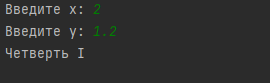
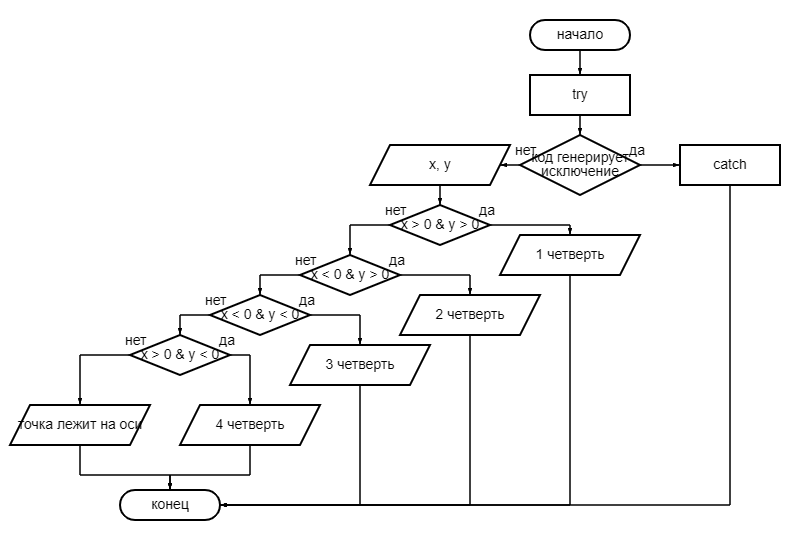




24. Определить, в какой координатной четверти находится заданная точка с координатами (х, у).

Входные данные: x, y – вещественные числа (double)  
Выходные данные: println – текстовый тип (string)

fun main()  
{  
 try  
 {  
 *print*("Введите x: ")  
 var x = *readLine*()!!.*toDouble*()  
 *print*("Введите y: ")  
 var y = *readLine*()!!.*toDouble*()  
 if (x > 0 && y > 0) *println*("Четверть I")  
 else if (x < 0 && y > 0) *println*("Четверть II")  
 else if (x < 0 && y < 0) *println*("Четверть III")  
 else if (x > 0 && y < 0) *println*("Четверть IV")  
 else *println*("Точка лежит на оси")  
 }  
 catch (ex: Exception)  
 {  
 *println*(ex.message)  
 }  
}

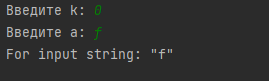
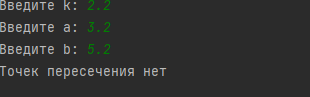
  


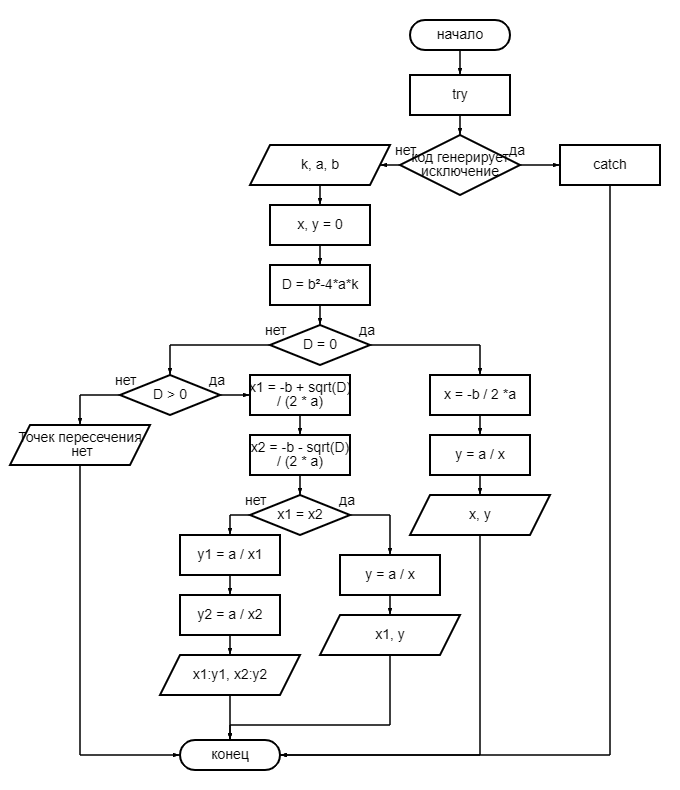
34. Заданы числа k, а, b. Определить число точек пересечения прямой у = kx + b с гиперболой у = a/x и их координаты.

Входные данные: k, a ,b – вещественные числа (double)

Выходные данные: x, y, x1, x2, y1, y2 – вещественные числа (double)

import kotlin.math.pow  
import kotlin.math.sqrt  
  
fun main()  
{  
 try {  
 *print*("Введите k: ")  
 val k = *readLine*()!!.*toDouble*()  
 *print*("Введите a: ")  
 val a = *readln*()!!.*toDouble*()  
 *print*("Введите b: ")  
 val b = *readln*()!!.*toDouble*()  
 var x: Double = 0.0  
 var y: Double = 0.0  
  
 val D = b.*pow*(2) - 4 \* a \* k  
  
 if (D == 0.0) {  
 x = -b / 2.0 \* a  
 y = a / x  
 *println*("Точка пересечения ($x;$y)")  
 } else if (D > 0) {  
 val x1 = -b + *sqrt*(D) / (2 \* a)  
 val x2 = -b - *sqrt*(D) / (2 \* a)  
 if (x1 == x2) {  
 val y = a / x  
 *println*("Точка пересечения ($x1;$y)")  
 } else {  
 val y1 = a / x1  
 val y2 = a / x2  
 *println*("Точки пересечения ($x1;$y1) ($x2;$y2)")  
 }  
  
 } else *println*("Точек пересечения нет")  
 }  
 catch (ex : Exception)  
 {  
 *println*(ex.message)  
 }  
  
}





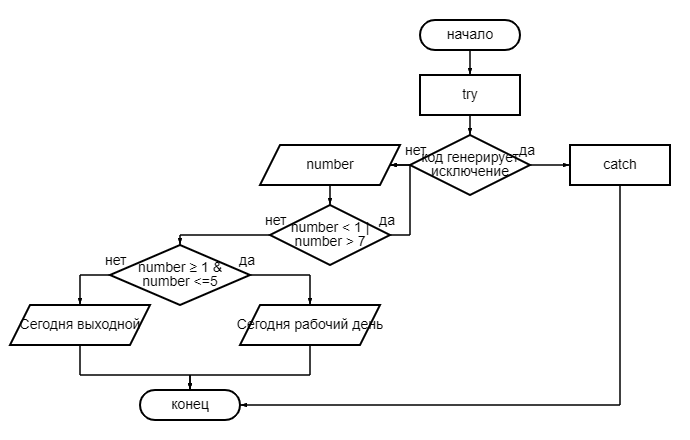
46. По номеру дня недели выдать сообщение, каким он является - рабочим или выходным.

Входные данные: number – целое число (int)

Выходные данные: println – текстовых тип (string)

fun main()  
{  
 try  
 {  
 var number : Int  
 do {  
 *print*("Введите номер дня недели: ")  
 number = *readLine*()!!.*toInt*()  
 }while (number < 1 || number > 7)  
  
 when  
 {  
 (number >= 1 && number <= 5) -> *println*("Сегодня рабочий день")  
 else -> *println*("Сегодня выходной")  
 }  
  
 }  
 catch (ex: Exception)  
 {  
 *println*(ex.message)  
 }  
}



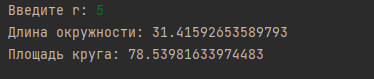


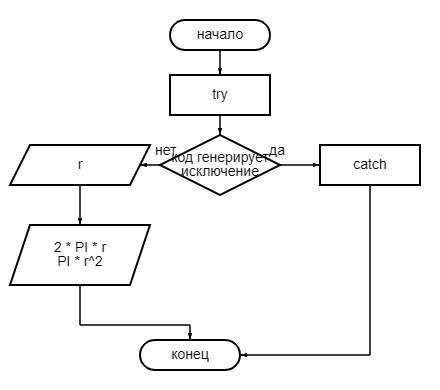
12. Найти длинну окружности и площадь круга заданного радиусу r

Входные данные: r – вещественное число (double)

Выходные данные: длина окр. Площадь круга – double

import kotlin.math.*PI*import kotlin.math.pow  
  
fun main()  
{  
 try  
 {  
 *print*("Введите r: ")  
 val r = *readLine*()!!.*toDouble*()  
  
 *println*("Длина окружности: ${2 \* *PI* \* r}")  
 *println*("Площадь круга: ${*PI* \* r.*pow*(2)}")  
 }  
 catch (ex: Exception)  
 {  
 *println*(ex.message)  
 }  
}

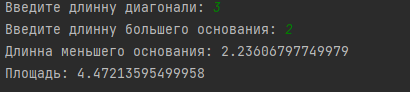


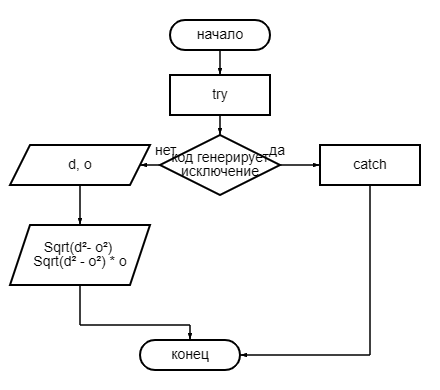


21. Найти площадь и меньшее основание прямоугольника по заданной диагонали и большему основанию.  
Входные данные: d, o – вещественные числа (double)

Выходные данные: Длина меньшего основания, Площадь - double

import kotlin.math.pow  
  
fun main()  
{  
 try {  
 *print*("Введите длинну диагонали: ")  
 val d = *readLine*()!!.*toDouble*()  
 *print*("Введите длинну большего основания: ")  
 val o = *readLine*()!!.*toDouble*()  
  
 *println*("Длинна меньшего основания: ${Math.sqrt(d.*pow*(2) - o.*pow*(2))}")  
 *println*("Площадь: ${(Math.sqrt(d.*pow*(2) - o.*pow*(2))) \* o}")  
 }  
 catch (ex: Exception)  
 {  
 *println*(ex.message)  
 }  
}





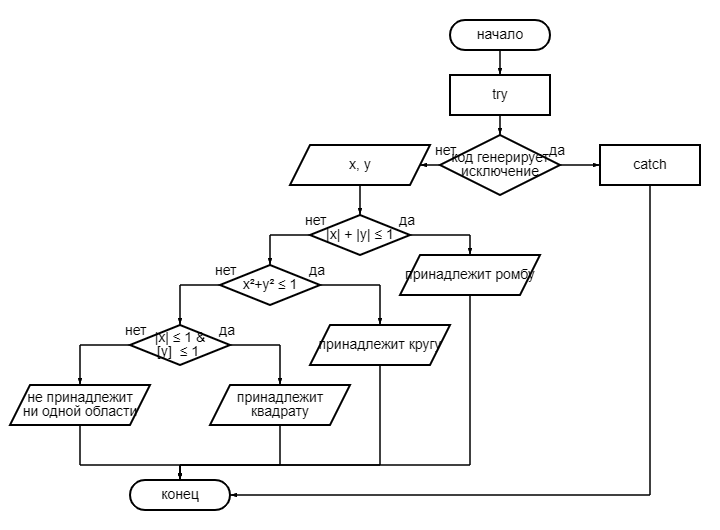
12. Даны координаты точки на плоскости. Выяснить, принадлежит ли эта точка одной из вложенных областей (ромбу, кругу, квадрату). Ответ выдать в виде строки текста.

Входные данные: x, y – вещественные числа (double)

Выходные данные: println – текстовое сообщение (string)

import kotlin.math.pow  
  
fun main()  
{  
 try {  
 *print*("Введите x: ")  
 val x = *readLine*()!!.*toDouble*()  
 *print*("Введите y: ")  
 val y = *readLine*()!!.*toDouble*()  
  
 if(Math.abs(x) + Math.abs(y) <= 1) *println*("Точка принадлежит ромбу")  
 else if(x.*pow*(2) + y.*pow*(2) <= 1) *println*("Точка принадлежит кругу")  
 else if(Math.abs(x) <= 1 && Math.abs(y) <= 1) *println*("Точка принадлежит квадрату")  
 else *println*("Точка не принадлежит ни одной области")  
 }  
 catch (ex: Exception)  
 {  
 *println*(ex.message)  
 }  
}





21. Выяснить, сколько точек пересечения имеют прямая y = kx + b и окружность радиуса r с центром в начале координат. Значения k, b, r ввести.

Входные данные: k, b, r – вещественные числа (double)

Выходные данные: x, x1, x2, y, y1, y2 – вещественные числа (double)

import kotlin.math.pow  
  
fun main()  
{  
 try {  
 *print*("Введите k: ")  
 val k = *readLine*()!!.*toDouble*()  
 *print*("Введите b: ")  
 val b = *readLine*()!!.*toDouble*()  
 *print*("Введите r: ")  
 val r = *readLine*()!!.*toDouble*()  
  
 var aa = k.*pow*(2) + 1  
 var bb = 2 \* k \* b  
 var cc = b.*pow*(2) - r.*pow*(2)  
 var D = bb.*pow*(2) - 4 \* aa \* cc  
  
 if(D < 0) *println*("Точек пересечения нет")  
 else if (D > 0)  
 {  
 var x1 = (-bb+Math.sqrt(D)) / (2 \* aa)  
 var x2 = (-bb-Math.sqrt(D)) / (2 \* aa)  
 var y1 = k \* x1 + b  
 var y2 = k \* x2 + b  
  
 *println*("Точи пересечения ($x1;$y1) ($x2;$y2)")  
 }  
 else{  
 var x = bb / (2 \* aa)  
 var y = k \* x + b  
 *println*("Точи пересечения ($x;$y)")  
 }  
 }  
 catch (ex: Exception)  
 {  
 *println*(ex.message)  
 }  
}

