Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

# Отчёт по программе «Практическое занятие 7.1 задание 2»

Выполнил: Лежень Андрей Игоревич

Группа: ПР-23

Преподаватель: Мирошниченко Г.В

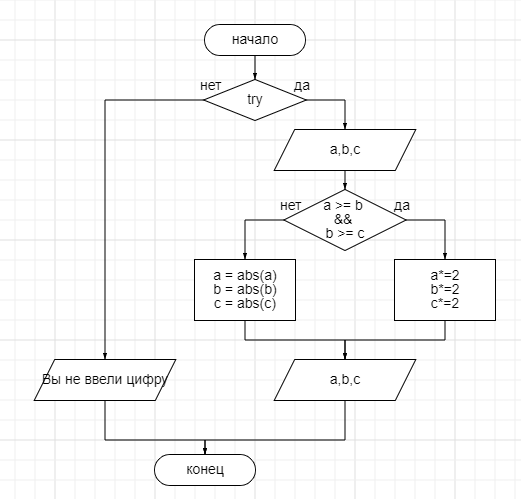
2023

**Задание:  
**

**Входные и выходные данные**

A,b,c – переменные содержащие числа, double

**Блок-схема**

****

**Тестовые ситуации**

if (a >= b && b >= c)

try {} catch (a: Exception)

{}

**Вывод**

Я узнал, как использовать if в kotlin

Листинг программы:

import kotlin.math.abs  
fun main() {  
 try {  
 var a = *readln*().*toDouble*()  
 var b = *readln*().*toDouble*()  
 var c = *readln*().*toDouble*()  
 if (a >= b && b >= c){  
 a\*=2  
 b\*=2  
 c\*=2  
 }  
 else{  
 a = *abs*(a)  
 b = *abs*(b)  
 c = *abs*(c)  
 }  
 *println*("$a,$b,$c")  
 }  
 catch (a: Exception) {  
 *println*("Вы не ввели цифру")}  
}

**Задание:**

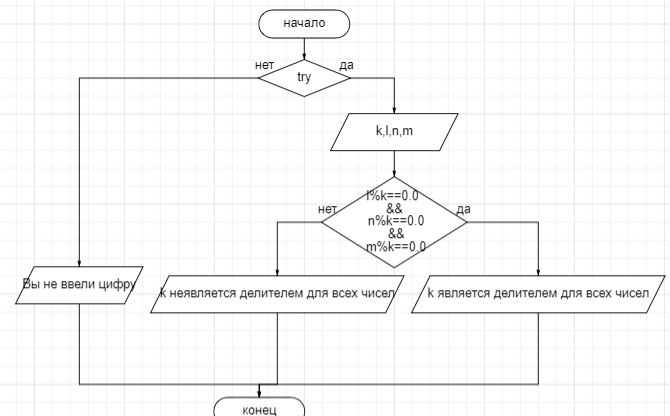
****

**Входные и выходные данные**

K – делитель, double

L,n,m – делимые, double

**Блок-схема**

****

**Тестовые ситуации**

if (l%k==0.0 && n%k==0.0 && m%k==0.0)

try {} catch (a: Exception)

{}

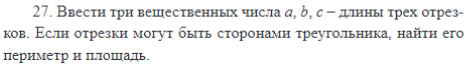
**Вывод**

Я узнал, как использовать if в kotlin

Листинг программы:

fun main(){  
 try {  
 var k = *readln*().*toDouble*()  
 var l = *readln*().*toDouble*()  
 var n = *readln*().*toDouble*()  
 var m = *readln*().*toDouble*()  
 if (l%k==0.0 && n%k==0.0 && m%k==0.0){  
 *println*("k является делителем для всех чисел")  
 }  
 else{  
 *println*("k не является делителем для всех чисел")  
 }  
 }  
 catch (a: Exception) {  
 *println*("Вы не ввели цифру")}  
}

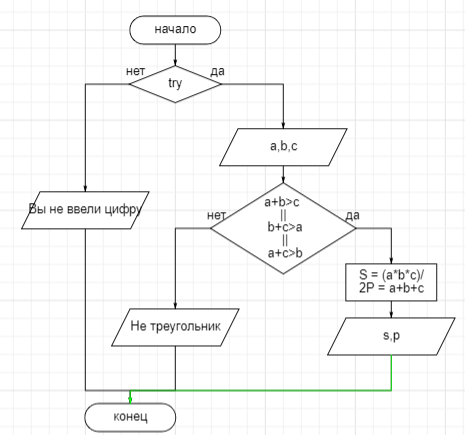
**Задание:**

****

**Входные и выходные данные**

A,b,c – стороны треугольника, double

**Блок-схема**

****

**Тестовые ситуации**

if (a+b>c || b+c>a || a+c>b)

try {} catch (a: Exception)

{}

**Вывод**

Я закрепил знания по использованию if в kotlin

Листинг программы:

fun main(){  
 try {  
 val a = *readln*().*toDouble*()  
 val b = *readln*().*toDouble*()  
 val c = *readln*().*toDouble*()  
 if (a+b>c || b+c>a || a+c>b){  
 val S = (a\*b\*c)/2  
 val P = a+b+c  
 *println*("Площадь = $S, периметр = $P")  
 }  
 else{  
 *println*("Не треугольник")  
 }  
 }  
 catch (a: Exception) {  
 *println*("Вы не ввели цифру")}  
}

**Задание:**

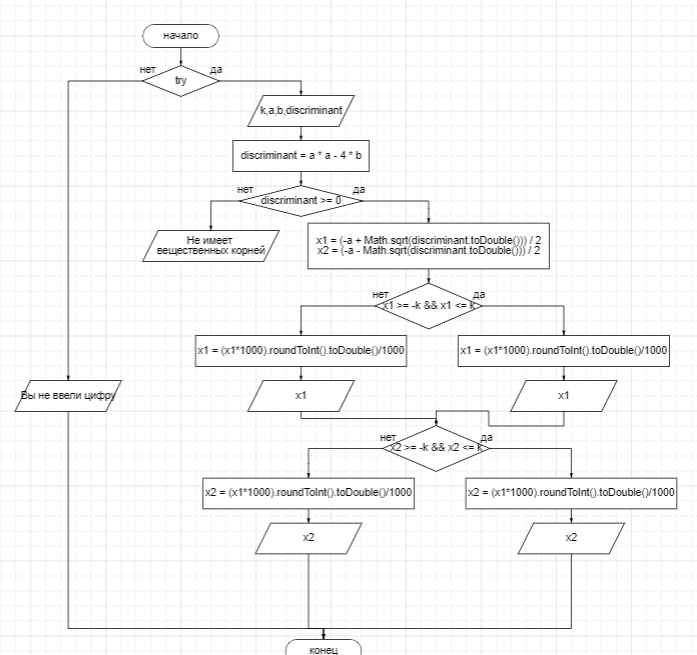
**Входные и выходные данные**

A,b,k – числа, double

Discriminant – дискриминант, double

X1, x2 - корни, double

**Блок-схема**

****

**Тестовые ситуации**

if (x1 >= -k && x1 <= k)

if (x2 >= -k && x2 <= k)

try {} catch (a: Exception)

{}

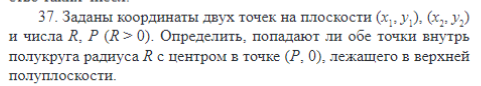
**Вывод**

Я закрепил знания использования if в kotlin

Листинг программы:

import kotlin.math.roundToInt  
  
fun main(){  
 try {  
 val k = *readln*().*toDouble*()  
 val a = *readln*().*toDouble*()  
 val b = *readln*().*toDouble*()  
 val discriminant = a \* a - 4 \* b  
 if (discriminant >= 0) {  
 var x1 = (-a + Math.sqrt(discriminant.toDouble())) / 2  
 var x2 = (-a - Math.sqrt(discriminant.toDouble())) / 2  
 if (x1 >= -k && x1 <= k) {  
 x1 = (x1\*1000).*roundToInt*().toDouble()/1000  
 *println*("x1 = $x1 в интервале (-$k, $k)")  
 }  
 else {  
 x1 = (x1\*1000).*roundToInt*().toDouble()/1000  
 *println*("x1 = $x1 не в интервале (-$k, $k)")  
 }  
 if (x2 >= -k && x2 <= k) {  
 x2 = (x2\*1000).*roundToInt*().toDouble()/1000  
 *println*("x2 = $x2 в интервале (-$k, $k)")  
 }  
 else {  
 x2 = (x2\*1000).*roundToInt*().toDouble()/1000  
 *println*("x2 = $x2 не в интервале (-$k, $k)")  
 }  
 }  
 else {  
 *println*("Не имеет вещественных корней")  
 }  
 }  
 catch (a: Exception) {  
 *println*("Вы не ввели цифру")}  
}

**Задание:**

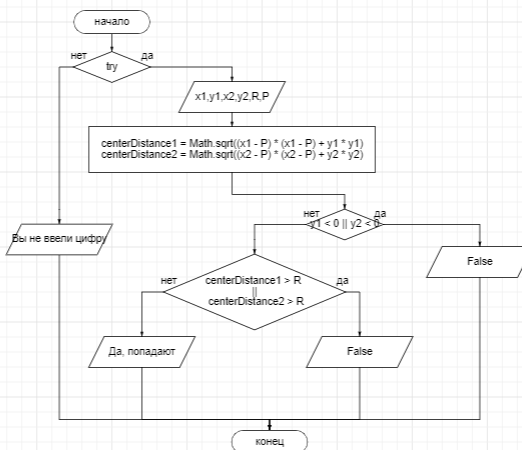
****

**Входные и выходные данные**

X1,y1,x2,y2 – координаты 1 и 2 точек соответственно, double

R,P – размеры полукруга ,double

**Блок-схема**

****

**Тестовые ситуации**

if (y1 < 0 || y2 < 0)   
else if (centerDistance1 > R || centerDistance2 > R) {  
try {} catch (a: Exception)

{}

**Вывод**

Я закрепил знания использования if в kotlin

Листинг программы:

fun main(){  
 try {  
 val x1 = *readln*().*toDouble*()  
 val y1 = *readln*().*toDouble*()  
 val x2 = *readln*().*toDouble*()  
 val y2 = *readln*().*toDouble*()  
 val R = *readln*().*toDouble*()  
 val P = *readln*().*toDouble*()  
 val centerDistance1 = Math.sqrt((x1 - P) \* (x1 - P) + y1 \* y1)  
 val centerDistance2 = Math.sqrt((x2 - P) \* (x2 - P) + y2 \* y2)  
 if (y1 < 0 || y2 < 0) {  
 *println*("False")  
 }  
 else if (centerDistance1 > R || centerDistance2 > R) {  
 *println*("False")  
 }  
 else{  
 *println*("Да, попадают");  
 }  
 }  
 catch (a: Exception) {  
 *println*("Вы не ввели цифру")}  
}