Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по программе «Программирование линейных и разветвляющихся алгоритмов»

Выполнил: Лазаренко Александр Алексеевич

Группа: ПР-24

Преподаватель: Мирошниченко Г.В

2023

**Задание: 1**



**Входные и выходные данные**

Int r1 – первое число

Int r2 – второе число

Int r3 – третье число

Ro-Общее сопротевление

**Блок-схема**

**Листинг программы (если есть)**

fun main(){  
 try {  
 *println*("Введите первое число R1 ")  
 var r1=*readLine*()!!.*toInt*()  
 *println*("Введите второе число R2 ")  
 var r2=*readLine*()!!.*toInt*()  
 *println*("Введите второе число R3 ")  
 var r3=*readLine*()!!.*toInt*()  
 var ro=1/((1/r1)+(1/r2)+(1/r3))  
 *println*("сопротивление при параллельном соединении $ro")  
 }catch (e: NumberFormatException)  
 {  
 *println*("Надо вводить только числа")  
 }  
}

**Тестовые ситуации**

**Try{}catch** (e: NumberFormatException) { *println*("Надо вводить только числа")}

**Задание: 2**



**Входные и выходные данные**

double x1 – 1 число

double x2 – 2 число

srk-Среднее арифметическое квадратов

srm-Среднее арифметическое модулей

**Блок-схема**

**Листинг программы (если есть)**

import kotlin.math.pow  
import kotlin.math.abs  
  
fun main(){  
 try {  
 *println*("Введите первое число x1 ")  
 var x1=*readLine*()!!.*toDouble*()  
 *println*("Введите второе число x2 ")  
 var x2=*readLine*()!!.*toDouble*()  
 var srk=x1.*pow*(2)+x2.*pow*(2)  
 srk=srk/2  
 var srm= *abs*(x1)+ *abs*(x2)  
 srm=srm/2  
 *println*("Срденее арифметическое квадратов $srk и Срденее арифметическое модулей $srm")  
 }catch (e: NumberFormatException)  
 {  
 *println*("Надо вводить только числа")  
 }  
}

**Тестовые ситуации**

**Try{}catch** (e: NumberFormatException) { *println*("Надо вводить только числа")}

**Задание: 3**



**Входные и выходные данные**

double r1 – внутренний радиус

double r2 – внешний радиус

s - площадь

**Блок-схема**

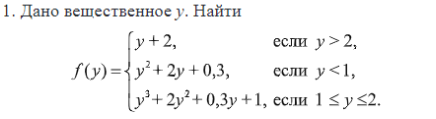
**Листинг программы (если есть)**

import kotlin.math.*PI*fun main(){  
 try {  
 *println*("Введите внутренний радиус кольца ")  
 var r1=*readLine*()!!.*toDouble*()  
 *println*("Введите внешний радиус кольца ")  
 var r2=*readLine*()!!.*toDouble*()  
 if (r2>r1)  
 {  
 var s= *PI*\*(r2-r1)  
 *println*("Площадь кольца = $s")  
 }else *println*("Внешний радиус кольца меньше внутреннего радиуса")  
 }catch (e: NumberFormatException)  
 {  
 *println*("Надо вводить только числа")  
 }  
}

**Тестовые ситуации**

**Try{}catch** (e: NumberFormatException) { *println*("Надо вводить только числа")}

**Задание: 4**



**Входные и выходные данные**

int y– число

**Блок-схема**

**Листинг программы (если есть)**

import kotlin.math.pow  
  
fun main(){  
 try {  
 *println*("Введите число y ")  
 var y=*readLine*()!!.*toDouble*()  
 var f=y  
 when {  
 (y>2)->y=y+2  
 (y<1)->y=y.*pow*(2)+(2\*y)+0.3  
 (y>=1 && y<=2)->y=(y.*pow*(3))+(2\*y.*pow*(2))+(0.3\*y)+1  
 }  
 *println*("f($f)=$y")  
 }catch (e: NumberFormatException)  
 {  
 *println*("Надо вводить только числа")  
 }  
}

**Тестовые ситуации**

**Try{}catch** (e: NumberFormatException) { *println*("Надо вводить только числа")}

**Задание: 5**



**Входные и выходные данные**

double a – 1 сторона

double b – 2 сторона

double c – 3 сторона

**Блок-схема**

**Листинг программы (если есть)**

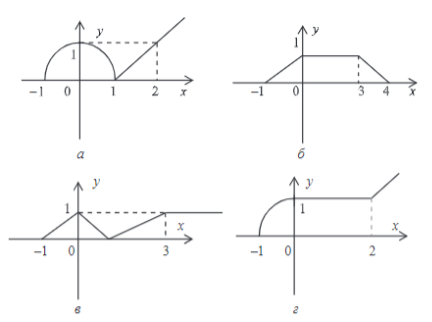
import kotlin.math.sqrt  
  
fun main(){  
 try {  
 *println*("Введите 1 сторону a ")  
 var a=*readLine*()!!.*toDouble*()  
 *println*("Введите 2 сторону b ")  
 var b=*readLine*()!!.*toDouble*()  
 *println*("Введите 3 сторону c ")  
 var c=*readLine*()!!.*toDouble*()  
 if (a+b>c && a+c>b && b+c>a)  
 {*println*("Треуголник можно построить")  
 if (*sqrt*(a)== *sqrt*(b)+ *sqrt*(c) || *sqrt*(b)== *sqrt*(a)+ *sqrt*(c) || *sqrt*(c)== *sqrt*(a)+ *sqrt*(b))  
 {  
 *println*("Треуголник будет прямоугольным")  
 }  
 }  
 else *println*("Треуголник нельзя построить")  
 }catch (e: NumberFormatException)  
 {  
 *println*("Надо вводить только числа")  
 }  
}

**Тестовые ситуации**

**Try{}catch** (e: NumberFormatException) { *println*("Надо вводить только числа")}

**Задание: 6**





**Входные и выходные данные**

double x – число

y-ответ

**Блок-схема**

**Листинг программы (если есть)**

import kotlin.math.pow  
import kotlin.math.sqrt  
  
fun main(){  
 try {  
 *println*("Введите число x ")  
 var x=*readLine*()!!.*toDouble*()  
 var y=0.0  
 when {  
 (x==-1.0)->y=0.0  
 (x>=-1 && x<0)->y=-1\*x.*pow*(2)+1  
 (x==0.0)->y=1.0  
 (x>0 && x<1)->y=-1\*x.*pow*(2)+1  
 (x==1.0)->y= 1.0  
 (x>1.0)->y=x-1  
 }  
 *println*("y($x)=$y")  
 }catch (e: NumberFormatException)  
 {  
 *println*("Надо вводить только числа")  
 }  
}

fun main(){  
 try {  
 *println*("Введите число x ")  
 var x=*readLine*()!!.*toDouble*()  
 var y=0.0  
 when {  
 (x==-1.0)->y=0.0  
 (x>=-1 && x<0)->y=1+x  
 (x==0.0)->y=1.0  
 (x>0 && x<=3)->y=1.0  
 (x>3.0 && x<=4)->y=-1\*x+4  
 }  
 *println*("y($x)=$y")  
 }catch (e: NumberFormatException)  
 {  
 *println*("Надо вводить только числа")  
 }  
}

import kotlin.math.pow  
  
fun main(){  
 try {  
 *println*("Введите число x ")  
 var x=*readLine*()!!.*toDouble*()  
 var y=0.0  
 when {  
 (x>=-1 && x<=0)->y=-1\*x.*pow*(2)+1  
 (x>0 && x<=1)->y=-1\*x+1  
 (x>1 && x<=3)->y=x/3  
 (x>3.0)->y=1.0  
 }  
 *println*("y($x)=$y")  
 }catch (e: NumberFormatException)  
 {  
 *println*("Надо вводить только числа")  
 }  
}

import kotlin.math.pow  
  
fun main(){  
 try {  
 *println*("Введите число x ")  
 var x=*readLine*()!!.*toDouble*()  
 var y=0.0  
 when {  
 (x>=-1 && x<=0)->y=-1\*x.*pow*(2)+1  
 (x>0 && x<=2)->y=1.0  
 (x>2.0)->y=x-2  
 }  
 *println*("y($x)=$y")  
 }catch (e: NumberFormatException)  
 {  
 *println*("Надо вводить только числа")  
 }  
}

**Тестовые ситуации**

**Try{}catch** (e: NumberFormatException) { *println*("Надо вводить только числа")}

**Вывод**

Проделав работу я выполнил задания и научился работать с командами в kotlin и Программирование линейных и разветвляющихся алгоритмов.