Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по программе «**Синтаксис языка Kotlin. Программирование линейных и разветвляющихся алгоритмов на языке Kotlin.**»

Выполнил: Плоских Александр Павлович

Группа: ПР-21

Преподаватель: Мирошниченко Г.В

2023

**Задача**

****

**Входные и выходные данные**

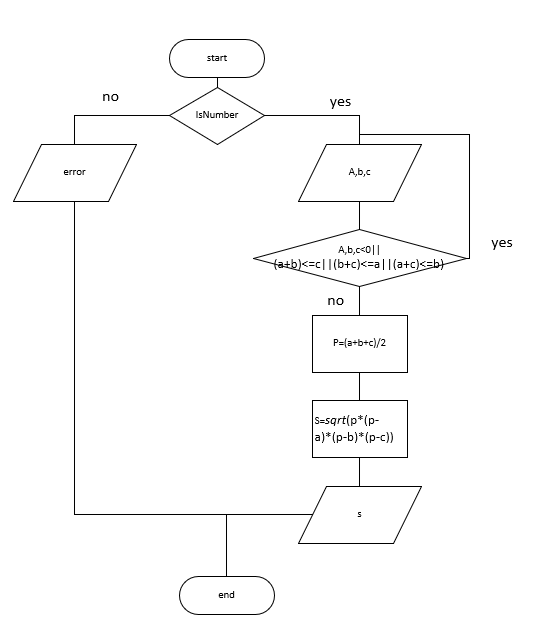
**Входные**

A,b,c-тип данных double,хранят в себе числа которые ввел пользователь

**Выходные**

S-тип данных double,вы водим число которое было получено по формуле из здания.

**Блок-схема**

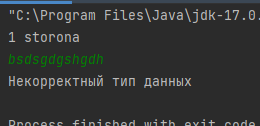
****

**Листинг программы**

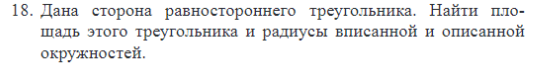
try{  
 var a: Double  
 var b: Double  
 var c: Double  
 var p: Double  
 var s:Double  
 do{  
 *println*("1 storona")  
 a= *readln*()!!.*toDouble*()  
 *println*("2 storona")  
 b=*readln*()!!.*toDouble*()  
 *println*("3 storona")  
 c= *readln*()!!.*toDouble*()  
 }while ((a<=0||b<=0||c<=0)||((a+b)<=c||(b+c)<=a||(a+c)<=b))  
 p=(a+b+c)/2  
 s= *sqrt*(p\*(p-a)\*(p-b)\*(p-c))  
 *println*("Площадь трекгольника равна:"+s)  
}catch (e :Exception){  
 *println*("Некорректный тип данных")  
}

**Тестовые ситуации**

****

****

**Задача**

****

**Входные и выходные данные**

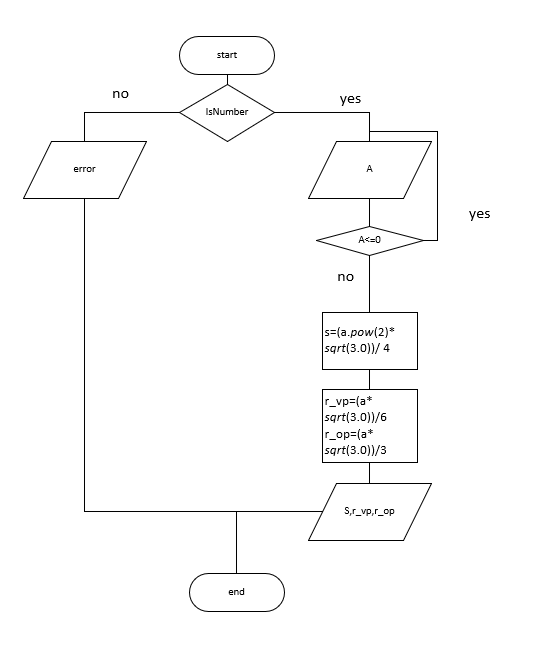
**Входные**

a- тип данных double,хранит число, которое ввел пользователь

**Выходные**

S,r\_vp, r\_op-тип переменной double

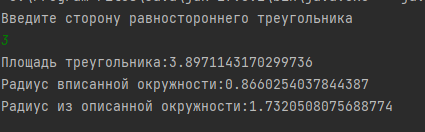
**Блок-схема**

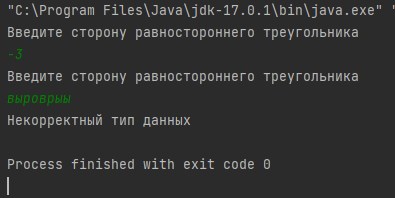
****

**Листинг программы**

try{  
 var a: Double  
 var s: Double  
 var r\_vp: Double  
 var r\_op: Double  
do{  
 *println*("Введите сторону равностороннего треугольника")  
a= *readln*()!!.*toDouble*()  
}while (a<=0)  
s=(a.*pow*(2)\* *sqrt*(3.0))/ 4  
 r\_vp=(a\* *sqrt*(3.0))/6  
 r\_op=(a\* *sqrt*(3.0))/3  
 *println*("Площадь треугольника:"+s)  
 *println*("Радиус вписанной окружности:"+r\_vp)  
 *println*("Радиус из описанной окружности:"+r\_op)  
}catch (e :Exception){  
 *println*("Некорректный тип данных")  
}

**Тестовые ситуации**

****

****

**Задача**

****

**Входные и выходные данные**

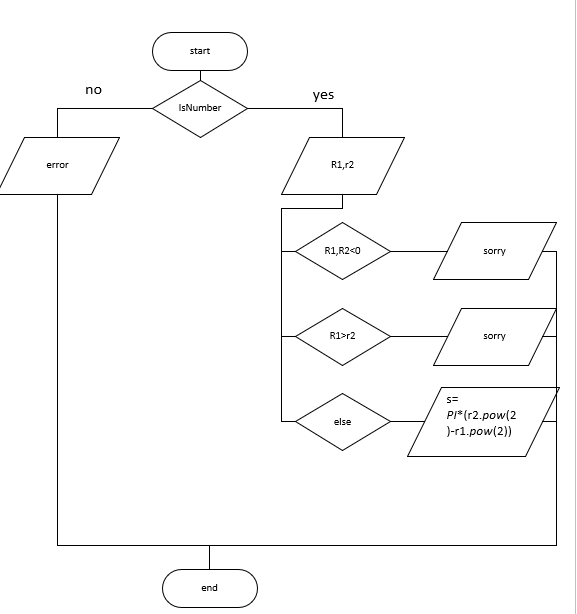
**Входные**

R1.R2-тип переменных double,хранят в себе длины внутреннего и внешнего радиуса

**Выходные**

String,либо же s-тип переменной double,хранит в себе величину кольца

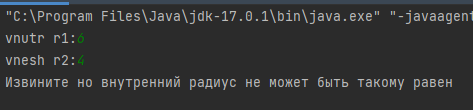
**Блок-схема**

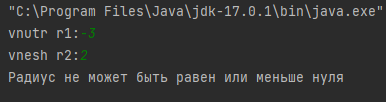
****

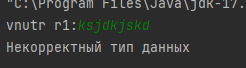
**Листинг программы**

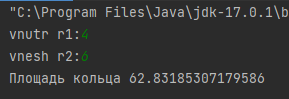
try{  
 var r1:Double  
 var r2: Double  
 var s: Double  
 *print*("vnutr r1:")  
 r1= *readln*()!!.*toDouble*()  
 *print*("vnesh r2:")  
 r2= *readln*()!!.*toDouble*()  
 when{  
 (r1<=0||r2<=0) -> *println*("Радиус не может быть равен или меньше нуля")  
 (r1>=r2) -> *println*("Извините но внутренний радиус не может быть такому равен")  
 else ->{  
 s= *PI*\*(r2.*pow*(2)-r1.*pow*(2))  
 *println*("Площадь кольца $s")  
 }  
 }  
}catch (e: Exception){  
 *println*("Некорректный тип данных")  
}

**Тестовые ситуации**

****

****

****

****

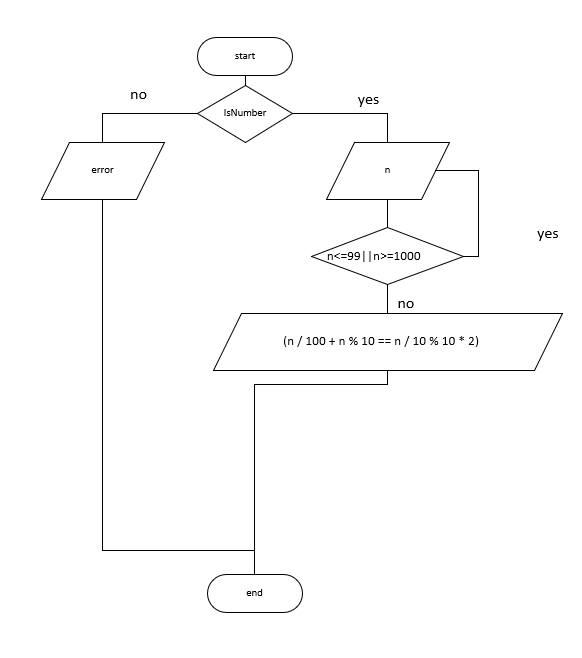
**Задача**

****

**Входные и выходные данные**

n,-тип данных int,переменная которая хранит число введенное пользователем

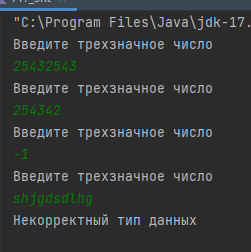
**Блок-схема**

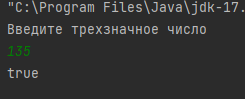
****

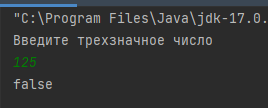
**Листинг программы**

try {  
 var n: Int  
do {  
 *println*("Введите трехзначное число")  
n= *readln*()!!.*toInt*()  
}while (n<=99||n>=1000)  
*println*(n / 100 + n % 10 == n / 10 % 10 \* 2)  
}catch (e : Exception){  
 *println*("Некорректный тип данных")  
}

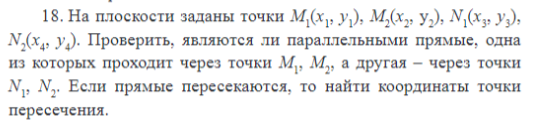
**Тестовые ситуации**

****

****

****

**Задача**

****

**Входные и выходные данные**

**Входные**

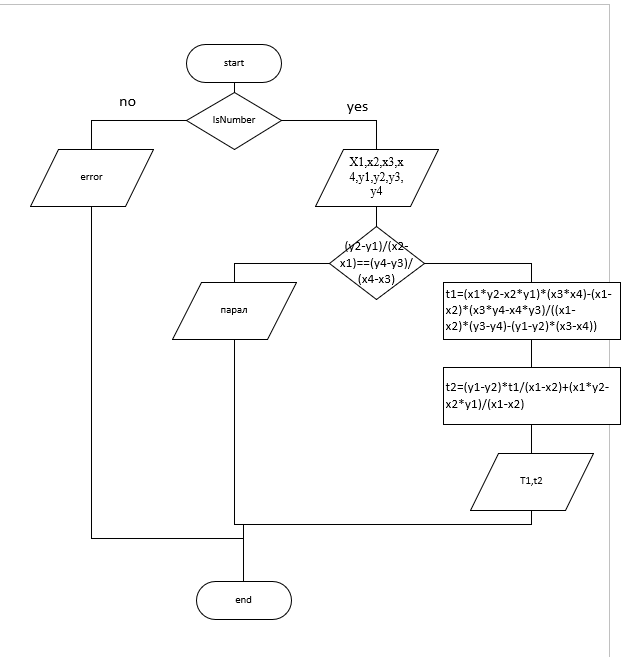
X1,x2,x3,x4,y1,y2,y3,y4- тип данных double, в которую пользователь заносит координаты

**Выходные**

String(В зависимости от введенного пользователем числа) либо же

T1,t2-переменные типа double,хранящие в себе координаты точки

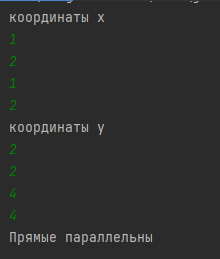
**Блок-схема**

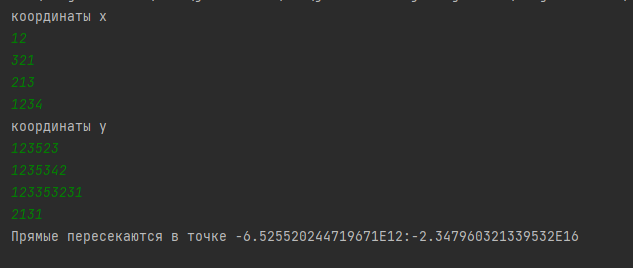
****

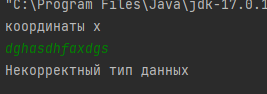
**Листинг программы**

try{  
 *println*("координаты x")  
 var x1= *readln*()!!.*toDouble*()  
 var x2= *readln*()!!.*toDouble*()  
 var x3= *readln*()!!.*toDouble*()  
 var x4= *readln*()!!.*toDouble*()  
 *println*("координаты y")  
 var y1= *readln*()!!.*toDouble*()  
 var y2= *readln*()!!.*toDouble*()  
 var y3= *readln*()!!.*toDouble*()  
 var y4= *readln*()!!.*toDouble*()  
 if ((y2-y1)/(x2-x1)==(y4-y3)/(x4-x3)){  
 *println*("Прямые параллельны")  
 }else{  
 var t1=(x1\*y2-x2\*y1)\*(x3\*x4)-(x1-x2)\*(x3\*y4-x4\*y3)/((x1-x2)\*(y3-y4)-(y1-y2)\*(x3-x4))  
 var t2=(y1-y2)\*t1/(x1-x2)+(x1\*y2-x2\*y1)/(x1-x2)  
 *println*("Прямые пересекаются в точке $t1:$t2")  
 }  
}catch (e :Exception){  
 *println*("Некорректный тип данных")  
}

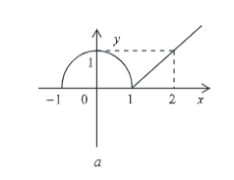
**Тестовые ситуации**

****

****

****

**Задача**

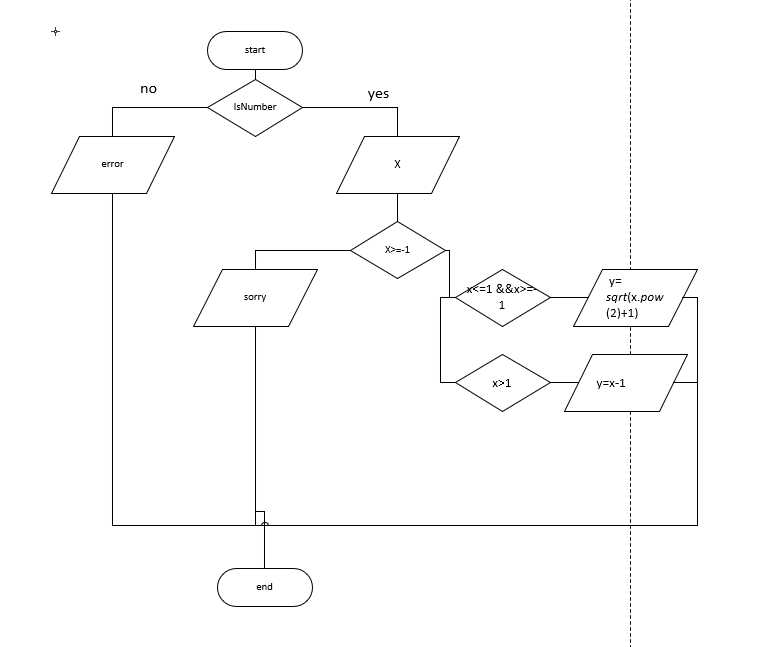
****

**Входные и выходные данные данные**

x-тип переменной double, хранит координаты

y-тип переменной double,хранит значение функции

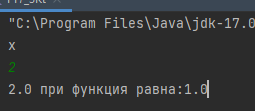
**Блок-схема**

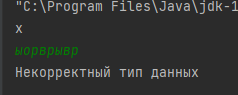
****

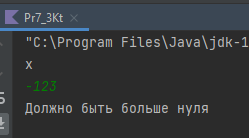
**Листинг программы**

try{  
 *println*("x")  
 var x= *readln*()!!.*toDouble*()  
 var y=0.0  
 if(x>=-1){  
 when{  
 x<=1 &&x>=-1 -> y= *sqrt*(x.*pow*(2)+1)  
 x>1 ->y=x-1  
 }  
 *println*("$x при функция равна:$y")  
 }  
 else *println*("Должно быть больше нуля")  
}catch ( e: Exception){  
 *println*("Некорректный тип данных")  
}

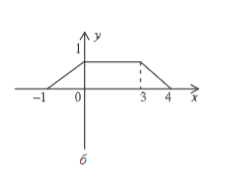
**Тестовые ситуации**

****

****

****

**Задача**

****

**Входные и выходные данные**

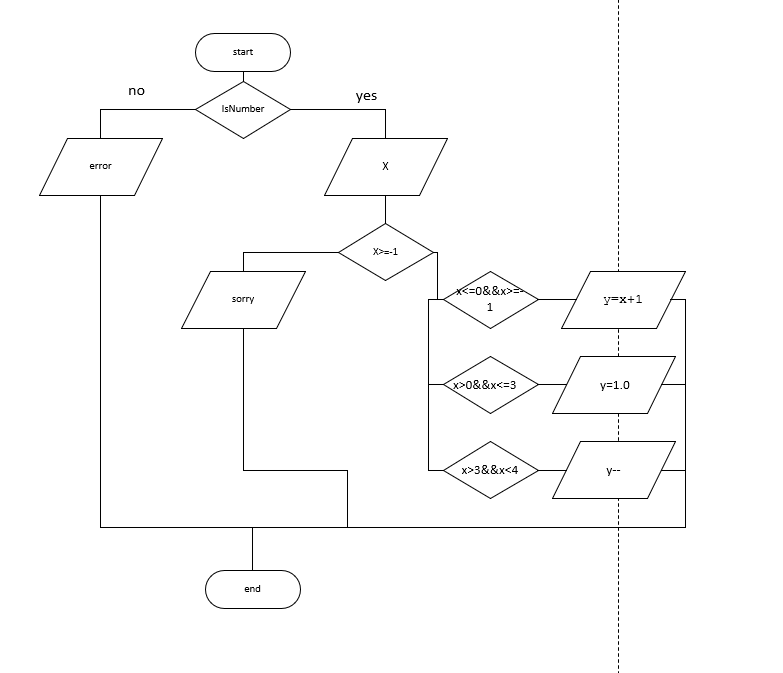
**Входные**

x-тип переменной double, хранит координаты

**Выходные**

y-тип переменной double,хранит значение функции

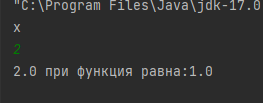
**Блок-схема**

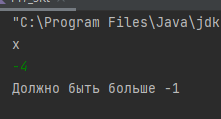
****

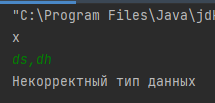
**Листинг программы**

try{  
 *println*("x")  
 var x= *readln*()!!.*toDouble*()  
 var y=2.0  
 if(x>=-1){  
 when{  
 x<=0&&x>=-1 -> y=x+1  
 x>0&&x<=3 -> y=1.0  
 x>3&&x<4 -> y--  
 }  
 *println*("$x при функция равна:$y")  
 }  
 else *println*("Должно быть больше -1")  
}catch (e : Exception){  
 *println*("Некорректный тип данных")  
}

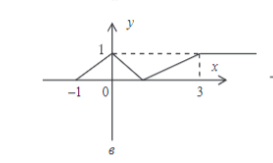
**Тестовые ситуации**

****

****

****

**Задача**

****

**Входные и выходные данные**

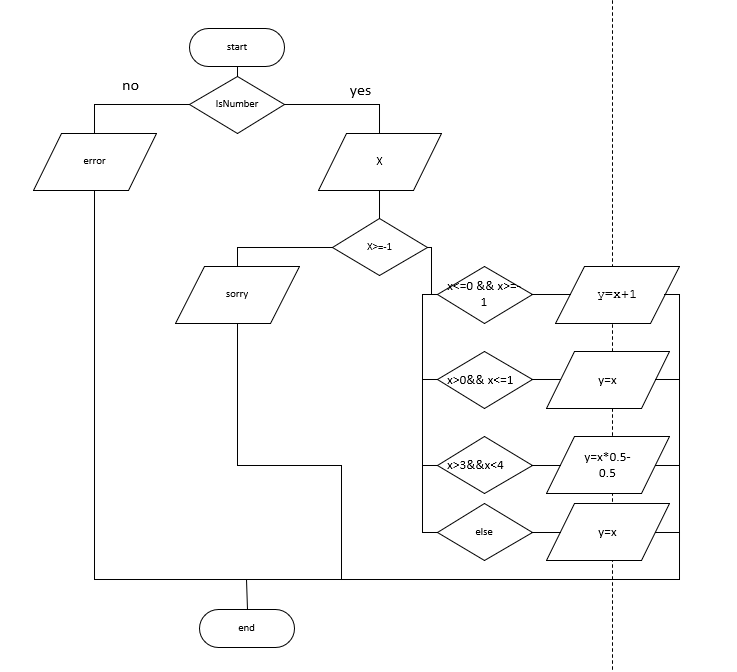
**Входные**

x-тип переменной double, хранит координаты

**Выходные**

y-тип переменной double,хранит значение функции

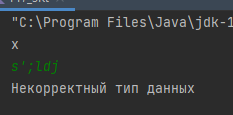
**Блок-схема**

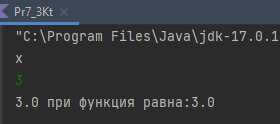
****

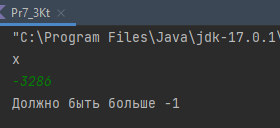
**Листинг программы**

try{  
 *println*("x")  
 var x= *readln*()!!.*toDouble*()  
 var y=0.0  
 if(x>=-1){  
 when{  
 x<=0 && x>=-1 -> y=x+1  
 x>0&& x<=1 -> y=x  
 x>3&&x<4-> y=x\*0.5-0.5  
 else -> y=x  
 }  
 *println*("$x при функция равна:$y")  
 }  
 else *println*("Должно быть больше -1")  
}catch (e : Exception){  
 *println*("Некорректный тип данных")  
}

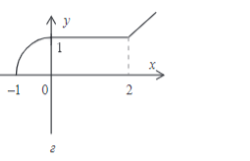
**Тестовые ситуации**

****

****

****

**Задача**

****

**Входные и выходные данные**

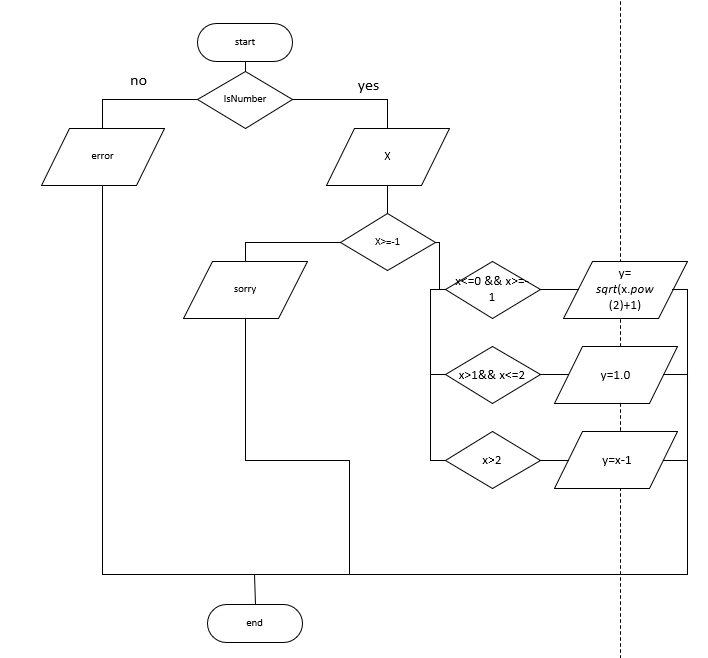
**Входные**

x-тип переменной double, хранит координаты

**Выходные**

y-тип переменной double,хранит значение функции

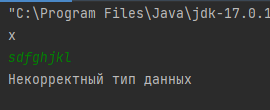
**Блок-схема**

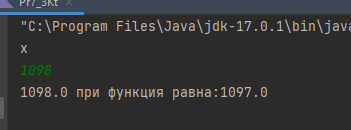
****

**Листинг программы**

try{  
 *println*("x")  
 var x= *readln*()!!.*toDouble*()  
 var y=0.0  
 if(x>=-1){  
 when{  
 x<=0 && x>=-1 -> y= *sqrt*(x.*pow*(2)+1)  
 x>1&& x<=2 -> y=1.0  
 x>2 -> y=x-1  
  
 }  
 *println*("$x при функция равна:$y")  
 }  
 else *println*("Должно быть больше -1")  
}catch (e : Exception){  
 *println*("Некорректный тип данных")  
}

**Тестовые ситуации**

****

****

**Вывод:** Выполняя эти задачи я кардинально улучшил свое знание языка программирования Kotlin. А в частности конструкции: when,try/catch,if/else,цикл do/while.