Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по программе «**Практическая 7**»

Выполнил: Валиев Айнур Ильдусович

Группа: ПР-22

Преподаватель: Мирошниченко Г.В

2024

**Задание:**

****

**Входные и выходные данные**

**R1,R2,R3 – Числа введенные пользователем радиус | Double**

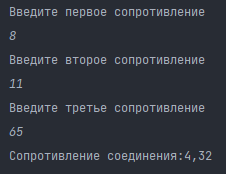
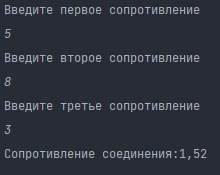
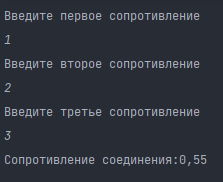
**total – сопротивление соединения | double**

**Блок-схема**

**Листинг программы (если есть)**

*fun* main(){  
 *try* {  
 *println*("Введите первое сопротивление")  
 *var* R1 = *readLine*()!!.*toDouble*()  
 *println*("Введите второе сопротивление")  
 *var* R2 = *readLine*()!!.*toDouble*()  
 *println*("Введите третье сопротивление")  
 *var* R3 = *readLine*()!!.*toDouble*()  
 *var* total = 1/(1/R1 + 1/R2 + 1/R3)  
 *println*("Сопротивление соединения:${String.*format*("%.2f",total)}")  
 }*catch* (e:Exception){*println*("Введено неверное значение")}  
}

**Тестовые ситуации**

****

**Задание:**

****

**Входные и выходные данные**

**R1,R2- радиусы введенные пользователем || Double**

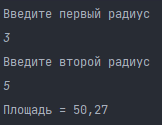
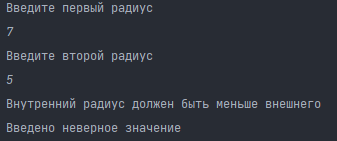
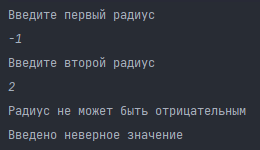
**Itog – вывод ответа**

**Блок-схема**

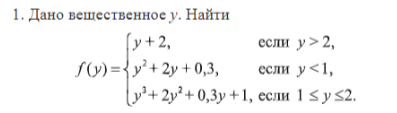
**Листинг программы (если есть)**

*import* kotlin.math.\*  
*fun* main ()  
{  
 *try* {  
 *println*("Введите первое радиус")  
 *var* r1 = *readLine*()!!.*toDouble*()  
 *println*("Введите второй радиус")  
 *var* r2 = *readLine*()!!.*toDouble*()  
 *var* itog = *when* {  
 r1<0 || r2<0 -> *println*("Радиус не может быть отрицательным")  
 r1>=r2 -> *println*("Внутренний радиус должен быть меньше внешнего")  
 *else* -> Math.*PI* \*(r2 \*r2-r1\*r1)  
 }  
 *println*("Площадь = ${String.*format*("%.2f",itog)}")  
 }*catch* (e:Exception) { *println*("Введено неверное значение")}  
}

**Тестовые ситуации**

**** **** 

**Задание:**

****

**Входные и выходные данные**

**Y- Число введенное пользователем || Double**

**Result-итоговое значение || Double**

**Блок-схема**

**Листинг программы (если есть)**

*fun* main() {  
 *try* {  
 *var* y = *readLine*()!!.*toDouble*()  
 *var* result = *when* {  
 y > 2 -> y + 2  
 y < 1 -> y \* y + 2 \* y + 0.3  
 y *in* 1.0..2.0 -> y \* y \* y + 2 \* y \* y + 0.3 \* y + 1  
 *else* ->*println*("Недопустимое значение y")  
 }  
 *println*("Результат: $result")  
 }*catch* (e:Exception) {  
 *println*("Введено неверное значение")}  
}

**Тестовые ситуации**

**** ****

**Задание:**

****

**Входные и выходные данные**

**A,b,c,d,u,v- данные веденные пользователем**

**minY- Итоговое значение минимального по отрезку**

**maxY- Итоговое значение максимального по отрезку**

**min- Итоговое значение минимального по отрезку**

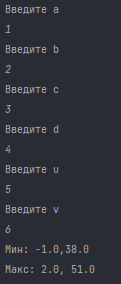
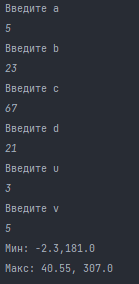
**max- Итоговое значение максимального по отрезку**

**Блок-схема**

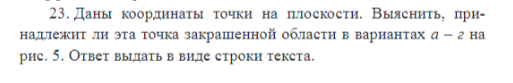
**Листинг программы (если есть)**

*import* kotlin.math.max  
*import* kotlin.math.min  
*fun* main() {  
 *try* {  
 *println*("Введите a")  
 *var* a = *readLine*()!!.*toDouble*()  
 *println*("Введите b")  
 *var* b = *readLine*()!!.*toDouble*()  
 *println*("Введите c")  
 *var* c = *readLine*()!!.*toDouble*()  
 *println*("Введите d")  
 *var* d = *readLine*()!!.*toDouble*()  
 *println*("Введите u")  
 *var* u = *readLine*()!!.*toDouble*()  
 *println*("Введите v")  
 *var* v = *readLine*()!!.*toDouble*()  
 *var* yu = a\*u\*u+b\*u+c  
 *var* yv = a\*v\*v+b\*v+c  
 *var* vertX = -b / (2\*a)  
 *var* vertY = a\*vertX\*vertX+b\*vertX+c  
 *var* minY= *min*(yu, yv)  
 *var* maxY=*max*(yu,yv)  
 *var* max = *max*(vertX,vertY)  
 *var* min = *min*(vertX,vertY)  
 *println*("Мин: $min,$minY")  
 *println*("Макс: $max, $maxY")  
 }*catch* (e:Exception) {*println*("Введено неверное значение")}  
}

**Тестовые ситуации**

**Задание:**

****

**Входные и выходные данные**

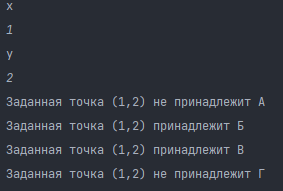
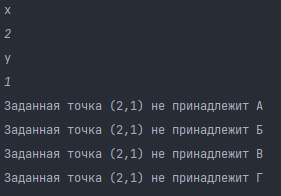
**X,y- данные веденные пользователем**

**Блок-схема**

**Листинг программы (если есть)**

*fun* main() {  
 *try* {  
 *print*("x")  
 *var* x = *readLine*()!!.*toInt*()  
 *print*("y")  
 *var* y = *readLine*()!!.*toInt*()  
 *when* (x>=-3&&x<=1&&y>=-1&&y<=1)  
 {  
 *true* -> *println*("Заданная точка ($x,$y) принадлежит А")  
 *false* -> *println*("Заданная точка ($x,$y) не принадлежит А")  
 }  
 *when* (x>=-1&&x<=1&&y>=-2&&y<=2) {  
 *true* -> *println*("Заданная точка ($x,$y) принадлежит Б")  
 *false* -> *println*("Заданная точка ($x,$y) не принадлежит Б")  
 }  
 *when* (x>=-1&&x<=1&&y>=-1&&y<=2) {  
 *true* -> *println*("Заданная точка ($x,$y) принадлежит В")  
 *false* -> *println*("Заданная точка ($x,$y) не принадлежит В")  
 }  
 *when* (x>=-2&&x<=1&&y>=-2&&y<=1) {  
 *true* -> *println*("Заданная точка ($x,$y) принадлежит Г")  
 *false* -> *println*("Заданная точка ($x,$y) не принадлежит Г")  
 }  
 }*catch* (e:Exception) {  
 *println*("Введено неправильное значение")}  
}

**Тестовые ситуации**

**** ****