Министерство образования и молодежной политики Свердловской области

ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по программе «**Практическое занятие №7.2**»

Выполнила: Яковлева Анастасия Николаевна

Группа: ПР-21

Преподаватель: Мирошниченко Г.В

2025

**Вариант 4**

**Задание:** По заданным вещественным числам a, b, c вычислить:

P = (max(a,b,c) + min(a,b,c))/2

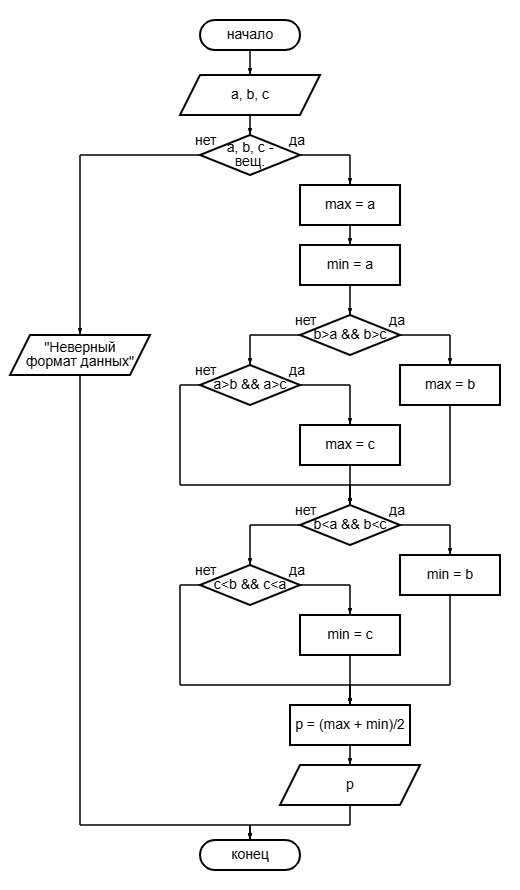
**Входные данные:**

*a, b, c –* числа, которые вводит пользователь, вещественный тип.

**Выходные данные:**

*p-* результат вычисление по формуле, вещественный тип.

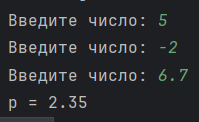
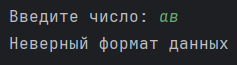
**Блок-схема**



**Листинг программы**

fun main() {  
 try {  
 *print*("Введите число: ")  
 val a: Double = *readln*().*toDouble*()  
 *print*("Введите число: ")  
 val b: Double = *readln*().*toDouble*()  
 *print*("Введите число: ")  
 val c: Double = *readln*().*toDouble*()  
 var max: Double = a  
 var min: Double = a  
 when {  
 (b>a && b>c) -> max = b  
 (c>b && c>a) -> max = c  
 }  
 when {  
 (b<a && b<c) -> min = b  
 (c<b && c<a) -> min = c  
 }  
 val p = (max + min)/2  
 *println*("p = $p")  
 }  
 catch (e: NumberFormatException){  
 *println*("Неверный формат данных")  
 }  
}

**Тестовые ситуации**

**Вариант 14**

**Задание:** Заданы целый числа x, y, m, n. Если разность x и y меньше остатка от деления m на n, увеличить x на 1.

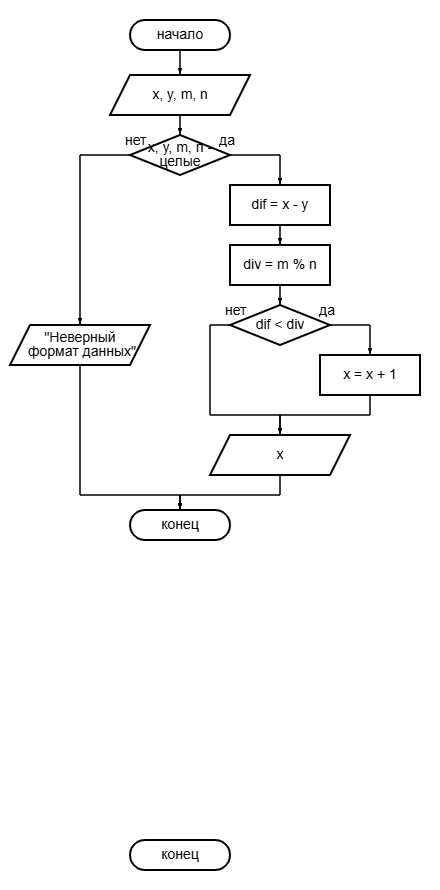
**Входные данные:**

*x, y, m, n -* числа, вводимые пользователем, целый тип

**Выходные данные:**

*x -* возвращающееся число, подвергающееся изменению при определенном условии, целый тип.

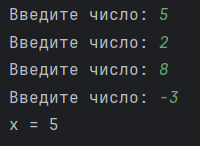
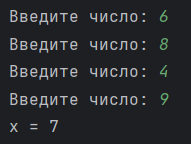
**Блок-схема:**



**Листинг программы:**

fun main() {  
 try {  
 *print*("Введите число: ")  
 var x: Int = *readln*().*toInt*()  
 *print*("Введите число: ")  
 val y: Int = *readln*().*toInt*()  
 *print*("Введите число: ")  
 val m: Int = *readln*().*toInt*()  
 *print*("Введите число: ")  
 val n: Int = *readln*().*toInt*()  
 val dif = x-y  
 val div = m%n  
 when{  
 dif<div -> x++  
 }  
 *println*("x = $x")  
 }  
 catch (e: NumberFormatException){  
 *println*("Неверный формат данных")  
 }  
}

**Тестовые ситуации:**

**Вариант 24**

**Задание:** Определить в какой координатной четверти находится точка с координатами (x, y).

**Входные данные:**

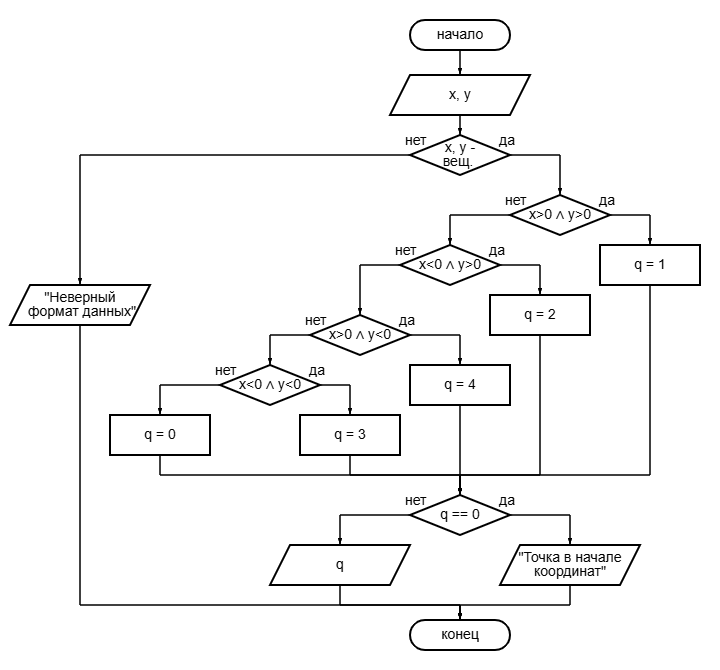
*х -* координата по оси Х, вещественный тип;

*у -* координата по оси У, вещественный тип;

**Выходные данные:**

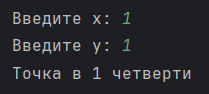
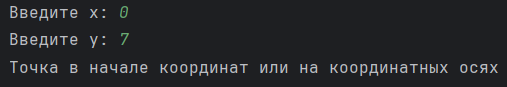
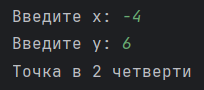
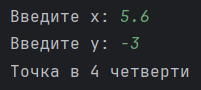
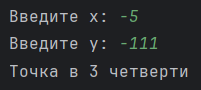
*q -* координатная четверть, целый тип.

**Блок-схема:**

**Листинг программы:**

fun main() {  
 try {  
 *print*("Введите x: ")  
 val x: Double = *readln*().*toDouble*()  
 *print*("Введите y: ")  
 val y: Double = *readln*().*toDouble*()  
 val q: Int  
 when {  
 (x>0 && y>0) -> q = 1  
 (x<0 && y>0) -> q = 2  
 (x>0 && y<0) -> q = 4  
 (x<0 && y<0) -> q = 3  
 else -> q = 0  
 }  
 when (q){  
 0 -> *println*("Точка в начале координат или на координатных осях")  
 else -> *println*("Точка в $q четверти")  
 }  
 }  
 catch (e: NumberFormatException){  
 *println*("Неверный формат данных")  
 }  
}

**Тестовые ситуации:**

**Вариант 34**

**Задание:** Заданы числа k, a, b. Определить количество точек пересечения прямой y = kx+b и гиперболы y = a/x и их координаты.

**Входные данные:**

*а -* коэффициент, отвечающий за положение ветвей гиперболы, вещественный

*b -* коэффициент, отвечающий за точку пересечения графика с осью ординат, вещественный

*k -* коэффициент наклона, вещественный

**Выходные данные:**

*x1-* координата по оси Х первой точки пересечения, вещественный;

*у1-* координата по оси У первой точки пересечения, вещественный;

*x2-* координата по оси Х второй точки пересечения, вещественный;

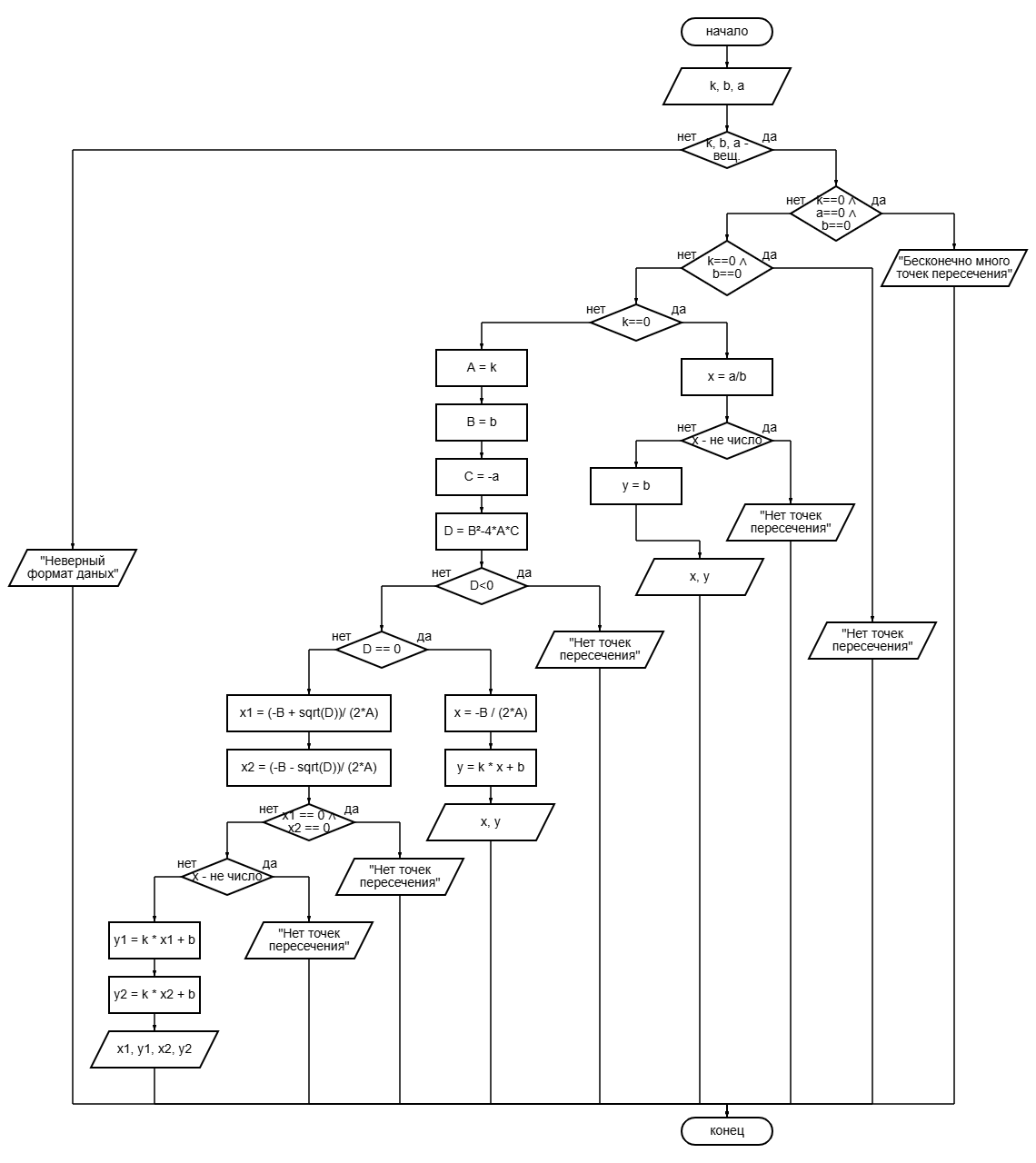
*у2-* координата по оси У второй точки пересечения, вещественный;

Либо

*x -* координата по оси Х единственной точки пересечения, вещественный;

*y -* координата по оси У единственной точки пересечения, вещественный.

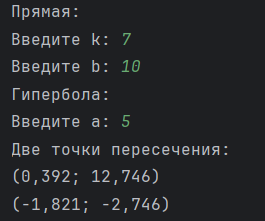
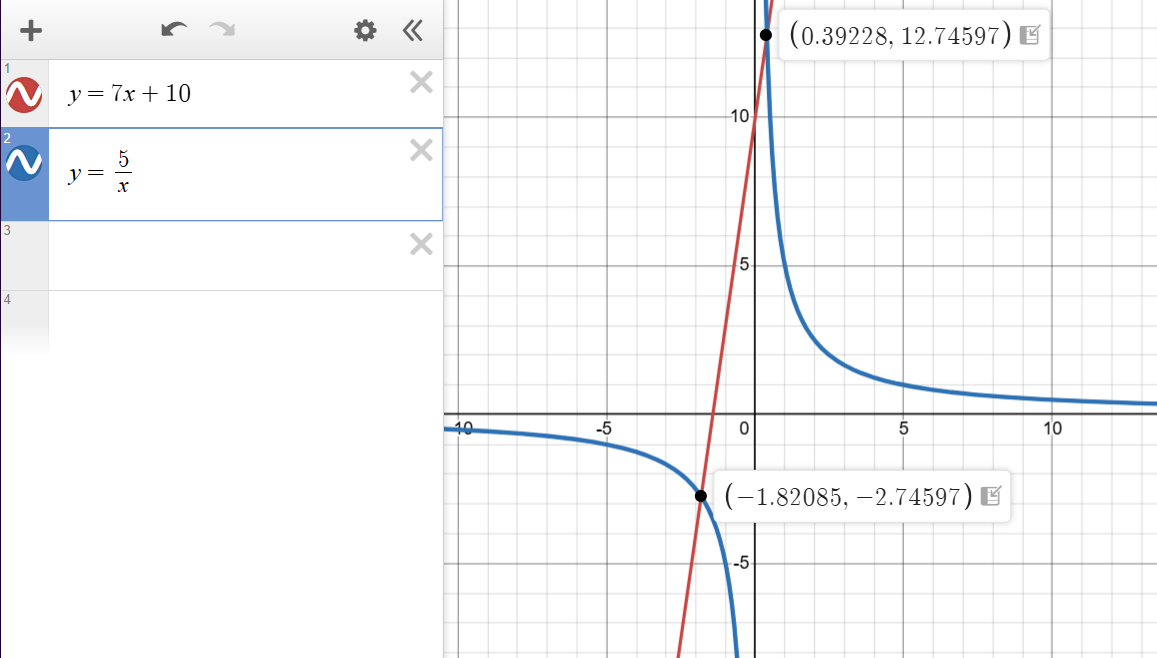
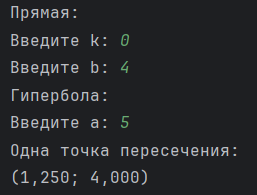
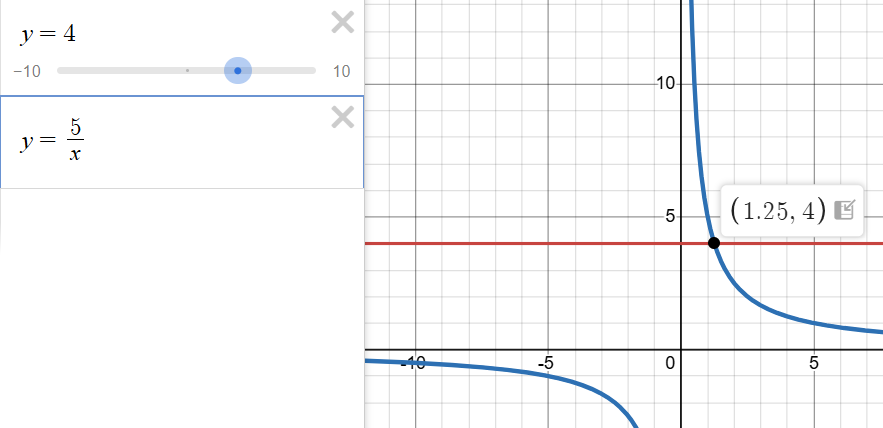
**Блок-схема:**

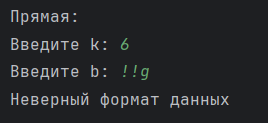
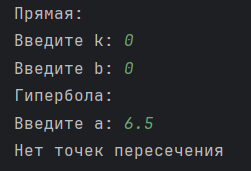
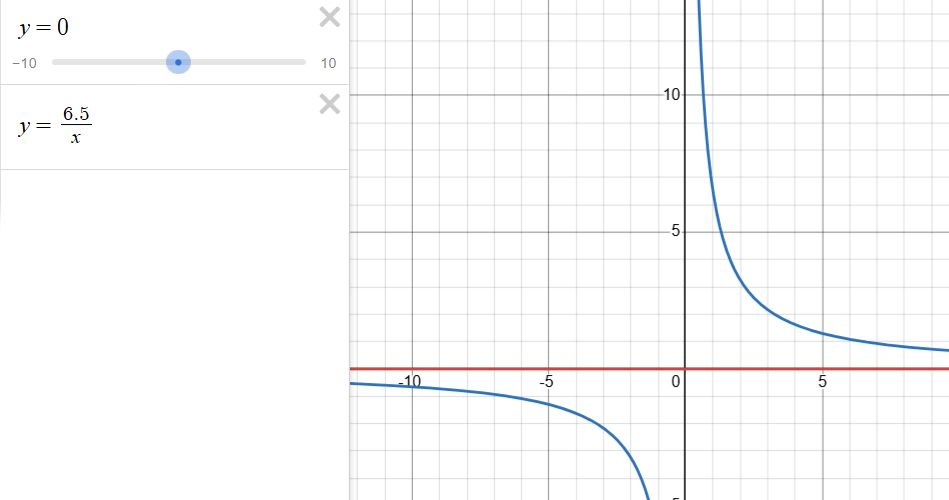


**Листинг программы:**

fun main() {  
 try {  
 *println*("Прямая:")  
 *print*("Введите k: ")  
 val k: Double = *readln*().*toDouble*()  
 *print*("Введите b: ")  
 val b: Double = *readln*().*toDouble*()  
 *println*("Гипербола:")  
 *print*("Введите a: ")  
 val a: Double = *readln*().*toDouble*()  
 when {  
 (k == 0.0 && a==0.0 && b==0.0)-> *println*("Бесконечно много точек пересечения")  
 (k == 0.0 && b==0.0)-> *println*("Нет точек пересечения")  
 (k == 0.0) -> {  
 val x = a/b  
 when {  
 x.*isNaN*() -> *println*("Нет точек пересечения")  
 else -> {  
 val y = b  
 *println*("Одна точка пересечения:\n(${"%.3f".*format*(x)}; ${"%.3f".*format*(y)})")  
 }  
 }  
 }  
 else -> {  
 val A = k  
 val B = b  
 val C = -a  
 val D = Math.pow(B, 2.0) - 4 \* A \* C  
 when {  
 D < 0 -> *println*("Нет точек пересечения.")  
 D == 0.0 -> {  
 val x = -B / (2 \* A)  
 val y = k \* x + b  
 *println*("Одна точка пересечения:\n(${"%.3f".*format*(x)}; ${"%.3f".*format*(y)})")  
 }  
 else -> {  
 val x1 = (-B + Math.sqrt(D)) / (2 \* A)  
 val x2 = (-B - Math.sqrt(D)) / (2 \* A)  
 when {  
 x1 == 0.0 && x2 == 0.0 -> *println*("Нет точек пересечения, корни равны нулю")  
 x1.*isNaN*() || x2.*isNaN*() -> *println*("Нет точек пересечения.")  
 else -> {  
 val y1 = k \* x1 + b  
 val y2 = k \* x2 + b  
 *println*("Две точки пересечения:\n(${"%.3f".*format*(x1)}; ${"%.3f".*format*(y1)})\n(${"%.3f".*format*(x2)}; ${"%.3f".*format*(y2)})")  
 }  
 }  
 }  
 }  
 }  
 }  
 }  
 catch (e: NumberFormatException){  
 *println*("Неверный формат данных")  
 }  
}

**Тестовые ситуации:**



**Вариант 44**

**Задание:** Даны n1, n2, n3 - количества пропущенных часов занятий в среднем на студента за неделю для трех групп. Вывести “Есть хорошая группа”, если min(n1, n2, n3) < 10.

**Входные данные:**

*n1 –* среднее количество пропущенных занятий первой группы, целый;

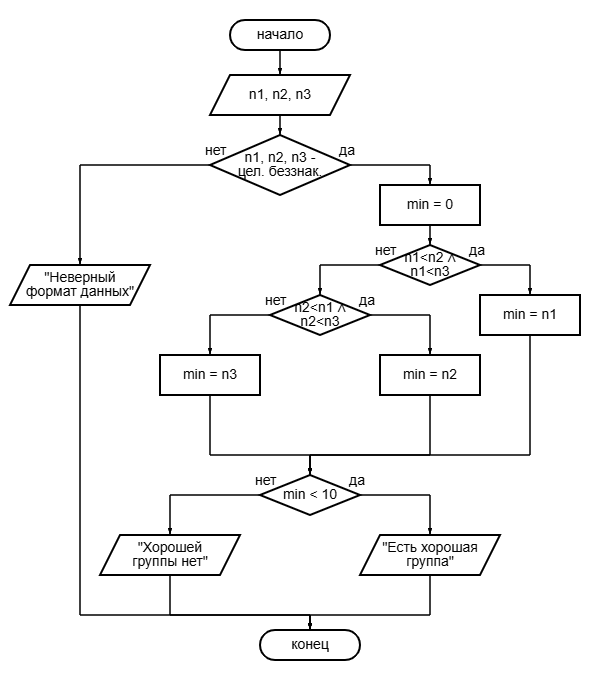
*n2 -* среднее количество пропущенных занятий второй группы, целый;

*n3 -* среднее количество пропущенных занятий третьей группы, целый;

**Выходные данные:**

*Сообщение* “Есть хорошая группа” или “Хорошей группы нет”, строковый.

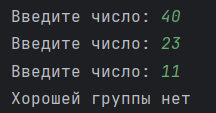
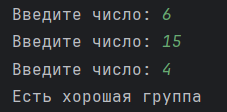
**Блок-схема:**



**Листинг программы:**

fun main() {  
 try {  
 *print*("Введите число: ")  
 val n1: UInt = *readln*().*toUInt*()  
 *print*("Введите число: ")  
 val n2: UInt = *readln*().*toUInt*()  
 *print*("Введите число: ")  
 val n3: UInt = *readln*().*toUInt*()  
 var min: UInt = 0u  
 when {  
 (n1<n2 && n1<n3) -> min = n1  
 (n2<n1 && n2<n3) -> min = n2  
 else -> min = n3  
 }  
 when {  
 min<10u -> *println*("Есть хорошая группа")  
 else -> *println*("Хорошей группы нет")  
 }  
 }  
 catch (e: NumberFormatException){  
 *println*("Неверный формат данных")  
 }  
}

**Тестовые ситуации:**



**Вывод:** я закрепила основы Kotlin, а также самостоятельно изучила string.format для округления чисел с плавающей запятой.