**Задание 1:**

Реализовать через корутины вызов функций нажатий и параллельного получения данных из n функций (n вводит пользователь). Затем, нужно их объединять и отображать.

**Class1:**

**fun Input():Int**

Входные данные: n-int, целое число

Выходные данные: 1, n-int, целое число

**fun Square()**

Входные данные: a, b, h – double, дробное число.

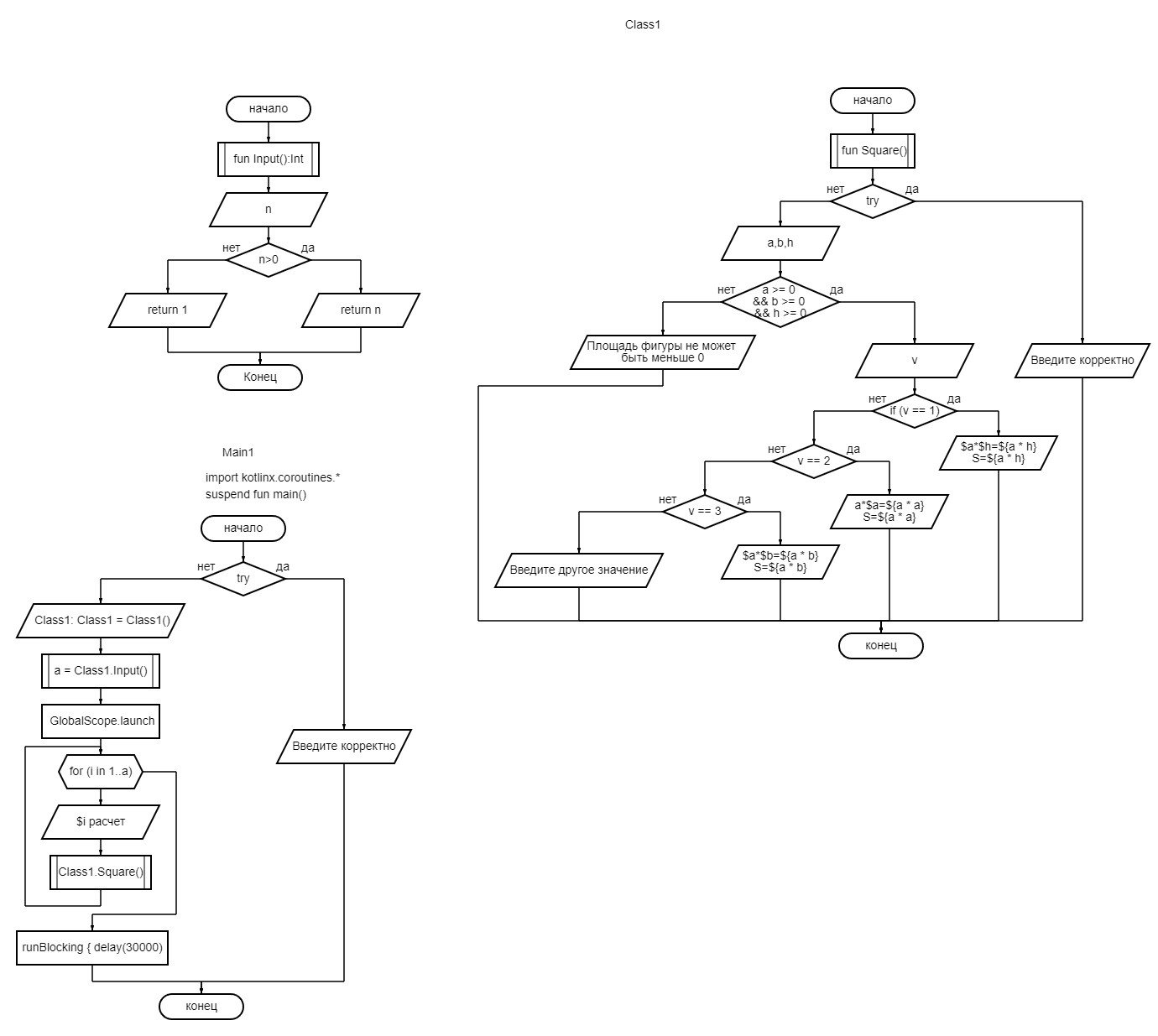
Выходные данные: a\*h, a\*a, a\*b-S фигур, string-текст.

**Main()**

Входные данные: Class1:Class=Class()

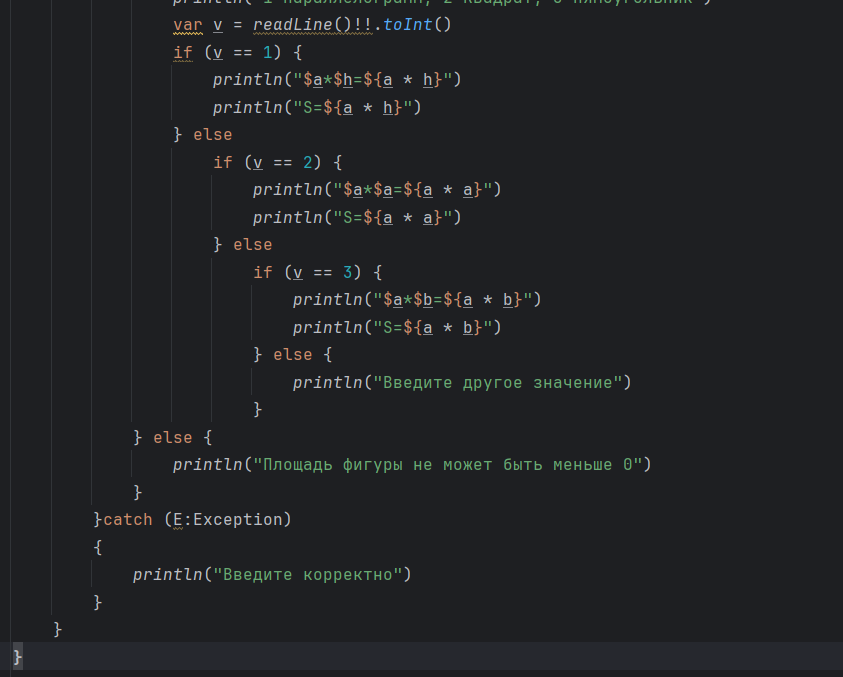
Выходные данные: i-кол-во вычислений.

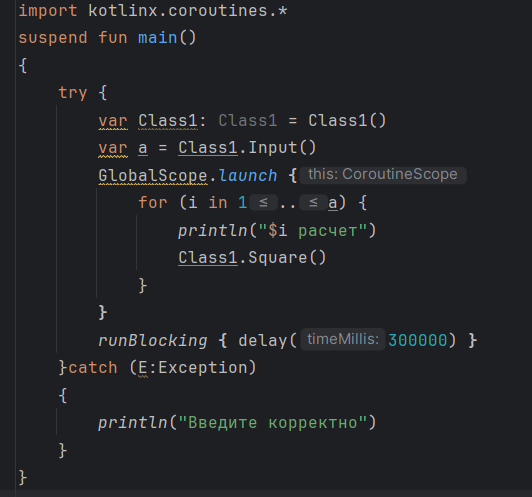
**Блок-схема:**



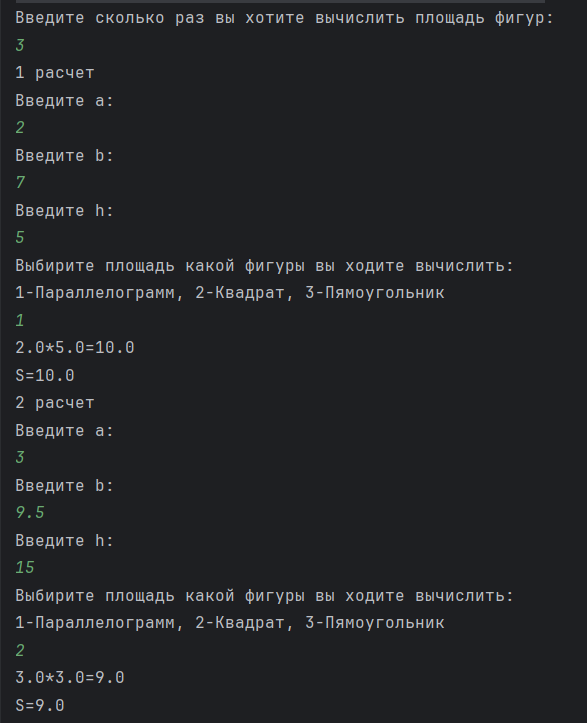
**Программа:**

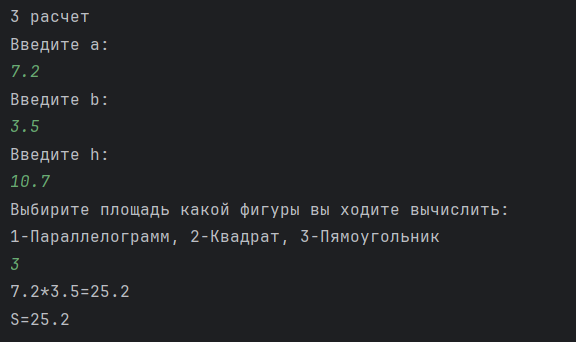
****

****

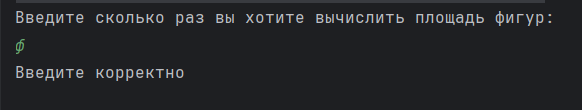
****

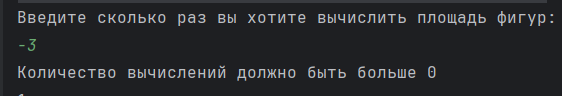
**Результат:**

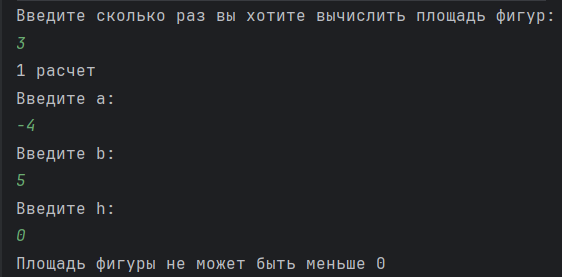
****

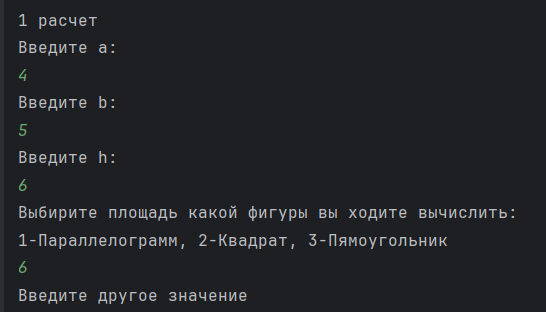
****

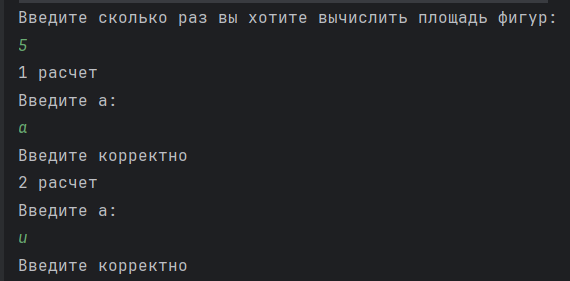
**Тестовые ситуации:**

****

****

****

****

****

**Задание 2:**

Создайте программу для загрузки участников для всех репозиториев данной организации, с сортировкой пользователей по количеству их репрозиториев. Вводится имя пользователя и токен (или пароль) GitHub в соответствующие поля. Если в раскрывающемся меню вариантов выбран вариант «БЛОКИРОВКА», нажмите «Загрузить участников». Пользовательский интерфейс должен зависнуть на некоторое время, а затем отобразить отсортированный список участников с указанием количества репрозиториев.

**Class2:**

Входные данные:

login-string, текст.

parol-string, текст.

reposit-int, целое число.

Выходные данные: login, parol, reposit, string-текст.

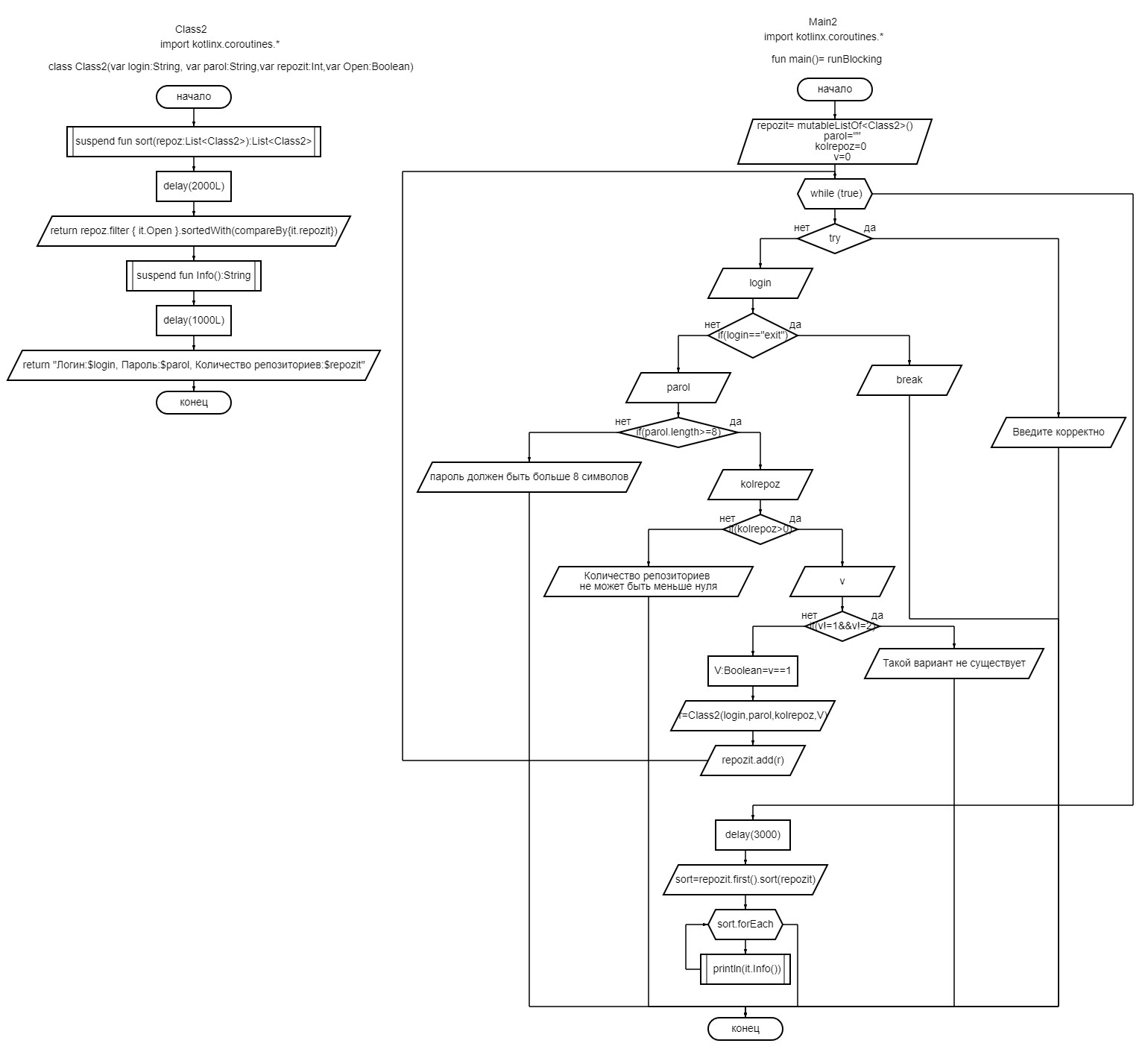
**Main()**

Входные данные:

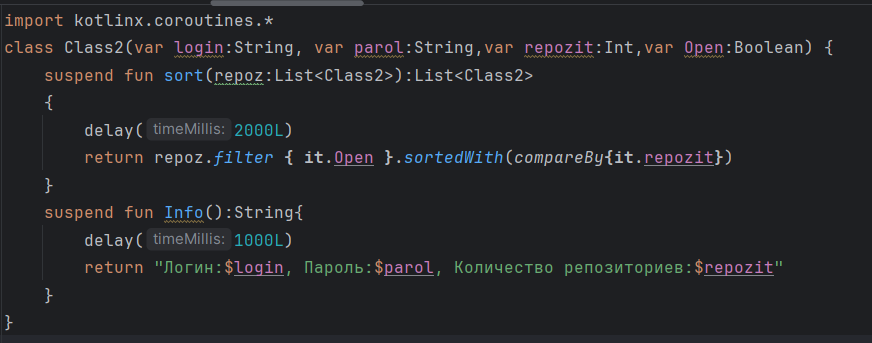
repozit= *mutableListOf*<Class2>()  
parol=""  
kolrepoz=0  
v=0

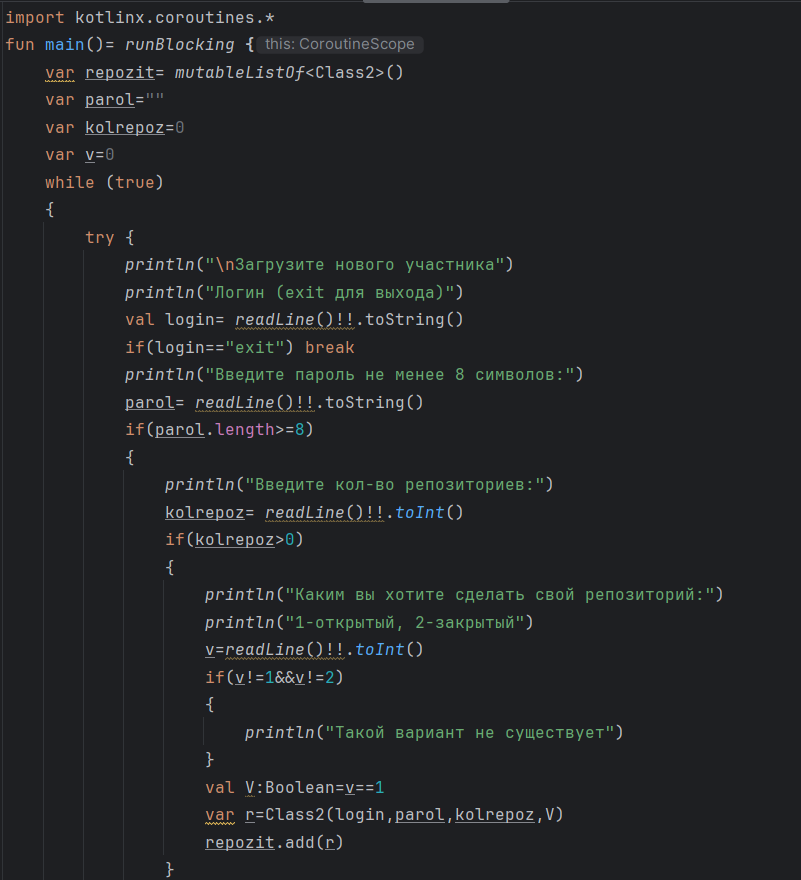
Выходные данные: repozit.add(r), **it**.Info()

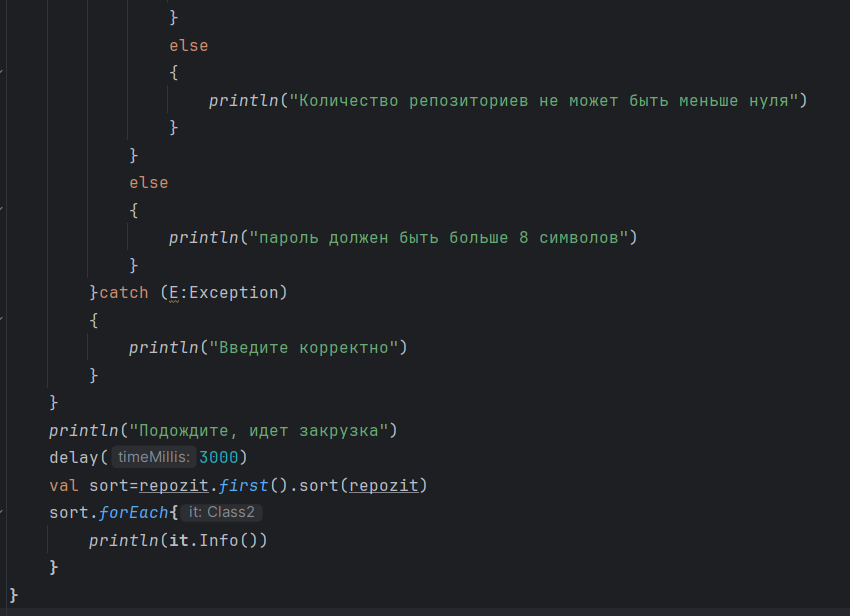
**Блок-схема:**



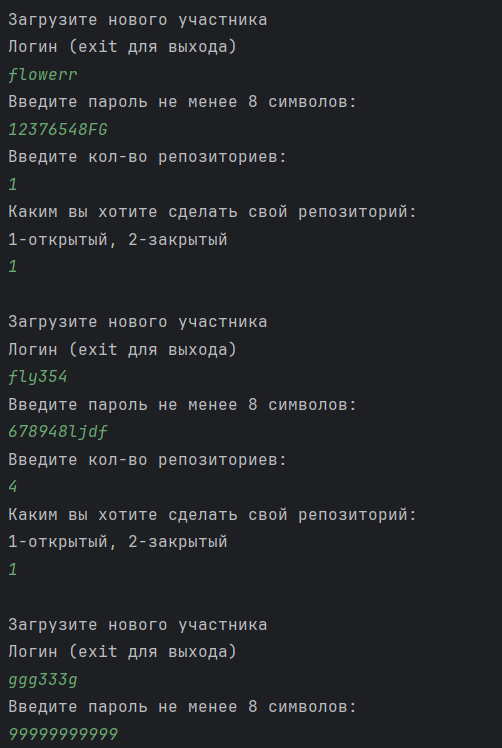
**Программа:**

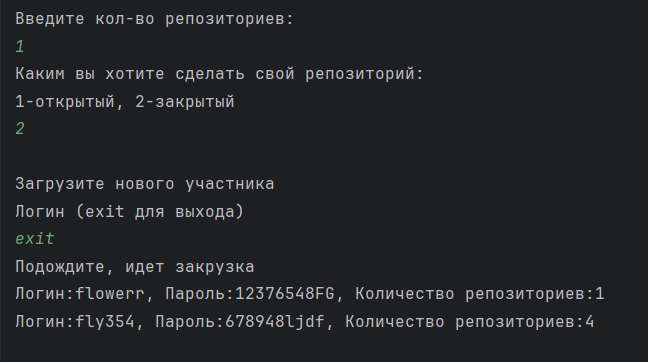
****

****

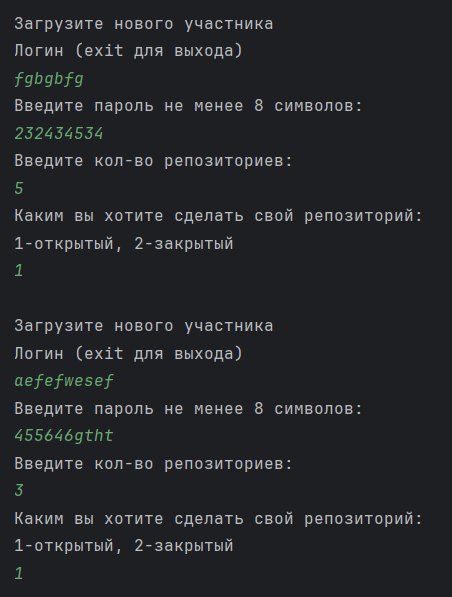
****

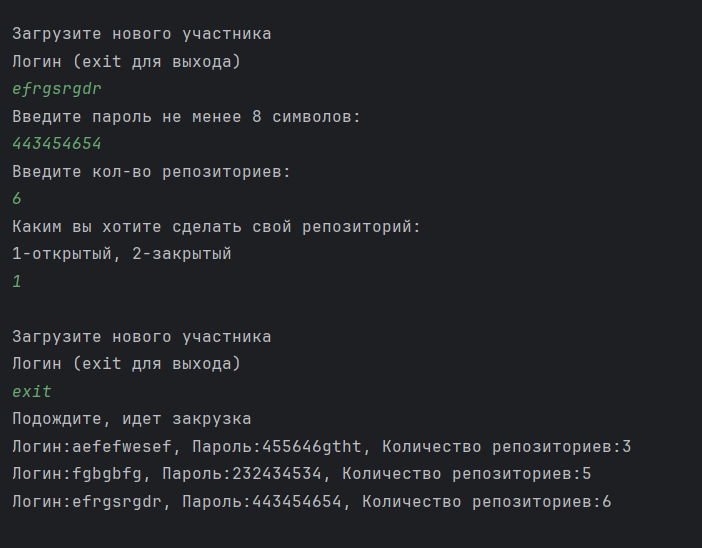
**Результат 1:**

****

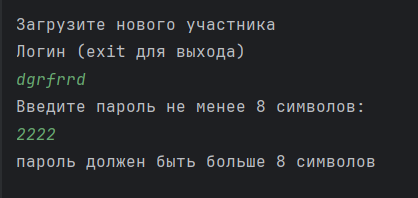
****

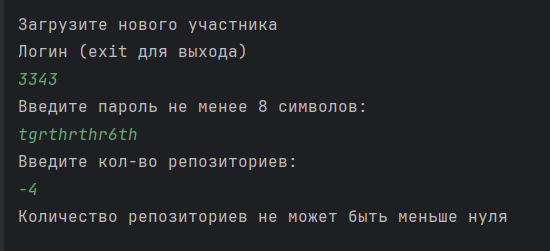
**Результат 2:**

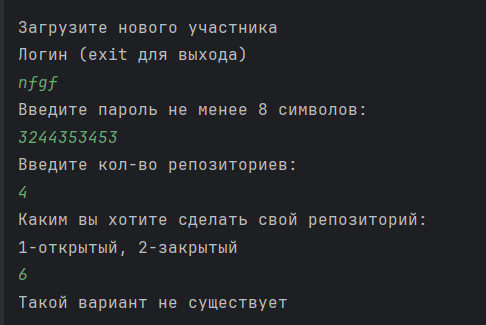
****

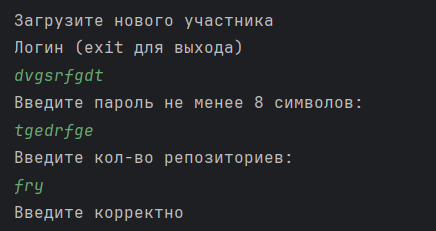
****

**Тестовые ситуации:**

****

****

****

****