Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по программе «**Практика 12**»

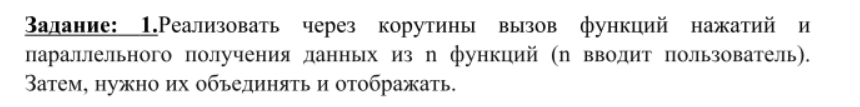
Выполнил: Молотова Виктория Анатольевна

Группа: ПР-21

Преподаватель: Мирошниченко Г.В

2023

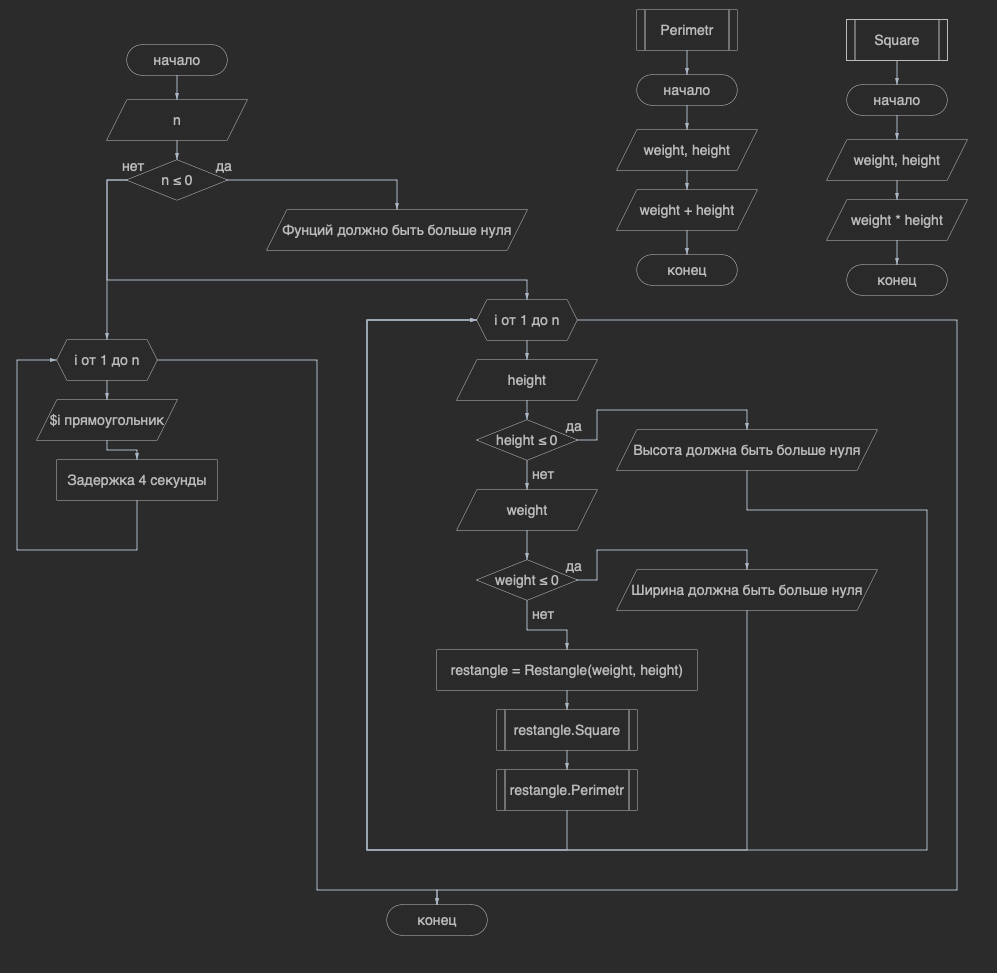
**Задание 1:**

****

**Входные и выходные данные**

* *Входные данные:* val height (double), val weight (double)
* *Выходные данные: текст (string)*

**Блок-схема**

****

**Листинг программы**

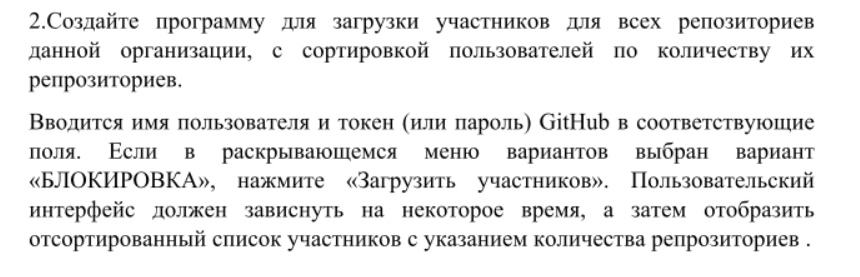
//Restangle.kt  
class Rectangle(var weight: Double, var height: Double) {  
 public fun Square() {  
 *println*("Площать прямоугольника: ${weight \* height}")  
 }  
  
 public fun Perimeter() {  
 *println*("Периметр прямоугольника: ${(weight + height) \* 2}")  
 }  
}

//zad1.kt  
import kotlinx.coroutines.\*  
import java.lang.NumberFormatException  
suspend fun main()= coroutineScope**{** *println*("Введите кол-во функций: ")  
 val n = *readLine*()!!.*toInt*()  
  
 if(n <=0){  
 *println*("Фунций должно быть больше нуля")  
 }  
 else{  
  
 *launch* **{** for (i in 1..n) {  
 try {  
 *println*("Введите высоту премоугольника:")  
 val height = *readLine*()!!.*toDouble*()  
 if(height <= 0) *println*("Высота должна быть больше нуля")  
 else{  
 *println*("Введите ширину премоугольника:")  
 val weight = *readLine*()!!.*toDouble*()  
 if(weight <= 0) *println*("Ширина должна быть больше нуля")  
 else{  
 val rectangle: Rectangle = Rectangle(weight, height)  
 rectangle.Square();  
 rectangle.Perimeter()  
 }  
 }  
 }  
  
 catch (e: NumberFormatException){  
 *println*("Необходимо ввести число")  
 }  
 }  
 **}** for (i in 1..n) {  
 *println*("$i прямоугольник")  
 delay(4000)  
 }  
 }  
**}**

**Тестовые ситуации**

Если пользователь введет ширину или высоту прямоугольника меньше или равной нулю – программа сообщит об ошибке. Так же если требуется ввод числа, а пользователь вводит строку – программа сообщает об ошибке.

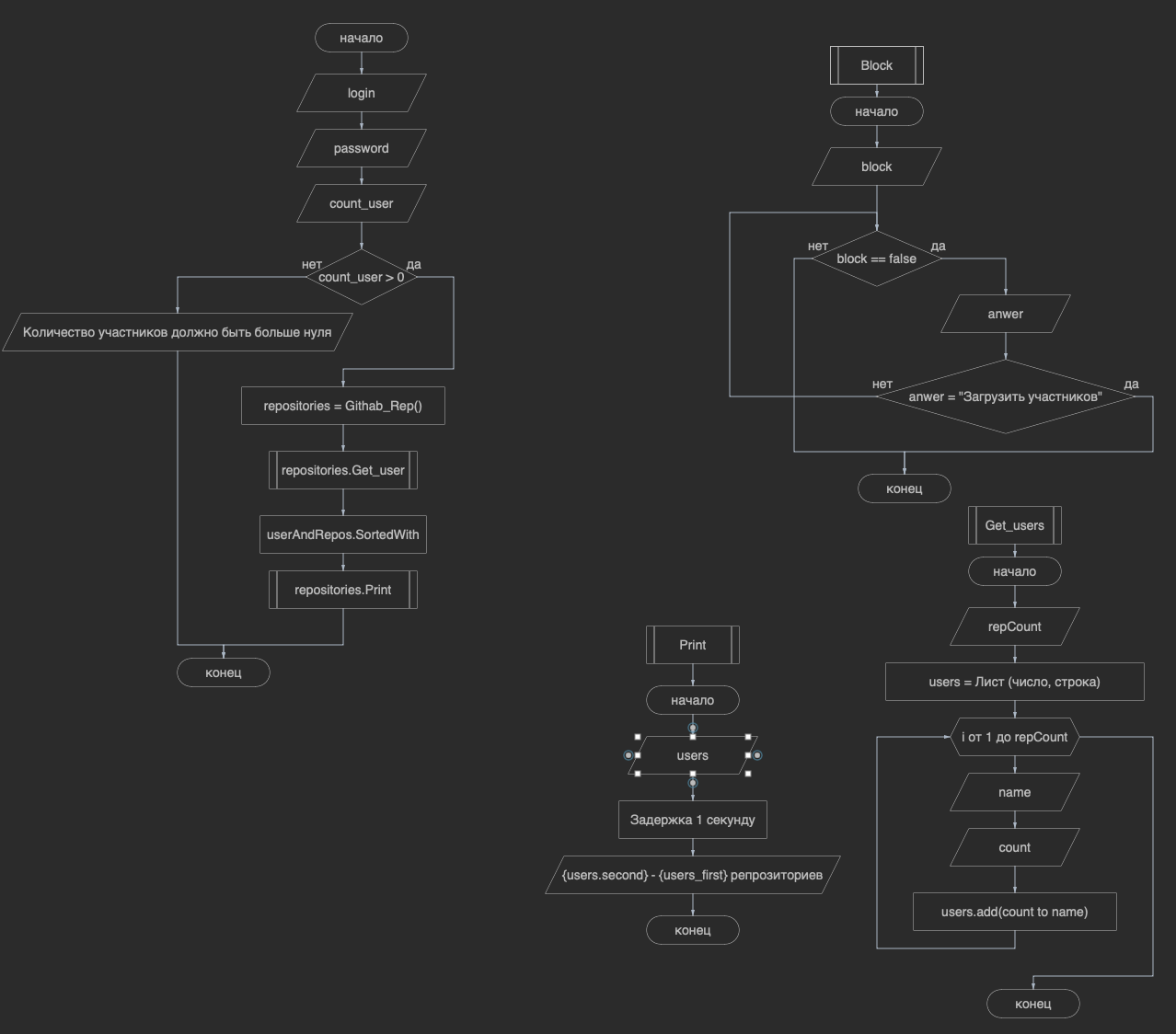
**Задание 2:**

****

**Входные и выходные данные**

* *Входные данные:* login (string), password (string), count\_user (int), name (string), count (int)
* *Выходные данные:* repositories.PrintUser(user)

**Блок-схема**

****

**Листинг программы**

//Githab\_Rep.kt  
import kotlinx.coroutines.\*  
  
class Githab\_Rep {  
 fun Block(block: Boolean){  
 while(block == false){  
 *println*("Заблокировано. Введите 'Загрузить участников'")  
 val anwer = *readLine*()!!  
 if(anwer == "Загрузить участников"){  
 break  
 }  
  
 }  
  
 }  
 suspend fun Get\_users(repCount: Int): List<Pair<Int, String>> = withContext(Dispatchers.IO)  
 **{** val users = *mutableListOf*<Pair<Int, String>>()  
 try {  
 for(i in 1..repCount)  
 {  
 *println*("Введите имя участника $i")  
 val name: String = *readLine*()!!  
 *println*("Введите количество его репозиториев")  
 val count: Int = *readLine*()!!.*toInt*()  
 users.add(count *to* name)  
 }  
 }  
 catch (e: Exception)  
 {  
 *println*("Вы ввели не число")  
 }  
 users  
 **}** suspend fun Print(user: Pair<Int, String>)  
 {  
 delay(1000)  
 *println*("${user.second } - ${user.first} репозиториев")  
 }  
}

//zad2.kt  
import kotlinx.coroutines.\*  
import java.lang.NumberFormatException  
  
fun main() {  
 *println*("Введите имя пользователя:")  
 var login = *readLine*()!!  
 *println*("Введите пароль:")  
 try{  
 var password = *readLine*()!!  
 *println*("Введите количество участников:")  
 val count\_user = *readLine*()!!.*toInt*()  
 if(count\_user > 0){  
 val repositories = Githab\_Rep()  
  
 val usersAndRepos = *runBlocking* **{** repositories.Get\_users(count\_user)  
 **}** *runBlocking* **{** repositories.Block(false)  
 *println*("Количество участников:")  
 usersAndRepos.*sortedWith*(*compareBy*(**{ it**.first **}**, **{ it**.second **}**))  
 .*forEach* **{** user **->** *launch* **{** repositories.Print(user) **} }** delay(3000)  
 **}** }  
 else{  
 *println*("Количество участников должно быть больше нуля")  
 }  
 }  
  
 catch (e: NumberFormatException){  
 *println*("Необходимо ввести число")  
 }  
  
}

**Тестовые ситуации**

Если требуется ввести строку, когда требуется ввести число – программа сообщит об ошибке. Так же количество участников репрозитория не может быть меньше или равно нулю.

**Вывод**

Благодаря этой практической я стала немного разбираться с корунтинами, а так же немного изучила работу с листами.