Міністерство освіти і науки України  
Національний технічний університет України

«Київський політехнічний інститут ім. І. Сікорського»

Кафедра інженерії програмного забезпечення в енергеніці

Лабораторна робота № 7

з курсу: "Розробка програмного забезпечення мобільних пристроїв"

**Виконав:**  
студентка 4-го курсу,  
групи ТІ-01  
Круть Катерина Олександрівна

**Перевірив:**

Недашківський О.Л.

Київ 2022/2023

Лабораторна робота № 7

**Завдання:**

**Завдання 1.**

Обов'язково використовувати рекурсію та рекурентність. Програма має рахувати числа Фібоначі і виводити результати на екран у формі:

0 - 0

1 - 1

2 - 1

3 - 2

і так далі

Кількість чисел Фібоначі, що необхідно вивести вводить користувач.

**Завдання 2.**

Обов'язково використовувати рекурсію та рекурентність. Програма має рахувати n! Число n вводить користувач.

**Завдання 3.**

Написати программу обчислення кількості (ми не знаємо кількість цифр) десяткових цифр натурального числа. Не використовувати стрічки, суто математичні операції.

Усі завданні мають бути реалізовані в одному класі.

**Хід виконання:**

Під час роботи програми користувача керує меню, яке надає можливість вводити число, з яким він бажає працювати, та вибирати завдання. Таким чином, користувач може самостійно визначати, яка функція йому необхідна в данний момент. Програма залишається активною навіть після виконання обчислень. Користувачу пропонується вирішити, чи бажає він завершити роботу. У випадку відмови програма продовжує роботу, доки користувач не вирішить її вимкнути.

Додатково, програма включає в себе механізм перевірки валідності даних: у випадку введення користувачем не натурального числа, програма просить його ввести дані ще раз.

Main.kt

fun main(args: Array<String>) {

try {

// Параметр для того щоб програма не вимикалась

var quit = "wait"

// цикл для того, щоб програма не вимикалась сама

while (!(quit == "q")) {

*println*("число для обробки:")

// користувач водить число, яке ми потім перевіряємо щоб воно було Натуральне

var choose = *readLine*()?.*toInt*()

var ind = true

while (ind) {

if (choose != null) if (choose <= 0) {

*println*("число має бути > 0")

choose = *readLine*()?.*toInt*()

} else ind = false

}

if (choose != null) {

Logic(choose)

}

*println*("'q' - quit, anything - continue")

quit = *readLine*()?.toString().*toString*()

}

} catch (e: Exception) {

// якщо введено не є цифри / знак мінус -- виводимо помилку

*println*("Помилка вводу")

}

}

logic.kt

*/\*\**

*\* Клас, що описує методи для роботи з числом:*

*\* - обрахунок числа Фібоначі*

*\* - обрахунок факторіалу*

*\* - обчислення кількості десяткових цифр натурального числа.*

*\**

*\** ***@param*** *[number] Параметр який зберігає число, який вів користувач*

*\** ***@param*** *menu1 Функція яка виводить меню для роботи з числом*

*\** ***@param*** *forFibonachi Функція для виводу всіх чисел Фібоначчі*

*\* включно до введеного числа користувача*

*\** ***@param*** *fibonachi Функція яка рахує число Фібоначі*

*\** ***@param*** *factorial Функція яка рахує факторіал числа*

*\** ***@param*** *sizenum Функція яка рахує кількість цифр в числі*

*\* \*/*

class Logic() {

*/\*\**

*\* Параметр який зберігає число користувача*

*\* \* \*/*

var number: Int = 0

*/\*\**

*\** ***@param*** *[num] Число, введене користувачем*

*\* \*/*

constructor(num: Int) : this() {

number = num

val choose = menu1()

when (choose) {

1 -> {

*println*("Fibonacci numbers for $number: ${forFibonachi(number)}")

}

2 -> {

*println*("$number! = " + factorial(number))

}

3 -> {

*println*("$number => " + sizenum(number))

}

}

}

*/\*\**

*\* Функція меню*

*\** ***@return*** *Функція повертає число (пункт меню який обрав користувач)*

*\* \*/*

fun menu1(): Int? {

*println*("1. Обчислити Фібоначі\n2. Знайти Факторіал\n3. Обчислення кількості десяткових цифр натурального числа")

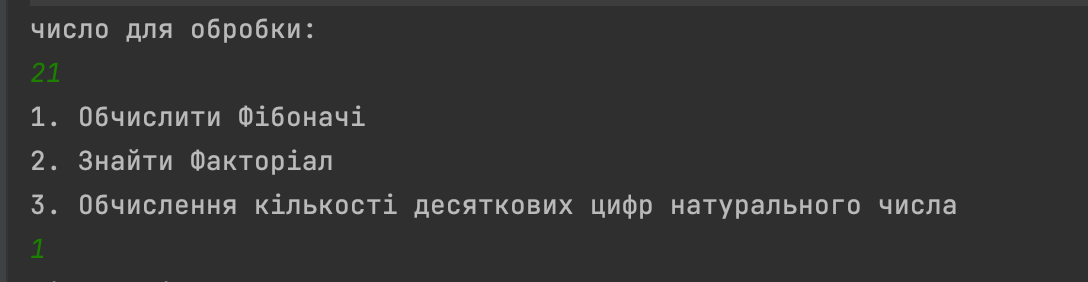
var choose = *readLine*()?.*toInt*()

while (!(choose == 1 || choose == 2 || choose == 3)) choose = *readLine*()?.*toInt*()

return choose

}

Виконання:



**Завдання 1:**

Обчислити числа Фібоначчі до вказаного числа

logic.kt

*/\*\**

*\* Функція яка допомогаю вивести усі числа Фібоначі включно до,*

*\* веденого юзером, числа користувача, використовуючи рекурсію*

*\** ***@param*** *[n] Число для обробки*

*\** ***@return*** *Результат обрахунку*

*\* \*/*

fun forFibonachi(n: Int): String {

val result = StringBuilder()

fun helper(n: Int) {

if (n >= 0) {

result.append("$n - ${fibonachi(n)}\n")

helper(n - 1)

}

}

helper(n)

return result.*split*("\n").*reversed*().*joinToString*("\n")

}

*/\*\**

*\* Функція обрахунку числа Фібоначчі, використовуючи рекурсію*

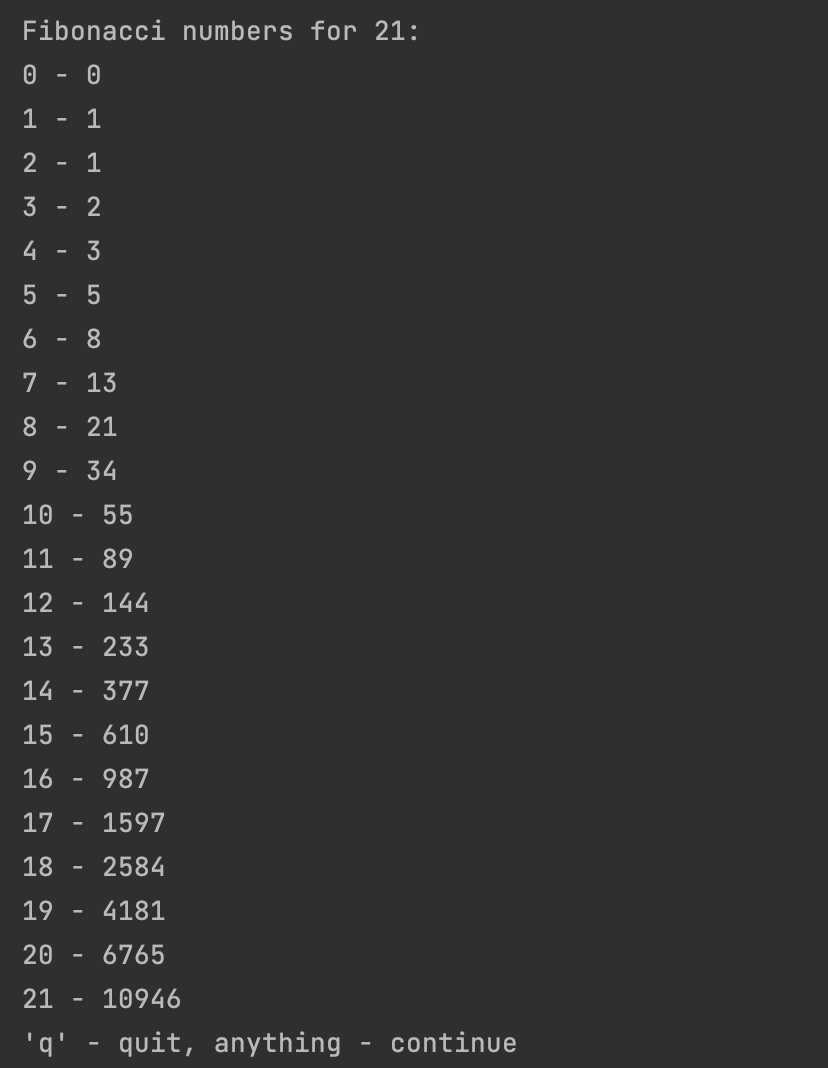
*\** ***@param*** *[n] Число для обробки*

*\** ***@return*** *Результат обрахунку фібоначчі*

*\* \*/*

fun fibonachi(n: Int): Int = if (n == 0) 0 else if (n == 1) 1 else fibonachi(n - 1) + fibonachi(n - 2)

Виконання:



**Завдання 2:**

Обчислити Факторіал числа

logic.kt

*/\*\**

*\* Функція яка рахує факторіал числа, використовуючи рекурсію*

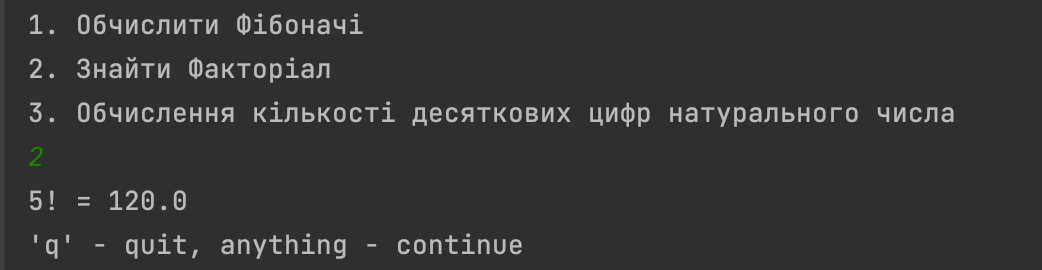
*\** ***@param*** *[n] Число для обробки*

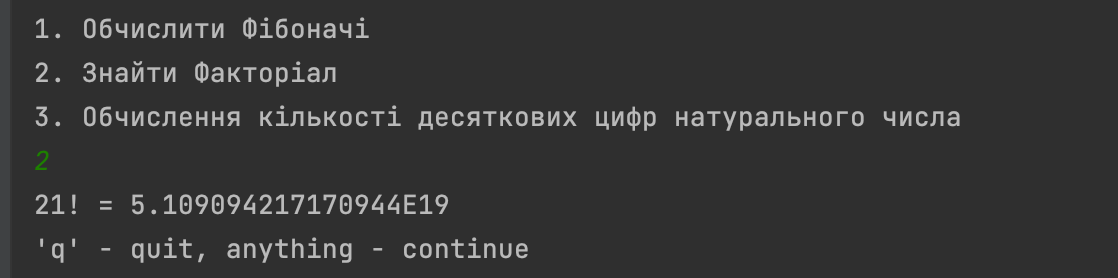
*\** ***@return*** *Результат обрахунку*

*\* \*/*

fun factorial(n: Int): Double = if (n < 2) 1.0 else n \* factorial(n - 1)

Виконання:





**Завдання 3:**

Обчислити кількість цифр в числі

logic.kt

*/\*\**

*\* Функція яка рахує кількість цифр в чіслі*

*\** ***@param*** *[n] Число для обробки*

*\** ***@return*** *Результат обрахунку*

*\* \*/*

fun sizenum(n: Int): Int {

var count: Int = 0

var x: Int = n

while (x > 0) {

x /= 10

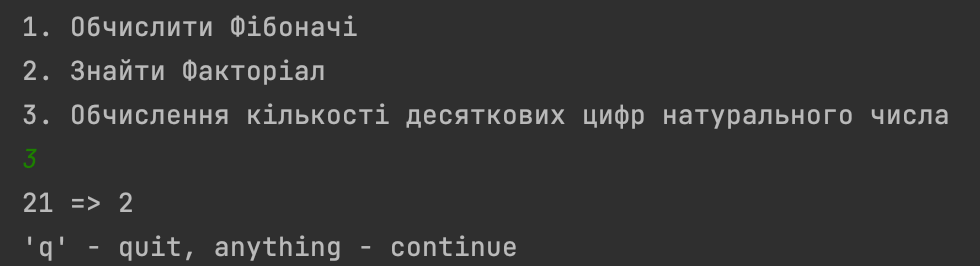
count++

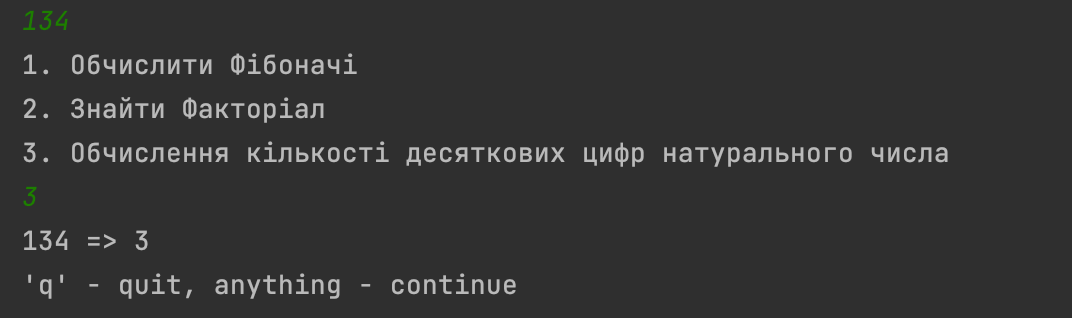
};

return count

}

Виконання:





**Висновок**

В результаті виконання лабораторної роботи №7 було освоєно основні принципи роботи з мовою Kotlin, використано їх на практиці: використання та створення функцій, виклик функцій з вказанням параметрів, робота з циклами, рекурсивні методи.