Міністерство освіти і науки України  
Національний технічний університет України

«Київський політехнічний інститут ім. І. Сікорського»

Кафедра інженерії програмного забезпечення в енергеніці

Лабораторна робота № 12

з курсу: "Розробка програмного забезпечення мобільних пристроїв"

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

**Перевірив:**

Недашківський О.Л.

Київ 2023/2024

Лабораторна робота № 12

**Завдання:**

1. Написати программу, що робить перевірку, чи є значення символьної змінної, що введена з клавіатури:
   * цифрою від ‘0’ до ‘9’;
   * малою латинською літерою;
   * латинською літерою (великою чи малою)
2. У розкладі рейсів літаків дні тижня позначаються номерами від 1 до 7. Припустимо, що в програмі дні тижня подаються enum типом Weekd. Написати програму, що у відповідь на введення номера дня виводить текстове подання дня тижня.
3. За понеділком іде вівторок тощо, а за неділею – понеділок. Написати функцію обчислення за днем тижня (типу Weekd) наступного за ним дня.
4. Написати “найпростіший калькулятор”, що отримує на вхід 2 числа і операцію над ними після чого повертає результат обчислень.
5. Написати процедуру обчислення за цілим N>3 таких натуральних A і B, що 5A-2B=N, причому A+B мінімально.

**Хід виконання:**

Було створено enum клас Weekd, який представляє дні тижня та головний модуль програми, де реалізовані всі фугкції для забезпечення юзер інтрефейсу та логіки обробки для всіх пунктів завдання.

**Weekd.kt**

*/\*\*  
 \* Enum клас Weekd - представляє дні тижня  
 \* Кожен день тижня асоційований з певним інтовим значенням, що відповідає його порядковому номеру у тижні  
 \*/*enum class Weekd(val num: Int) {  
 *Monday*(1),  
 *Tuesday*(2),  
 *Wednesday*(3),  
 *Thursday*(4),  
 *Friday*(5),  
 *Saturday*(6),  
 *Sunday*(7)  
}

**Main.kt**

**Робота з інтерфейсом**

Головна функція програми – main()

Функція меню – menu()

Функція зчитування символу з консолі readChar()

*/\*\*  
 \* Опис головного меню програми  
 \*/*const val *GENERAL\_MENU* =  
 "1. Перевірка, чи є значення символьної змінної, що введена з клавіатури: \n\tцифрою від 0 до 9\n\tмалою латинською буквою" +  
 "\n\tлатинською літерою (великою чи малою)" +  
 "\n2. Відповідь на введення номера дня виводить текстове подання дня тижня" +  
 "\n3. Функція обчислення за днем тижня (типу Weekd) наступного за ним дня\n" +  
 "4. Найпростіший калькулятор для двох чисел\n" +  
 "5. Процедура обчислення за цілим N>3 таких натуральних A і B, що 5A-2B=N, причому A+B мінімально"  
  
*/\*\*  
 \* Головна функція програми. Запуск меню та вивід результату обробки відповідно до вибраної опції  
 \*/*fun main(args: Array<String>) {  
 try {  
 var quit = "wait"  
 while (quit != "q") {  
 when (*menu*()) {  
 1 -> *println*(*symbolIdentify*())  
 2 -> *println*(*getWeekdDayName*())  
 3 -> *println*(*getWeekdNextDay*())  
 4 -> *println*(*expressionCalculation*())  
 5 -> *println*(*calculateABNExperssion*())  
 }  
 *println*("завершити роботу - 'q', якщо ні - будь що")  
 quit = *readLine*()?.toString() ?: ""  
 }  
 } catch (e: Exception) {  
 *println*("Введено некоректні данні")  
 }  
}  
  
*/\*\*  
 \* Функція для виводу головного меню та отримання вибору користувача  
 \** ***@return*** *номер обраного користувачем пункту меню  
 \*/*fun menu(): Int? {  
 *println*(*GENERAL\_MENU*)  
 var choose = *readLine*()?.*toInt*()  
 val validChoices = *setOf*(1, 2, 3, 4, 5)  
 while (choose !in validChoices) choose = *readLine*()?.*toInt*()  
 return choose  
}

*/\*\*  
 \* Функція для зчитування символу з консолі  
 \** ***@return*** *символ, який ввів користувач, або null, якщо введено більше одного символу  
 \*/*fun readChar(): Char? {  
 while (true) {  
 val char = *readLine*()?.*toCharArray*()  
 return if (char?.size == 1) char[0] else null  
 }  
}

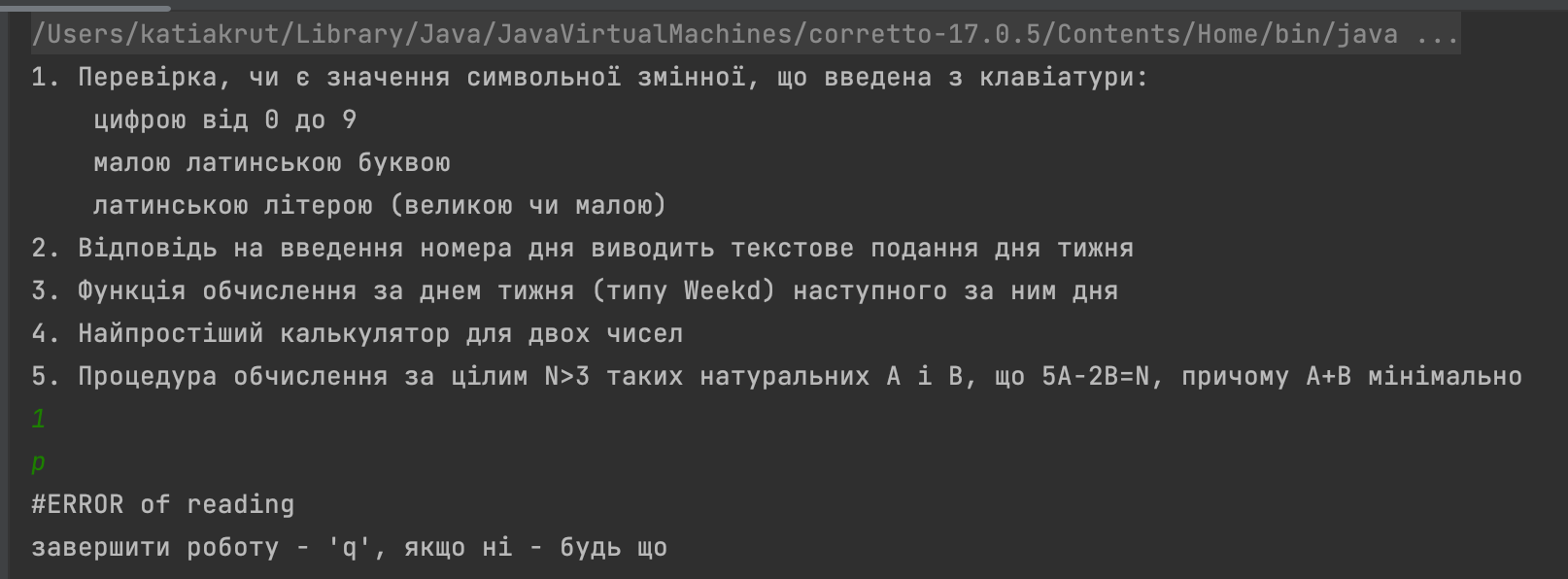
**Завдання 1:** Написати программу, що робить перевірку, чи є значення символьної змінної, що введена з клавіатури:

* + цифрою від ‘0’ до ‘9’;
  + малою латинською літерою;
  + латинською літерою (великою чи малою)

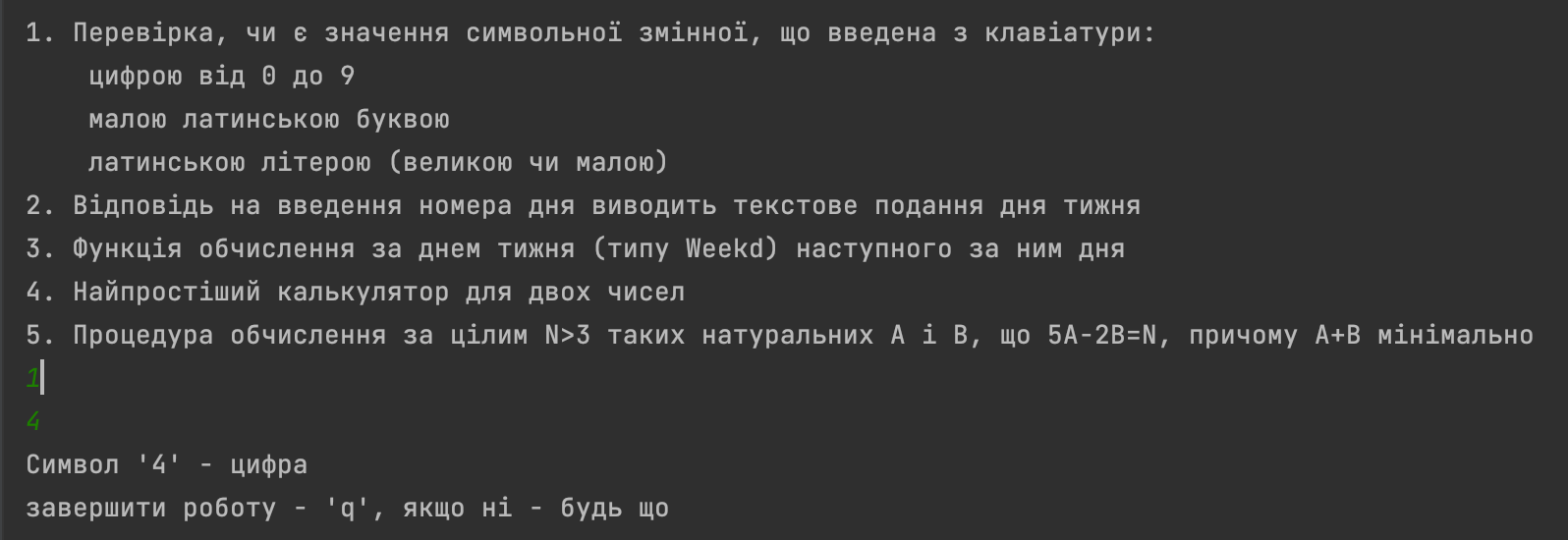
*/\*\*  
 \* Функція для перевірки символу на відповідність певним критеріям (цифра, латинська буква тощо)  
 \** ***@return*** *результат перевірки у текстовому форматі  
 \*/*fun symbolIdentify(): String {  
 val char = *readChar*() ?: return "#ERROR of reading"  
  
 return when (char) {  
 in '0'..'9' -> "Символ '$char' - цифра"  
 in 'a'..'z' -> "Символ '$char' - мала латинська буква."  
 in 'A'..'Z' -> "Символ '$char' - велика латинська буква"  
 else -> "#ERROR of reading"  
 }  
}

**Приклади виконання**

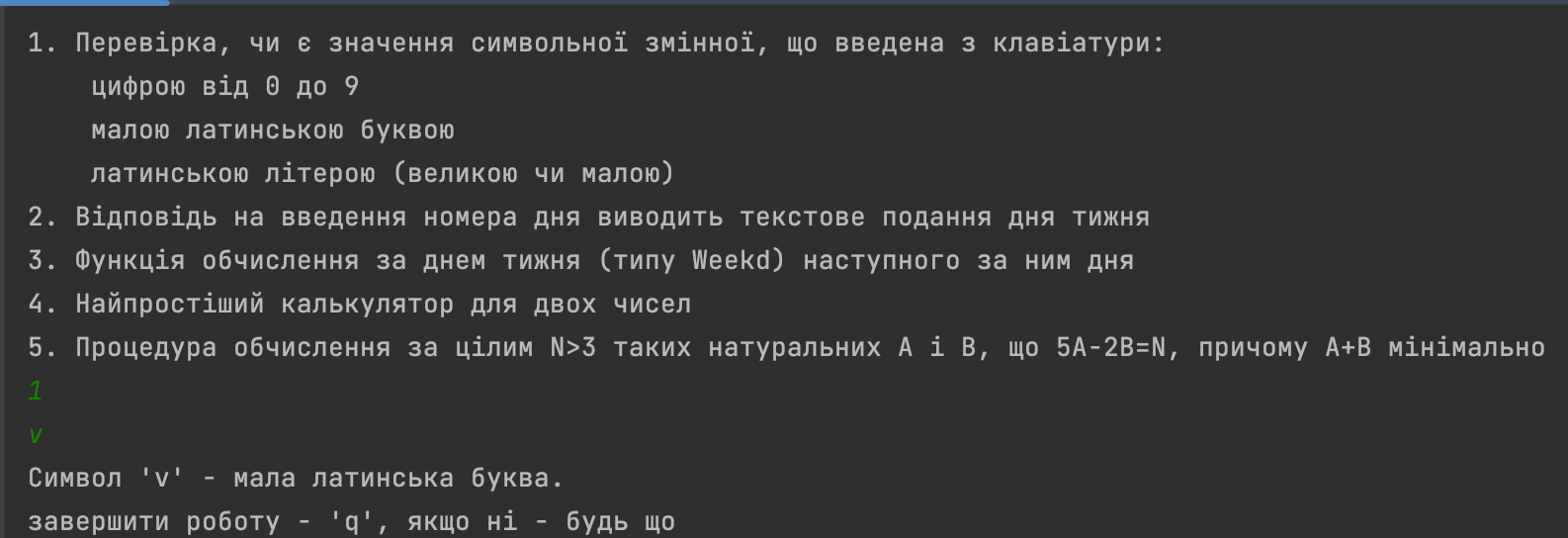
Українська літера



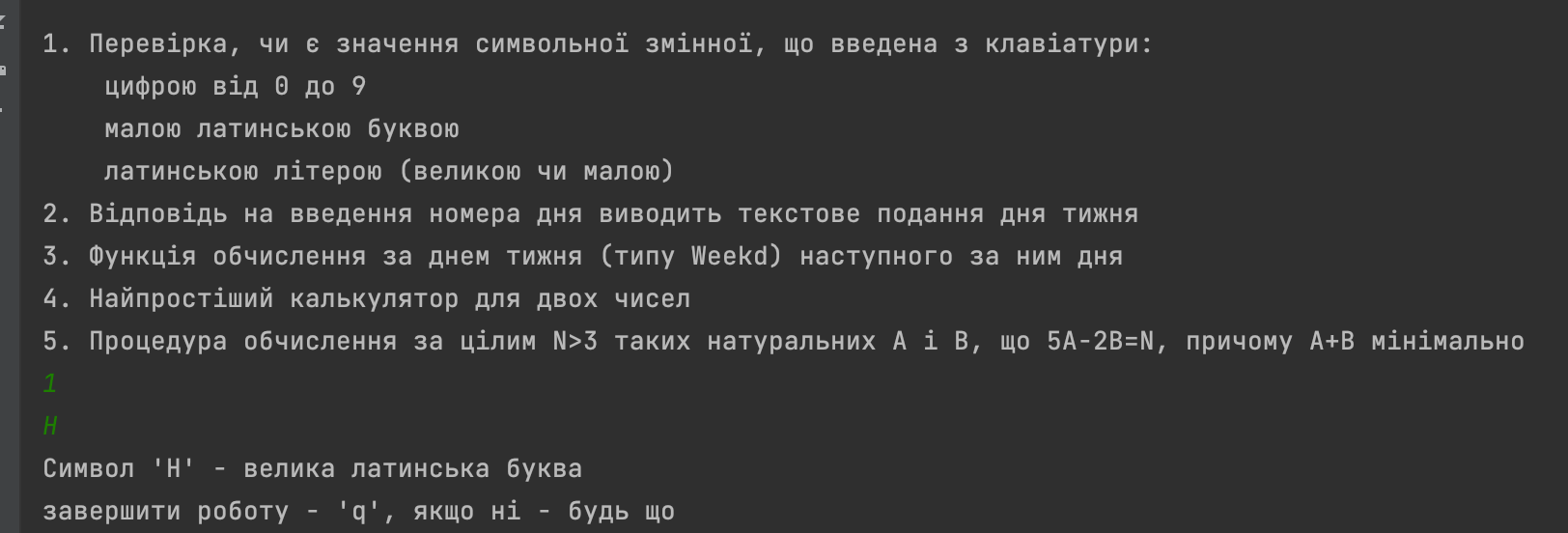
Цифра

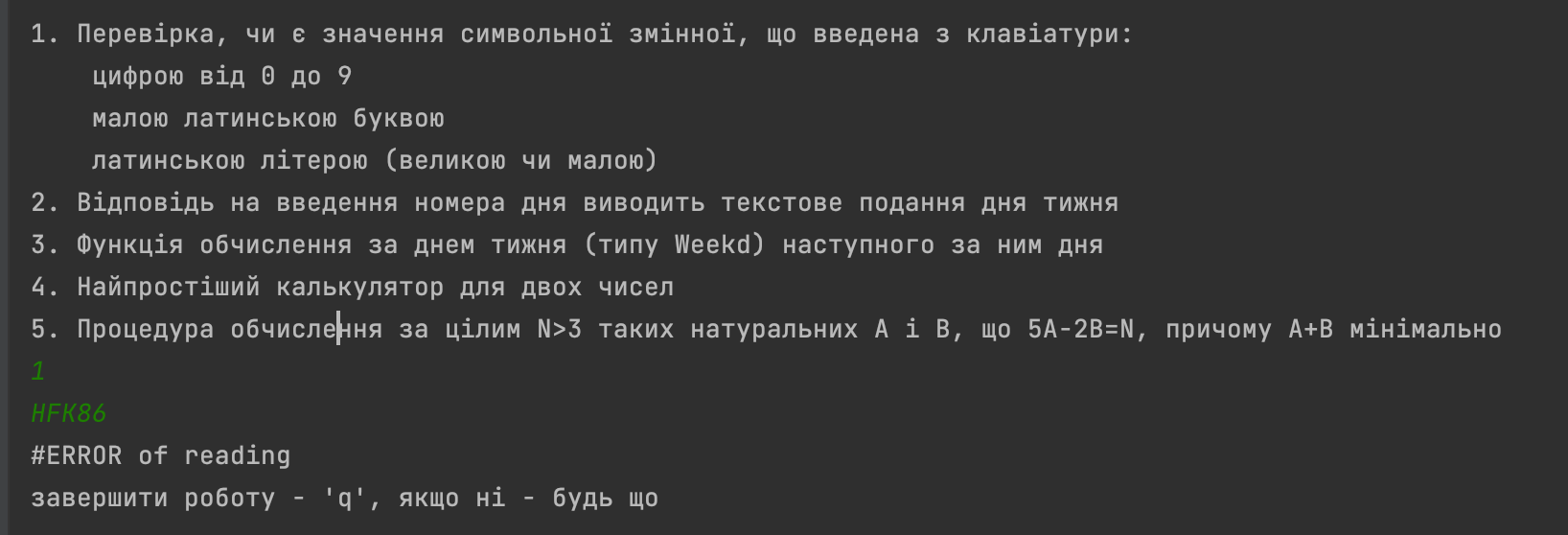


Латинська маленька літера



Латинська велика літера



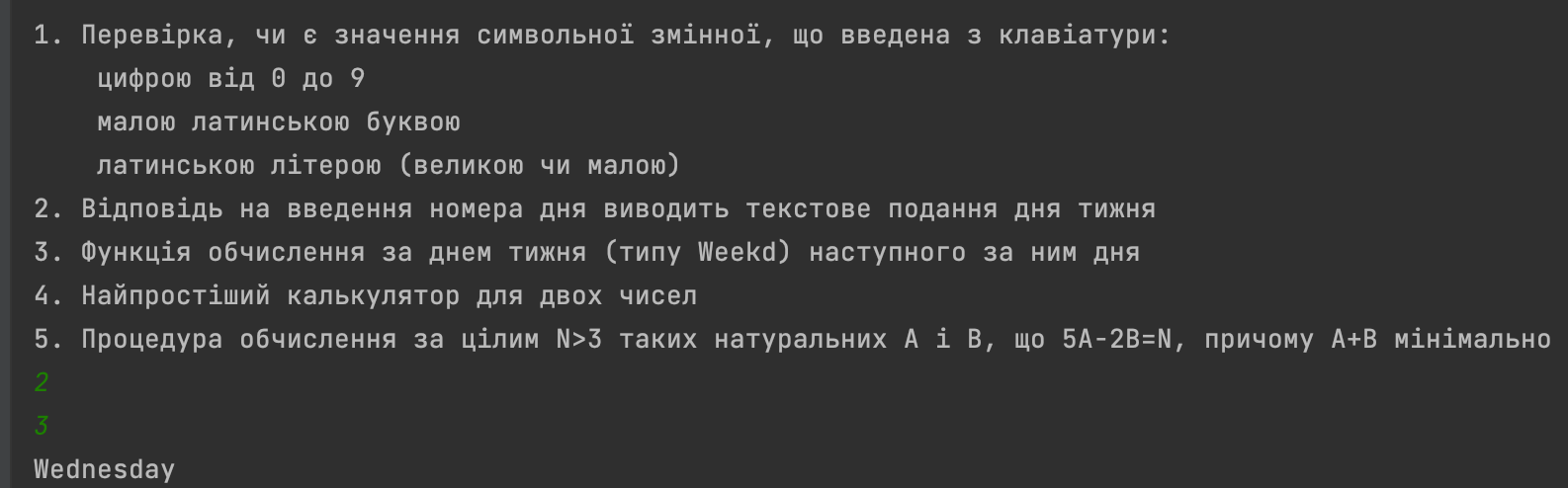
Декілька символів

**Завдання 2:** У розкладі рейсів літаків дні тижня позначаються номерами від 1 до 7. Припустимо, що в програмі дні тижня подаються enum типом Weekd. Написати програму, що у відповідь на введення номера дня виводить текстове подання дня тижня.

*/\*\*  
 \* Функція для отримання дня тижня за його номером  
 \** ***@param*** *num номер дня тижня  
 \** ***@return*** *об'єкт типу Weekd, що відповідає введеному номеру, або null, якщо такого дня немає  
 \*/*fun getWeekDay(num: Int): Weekd? {  
 return Weekd.values().*firstOrNull* **{ it**.num == num **}**}

*/\*\*  
 \* Функція для отримання назви дня тижня за його номером  
 \** ***@return*** *назва дня тижня або повідомлення про помилку  
 \*/*fun getWeekdDayName(): String {  
 val value = *readChar*()  
 return if (value != null && value in '1'..'7') "${*getWeekDay*(Character.getNumericValue(value))}" else "Помилка. Некоректний інпут"  
}

**Приклади виконання**

****



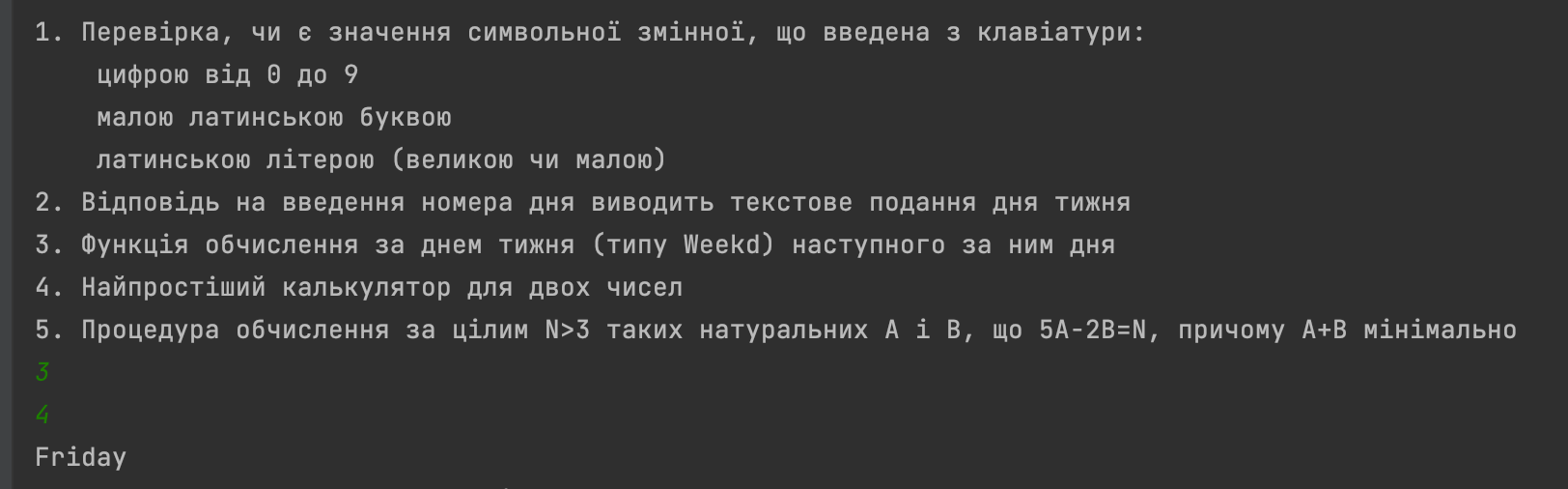
**Завдання 3:** За понеділком іде вівторок тощо, а за неділею – понеділок. Написати функцію обчислення за днем тижня (типу Weekd) наступного за ним дня.

*/\*\*  
 \* Функція для отримання дня тижня за його номером  
 \** ***@param*** *num номер дня тижня  
 \** ***@return*** *об'єкт типу Weekd, що відповідає введеному номеру, або null, якщо такого дня немає  
 \*/*fun getWeekDay(num: Int): Weekd? {  
 return Weekd.values().*firstOrNull* **{ it**.num == num **}**}

*/\*\*  
 \* Функція для отримання наступного дня тижня після заданого  
 \** ***@param*** *weekDay поточний день тижня  
 \** ***@return*** *назва наступного дня тижня  
 \*/*fun getNextWeekDay(weekDay: Weekd): String {  
 return "${Weekd.values().*first* **{ it**.num == if (weekDay.num == 7) 1 else weekDay.num + 1 **}**}"  
}

*/\*\*  
 \* Функція для отримання наступного дня тижня після заданого за його номером  
 \** ***@return*** *назва наступного дня тижня або повідомлення про помилку  
 \*/*fun getWeekdNextDay(): String {  
 val value = *readChar*()  
 return if (value != null && value in '1'..'7') "${  
 *getWeekDay*(Character.getNumericValue(value))?.*let* **{** *getNextWeekDay*(  
 **it** )  
 **}** }" else "Введено неккоректні данні"  
}

**Приклади виконання**



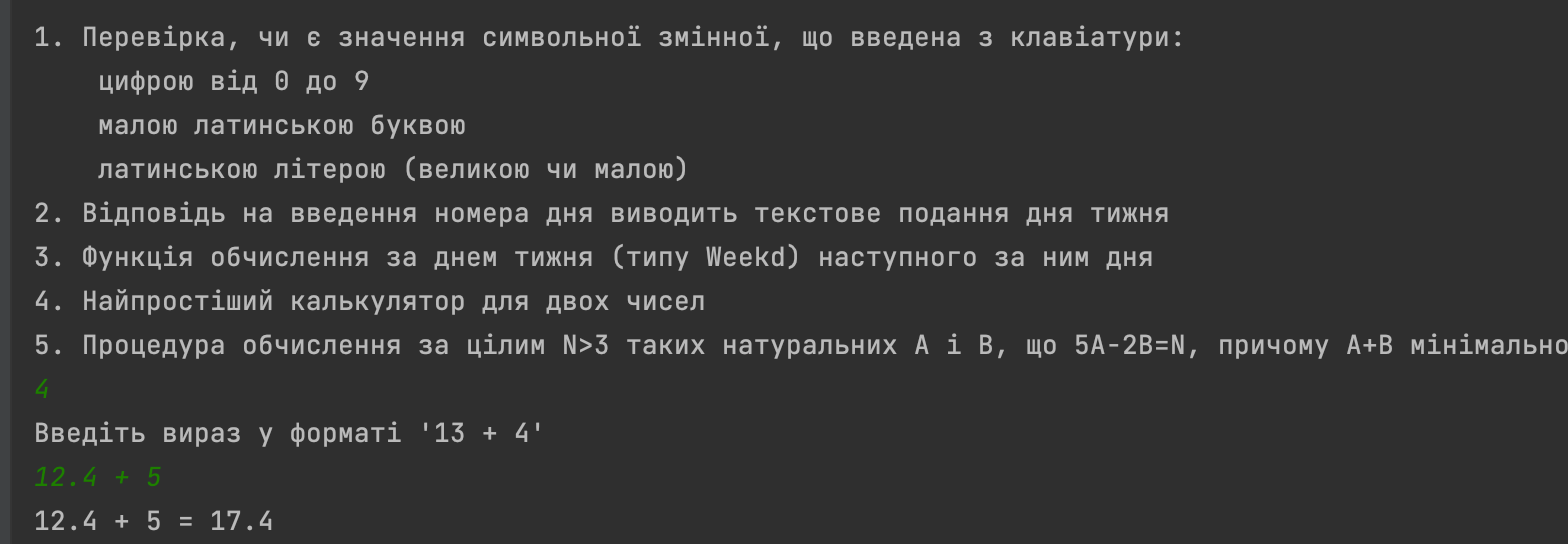


**Завдання 4:** Написати “найпростіший калькулятор”, що отримує на вхід 2 числа і операцію над ними після чого повертає результат обчислень.

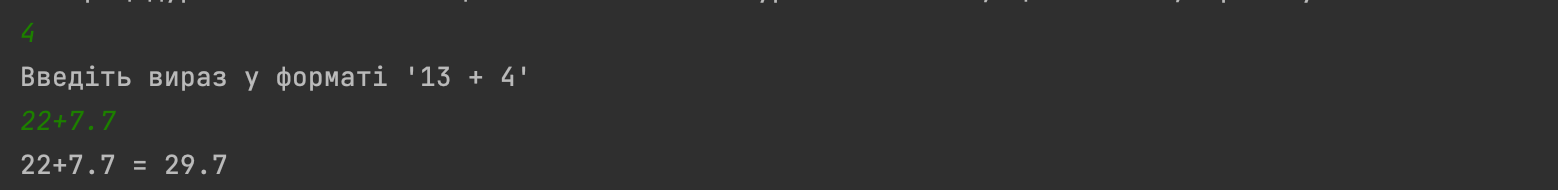
*/\*\*  
 \* Функція-калькулятор для обчислення результату арифметичної операції між двома числами  
 \** ***@param*** *a перше число  
 \** ***@param*** *b друге число  
 \** ***@param*** *operation символ арифметичної операції  
 \** ***@return*** *результат обчислення  
 \*/*fun calculate(a: Double, b: Double, operation: Char): Double {  
 return when (operation) {  
 '+' -> a + b  
 '-' -> a - b  
 '\*' -> a \* b  
 '/' -> a / b  
 else -> throw IllegalArgumentException("Некоректна операція: $operation")  
 }  
}

*/\*\*  
 \* Функція для розбору введеного користувачем виразу на складові (числа та операцію)  
 \** ***@param*** *input введений користувачем вираз  
 \** ***@return*** *значення: два числа та операція, або null, якщо вираз некоректний  
 \*/*fun getComponents(input: String): Triple<Double, Double, Char>? {  
 val regex = "([-+]?[0-9]\*\\.?[0-9]+)([+\\-\*/])([-+]?[0-9]\*\\.?[0-9]+)".*toRegex*()  
 val matchResult = regex.matchEntire(input.*replace*("\\s+".*toRegex*(), ""))  
  
 if (matchResult != null) {  
 val (aStr, operation, bStr) = matchResult.destructured  
 val a = aStr.*toDoubleOrNull*()  
 val b = bStr.*toDoubleOrNull*()  
  
 if (a != null && b != null) {  
 return Triple(a, b, operation[0])  
 }  
 }  
 *println*("Некоректний інпут: $input")  
 return null  
}  
  
*/\*\*  
 \* Функція для обчислення результату введеного користувачем виразу  
 \** ***@return*** *результат обчислення у текстовому форматі  
 \*/*fun expressionCalculation(): String {  
 *println*("Введіть вираз у форматі '13 + 4'")  
 val input = *readLine*() ?: return "Line Reading Error"  
 val components = *getComponents*(input)  
 if (components != null) {  
 val (a, b, operation) = components  
 try {  
 return "$input = ${*calculate*(a, b, operation)}"  
 } catch (e: IllegalArgumentException) {  
 *println*(e.message)  
 }  
 }  
 return "Помилка обробки"  
}

**Приклади виконання**



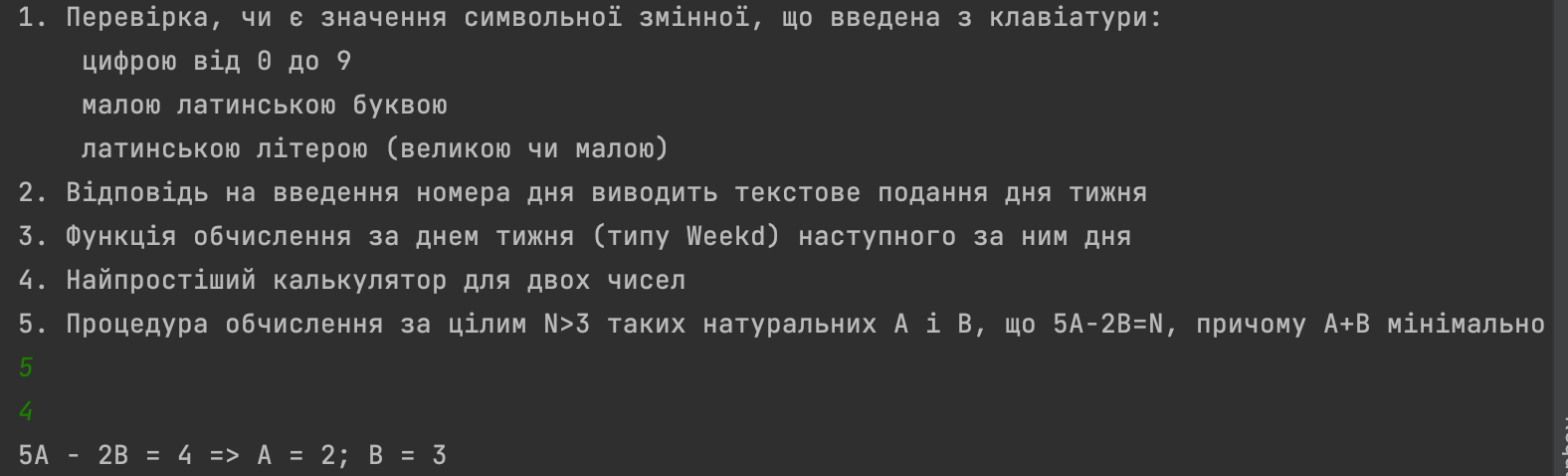




**Завдання 5:** Написати процедуру обчислення за цілим N>3 таких натуральних A і B, що 5A-2B=N, причому A+B мінімально.

*/\*\*  
 \* Функція для обчислення значень A та B за заданим числом N за умовою 5A - 2B = N  
 \** ***@return*** *значення A та B у текстовому форматі  
 \*/*fun calculateABNExperssion(): String {  
 val num = *readLine*()?.*toInt*()  
  
 if (num == null || num < 4) {  
 *println*("Число менше 4")  
 return "Некоректний інпут"  
 }  
  
 val a = when (num.rem(5)) {  
 1, 3 -> (num / 5) + 1  
 else -> num / 5 + 2  
 }  
  
 val b = (a \* 5 - num) / 2  
  
 return "5A - 2B = $num => A = $a; B = $b"  
}

**Приклади виконання**

****

**Висновок**

В результаті виконання лабораторної роботи №12 було поглиблено знання та навички роботи з мовою програмування Kotlin. Було отримано знання та навички в роботі з enum класами в Kotlin, їх декларуванням, ініціалізацією та використанням у програмі.