Лабораторна робота 3.

Використання типів, що пітримують null значення.

**Завдання**

Створіть клас "Користувач", який має ім'я (String), вік (Int?), імейл (String?) та адресу (String?). Реалізуйте функцію, яка виводить інформацію про користувача, ігноруючи null значення.

**Хід роботи**

Для реалізації завдання створення класу "Користувач" почнемо з визначення структури класу, який буде містити чотири поля: ім'я, вік, імейл та адресу. Клас буде мати функцію для виведення інформації про користувача, при цьому ігноруючи значення полів, які мають значення null. Це дозволяє уникнути виведення непотрібної інформації та зробити результат більш читабельним.

Створюємо клас User, який має чотири властивості: name типу String?, age типу Int?, email типу String?, та address типу String?. Усі властивості є nullable, оскільки деякі з них можуть бути відсутніми. Для виведення інформації про користувача створюємо метод displayUserInfo, який перевіряє кожну властивість на null і виводить відповідне повідомлення, якщо значення властивості відсутнє.

Основна програма використовує клас Scanner для зчитування вводу користувача з клавіатури. Спочатку ми запитуємо у користувача його ім'я і зчитуємо його за допомогою scanner.nextLine(). Потім запитуємо вік, але оскільки це необов'язкове поле, перевіряємо, чи було введено значення. Якщо поле залишилося порожнім, встановлюємо значення null, інакше перетворюємо введене значення на ціле число.

Аналогічно обробляємо введення для імейлу та адреси користувача. Використовуємо функцію takeIf { it.isNotBlank() }, щоб встановити значення null у випадку, якщо відповідне поле залишилося порожнім.

Після зчитування усіх даних створюємо об'єкт User, передаючи зчитані значення в конструктор. Далі викликаємо метод displayUserInfo для виведення інформації про користувача, ігноруючи значення null. В кінці закриваємо об'єкт Scanner, щоб звільнити ресурси.

Повний код програми:

import java.util.Scanner

class User(

val name: String?,

val age: Int?,

val email: String?,

val address: String?

) {

fun displayUserInfo() {

println("Ім'я: ${name ?: "Немає імені"}")

println("Вік: ${age ?: "Невідомий"}")

println("Email: ${email ?: "Немає email"}")

println("Адреса: ${address ?: "Немає адреси"}")

}

}

fun main() {

val scanner = Scanner(System.`in`)

// Зчитування даних про користувача з клавіатури

println("Введіть ім'я користувача:")

val name = scanner.nextLine()

println("Введіть вік користувача (якщо невідомий, натисніть Enter):")

val ageInput = scanner.nextLine()

val age = if (ageInput.isBlank()) null else ageInput.toIntOrNull()

println("Введіть email користувача (якщо немає, натисніть Enter):")

val email = scanner.nextLine().takeIf { it.isNotBlank() }

println("Введіть адресу користувача (якщо немає, натисніть Enter):")

val address = scanner.nextLine().takeIf { it.isNotBlank() }

// Створення об'єкта користувача з введеними даними

val user = User(name.takeIf { it.isNotBlank() }, age, email, address)

// Виведення інформації про користувача, ігноруючи null значення

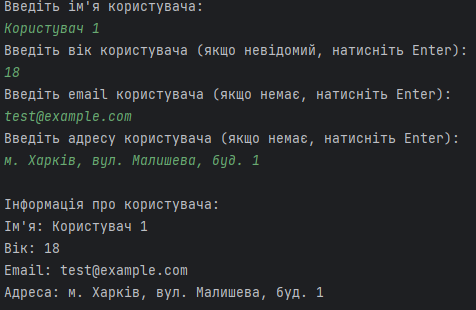
println("\nІнформація про користувача:")

user.displayUserInfo()

scanner.close()

}

**Результати роботи**

****

Тестування програми