МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Звіт
з практичної роботи №3
із дисципліни «Аналіз та рефакторинг коду програмного забезпечення»
на тему “ Refactoring Methods ”

Виконав:
ПЗПІ-22-1 Тимощук Денис

Перевірив:
доцент кафедри програмної інженерії, к.т.н.
Лещинський Володимир Олександрови.

Харків 2024

Мета роботи

Метою даної практичної роботи є засвоєння основних методів рефакторингу коду на основі реальних прикладів з моїх програмних проєктів. Я прагну навчитися ідентифікувати проблеми в коді та використовувати відповідні методи рефакторингу, такі як виділення методу, спрощення умовних виразів і заміна тимчасових змінних на запити, для покращення якості коду. Ця робота допомогла мені розвинути навички написання чистого, ефективного та підтримуваного коду, що є важливим аспектом програмування.

Висновки

Застосовані методи рефакторингу значно покращили якість мого коду, зробивши його більш читабельним і легким для підтримки. Виділення методу (Extract Method) дозволило розділити складний код на окремі частини, що поліпшило його модульність і спростило тестування. Заміна тимчасових змінних на запити (Replace Temp with Query) усунула непотрібні змінні, що зробило код простішим і менш схильним до помилок. Спрощення умовних виразів (Simplify Conditional Expression) допомогло скоротити складну логіку, полегшуючи розуміння коду та підвищуючи його підтримку і масштабованість у майбутньому. Ці рефакторингові методи підкреслюють важливість підтримки чистоти та зрозумілості коду, що сприяє його ефективності та зручності в подальшій роботі.

Додаток А

Скріншоти презинтації представленої на практичному занятті

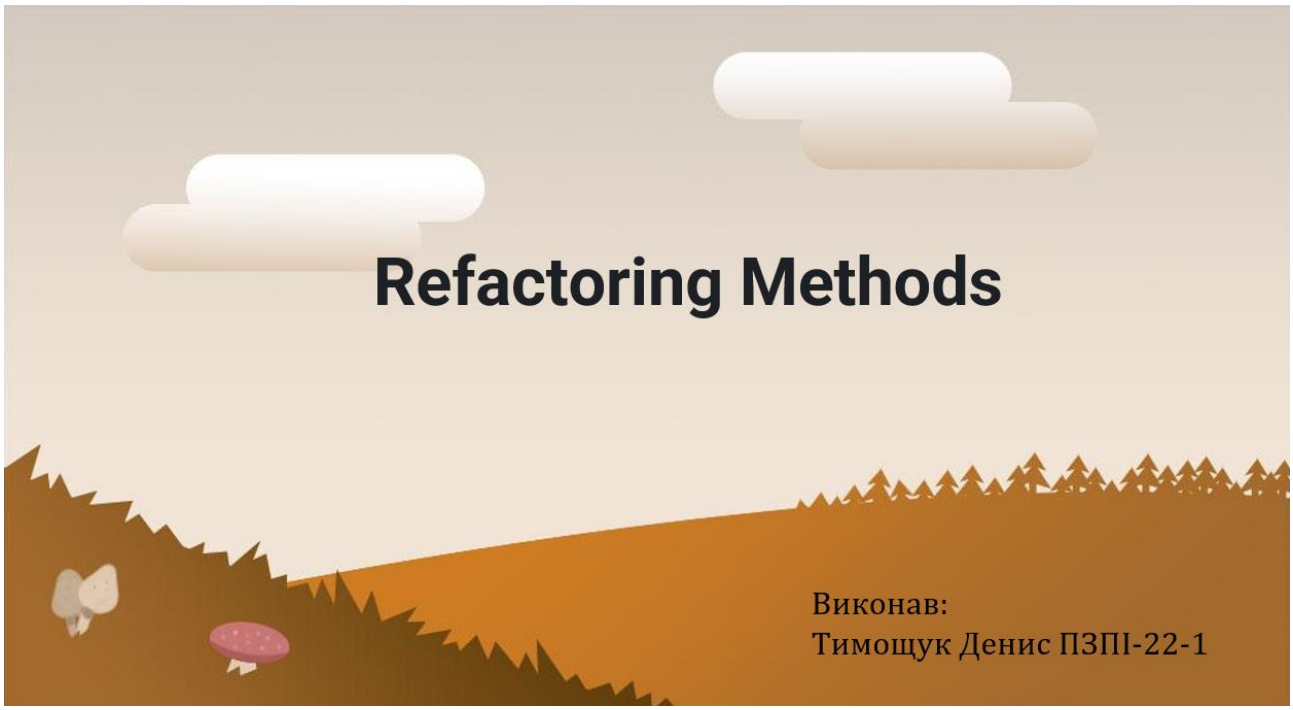

Рисунок 1 – Титульна сторінка

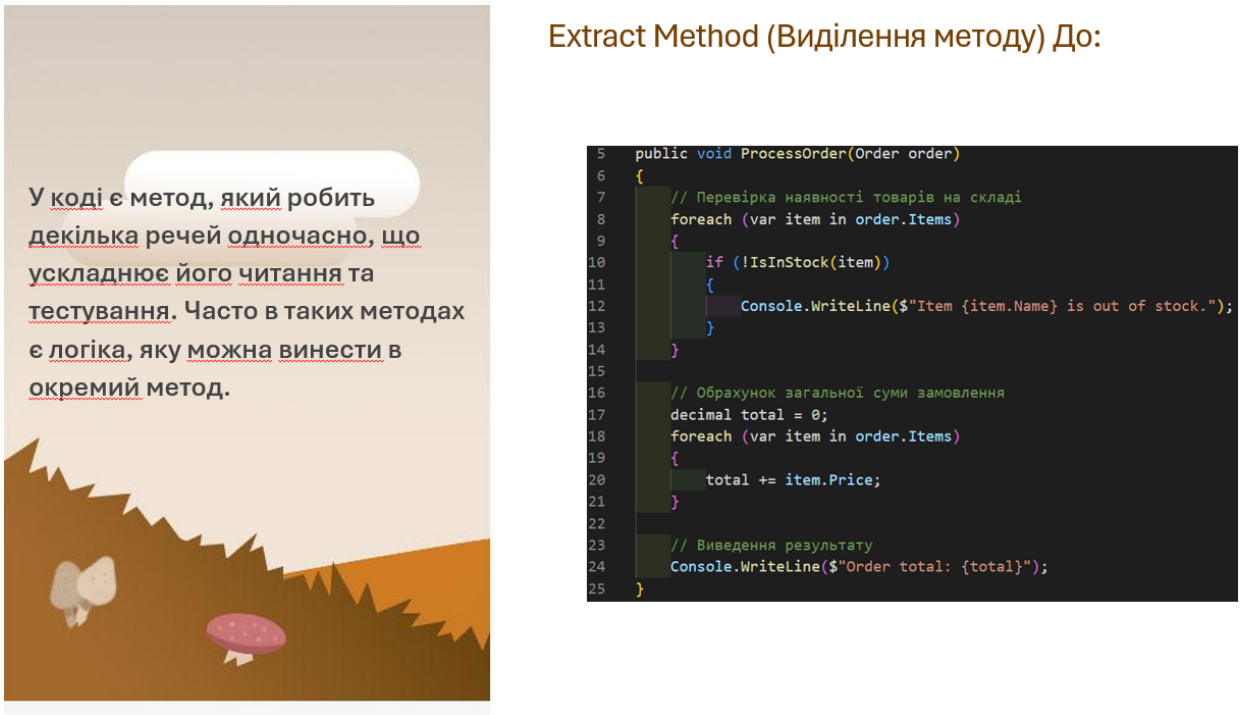

Рисунок 2 – Перший метод “Extract Metod” до редагування

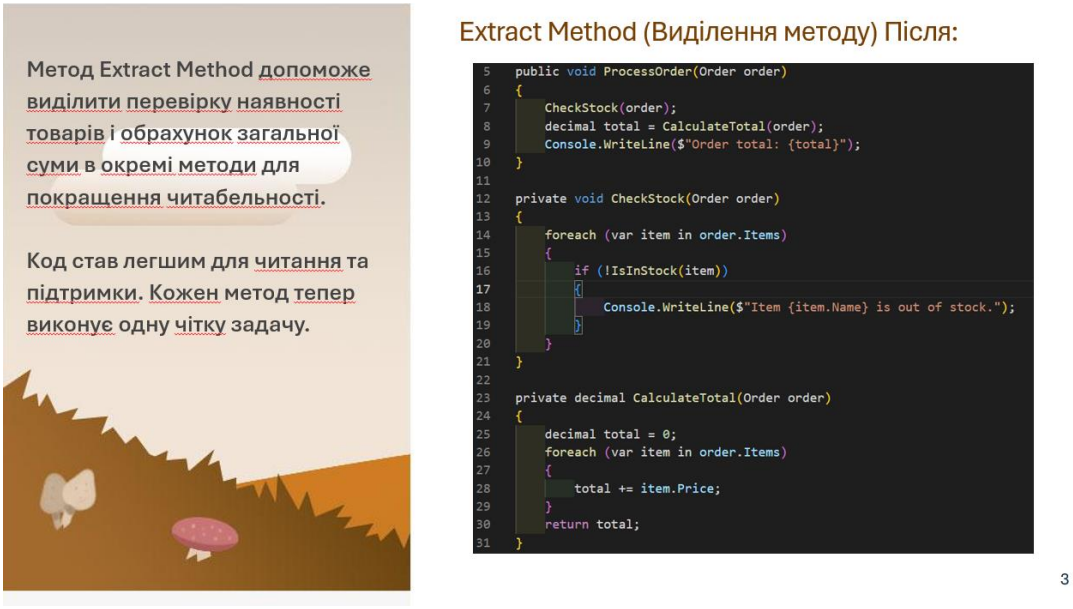

Рисунок 3 – Перший метод “Extract Metod” після редагування

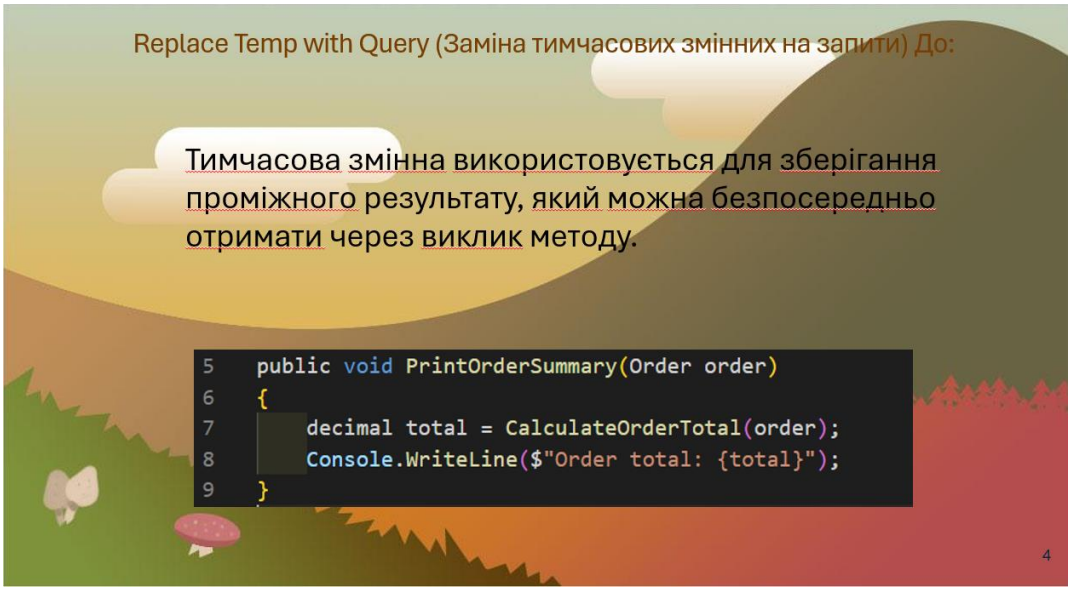

Рисунок 4 – Другий метод “Replace Temp with Query” до редагування

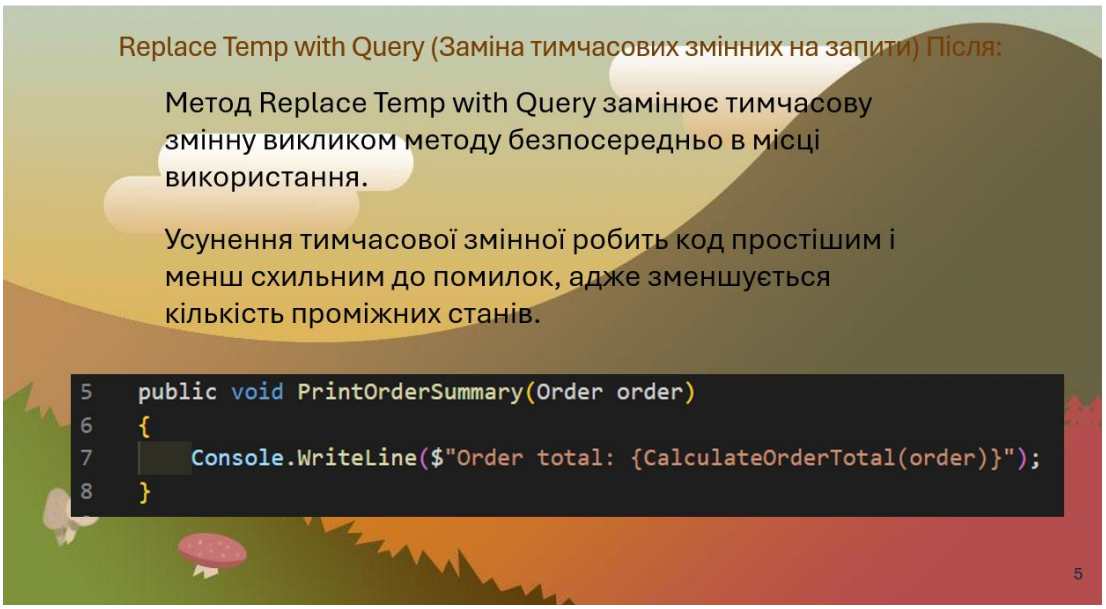

Рисунок 5 – Другий метод “Replace Temp with Query” після редагування

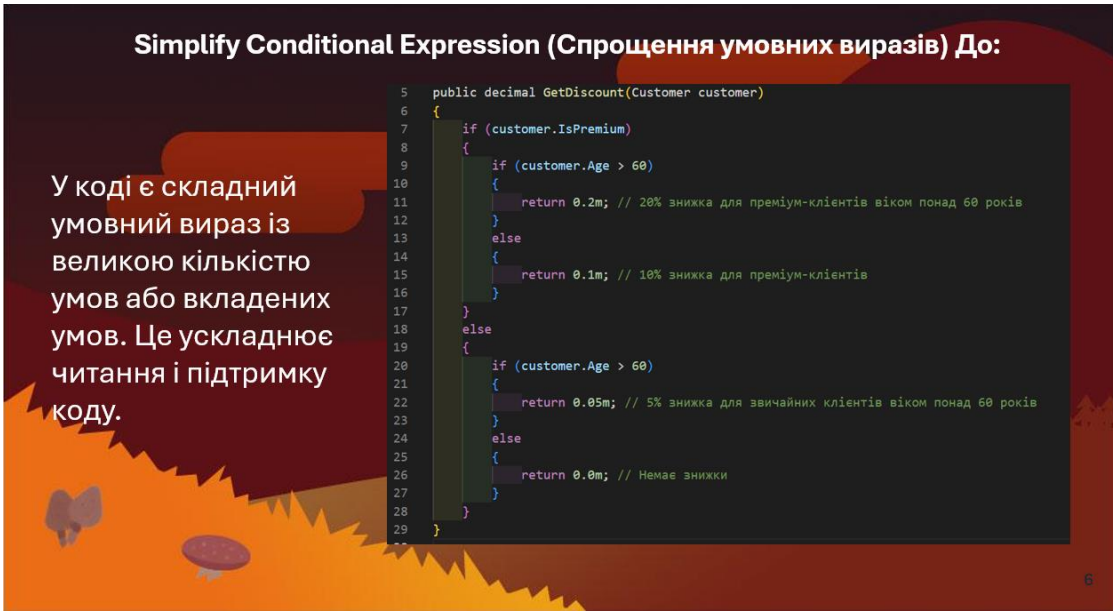

Рисунок 6 – Третій метод “Simplify Conditional Expression” до редагування

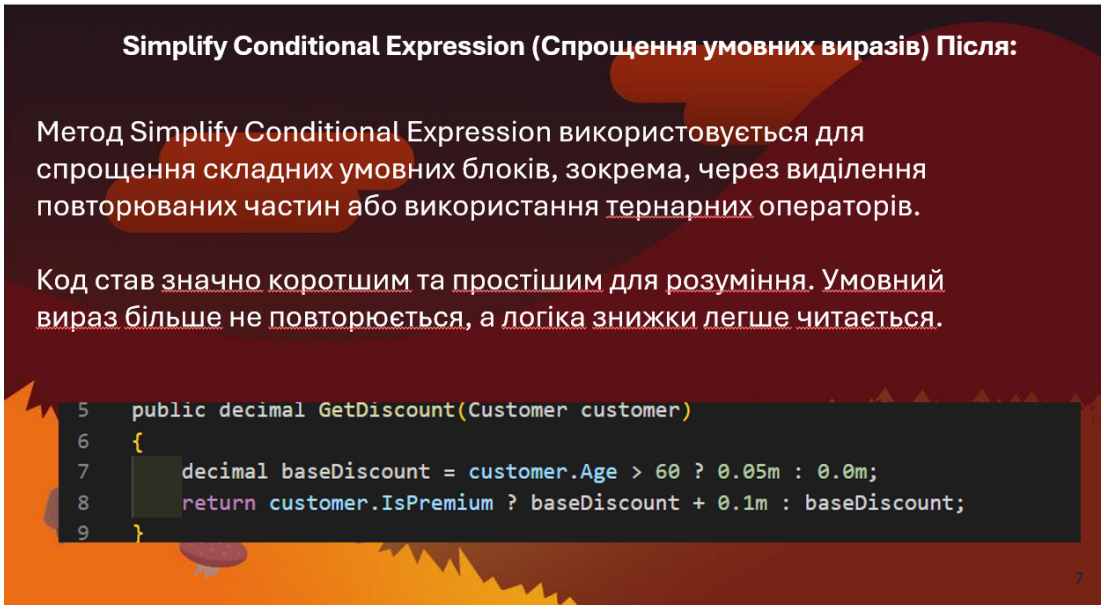

Рисунок 7 – Третій метод “Simplify Conditional Expression” після редагування

Висновки

Застосовані методи рефакторингу значно покращили якість коду, зробивши його більш читабельним і легким для підтримки. Виділення методу (Extract Method) розділило складний код на окремі частини, що поліпшило його модульність і спростило тестування. Заміна тимчасових змінних на запити (Replace Temp with Query) усунула непотрібні змінні, зробивши код простішим і менш схильним до помилок, завдяки усуненню проміжних станів. Спрощення умовних виразів (Simplify Conditional Expression) скоротило складну логіку, полегшивши розуміння коду та підвищивши його підтримку і масштабованість у майбутньому.

Рисунок 8 – Слайд із висновками

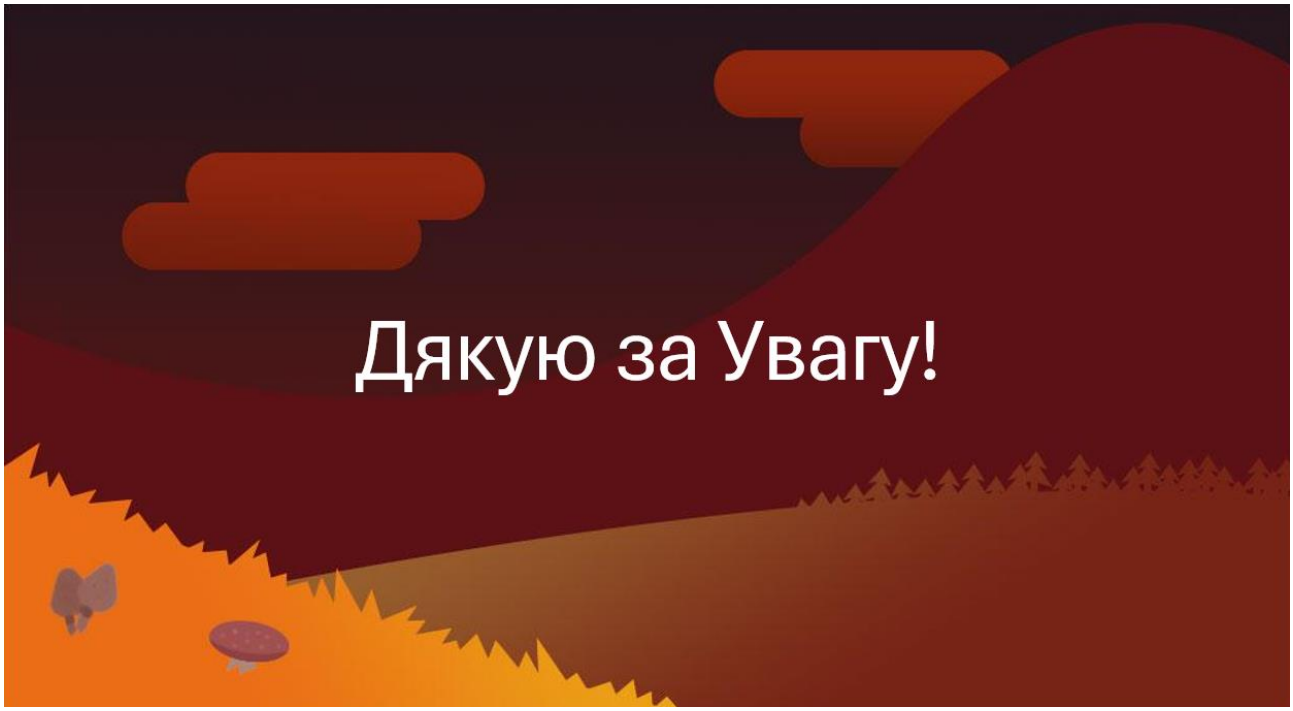

Рисунок 9– Слайд із подякоую за увагу