НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕЇНІКИ

ЛАБОРАТОРНА РОБОТА 2

з дисципліни:

«Сучасні методології і технології розробки програмного забезпечення»

на тему:

«Багатошарова архітектура програмних додатків. Використання системи контролю версій. Основи UML.»

Студента 3 курсу групи ІТ-81

Венделовського Івана Сергійовича

Кількість балів:\_\_\_\_\_\_\_\_\_\_\_\_ Оцінка\_\_\_\_\_\_\_\_

Викладач:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_к.т.н. Штифурак Юрій Михайлович

(оцінка)

Київ – 2021

ЗМІСТ

[1 Постанока задачі 3](#_Toc67165175)

[1 Реалізація основних інтерфейсів 4](#_Toc67165176)

[1.1 Інтерфейси рівня бізнес логіки 4](#_Toc67165177)

[1.2 Інтерфейси рівня доступу до даних 5](#_Toc67165178)

[2 реалізація інверсії залежностей 7](#_Toc67165179)

[2.1 Пояснення прийнятого рішення 7](#_Toc67165180)

[2.2 Конфігурація та налаштування інверсії залежностей 7](#_Toc67165181)

[Висновки 8](#_Toc67165182)

[Список використаних джерел 9](#_Toc67165183)

# 1 Постанока задачі

1) Реалізувати основні інтерфейси відповідно до розробленої у попередній лабораторній роботі UML моделі проекту.

2) Реалізувати п’ятий принцип солід «інверсії залежностей», з допомогою IoС контейнеру чи іншої сутності що відповідає за впровадження залежностей між модулями, шарами, сервісами та дозволяє розбирати тестопридатне програмне забезпечення.

# 1 Реалізація основних інтерфейсів

## 1.1 Інтерфейси рівня бізнес логіки

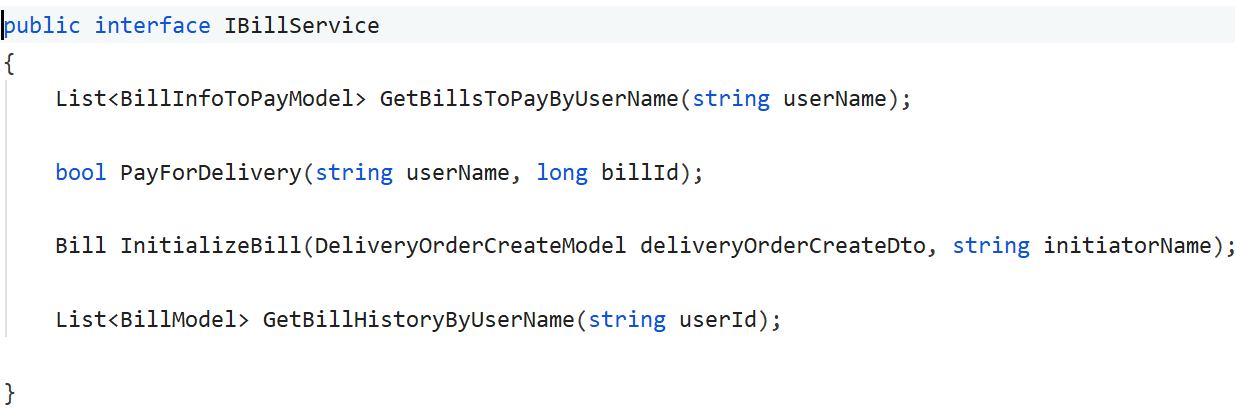


Рисунок 1.1 – Інтерфейс сервісу рахунків.

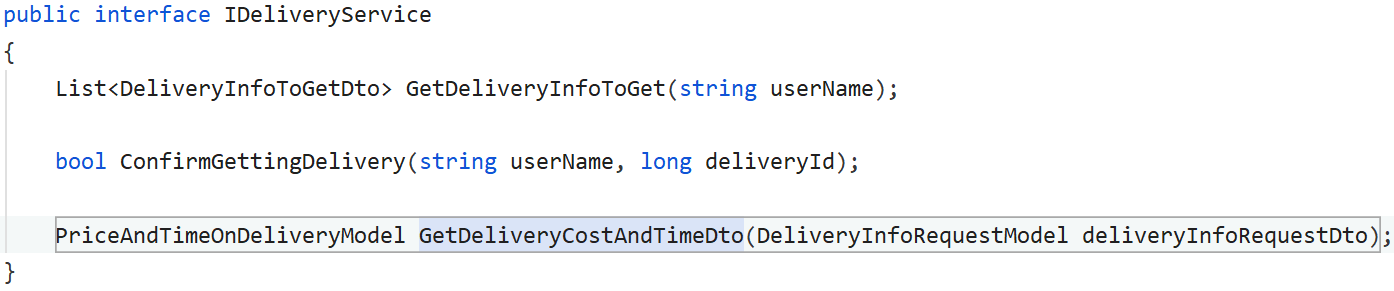


Рисунок 1.2 – Інтерфейс сервісу доставок.

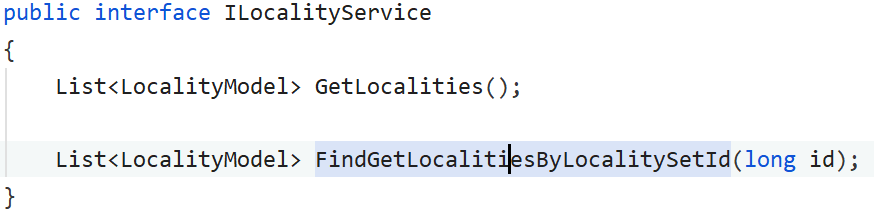


Рисунок 1.3 – Інтерфейс сервісу локацій.

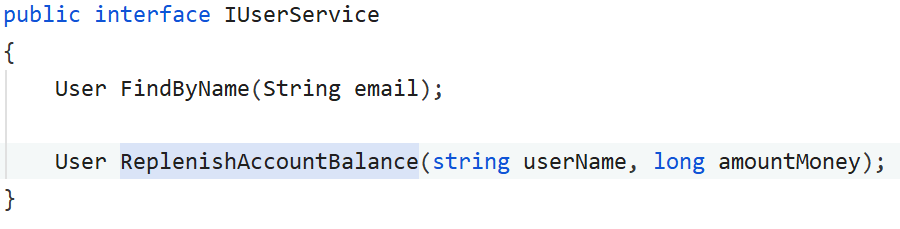


Рисунок 1.4 – Інтерфейс сервісу користувачів.

## 1.2 Інтерфейси рівня доступу до даних

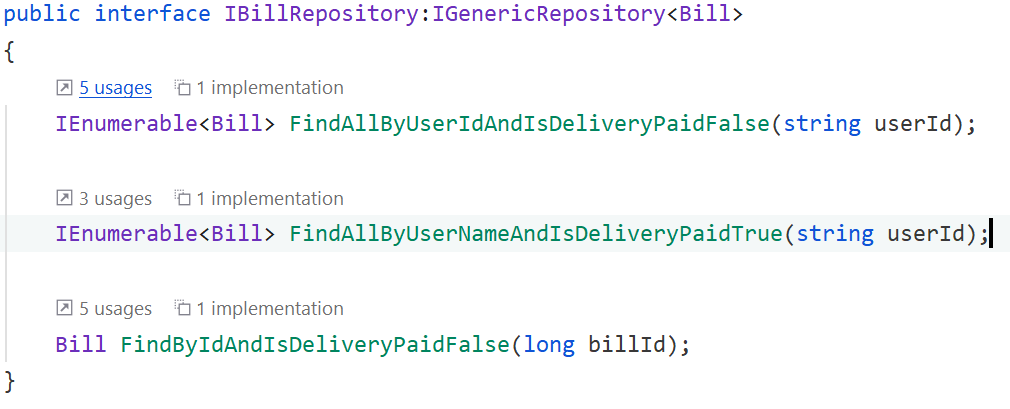


Рисунок 1.5 – Інтерфейс репозиторію рахунків.

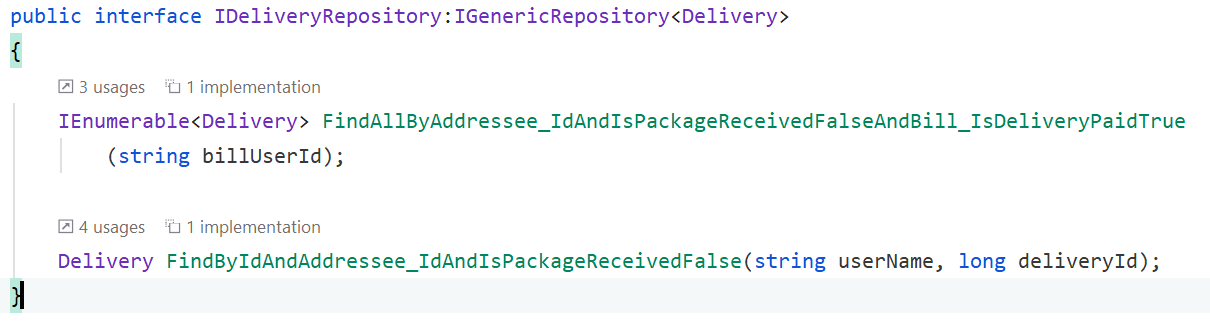


Рисунок 1.6 – Інтерфейс репозиторію доставок.

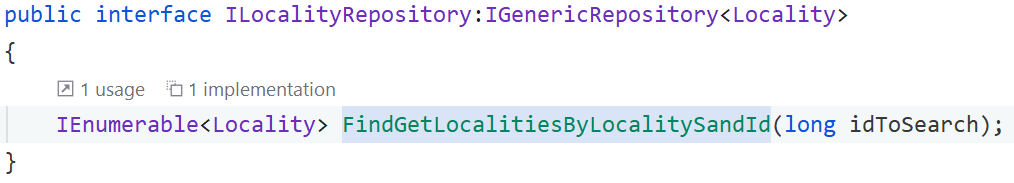


Рисунок 1.7 – Інтерфейс репозиторію локацій.

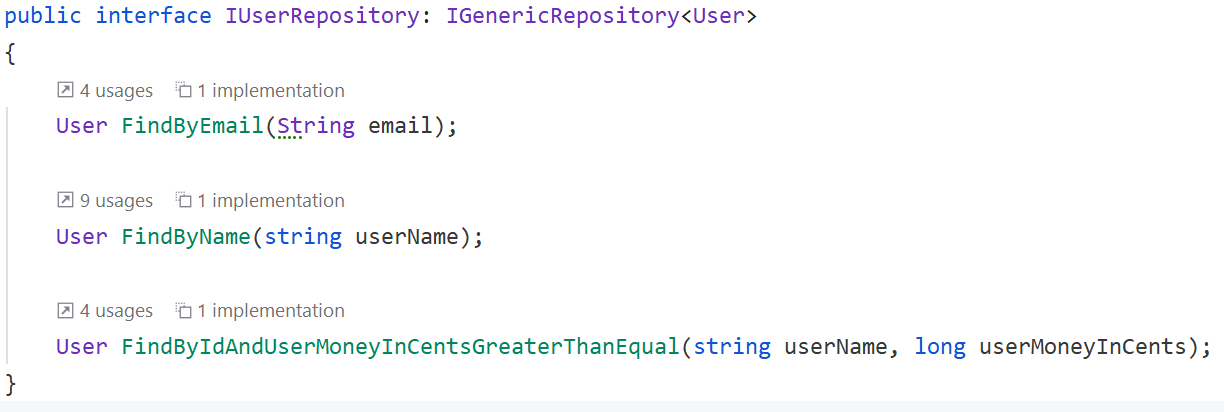


Рисунок 1.8 – Інтерфейс репозиторію користувачів.

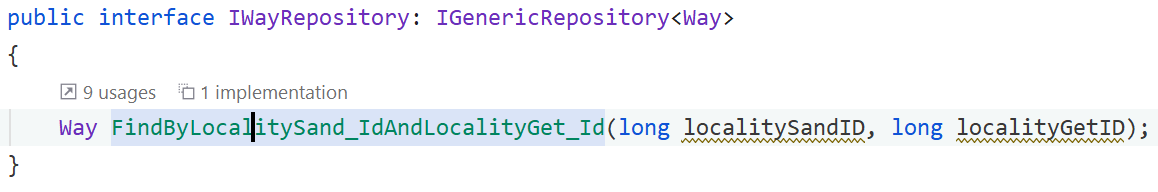


Рисунок 1.9 – Інтерфейс репозиторію маршрутів.

# 2 реалізація інверсії залежностей

## 2.1 Пояснення прийнятого рішення

В якості реалізації контролю інверсії залежностей було вирішено застосувати інстроменти що пропунує .NET core фреймворк. Таке рішення було прийнято оскільки, за загальною практикою краще використовувати перевіренів бібліотеки для будь-яких шаблонних задач. В джава реалізації даного проекту запропоновано ручну реалізацію IOC контейнеру. ЇЇ можна знайти за посиланням «https://github.com/VINIPOOH/ServletFinalProject»

## 2.2 Конфігурація та налаштування інверсії залежностей

На рисунку 2.1 запропоновано частину конфігурації із «Startup.cs» файлу.



Рисунок 2.1 – Конфігурація інверсії залежностей в стартап файлі.

# Висновки

В дані роботі було створено основні інтерфейси розроблені в процесі попередньої лабораторної роботи, а також проаналізовано різні варіанти досягнення інверсії залежностей та обрано ASP net в якості бібліотеки для забезпечення інверсії залежностей.

# Список використаних джерел

1. Ноубл, Дж., Андерсон, Т., Брэйтуэйт, Г., Казарио, М., Третола, Р. Flex 4. Рецепты программирования. — БХВ-Петербург, 2011. — С. 548. — 720 с
2. Самоучитель UML 2. – СПб.: БХВ-Петербург, 2007. – 567 с.: ил. ISBN 978-5-94157-878-8
3. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. П75 Приемы объектно-ориентированного проектирования. Паттерны проектирования. — СПб: Питер, 2001. — 368 с.: ил. (Серия «Библиотека программиста») ISBN 5-272-00355-1
4. Мартин Фаулер., Чистий код: создание, анализ и рефакторинг. – СПб.: Питер, 2019. – 464 с.: ил.