МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ПОЛІТЕХНІКА»

Кафедра інформаційних технологій

**Звіт**

Практична робота 2.

З дисципліни:  Об’єктно-орієнтоване програмування

Тема: «Розробка прототипів графічного інтерфейсу. Виявлення класів і зв'язків за допомогою CRC-карток»

Виконав:

студент групи АД 221

Прудкий Дмитро Павлович

Перевірив:

доцент Рудніченко Н.Д.

Оцінка\_\_\_\_\_\_\_\_\_\_\_\_

Одеса 2024

ЗМІСТ

[ВСТУП 2](#_Toc161584168)

[ТЕОРЕТИЧНА ЧАСТИНА 4](#_Toc161584169)

[ПРАКТИЧНА ЧАСТИНА 5](#_Toc161584170)

[ВИСНОВОК 8](#_Toc161584171)

[ЛІТЕРАТУРА 9](#_Toc161584172)

# ВСТУП

Мета роботи:

* розробити прототипи графічного інтерфейсу на основі користувацьких історій;
* визначити користувачів і написати user stories (користувацькі історії) для розроблюваного застосунку;
* сформувати беклог продукту на основі написаних користувацьких історій.

# ТЕОРЕТИЧНА ЧАСТИНА

Розробка графічного інтерфейсу (GUI) в сучасному програмуванні є ключовим етапом в процесі створення програмного продукту. GUI забезпечує взаємодію користувача з програмою, тому важливо вже на ранніх етапах розробки ретельно визначити його структуру та функціональність. Один із способів цього досягнення - використання CRC-карток.

CRC-картки (Class-Responsibility-Collaborator) - це інструмент для аналізу та проектування об'єктно-орієнтованих систем. Кожна CRC-карта визначає клас у системі, його відповідальності (responsibilities) та співпрацюючі класи (collaborators).

Кроки розробки прототипів графічного інтерфейсу з використанням CRC-карток:

1. Визначення класів GUI: Спочатку визначаємо основні класи, які будуть присутні у GUI. Наприклад, можуть бути класи для вікон, кнопок, полів введення, тощо.
2. Визначення відповідальностей: Для кожного класу визначаємо його основні відповідальності у системі. Наприклад, клас вікна може відповідати за його відображення та інтерактивність.
3. Визначення співпрацюючих класів: Для кожного класу визначаємо інші класи, з якими він взаємодіє. Наприклад, вікно може співпрацювати з класом кнопки для обробки подій натискання.
4. Побудова прототипу: На основі визначених CRC-карток будуємо прототип GUI, використовуючи відомі фреймворки чи бібліотеки.
5. Тестування та вдосконалення: Після побудови прототипу важливо протестувати його з метою виявлення недоліків та можливостей для вдосконалення.

# ПРАКТИЧНА ЧАСТИНА

1. Разработать для своей курсовой работы Storyboard и макеты окон.
2. На основе разработанных пользовательских историй сформировать CRC-карточки.
3. Продемонстрировать результаты первого спринта.

1. Storyboard:

1. Користувач відкриває додаток каси для придбання железничного квитка.
2. Користувач вибирає маршрут та дату подорожі.
3. Після вибору маршруту користувач переходить до сторінки з вибором доступних поїздів.
4. Користувач обирає певний поїзд і клас вагону.
5. Після вибору поїзда користувач обирає кількість та тип квитків (наприклад, дорослий, дитячий, плацкарт, купе тощо).
6. Користувач вводить особисті дані та здійснює оплату квитків.
7. Після успішної оплати користувач отримує підтвердження покупки та можливість зберегти квитки на мобільний пристрій або роздрукувати.

*Макети вікон:*

* Головна сторінка додатку (включає форму пошуку маршруту та вибір дати).
* Сторінка з вибором поїзда (показує доступні поїзди для обраного маршруту та дати).
* Сторінка з вибором квитків (форма для вибору класу вагону та типу квитків).
* Сторінка введення особистих даних та оплати.
* Сторінка підтвердження покупки.

2. CRC-карточки:

1. Клас Passenger (пасажир):

Відповідальності: Збереження особистих даних пасажира (ім'я, прізвище, дата народження, номер паспорта), управління інформацією про квитки (дата і час купівлі, маршрут, клас, місце).

Співпрацює з: Класом TicketPurchase (купівля квитка), Класом PaymentSystem (платіжна система), Класом Ticket (квиток).

1. Клас Train (поїзд):

Відповідальності: Завантаження та відображення інформації про доступні поїзди (номер поїзда, назва, час відправлення та прибуття, кількість вільних місць).

Співпрацює з: Класом TrainSchedule (розклад поїздів), Класом TicketPurchase (купівля квитка), Класом Ticket (квиток).

1. Клас TicketPurchase (купівля квитка):

Відповідальності: Оформлення та оплата квитка, врахування різних акцій і знижок.

Співпрацює з: Класом Passenger (пасажир), Класом PaymentSystem (платіжна система), Класом Ticket (квиток).

1. Клас PaymentSystem (платіжна система):

Відповідальності: Проведення оплати за допомогою електронних платіжних систем (оплата кредитною карткою, електронним гаманцем тощо).

Співпрацює з: Класом TicketPurchase (купівля квитка).

1. Клас Ticket (квиток):

Відповідальності: Збереження інформації про куплений квиток (номер квитка, маршрут, дата і час відправлення, клас, місце).

Співпрацює з: Класом Passenger (пасажир), Класом TicketPurchase (купівля квитка).

3.Результати першого спринта:

* Розроблено головну сторінку додатку з формою пошуку маршруту та вибору дати.
* Виконано інтеграцію з базою даних поїздів, щоб показати доступні поїзди для обраного маршруту та дати.
* Розроблено функціонал вибору квитків з можливістю вибору класу вагону та типу квитків.
* Реалізовано форму для введення особистих даних та оплати квитків за допомогою електронних платіжних систем.
* Проведено тестування функціоналу купівлі квитків з різними сценаріями (наприклад, з врахуванням акцій та знижок).

Ці результати першого спринта вказують на успішне розпочаття розробки програмного забезпечення для покупки железничних квитків і виконання перших етапів проекту.

Посилання на репозиторій:

# ВИСНОВОК

В ході роботи я, розробив прототипи графічного інтерфейсу на основі користувацьких історій, визначив користувачів і написав user stories (користувацькі історії) для розроблюваного застосунку, сформував беклог продукту на основі написаних користувацьких історій.

# ЛІТЕРАТУРА

1. <http://www.javable.com/tutorials/fesunov/lesson1/>
2. <https://ru.wikibooks.org/wiki/Java/Hello_World>!
3. <http://jarlad.blogspot.com/2011/04/java.html>
4. <http://www.intuit.ru/studies/courses/569/425/lecture/9667?page=4>
5. http://www.oracle.com/technetwork/java/javase/downloads/jdk-netbeans-jsp-142931.html