Звіт

З лабораторної роботи №6

Студента групи МІТ-31

Добровольського Арсенія Михайловича

Варіант №9

**Тема роботи:** Паттерни проєктування

**Мета роботи:** розглянути поняття і призначення щаблонів проєктування. Розглянути породжуючі, структурні та поведінкові паттерни, дослідити їх призначення та особливості застосування. Навчитися застосовувати шаблони проєктування при розробці програмного забезпечення.

Завдання 1

Проаналізувавши код програми, розробленої на попередній лабораторній роботі, я дійшов таких висновків:

* Обмежена реалізація черги. Поточна реалізація черги дітей на зарахування до 1 класу обмежена принципом FIFO (першим зайшов, першим вийшов). Додавання нових типів черг, які працюють за іншими принципами (наприклад, пріоритетні черги), вимагає доволі суттєвих змін у вже існуючому коді.
* Обмежена гнучкість алгоритмів прийому. Наразі моя програма містить лише два алгоритми зарахування дітей на навчання, проте реалізація нових алгоритмів може бути складною через жорстке прив'язування до інтерфейсу.
* Управління пам'яттю. Наприклад, клас School створює об'єкти черги у конструкторі за допомогою спискового включення. Такий підхід може призвести до проблем з управлінням пам'яттю у випадку великої кількості класів, оскільки всі ці об'єкти будуть зберігатися в оперативній пам'яті одночасно.

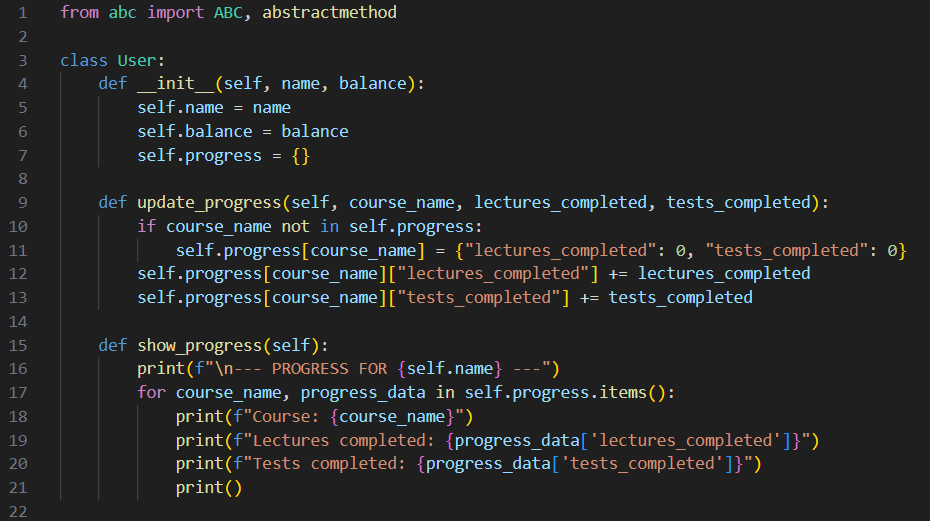
На мою думку, для покращення структуризації коду програми можна використати такі шаблони проєктування:

* Стратегія – для алгоритмів прийому. Замість прив'язки до конкретних класів алгоритмів можна визначити інтерфейс стратегії та реалізувати кілька алгоритмів, які відповідають цьому інтерфейсу.
* Фабричний метод – для створення різних типів черг. Цей паттерн дозволяє динамічно створювати об'єкти без прив'язки до конкретних класів, що полегшує розширення програми.
* Стан – для керування станами заявок. Наприклад, зміна статусу з "pending" на "approved" може бути реалізована за допомогою різних станів та методів переходу між ними.

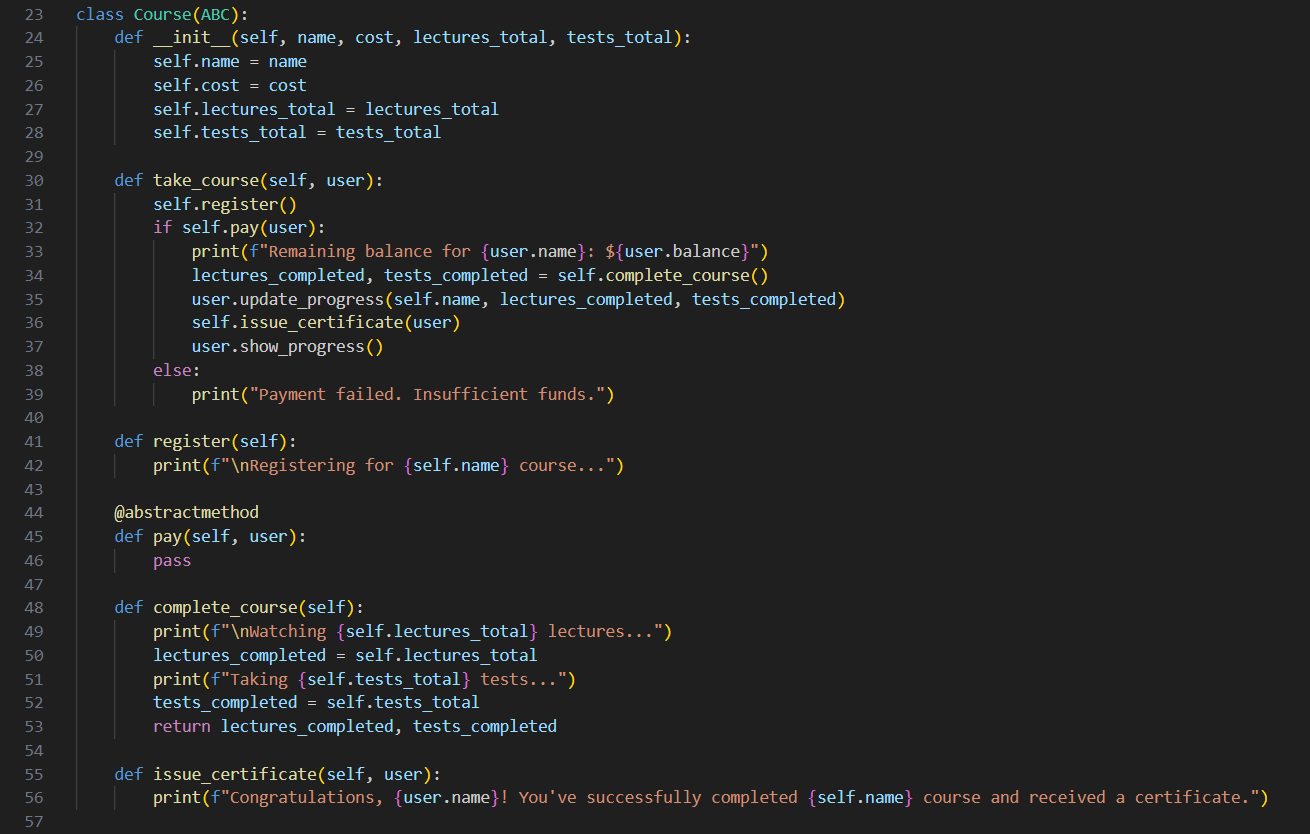
Завдання 2

За основу для виконання цього завдання я обрав тему реєстрації користувача онлайн-платформи на різні курси і оплати за них.

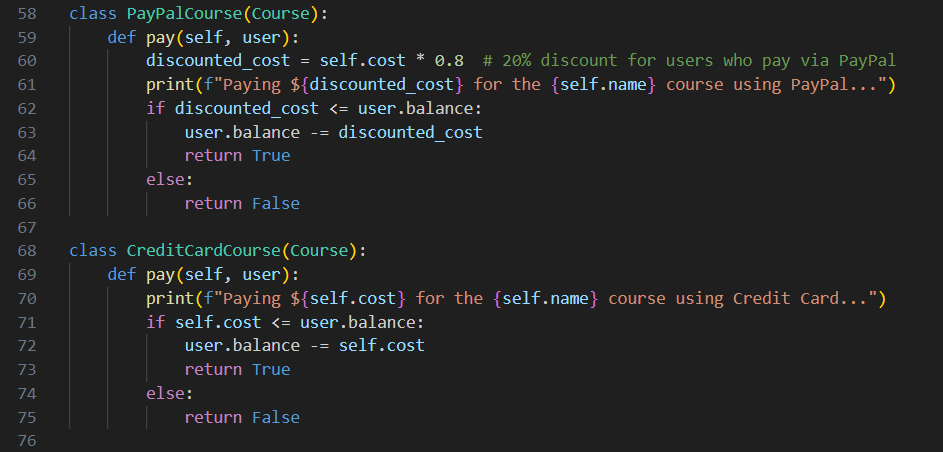
Тож спершу реалізовано клас користувача, який має ім'я, баланс акаунту та словник, що містить його прогрес з усіх курсів. Забезпечено можливості оновлення прогресу користувача та його виведення на екран:



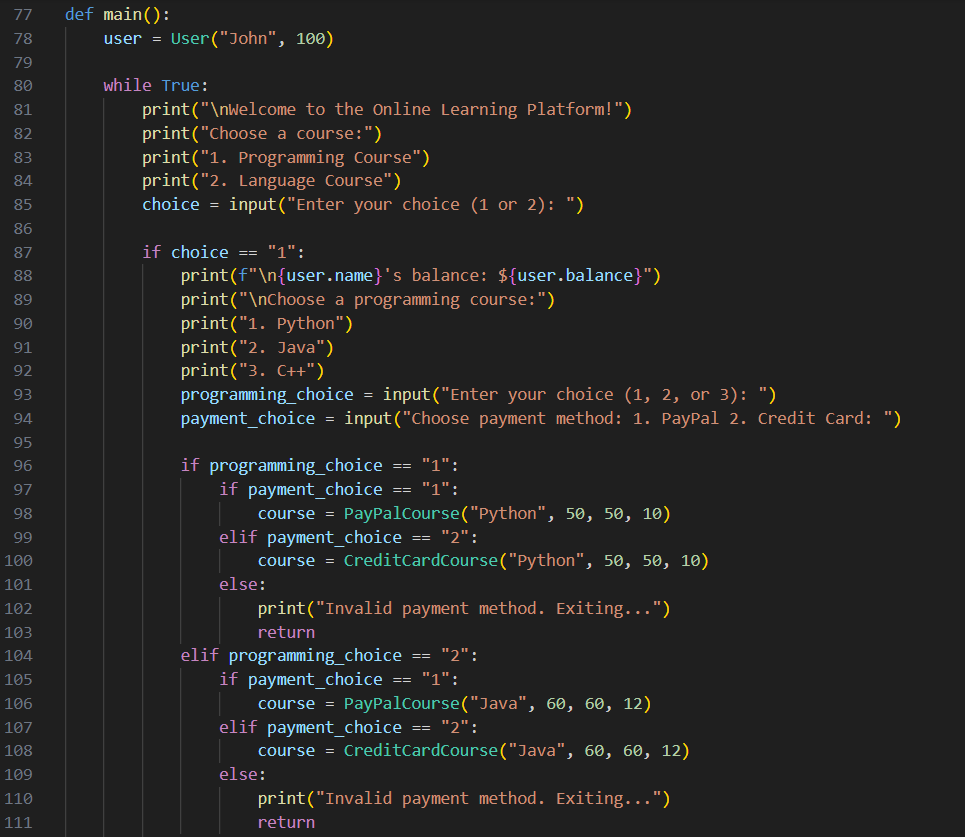
Клас Курс, що має назву, ціну, кількість лекцій і підсумкових тестів. Він містить шаблонний метод take\_course, який в свою чергу складається з методів реєстрації на курс, оплати за нього (поки абстрактний, буде реалізований у підкласах), процесу проходження курс, оновлення прогресу користувача, видача сертифікату і наостанок виведення прогресу користувача:

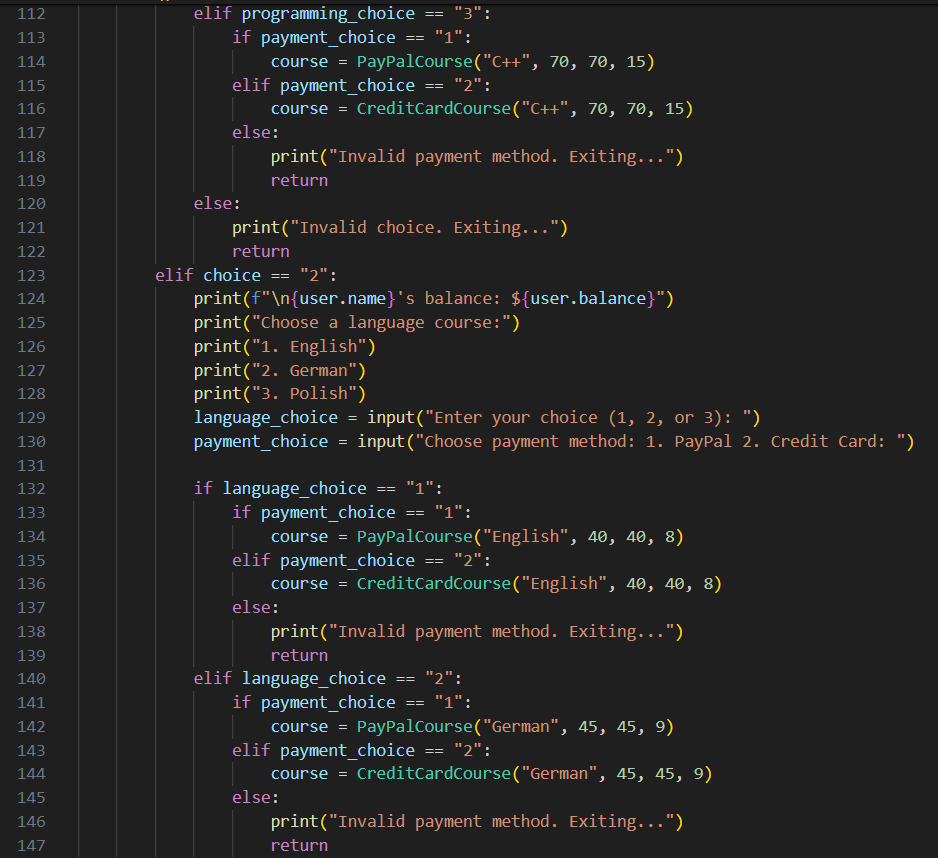


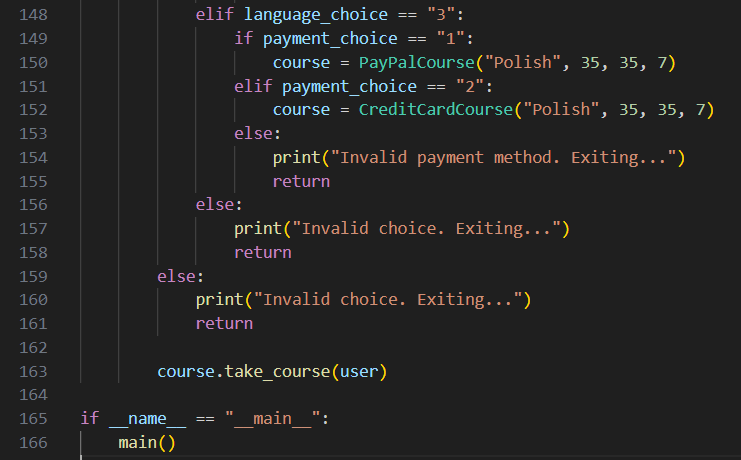
Далі метод pay реалізується у підкласах PayPalCourse та CreditCardCourse. Основна відмінність реалізації цього методу полягає у наявності знижки на курс (таку можливість мають лише користувачі, які сплачують за курс через PayPal):



Код основної програми має вигляд:

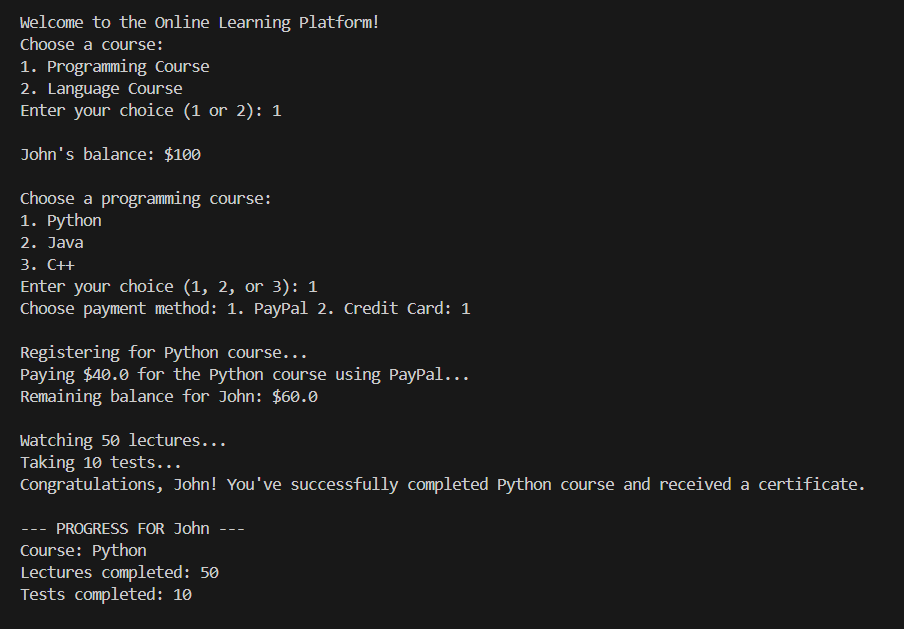




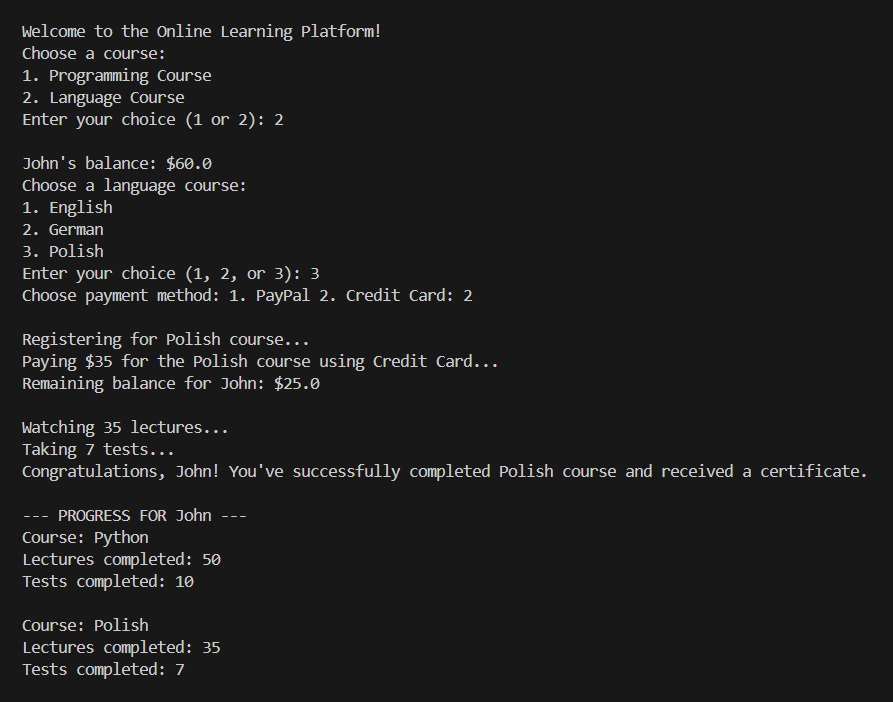


Демонстрація роботи програми:

А) За умови оплати за курс через PayPal (в даному випадку надається знижка на курс Python, тобто Джон сплатив $40 замість повної вартості у $50):

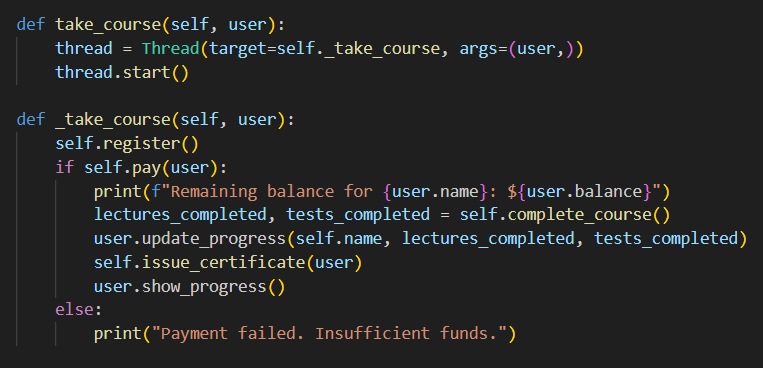


Б) За умови оплати за курс кредитною карткою (знижка вже відсутня):



В цьому випадку Джон сплатив повну вартість у $35 за курс з польської мови. Також можна помітити, що прогрес користувача оновився, і тепер Джон має два пройдених курси.

Для реалізації багатопоточної програми використано модуль threading, а саме клас Thread. Коли користувач обирає курс і спосіб оплати, програма створює новий об'єкт потоку для відповідного курсу. Деяких змін зазнав клас Course, а саме:



Такий підхід дозволяє програмі виконувати кожен курс паралельно, не чекаючи завершення проходження попереднього курсу.

**Висновок:** в ході виконання лабораторної роботи я ознайомився з різноманітними паттернами проєктування, дослідив їх призначення та особливості застосування, а також на практиці застосував паттерн Template Method для розробки власної програми.