МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

імені ІГОРЯ СІКОРСЬКОГО”

Факультет прикладної математики

Кафедра програмного забезпечення комп’ютерних систем



**Лабораторна робота №3**

з дисципліни “Об’єктно-орієнтовне програмування”

***тема:*** «Реалізація основних принципів ООП мовою C#»

|  |  |
| --- | --- |
|  | Виконав Вернік Данило  студент II курсу  групи КП-22 |

Київ 2023

Мета роботи:

Ознайомитися з шаблонами проєктування у мові програмування C#. Вирішити поставлені задачі, використовуючи їх.

Постановка задачі (Варіант №2):

1. За допомогою відповідного шаблона проєктування створити

структуру класів, до якої будуть входити класи, які реалізуються у

об’єкти «Студент» та «Група студентів». Обидва об’єкти повинні

мати імена та метод «Показати інформацію». Крім того, забезпечити

підтримку об’єктами метода «Скласти іспит».

2. В системі дистанційної освіти реалізувати механізм, який буде

дозволяти користувачу он-лайн курсу виконувати завдання з

поточного модуля (генерувати список завдань з наявного загального

масиву завдань) тільки за умови наявності виконаних завдань

(позитивних оцінок) з минулих модулів.

3. За допомогою шаблону проєктування реалізувати процес випуску

книжок за заготовленими раніше матрицями. Кожна книга

характеризується іменем автора, видавництвом, де вона була

надрукована, серією та номером видання. Масив сторінок є тим

об’єктом, який необхідно розтиражувати для кожної книги.

4. За допомогою шаблону проєктування реалізувати процес формування

штучної водойми, який складається з таких етапів як побудова

котловану, заповнення його водою, висадження рослин, запуск

водних мешканців. В залежності від типу ділянки котлован може

бути як бетонним (каркасним), так і виконаний як пластикова чаша

чи просто яма, застелена плівкою.

5. Різні люди по-різному створюють свої фінансові накопичення. Одні

люди купують валюту, інші - кладуть гроші на депозит в банку, треті

– купують акції чи облігації. Забезпечити виконання функції

«Відкласти гроші» відповідно до обраної людиною стратегії

накопичення.

6. За допомогою шаблону проєктування змоделювати роботу

викладачів, які перевіряють екзаменаційні роботи студентів з певного

предмету. Кожна робота перевіряється тільки одним викладачем.

Якщо перший викладач зайнятий перевіркою роботи, вона

передається для перевірки другому і т.д. У разі, якщо були зайняті всі

викладачі і екзаменаційна робота залишилась неперевіреною, вона

надходить на перевірку повторно до тих пір, поки її не перевірить

якийсь викладач.

Протокол має містити: титульний аркуш, постановку задачі, UML діаграму класів, фрагменти коду, які демонструють виконання поставлених задач, висновки.

**Всього –** **10 балів**

**Хід роботи:**

Визначимо, до якого шаблону стосується задача:

1. Композиція, бо у нас є клас «Група студентів», що містить список студентів. Також через те, що можна розвинути його і враховувати кураторів, тд.
2. Проксі
3. Легковик, бо ми на основі опису книги маємо багато форматів книг з різною кількістю сторінок -- Прототип
4. Будівельник, бо є конкретний результат, різні матеріали/проміжні кроки
5. Стратегія, бо одна й та сама операція містить різну реалізацію
6. Ланцюг відповідальності

**1. Задача**

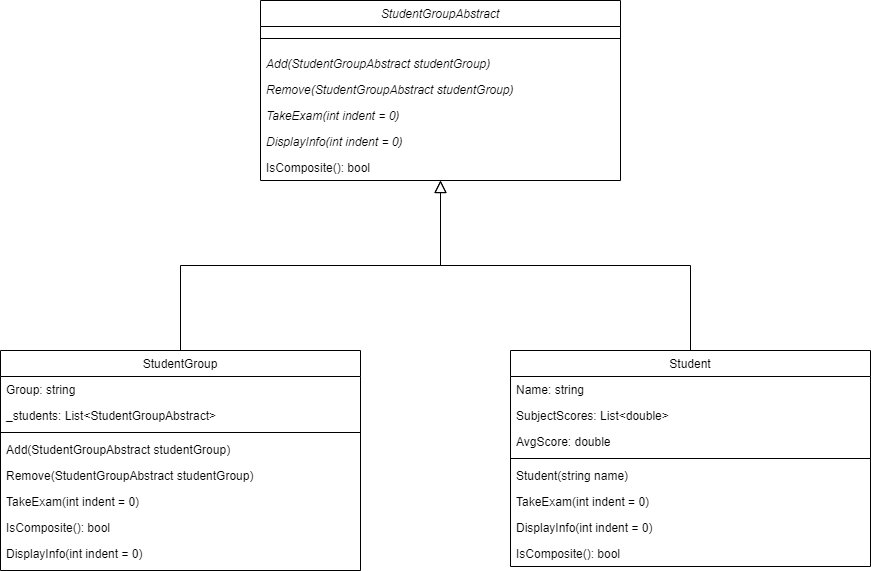
**Вибір шаблону:** Після вивчення шаблонів проєктування було вирішено використовувати шаблон Компонувальник. Його особливості дозволять створити ієрархію класів для об'єктів "Студент" та "Група студентів" та легко керувати операціями на різних рівнях.

**Особливості шаблону Компонувальник:**

* Ієрархія класів: Компонувальник дозволяє створювати ієрархію класів, де окремі об'єкти та їх групи можуть взаємодіяти як єдине ціле.
* Операції на різних рівнях: Компонувальник дозволяє виконувати операції як на окремих студентах, так і на групі студентів, при цьому операції можуть бути розширені та модифіковані без змін в інших частинах системи.
* Загальний інтерфейс: Шаблон дозволяє використовувати однаковий інтерфейс для всіх об'єктів незалежно від того, чи це окремий студент, чи група студентів.

**Застосування шаблону:** Компонувальник використовувався для створення ієрархії класів, де кожен студент може бути окремим об'єктом чи частиною групи студентів. Обидва об'єкти мають метод "Показати інформацію", а група студентів може виконувати операції, такі як "Скласти іспит", що буде автоматично застосовано до всіх студентів у групі.

**Діаграма виглядає так:**



**2. Задача**

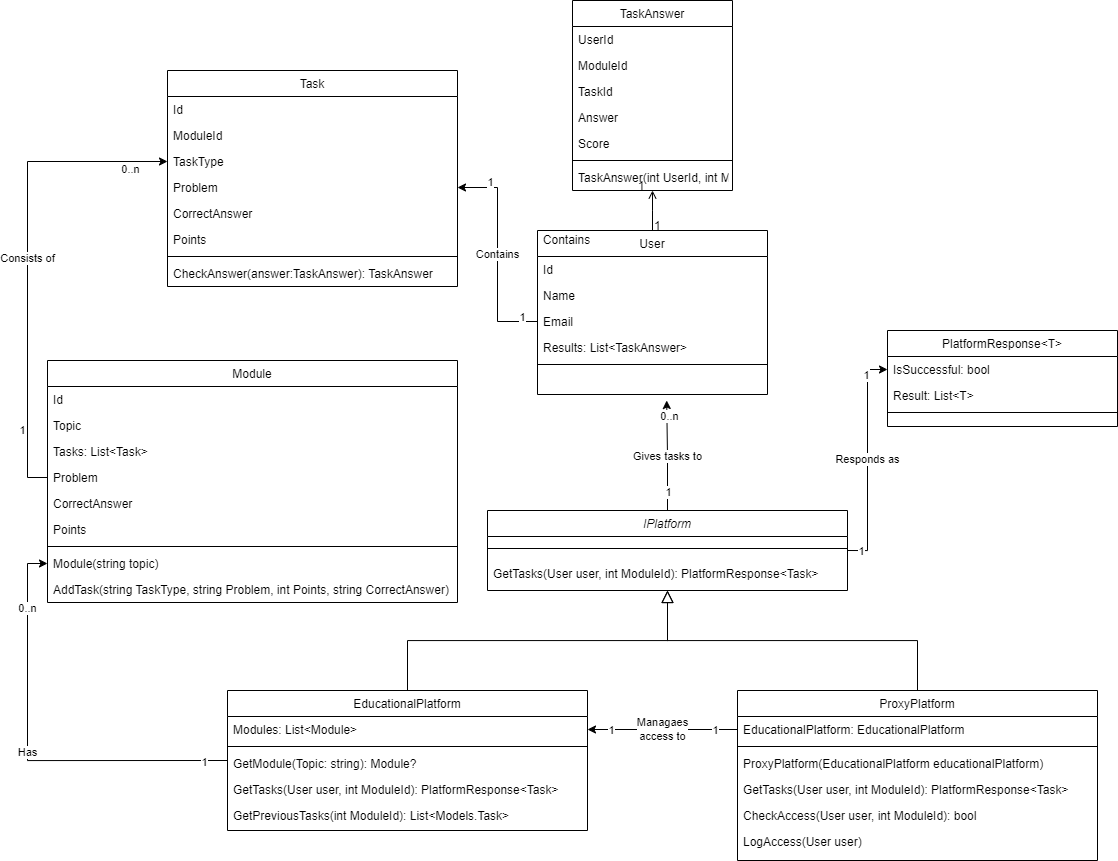
**Вибір шаблону**: Після аналізу шаблонів проєктування визначено, що шаблон Proxy є найбільш підходящим для цієї задачі. Proxy дозволяє контролювати доступ до об'єкта та реагувати на певні події, такі як перевірка наявності позитивних оцінок перед генерацією завдань.

**Особливості шаблону Proxy**:

* Контроль доступу: Proxy надає можливість контролювати доступ до об'єкта, в даному випадку - доступ до генерації списку завдань.
* Ліниве створення об'єкта: Proxy може використовувати ліниве створення об'єкта, що є важливим для оптимізації ресурсів та ефективного управління.
* Можливість виклику методів до та після обробки: Proxy надає можливість викликати методи перед та після виклику основного об'єкта, дозволяючи виконати додаткові дії, такі як перевірка умов перед генерацією завдань.

**Застосування шаблону:** Шаблон Proxy використовувався для реалізації механізму перевірки умов перед генерацією списку завдань. Проксі-об'єкт контролює доступ користувача до генерації завдань, перевіряючи наявність позитивних оцінок з попередніх модулів.

**Діаграма виглядає так:**



**3. Задача**

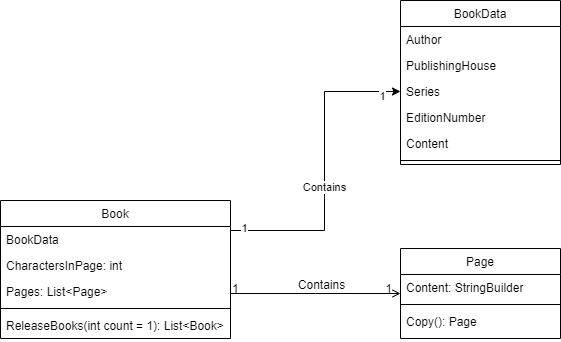
**Вибір шаблону:** Після ретельного аналізу шаблонів проєктування визначено, що шаблон Prototype є ідеальним варіантом для реалізації цієї задачі. Prototype дозволяє створювати нові об'єкти, використовуючи вже наявні екземпляри, що відповідає процесу виготовлення книжок з використанням заготовлених матриць.

**Особливості шаблону Prototype:**

* **Копіювання об'єктів:** Шаблон Prototype дозволяє створювати нові об'єкти на основі існуючих, використовуючи механізм клонування.
* **Зменшення витрат на створення об'єктів:** Prototype дозволяє ефективно використовувати існуючі об'єкти для створення нових, що зменшує витрати на створення об'єктів з нуля.
* **Можливість зміни класу об'єкта:** Prototype дозволяє змінювати клас об'єкта за допомогою клонування, що важливо для моделювання властивостей книг, що можуть варіюватися.

**Застосування шаблону:** Шаблон Prototype використовувався для створення копій об'єкта "масив сторінок", який є основною частиною кожної книги. Кожен новий об'єкт книги клонувався з вже існуючого об'єкта "масив сторінок".

**Діаграма виглядає так:**

**4. Задача**

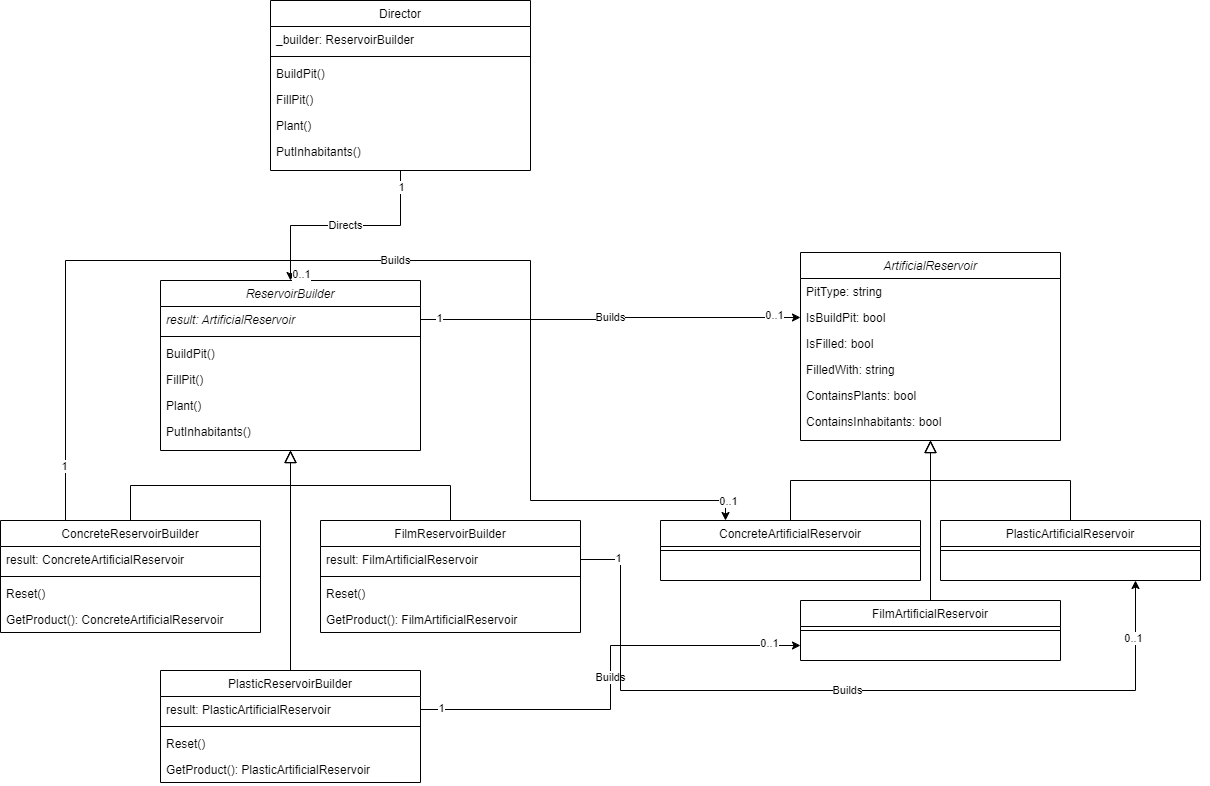
**Вибір шаблону:** Після уважного аналізу доступних шаблонів проєктування визначено, що шаблон Builder найкраще підходить для вирішення поставленої задачі. Його особливості дозволяють створити об'єкт із складовими частинами відповідно до визначених параметрів та етапів, що дозволяє легко модифікувати та розширювати процес будівництва.

**Особливості шаблону Builder:**

* **Розділення процесу будівництва**: Шаблон Builder дозволяє розділити процес формування водойми на окремі етапи (побудова котловану, заповнення водою, висадження рослин, запуск водних мешканців).
* **Гнучкість конструювання**: Builder дозволяє гнучко вибирати та комбінувати різні елементи під час конструювання, враховуючи вимоги до типу ділянки та індивідуальні вимоги.
* **Модульність**: Шаблон дозволяє використовувати різні модулі для кожного етапу, роблячи процес формування водойми більш модульним та легко змінюваним.

**Застосування шаблону**: На практиці, шаблон Builder використовувався для послідовного формування водойми за етапами. Об'єкт, що представляє водойму, побудовано з врахуванням усіх особливостей вибраного типу ділянки.

**Діаграма виглядатиме так:**



**5. Задача**

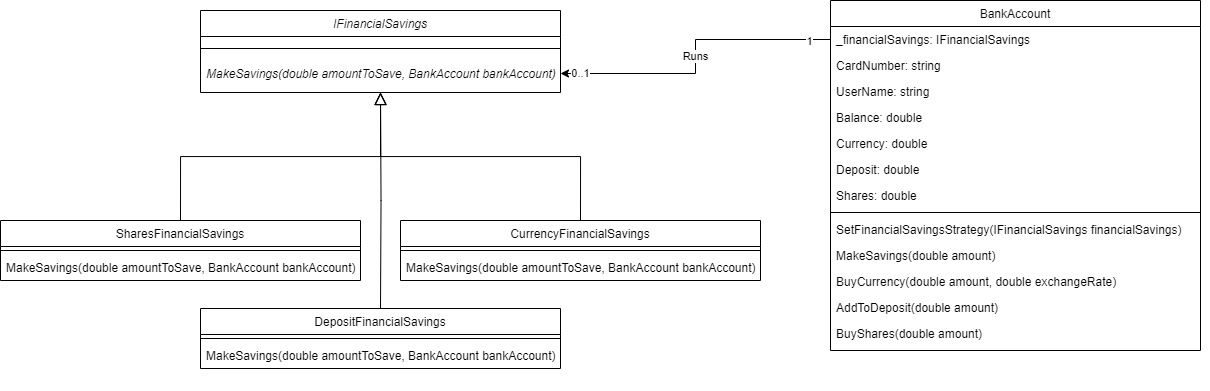
**Вибір шаблону:** Під час аналізу шаблонів проєктування визначено, що шаблон Strategy найбільш відповідає вимогам для вирішення цієї задачі. Strategy дозволяє визначити родину алгоритмів, і вибір конкретного алгоритму може відбуватися динамічно під час виконання програми, що ідеально підходить для управління різними стратегіями фінансових накопичень.

**Особливості шаблону Strategy:**

* **Визначення родини алгоритмів:** Strategy дозволяє визначити різні алгоритми для виконання конкретної функції. У даному випадку, це відкладання грошей.
* **Замінюваність алгоритмів:** Стратегії можуть бути замінені одна одною без зміни класу, що дає можливість легко переходити від одного методу накопичення до іншого, залежно від вибору користувача.
* **Гнучкість конфігурації:** Шаблон Strategy забезпечує гнучкість в конфігурації системи, дозволяючи додавати нові стратегії чи модифікувати існуючі без значних змін в основній структурі.

**Застосування шаблону:** Шаблон Strategy використовувався для реалізації функції "Відкласти гроші" враховуючи різні стратегії. Кожна стратегія представляла конкретний метод накопичення, такий як валютний обмін, депозити в банку, інвестування в акції чи облігації.

**Діаграма виглядатиме так:**



**6. Задача**

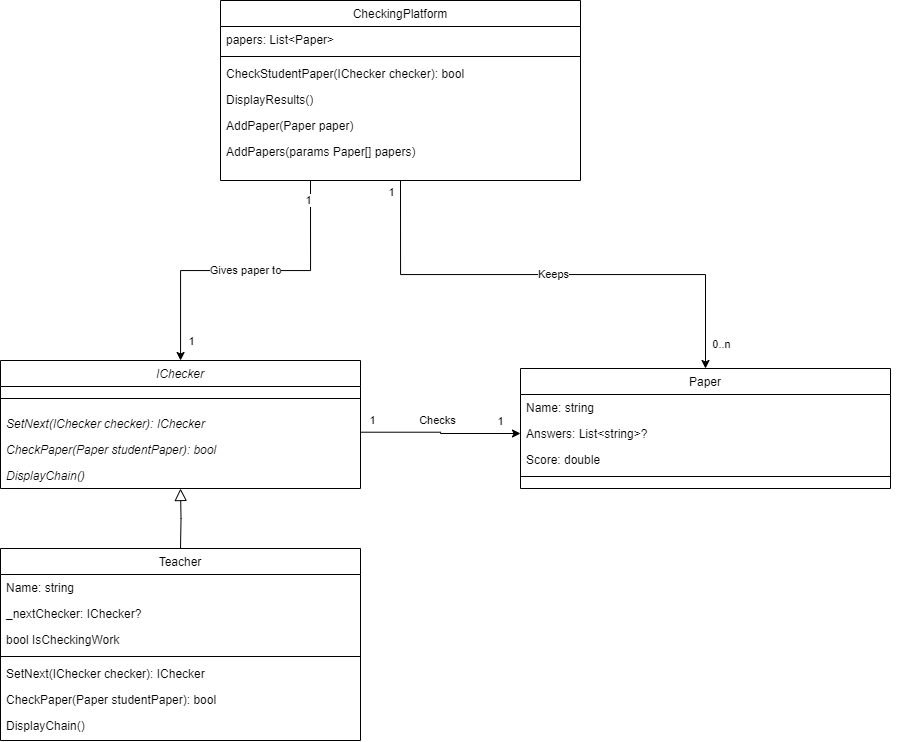
**Вибір шаблону:** В результаті аналізу шаблонів проєктування визначено, що шаблон Chain of Responsibility найкраще відповідає вимогам задачі. Його особливості ідеально підходять для передачі обробки від одного об'єкта до іншого в ланцюговому порядку.

**3. Особливості шаблону Chain of Responsibility:**

* **Ланцюжок відповідальності:** Кожен викладач утворює ланцюг, де кожен наступний викладач у ланцюгу може взяти на себе відповідальність за обробку роботи, якщо попередній викладач зайнятий.
* **Динамічна зміна ланцюгу:** Ланцюг може динамічно змінюватися в залежності від стану викладачів, їх доступності та зайнятості.
* **Зменшення зв'язності:** Кожен викладач в ланцюгу незалежний від решти, що дозволяє зменшити зв'язність між елементами системи.

**4. Застосування шаблону:** Шаблон Chain of Responsibility використовувався для моделювання процесу перевірки екзаменаційних робіт викладачами. Кожен викладач в ланцюгу мав відповідальність перевіряти роботу, і у випадку їхньої зайнятості, робота передавалася на перевірку наступному викладачу.

**Діаграма виглядатиме так:**



**Висновки:**

У даній лабораторній роботі я ознайомився з шаблонами проєктування у мові програмування C#. Вирішив поставлені задачі, використовуючи:

1. Композиція, бо у нас є клас «Група студентів», що містить список студентів. Також через те, що можна розвинути його і враховувати кураторів, тд.
2. Ланцюг відповідальності
3. Легковик, бо ми на основі опису книги маємо багато форматів книг з різною кількістю сторінок
4. Будівельник, бо є конкретний результат, різні матеріали/проміжні кроки
5. Стратегія, бо одна й та сама операція містить різну реалізацію
6. Ланцюг відповідальності