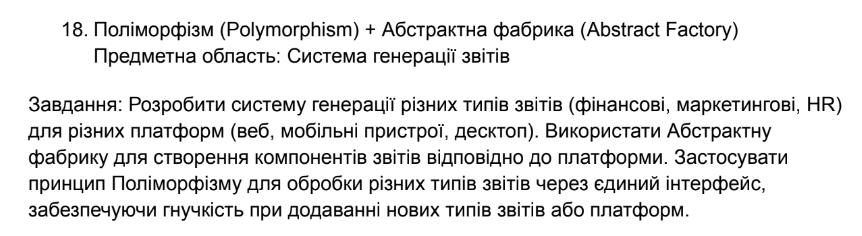
Лабораторна робота №4

Виконала: Сивоконь Валерія, КМ-33

Завдання:  


Теоретичні питання

№1

1. Яка основна мета застосування шаблонів проектування GRASP? Як вони пов'язані з принципами ООП?

Мета шаблонів GRASP це полегшити розподіл відповідальностей між класами для покращення якості коду. Вони узгоджені з принципами ООП, такими як інкапсуляція та поліморфізм, щоб забезпечити чіткі зв'язки між об'єктами.

1. Поясніть різницю між шаблонами проектування GRASP та шаблонами-творцями (Creational patterns). Наведіть приклади ситуацій, коли доцільно використовувати кожен з них.

GRASP спрямовані на загальний розподіл відповідальностей, тоді як Creational patterns допомагають створювати об’єкти. Наприклад, GRASP корисний для вибору відповідального класу, а Creational patterns, як Фабричний метод, корисні для динамічного створення об'єктів.

1. Як принцип Інформаційного експерта (Information Expert) допомагає розподілити відповідальність між класами? Наведіть приклад його застосування.

Принцип Information Expert призначає відповідальність класу, який має необхідну інформацію. Наприклад, клас Order може обчислювати загальну вартість замовлення, якщо має доступ до деталей продуктів.

1. У чому полягає суть принципу Низької зв'язаності (Low Coupling)? Як його дотримання впливає на якість та підтримку програмного коду?

Принцип Low Coupling передбачає мінімізацію залежностей між класами. Це покращує підтримуваність та адаптацію коду, оскільки зміни в одному класі менше впливають на інші.

1. Поясніть, як шаблон Одинак (Singleton) забезпечує єдину точку доступу до ресурсу. Які потенційні проблеми можуть виникнути при його використанні?

Шаблон Singleton забезпечує єдину точку доступу до об’єкта, але може створити проблеми з тестуванням і багатопотоковістю через глобальний стан.

1. Яка різниця між Абстрактною фабрикою (Abstract Factory) та Фабричним методом (Factory Method)? У яких ситуаціях краще використовувати кожен з цих шаблонів?

Abstract Factory створює набори пов’язаних об’єктів, тоді як Factory Method створює окремі об’єкти. Abstract Factory доцільний для створення цілих сімейств об'єктів, тоді як Factory Method для окремих підкласів.

1. Як принцип Високого зачеплення (High Cohesion) сприяє створенню більш модульної та зрозумілої архітектури програми?

Принцип High Cohesion сприяє розробці класів з чіткими функціями, що підвищує модульність і зрозумілість коду.

1. Поясніть, як шаблон Будівельник (Builder) допомагає у створенні складних об'єктів. Наведіть приклад ситуації, коли його використання є доцільним.

Шаблон Builder дозволяє створювати складні об'єкти покроково. Наприклад, корисний при створенні об'єктів з великою кількістю параметрів.

1. Як принцип Стійкості до змін (Protected Variations) допомагає зробити систему більш гнучкою та стійкою до майбутніх змін?

Принцип Protected Variations допомагає системі бути стійкою до змін, захищаючи частини системи від змін через інтерфейси та абстракції.

1. Поясніть, як використання шаблону Прототип (Prototype) може оптимізувати створення об'єктів у системі. Які переваги він надає порівняно з прямим створенням об'єктів через конструктор?

Шаблон Prototype забезпечує копіювання існуючого об’єкта для створення нового, що може бути швидше, ніж створення нового через конструктор, особливо якщо об'єкт складний для ініціалізації.