Лабораторна робота № 4

Звіт

**Семенченко Олександр КМ-33**

1. Яка основна мета застосування шаблонів проектування GRASP? Як вони

Пов’язані з принципами ООП? Їхня основна мета – забезпечити зрозумілість, гнучкість та легкість підтримки програмного коду шляхом раціонального розподілу завдань між об'єктами. Інкапсуляція: GRASP допомагає приховати внутрішню реалізацію класів і забезпечувати доступ до даних через чітко визначений інтерфейс.

Спадкування: шаблони GRASP можуть використовуватися для створення ієрархій класів і реалізації поліморфізму.

Поліморфізм: GRASP активно використовує поліморфізм для створення гнучких і розширюваних систем.

Абстракція: шаблони GRASP допомагають фокусуватися на сутності проблеми, відволікаючись від деталей реалізації.

2. Поясніть різницю між шаблонами проектування GRASP та шаблонами-творцями

(Creational patterns). Наведіть приклади ситуацій, коли доцільно

використовувати кожен з них. GRASP фокусується на розподілі відповідальності між класами. Він допомагає визначити, який клас повинен виконувати яку дію, щоб система була більш зрозумілою і легко підтримувалася. Шаблони-творці ж займаються створенням об'єктів. Вони дозволяють абстрагуватися від деталей створення об'єктів, роблячи код більш гнучким і керованим. На початкових етапах проектування, коли ви визначаєте структуру класів і їх взаємодію. Коли процес створення об'єкта є складним або залежить від зовнішніх факторів.

3. Як принцип Інформаційного експерта (Information Expert) допомагає розподілити

відповідальність між класами? Наведіть приклад його застосування. Згідно з цим принципом, відповідальність за виконання дії повинна бути делегована тому класу, який має найбільше інформації, необхідної для виконання цієї дії. Уявімо систему управління бібліотекою. Ми хочемо визначити, який клас повинен обчислювати загальну кількість сторінок у книзі.

Для обчислення кількості сторінок нам потрібна інформація про кількість сторінок в кожному розділі книги. Клас "Розділ" має інформацію про кількість сторінок у собі. Отже, клас "Розділ" повинен мати метод для обчислення кількості сторінок, а клас "Книга" може викликати цей метод для всіх своїх розділів, щоб отримати загальну кількість сторінок.

4. У чому полягає суть принципу Низької зв’язаності (Low Coupling)? Як його

дотримання впливає на якість та підтримку програмного коду? Принцип низької зв'язаності в програмуванні означає, що різні частини програмного коду мають мінімальну залежність одна від одної. Іншими словами, зміни в одній частині коду повинні мінімально впливати на інші частини. Він допомагає створювати більш гнучкі, підтримувані і легкі для розуміння системи.

5. Поясніть, як шаблон Одинак (Singleton) забезпечує єдину точку доступу до

ресурсу. Які потенційні проблеми можуть виникнути при його використанні? Клас має приватний конструктор, щоб запобігти створенню екземплярів ззовні. Клас надає статичний метод, який повертає єдиний екземпляр класу. Якщо екземпляр ще не створено, він створюється всередині цього методу і зберігається в статичному полі класу. Ускладнення тестування, порушення принципу відкритості/закритості, можливі проблеми з паралелізмом, залежність від глобального стану

6. Яка різниця між Абстрактною фабрикою (Abstract Factory) та Фабричним

методом (Factory Method)? У яких ситуаціях краще використовувати кожен з цих

шаблонів? Абстрактна фабрика створює сімейства, а фабричний метод – окремі об'єкти. Абстрактна фабрика: для складних систем з різними конфігураціями, коли потрібно створювати сімейства об'єктів. Фабричний метод: для більш простих систем, коли потрібно створювати різні варіанти одного типу об'єктів.

7. Як принцип Високого зачеплення (High Cohesion) сприяє створенню більш

модульної та зрозумілої архітектури програми? Він сприяє створенню більш модульної та зрозумілої архітектури програми через такі характеристики: зрозумілість, легкість підтримки, повторне використання, легше тестування.

8. Поясніть, як шаблон Будівельник (Builder) допомагає у створенні складних

Об’єктів. Наведіть приклад ситуації, коли його використання є доцільним. Шаблон Будівельник дозволяє ізолювати код, що відповідає за створення об'єкта, від того, як об'єкт виглядає або яку саме інформацію має. Це дозволяє розробнику змінювати спосіб побудови об'єкта, не змінюючи його структуру. Будівельник може містити кілька етапів побудови об'єкта. Завдяки шаблону можна створювати різні варіанти одного і того ж об'єкта, змінюючи лише його складові частини або порядок їх створення. Шаблон є доцільним для комплексних об’єктів, різних варіантів об’єкта та вибіркового налаштування.

9. Як принцип Стійкості до змін (Protected Variations) допомагає зробити систему

більш гнучкою та стійкою до майбутніх змін? Зміни в одній частині (наприклад, платіжному методі) не вимагають змін в інших частинах системи. Легко додавати нові функції або заміняти старі, не змінюючи основний код. Зміни обмежені окремими модулями, що робить систему більш зрозумілою та легкою для підтримки.

10. 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? Копіювання об'єкта зазвичай швидше, ніж його створення через конструктор, оскільки не потрібно повторно ініціалізувати всі властивості. Клонування об'єкта дозволяє легко створювати варіанти об'єктів без необхідності вручну налаштовувати всі параметри.

Переваги порівняно з конструктором: економія ресурсів та легкість у модифікації