

Дисципліна: «Розробка програмного забезпечення з використанням патернів проектування»

Поняття та види патернів проектування

Баран С.В.

Проектування об'єктно-орієнтованих програм – складна справа. Необхідно підібрати відповідні об'єкти, віднести їх до різних класів, дотримуючись розумного ступіню деталізації, визначити інтерфейси класів і ієрархію наслідування і встановити суттєві відносини між класами. Дизайн повинен, з одного боку, відповідати задачі, з іншого - бути загальним, щоб вдалося врахувати всі вимоги, які можуть виникнути в майбутньому.

Перш за все, досвідчений розробник намагається повторно скористатися тими рішеннями, які виявилися вдалими в минулому. У багатьох об'єктно-орієнтованих системах зустрічаються патерни, що повторно використовуються і складаються з класів і об'єктів, які взаємодіють. З їх допомогою вирішуються конкретні завдання проектування, в результаті чого об'єктно-орієнтований дизайн стає більш гнучким, елегантним, і їм можна скористатися повторно.

Патерни проектування спрощують повторне використання вдалих проектних і архітектурних рішень. За допомогою патернів можна поліпшити якість документації і супроводу існуючих систем, дозволяючи явно описати взаємодії класів і об'єктів, а також причини, за якими система була побудована так, а не інакше.

Мабуть, створення нових об'єктів є найбільш поширеним завданням, що встає перед розробниками програмних систем. Патерни проектування, що породжують об'єкти, призначені для створення об'єктів, дозволяючи системі залишатися незалежною як від самого процесу породження, так і від типів об'єктів, що створюються.

У структурних патернах розглядається питання про те, як із класів та об'єктів утворюються більші структури. Структурні патерни рівня класу використовують успадкування для складання композицій з інтерфейсів та реалізацій. Замість композиції інтерфейсів чи реалізацій структурні патерни рівня об'єкта компонують об'єкти для отримання нової функціональності. Додаткова гнучкість у разі пов'язані з можливістю змінити композицію об'єктів під час виконання, що неприпустимо для статичної композиції класів.

Паттерни поведінки пов'язані з алгоритмами та розподілом обов'язків між об'єктами. Річ в них йде не лише про самі об'єкти та класи, а й про типові способи взаємодії. Патерни поведінки

характеризують помилковий потік управління, який важко простежити під час виконання програми. Увага акцентована не на потоці управління як такому, а на зв'язках між об'єктами.