

Введение в паттерны ООП

Пахаев Х.Х.

Немного истории

Впервые, в конце 1970-х годов Кристофером Александром был разработан каталог паттернов, предназначенных для проектирования зданий и городов. В конце 1980-х годов Кент Бек и Вард Каннингем попытались перенести идеи Александра в область разработки ПО, составив 5 небольших паттернов для проектирования пользовательских интерфейсов на языке Smalltalk. В 1989 Джеймс Коплиен в целях обучения C++ внутри компании AT&T составил каталог идиом C++ (разновидность паттернов, специфичных для языка программирования), а в 1991 на его основе вышла в свет книга "Advanced C++ Programming Styles and Idioms" (Продвинутые стили и идиомы программирования на C++).

Однако по настоящему популярным применение паттернов в индустрии разработки программного обеспечения стало после того, как в 1994 был опубликован каталог, включающий 23 паттерна объектно-ориентированного проектирования. Этот каталог настолько популярен, что часто упоминается как паттерны GoF ("Gang of Four" или "банда четырех" по числу авторов).

Что такое Паттерн?

- Паттерн проектирования — это часто встречающееся решение определённой проблемы при проектировании архитектуры программ.
- В отличие от готовых функций или библиотек, паттерн нельзя просто взять и скопировать в программу. Паттерн представляет собой не какой-то конкретный код, а общую концепцию решения той или иной проблемы, которую нужно будет ещё подстроить под нужды вашей программы.
- Паттерны часто путают с алгоритмами, ведь оба понятия описывают типовые решения каких-то известных проблем. Но если алгоритм — это чёткий набор действий, то паттерн — это высокоуровневое описание решения, реализация которого может отличаться в двух разных программах.
- Если привести аналогии, то алгоритм — это кулинарный рецепт с чёткими шагами, а паттерн — инженерный чертёж, на котором нарисовано решение, но не конкретные шаги его реализации.

Из чего состоит паттерн?

Описания паттернов обычно очень формальны и чаще всего состоят из таких пунктов:

- проблема, которую решает паттерн;
- мотивации к решению проблемы способом, который предлагает паттерн;
- структуры классов, составляющих решение;
- примера на одном из языков программирования;
- особенностей реализации в различных контекстах;
- связей с другими паттернами.

Основы паттернов проектирования в объектно-ориентированном программировании (ООП)

- Паттерны ООП представляют собой проверенные шаблоны решений для общих проблем в разработке программного обеспечения;
- ООП основано на принципах инкапсуляции, наследования и полиморфизма, и паттерны служат эффективным средством повышения гибкости и повторного использования кода;
- Задача каждого паттерна - дать четкое описание проблемы и ее решения в соответствующей области.

Зачем использовать паттерны?

- Паттерны предоставляют архитектурные решения, способствующие лучшему пониманию и обеспечивающие структурирование кода.
- Облегчают поддержку, модификацию и масштабирование кода.
- Повышают читаемость и ясность программы, делая ее более поддерживаемой для разработчиков.

Типы паттернов

- Порождающие паттерны: Отвечают за механизм создания объектов.
- Структурные паттерны: Занимаются композицией классов и объектов.
- Поведенческие паттерны: Управляют взаимодействием между объектами и ответственны за распределение обязанностей.

Паттерны

Структурные паттерны проектирования

- Адаптер;
- Мост;
- Компоновщик;
- Декоратор;
- Фасад;
- Легковес;
- Заместитель.

Порождающие паттерны проектирования

- Фабричный метод;
- Абстрактная фабрика;
- Строитель;
- Прототип;
- Одиночка.

Поведенческие паттерны проектирования

- Команда;
- Итератор;
- Посредник;
- Снимок;
- Наблюдатель;
- Состояние;
- Стратегия.