Паттерны проектирования

Паттерны проектирования (шаблоны, образцы) - описание подхода к решению проблемы, который можно применять в разных ситуациях.

Паттерны проектирования - это многократно применяемое решение регулярно возникающей проблемы в рамках определённого контекста архитектуры приложения.

Паттерны проектирования – это комбинация основных инструментов: наследования, интерфейсов, вложенности, универсальности.

При написании программ мы можем формализовать проблему в виде классов и объектов и связей между ними. И применить один из существующих паттернов для ее решения. В итоге нам не надо ничего придумывать. У нас уже есть готовый шаблон, и нам только надо его применить в конкретной программе.

Существует множество различных паттернов, которые решают разные проблемы и выполняют различные задачи. Но по своему действию их можно объединить в ряд групп.

**1. Порождающие паттерны**

Порождающие паттерны — это паттерны, которые абстрагируют процесс инстанцирования или, иными словами, процесс порождения классов и объектов. Среди них выделяются следующие:

Абстрактная фабрика (Abstract Factory)

Строитель (Builder)

Фабричный метод (Factory Method)

Прототип (Prototype)

Одиночка (Singleton)

**2. Структурные паттерны**

Структурные паттерны – это группа паттернов рассматривающая, как классы и объекты образуют более крупные структуры - более сложные по характеру классы и объекты. К таким шаблонам относятся:

Адаптер (Adapter)

Мост (Bridge)

Компоновщик (Composite)

Декоратор (Decorator)

Фасад (Facade)

Приспособленец (Flyweight)

Заместитель (Proxy)

**3. Поведенческие паттерны**

Поведенческие паттерны - они определяют алгоритмы и взаимодействие между классами и объектами, то есть их поведение. Среди подобных шаблонов можно выделить следующие:

Цепочка обязанностей (Chain of responsibility)

Команда (Command)

Интерпретатор (Interpreter)

Итератор (Iterator)

Посредник (Mediator)

Хранитель (Memento)

Наблюдатель (Observer)

Состояние (State)

Стратегия (Strategy)

Шаблонный метод (Template method)

Посетитель (Visitor)

Существуют и другие классификации паттернов.

**Как выбрать нужный паттерн?**

Прежде всего при решении какой-нибудь проблемы надо выделить все используемые сущности и связи между ними и абстрагировать их от конкретной ситуации. Затем надо посмотреть, вписывается ли абстрактная форма решения задачи в определенный паттерн. Например, суть решаемой задачи может состоять в создании новых объектов. В этом случае, возможно, стоит посмотреть на порождающие паттерны. Причем лучше не сразу взять какой-то определенный паттерн - первый, который показался нужным, а посмотреть на несколько родственных паттернов из одной группы, которые решают одну и ту же задачу.

При этом важно понимать смысл и назначение паттерна, явно представлять его абстрактную организацию и его возможные конкретные реализации. Один паттерн может иметь различные реализации, и чем чаще вы будете сталкиваться с этими реализациями, тем лучше вы будете понимать смысл паттерна. Но не стоит использовать паттерн, если вы его не понимаете, даже если он на первый взгляд поможет вам в решении задачи.