Патерни GoF.

**Патерн проектування** — це типовий спосіб вирішення певної проблеми, що часто зустрічається при проектуванні архітектури програм.

На відміну від готових функцій чи бібліотек, патерн не можна просто взяти й скопіювати в програму. Патерн являє собою не якийсь конкретний код, а загальний принцип вирішення певної проблеми, який майже завжди треба підлаштовувати для потреб тієї чи іншої програми.

Патерни часто плутають з алгоритмами, адже обидва поняття описують типові рішення відомих проблем. Але якщо алгоритм — це чіткий набір дій, то патерн — це високорівневий опис рішення, реалізація якого може відрізнятися у двох різних програмах.

Якщо провести аналогії, то алгоритм — це кулінарний рецепт з чіткими кроками, а патерн — інженерне креслення, на якому намальовано рішення без конкретних кроків його отримання.

Описи патернів зазвичай дуже формальні й найчастіше складаються з таких пунктів:

* проблема, яку вирішує патерн;
* мотивація щодо вирішення проблеми способом, який пропонує патерн;
* структура класів, складових рішення;
* приклад однією з мов програмування;
* особливості реалізації в різних контекстах;
* зв’язки з іншими патернами.

=======================================================

Шаблон (Template).

**Шаблонний метод** — це поведінковий патерн проектування, який визначає кістяк алгоритму, перекладаючи відповідальність за деякі його кроки на підкласи. Патерн дозволяє підкласам перевизначати кроки алгоритму, не змінюючи його загальної структури.

Аналог із життя

Під час будівництва типових будинків будівельники використовують підхід, схожий на шаблонний метод. У них є основний архітектурний проект, в якому розписані кроки будівництва: заливка фундаменту, витягування стін, покриття даху, встановлення вікон тощо.

Але, незважаючи на стандартизацію кожного етапу, будівельники можуть робити невеликі зміни на кожному з етапів, щоб зробити будинок трішечки не схожим на інші.