

**GoF патерни** — це шаблони проектування, які забезпечують рішення загальних проблем проектування програмного забезпечення. У разі об'єктно-орієнтованого програмування шаблони проектування зазвичай націлені на вирішення проблем взаємодії об'єктів, а не на більш масштабні проблеми загальної архітектури програмного забезпечення. Вони дають узагальнені рішення у формі шаблонів, які можуть бути застосовані до реальних проблем

В колекції є 23 шаблони, які поділяються на поведінкові(забезпечують кращу взаємодію між об'єктами), структурні(стосуються структури класів, таких як успадкування та композиція), створюючі(стосуються створення об'єкта).

**Абстрактна фабрика** — це породжувальний патерн проектування, що дає змогу створювати сімейства пов'язаних об'єктів, не прив'язуючись до конкретних класів створюваних об'єктів.

## Структура

1. Абстрактні продукти оголошують інтерфейси продуктів, що пов'язані один з одним за змістом, але виконують різні функції.
2. Конкретні продукти - великий набір класів, що належить до різних абстрактних продуктів, але мають одні й ті самі варіації.
3. Абстрактна фабрика оголошує методи створення різних абстрактних продуктів.
4. Конкретні фабрики належить до своєї варіації продукту і реалізує метод абстрактної фабрики, даючи змогу створювати всі продукти певної варіації.
5. Конкретні фабрики породжують конкретні продукти. Це дозволить коду, що використовує фабрику не прив'язуватись до конкретних класів продуктів.

## Переваги та недоліки

### Переваги:

1. Гарантує поєднання створюваних продуктів.
2. Звільняє код від прив'язки до конкретних класів.
3. Виділяє код виробництва продуктів в одне місце, спрощуючи підтримку коду.
4. Спрощує додавання нових продуктів до програми.
5. Реалізує принцип відкритості/закритості.

### Недоліки:

1. Ускладнює код програми внаслідок введення великої кількості додаткових класів.
2. Вимагає наявності всіх типів продукту в кожній варіації.

### **Застосування**

1. Коли програма повинна працювати з різними видами пов'язаних один з одним продуктів, незалежно від конкретних класів продуктів.
2. Клієнтський код може працювати з усіма типами створюваних продуктів.
3. Коли в програмі вже використовується Фабричний метод, але зміни передбачають введення нових типів.