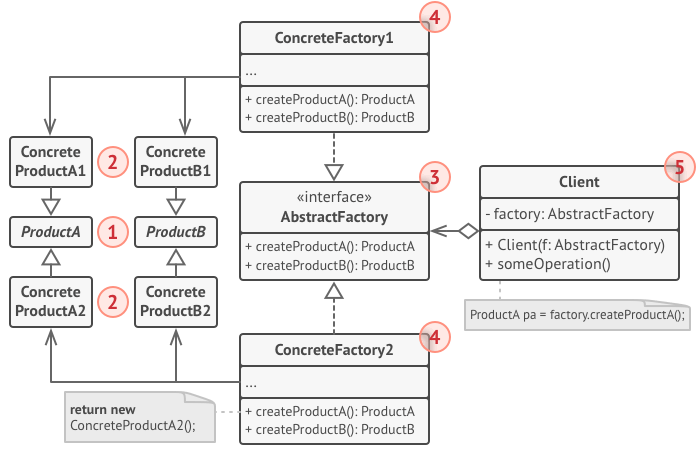
**АБСТРАКТНАЯ ФАБРИКА -** предоставляет интерфейс создания семейств взаимосвязанных продуктов, не привязываясь к конкретным классам создаваемых продуктов. Содержит несколько методов-создателей, каждый из которых создает определенный продукт для данного семейства. По сути это набор абстрактных Фабричных методов. Какое конкретно семейство продуктов будет создаваться методами - определяется реализацией абстрактной фабрики - т.о. образом создаются фабрики, производящие продукты для разных контекстов: разных регионов, разных операционных систем и т.д. Отделение кода позволяет динамически переключать фабрики для изменения поведения. У каждого из продуктов свой интерфейс, и каждый метод возвращает продукт как абстракцию. Основана на композиции: создание продуктов реализуется в методах фабрики, доступ к которым осуществляется через ее интерфейс

ПРИМЕНЕНИЕ:

- когда бизнес-логика программы должна работать с разными видами связанных друг с другом продуктов, не завися от конкретных классов продуктов

- когда в программе уже используется Фабричный метод, но очередные изменения предполагают введение новых типов продуктов

РЕАЛИЗАЦИЯ:

**1. АБСТРАКТНЫЕ ПРОДУКТЫ**: объявляют интерфейсы продуктов, которые связаны друг с другом по смыслу, но выполняют разные функции. Например, "соус", "овощ", "сыр".

**2. КОНКРЕТНЫЕ ПРОДУКТЫ**: большой набор классов, которые относятся к различным абстрактным продуктам, но имеют одни и те же вариации. Например, "соус майонез", "соус кетчуп", "овощ ананас", "сыр диетический", "сыр моцарелла".

**3. АБСТРАКТНАЯ ФАБРИКА**: объявляет методы создания различных абстрактных продуктов. Содержит набор методов-создателей, каждый возвращает свою абстракцию продукта. Например, "фабрика ингридиентов пиццы".

**4. КОНКРЕТНЫЕ ФАБРИКИ**: относятся каждая к своей вариации продуктов и реализуют методы абстрактной фабрики, позволяя создавать все продукты определённой вариации. Методы возвращают свои версии продуктов, которая должна производить данная конфигурация фабрики. Например, "нью-йоркская фабрика ингридиентов пиццы".

**5. КОНКРЕНТНЫЙ КЛИЕНТ**: обращается к абстрактной фабрике, чтобы получить продукты. Получает в конструктор ссылку на конкретную фабрику, с которой будет работать. Работает с ее методами для получения конфигурации продуктов. Например, "пиццерия".

ПЛЮСЫ:

- гарантирует сочетаемость создаваемых продуктов

- избавляет клиентский код от привязки к конкретным классам продуктов

- выделяет код производства продуктов в одно место, упрощая поддержку кода

- упрощает добавление новых продуктов в программу

- реализует принцип открытости/закрытости

МИНУСЫ:

- усложняет код программы из-за введения множества дополнительных классов

- требует наличия всех типов продуктов в каждой вариации