

Дисципліна: «Розробка програмного забезпечення з використанням патернів проектування»

## **Патерн проектування «Абстрактна фабрика»**

**Баран С.В.**

Для патернів, що породжують об'єкти, актуальні дві теми. По-перше, ці патерни інкапсулюють знання про конкретні класи, які застосовуються в системі. По-друге, приховують деталі того, як ці класи створюються і взаємодіють. Єдина інформація про об'єкти, відома системі, - це їх інтерфейси, визначені за допомогою абстрактних класів. Отже, патерни, що породжують об'єкти, забезпечують більшу гнучкість при вирішенні питання про те, що створюється, хто це створює, як і коли. Можна зібрати систему з «готових» об'єктів з самої різною структурою і функціональністю статично (на етапі компіляції) або динамічно (під час виконання).

Призначення патерну «Абстрактна фабрика» (Abstract Factory) в тому, що він надає інтерфейс для створення сімейств взаємопов'язаних або взаємозалежних об'єктів не специфікуючи їх конкретних класів.

Цей патерн використовується коли:

система не повинна залежати від того, як створюються, компонуються і представляються об'єкти, що входять до неї;

взаємопов'язані об'єкти, що входять в сімейство, повинні використовуватися разом і вам необхідно забезпечити виконання цього обмеження;

система повинна конфігуруватися одним з сімейств об'єктів, з яких вона складається; необхідно створювати групи або сімейства взаємопов'язаних об'єктів, виключаючи можливість одночасного використання різнорідних об'єктів в одному контексті;

необхідно надати бібліотеку об'єктів, розкриваючи тільки їх інтерфейси, але не реалізацію.

Патерн володіє наступними плюсами і мінусами:

ізолює конкретні класи. Допомагає контролювати класи об'єктів, що створюються додатком. Оскільки фабрика інкапсулює відповідальність за створення класів і сам процес їх створення, то вона ізолює клієнта від деталей реалізації класів. Клієнти маніпулюють екземплярами через їх абстрактні інтерфейси. Імена класів відомі тільки конкретній фабриці, в коді клієнта вони не згадуються;

спрощує заміну сімейств продуктів. Клас конкретної фабрики з'являється в додатку тільки один раз: при інстанціюванні. Це полегшує заміну додатком конкретної фабрики. Додаток може змінити конфігурацію продуктів, просто підставивши нову конкретну фабрику. Оскільки абстрактна фабрика створює все сімейство продуктів, то і замінюється відразу все сімейство;

гарантує сполучуваність продуктів. Якщо продукти деякого сімейства спроектовані для спільного використання, то важливо, щоб додаток в кожен момент часу працював тільки з продуктами єдиного сімейства;

недоліком є те, що підтримувати новий вид продуктів важко. Розширення абстрактної фабрики для виготовлення нових видів продуктів – складний процес.