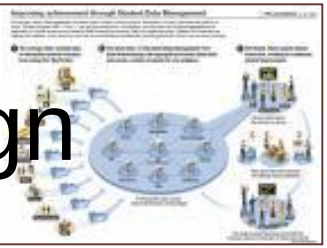
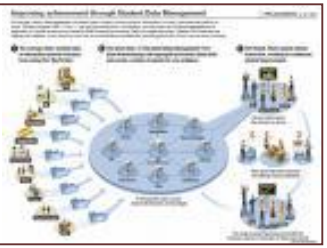
Systems Analysis and Design



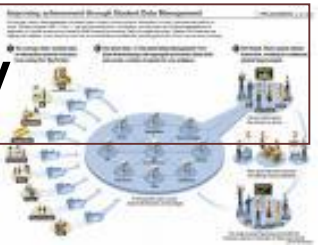
Шаблони проектування

План



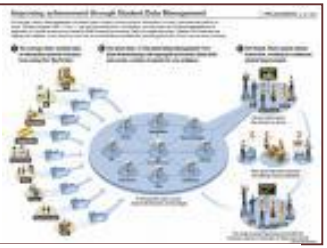
- Повторне використання коду
- Паттерни проектування
- Породжуючі паттерни
  - Singleton
  - Factory Method
- Структурні паттерни
  - Adapter
  - Decorator
  - Proxy
- Паттерни поведінки
  - Iterator
  - Observer

Повторне використання коду



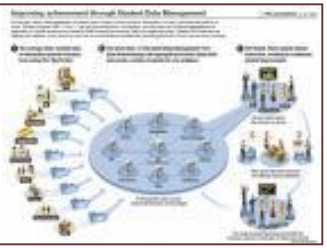
- Для додатків, які і снують
- Для нових додатків
- Принципи створення повторного використання коду
  - Модульність (modularity)
  - Слабка звязність (low coupling)
  - Висока сфокусованість (high cohesion)
  - Приховування інформації (information hiding)
  - Розподіл відповідальності (separation of concerns)
- Приклади повторного використання
  - «Копіпаста» (copy-and-paste)
  - Бібліотеки (software libraries)
  - Паттерни проектування (design patterns)
  - Фреймворки (software frameworks)

Паттерни проектування



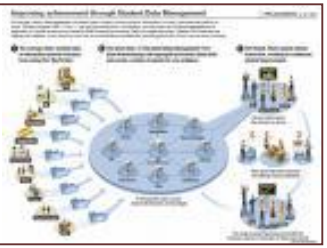
- Патерн проектування (Design Pattern, зразок проектування, шаблон проектування) - опис взаємодії об'єктів і класів, адаптованих для вирішення загальної задачі проектування в конкретному контексті
- Патерн проектування іменує, абстрагує і ідентифікує ключові аспекти структури спільного рішення, які і дозволяють застосувати його для створення повторно використовуваного проектного рішення

Паттерни в ООП



- Результат проектування на рівні ООП - розподіл відповідальностей і активностей по класах
- Паттерн - іменована конфігурація розподілу відповідальності за класами

Описи паттернів

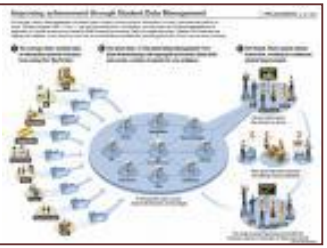


- **GoF**
  Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides (1995). Design Patterns: Elements of Reusable Object-Oriented Software
- **POSA1**
  Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad (1996). Pattern-Oriented Software Architecture, Volume 1: A System of Patterns
- **POSA2**
  Douglas Schmidt, Michael Stal, Hans Rohnert, Frank Buschmann (2000). Pattern-Oriented Software Architecture, Volume 2: Patterns for Concurrent and Networked Objects
- **PoEAA**
  Martin Fowler (2002). Patterns of Enterprise Application Architecture
- …

Породжуючі паттерни

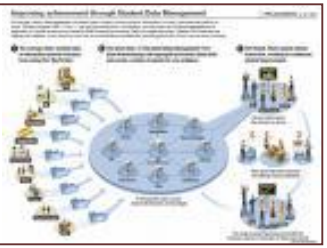
- Abstract Factory
- Builder
- Factory Method
- Prototype
- Singleton

Singleton

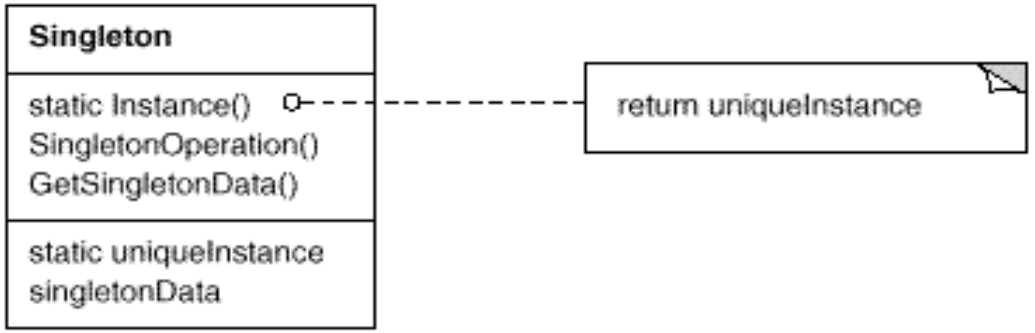


- Назва та класифікація
  - Одинак - патерн, який породжує об'єкти
- Призначення
  - Гарантує, що у класу є тільки один екземпляр, і надає до нього глобальну точку доступу
- Застосовність
  - Повинен бути рівно один екземпляр деякого класу, легко доступний всім клієнтам
  - Єдиний екземпляр повинен розширюватися шляхом породження підкласів, і клієнтам потрібно мати можливість працювати з розширеним екземпляром без модифікації свого коду

Singleton: структура



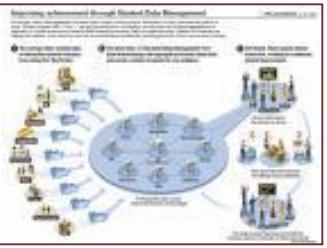
Структура



Учасники

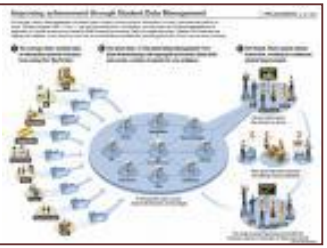
- Singleton – одинак, Определяет операцивизначає операцію **`Instance`** , яка дозволяє клієнтам отримати доступ до єдиного екземпляру

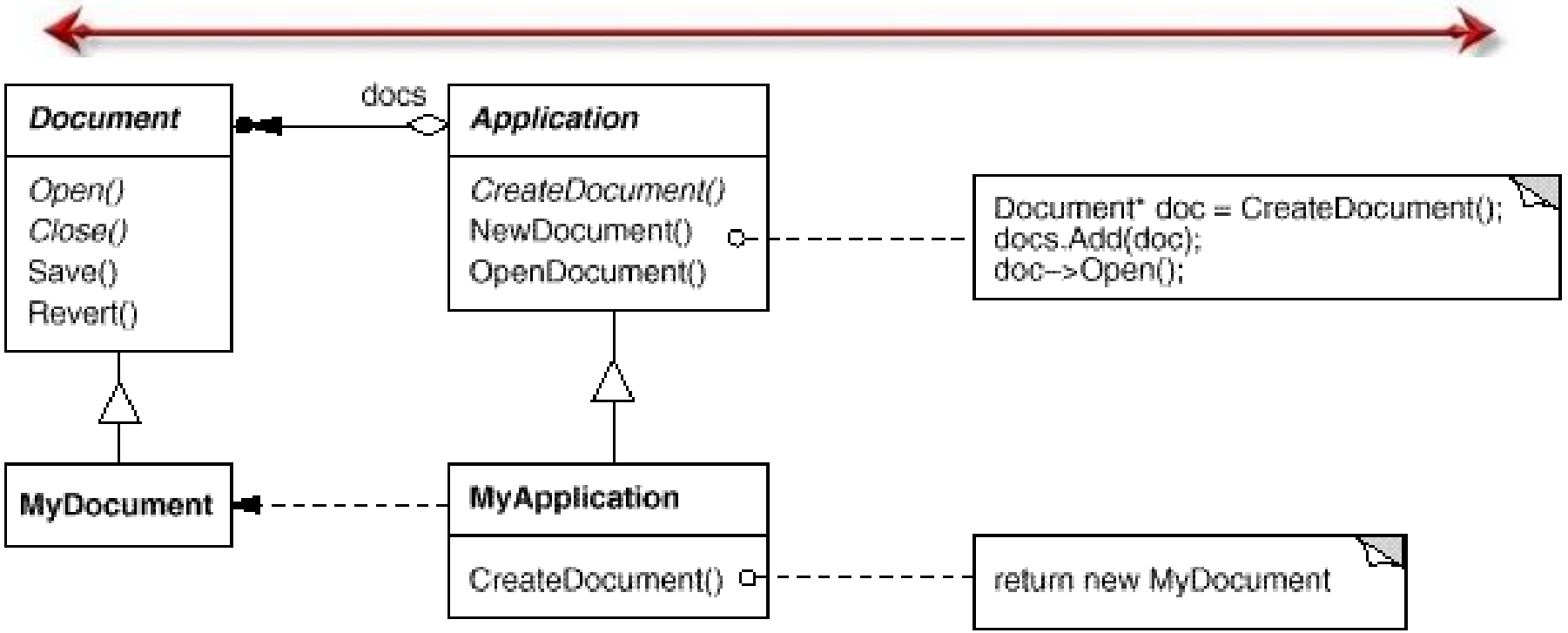
Factory Method



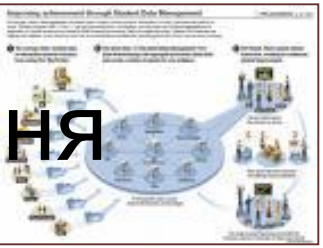
- Назва та класифікація
  - Фабричний метод - патерн, який породжує об'єкти
- Призначення
  - Визначає інтерфейс для створення об'єкта, але залишає подклассам рішення про те, який клас інстанціювати. Фабричний метод дозволяє класу делегувати інстанціювання в підкласи
- Відомий також під ім'ям Virtual Constructor

Factory Method





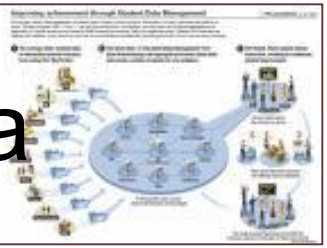
Factory Method: Застосування

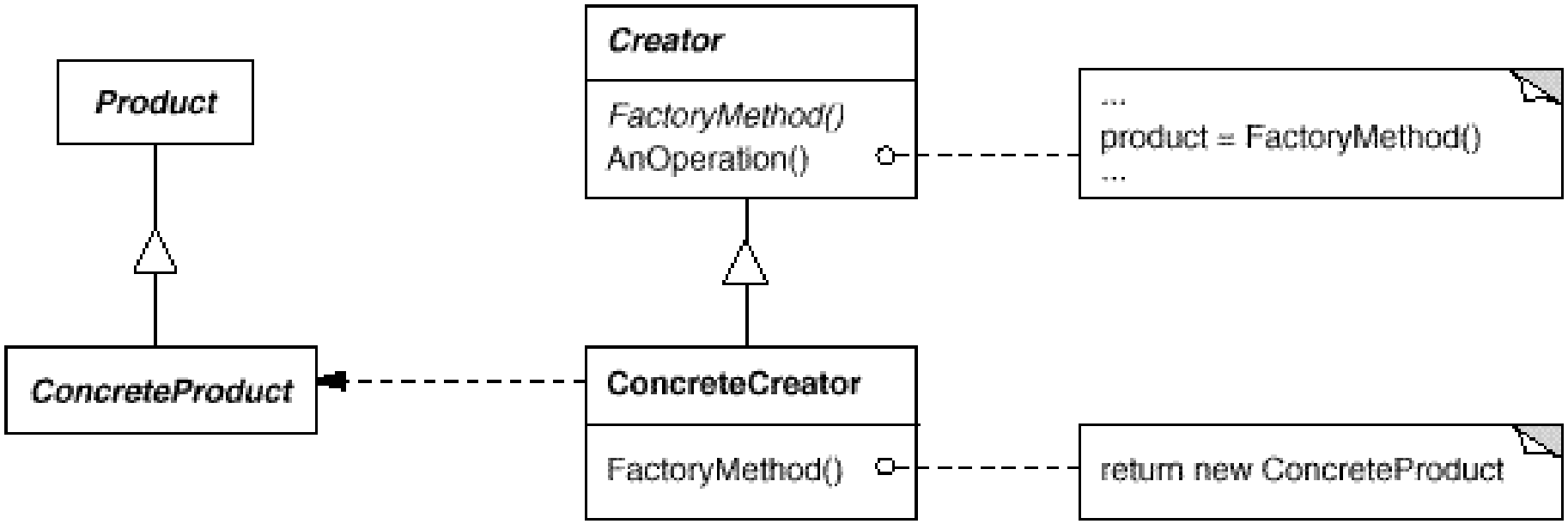


Використовуйте патерн фабричний метод, коли:

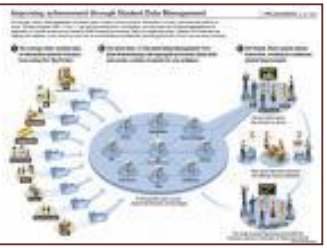
- класу заздалегідь невідомо, об'єкти яких класів йому потрібно створювати
- клас спроектований так, щоб об'єкти, які він створює, специфіковані підкласами
- клас делегує свої обов'язки одного з декількох допоміжних підкласів, і ви плануєте локалізувати знання про те, який клас приймає ці обов'язки на себе

Factory Method: Структура



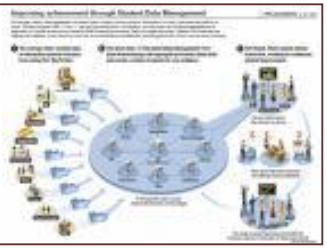


Структурні паттерни



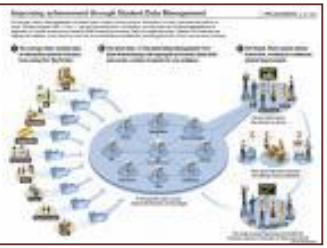
Adapter Адаптер	Зміна інтерфейсу
Bridge Міст	Розділення реалізації
Composite Компонувальник	Складна структура
Decorator Декоратор	Зміна обовязків
Facade Фасад	Інтерфейс до підсистеми
Flyweight Пристосуванець	Зниження видатків на зберігання
Proxy Замісник	Спосіб доступу до обєкта

Паттерни поведінки



Interpreter Інтерпретатор	Граматика та інтерпретація мови
Iterator Ітератор	Спосіб обходу елементів агрегата
Command Команда	Час і спосіб виконання запиту за рахунок включення запиту в объект
Observer Спостерігач	Спосіб, яким залежні об’кти підтримують себе в актуальному стані
Visitor Відвідувач	Операції, які можна застосувати до об’єкта (додавання операцій)

Патерни поведінки



Mediator Посередник	Спосіб кооперації взаємодіючих обєктів через проміжний
State Стан	Варіювання поведінки обєкта в залежності від його стану
Strategy Стратегія	Заключення алгоритма в обєкт, можливість заміни алгоритмів
Memento Зберігач	Закрита інформація, зовні обєкта, і час її зберігання
Chain of Responsibility Цепочка обовязків	Набір обєктів, які виконують запит
Template Method Шаблонний метод	Виділення в абстракцію кроків алгоритма