Wzorce projektowe w języku C# oraz platformie .NET

Spis treści

1. Krótki zbiór użytecznej informacji
2. Najbardziej użyteczne wzorce z wewnątrz
3. Tworzenie web aplikacji z użyciem wzorców projektowych

Początek rozdziałów

## Rozdział 1: Krótki zbiór użytecznej informacji

W tym rozdziale:

* .NET oraz C#
* Wstęp do programowania obiektowego
* Wstęp do UML
* Wstęp do wzorcu projektowego
* Wstęp automatów skończonych
* Tablica wzorców, klasyfikacja, przeznaczenie, sposób wykorzystania w .NET
* Wzorce według częstotliwości użycia
* Wzorce według ważności

### Podrozdział 1.1: Wiedza wstępna

Żeby ładnie sobie radzić z pojęciem wzorcu projektowego oraz używać wzorce projektowe skutecznie, koniecznym jest rozumienie podstaw programowania obiektowego, języka programowania oraz używanego frameworku. Także dobrą zaletą jest umiejętność odczytu diagram UML oraz znajomość podstaw automatów skończonych.

{ PLACEHOLDER DLA OOP, UML, SM, DP basics }

## Basic Design Patterns and Groups

Design patterns fall into groups, based on the type and aims of the pattern. For example, some patterns provide presentation logic for displaying specific views that make up the user interface. Others control the way that the application behaves as the user interacts with it. There are also groups of patterns that specify techniques for persisting data, define best practices for data access, and indicate optimum approaches for creating instances of objects that the application uses. The following list shows some of the most common design patterns within these groups:

**Presentation Logic**

* + Model-View-Controller (MVC)
  + Model-View-Presenter (MVP)
  + Use Case Controller

**Host or Behavioral**

* + Command
  + Publish-Subscribe / Observer
  + Plug-in / Module / Intercepting Filter

**Structural**

* + Service Agent / Proxy / Broker
  + Provider / Adapter

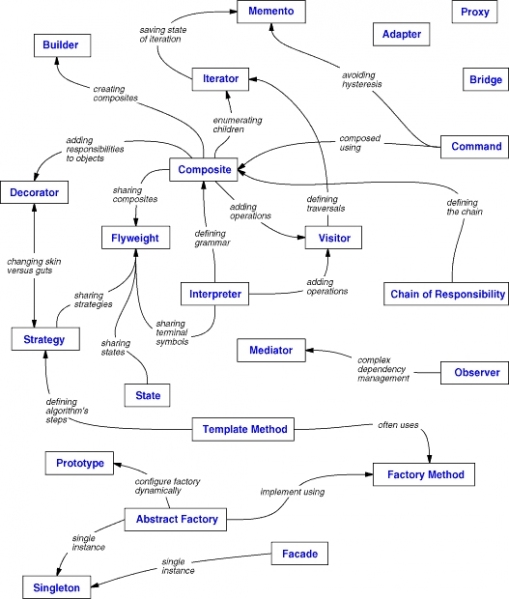
**Creational**

* + Factory / Builder / Injection
  + Singleton

**Persistence**

* + Repository





## Rozdział 3: Krótki zbiór użytecznej informacji

Użyta literatura

* Design Patterns via C# - Aleksandr Szewczuk, Dimitri Ochrimenko, Andrzej Kasjanow
* Katalog GoF – banda 4-ch