.NET Design Patterns

Description

Developers face some problems while developing applications or in the software application Lifecycle which are common and repeated, for example, creation and disposal of objects, interaction between objects, the structure of classes that enhance cohesion and loose coupling, fixing of bugs in a way that minimize changing of source codes, etc.

Design Patterns in the object-oriented world are a reusable solution to common software design problems that repeatedly occur in real-world application development. It is a template or description of how to solve problems that can be used in many situations.

Outline

Introduction to Design Patterns

- What Are Design Patterns?
- The Importance of Design Patterns
- Design Patterns Categories

SOLID Principles

- Single responsibility(SRP)
- Open/Close (OCP)
- Liskov Subtition(LSP)

- Interface Seggregation(ISP)
- Dependency Inversion(DIP)

Creative Design Patterns

- Singleton Design Pattern (SRP)
- Factory Design Pattern (OCP)
- Abstract Factory Design Pattern
- Builder Design Pattern
- Prototype Design Pattern

Structural Design Molds

- Adapter Design Pattern
- Bridge Design Pattern:
- Composite Design Pattern
- Decorator Design Pattern
- Facade Design Pattern.
- Flyweight Design Pattern
- Proxy Design Pattern

Behavioral Design Patterns

- Chain Of Responsibility Design Pattern
- Command Design Pattern
- Interpreter Design Pattern.
- Iterator Design Pattern
- Mediator Design Pattern
- Memento Design Pattern
- Observer Design Pattern
- State Design Pattern
- Strategry Design Pattern.
- Template Design Pattern.
- Visitor Design Pattern

Prerequisites

There is no prerequisite