C# 디자인패턴

디자인패턴이란?

- 많은 개발자들에 의해 입증된 방법
- 사용하는 이유
  - 개발자와 설계자의 공통 언어
  - 좋은 사례의 모음
- 객체지향 언어의 기본원칙을 기본으로 함

디자인 패턴 설계 원칙

- **단일 책임 원칙**
  - 하나의 클래스는 하나의 역할만 담당하도록 설계
- **개방 폐쇄의 원칙**
  - 확장에 대해서는 열려있고, 변형에 대해서는 닫혀있어야 한다.
- **리스코프의 대체 원칙**
  - 하위타입은 기반 타입에 대해 대체 가능해야한다.
- **인터페이스 분리의 원칙**
  - 하나의 일반적인 인터페이스 보다 구체적인 여러 인터페이스를 사용해야 한다
- **의존 역전의 원칙**
  - 구현 클래스의 필요에 의해 추상클래스를 변경할 수 있다.

디자인 패턴의 분류

		목적		
		생성	구조	행동
범주	클래스	Factory Method	Adapter	Interpreter Template Method
	객체	Abstract Factory Builder Prototype Singleton	Adapter Bridge Composite Decorator Façade Flyweight Proxy	Chain of Responsibility Command Iterator Mediator Memento Observer State Strategy Visitor