

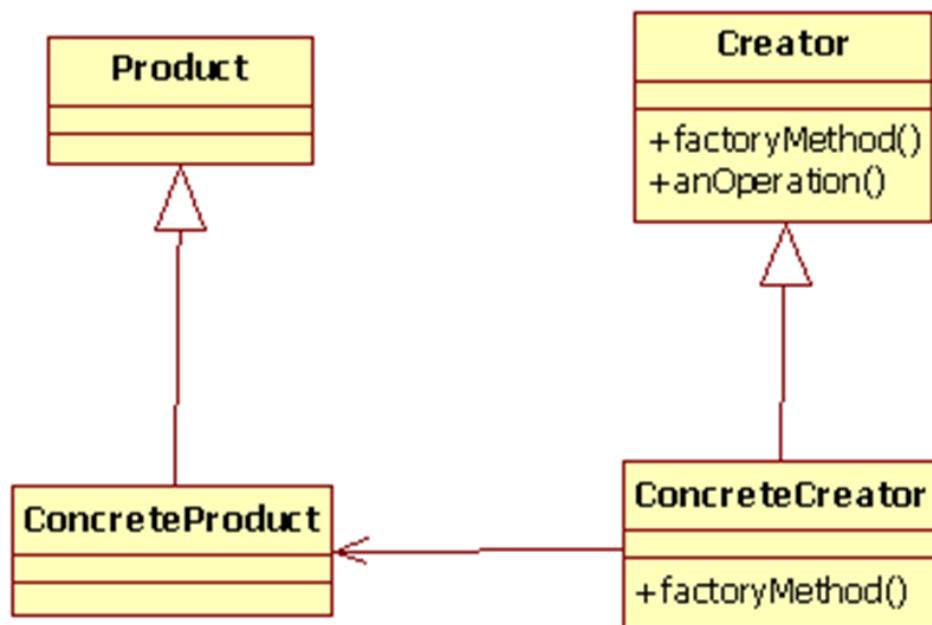
# 디자인패턴

1. 싱글톤 패턴에 대해 설명하고 DBCP(database connection pool)에서 어떤 방식으로 적용이 되는 지 간단하게 설명하세요

: 싱글톤 패턴은 단 하나의 인스턴스를 생성해 사용하는 디자인 패턴이다. 이렇게 만들어진 하나의 클래스 인스턴스는 전역 인스턴스 이기 때문에 다른 모듈들이 어디에서든 공유하며 접근, 사용할 수 있다. 또한 싱글톤 패턴으로 만들어진 클래스는 생성자가 여러번 호출되더라도 실제로 생성되는 객체는 하나고, 최초 생성 이후에 호출된 생성자는 최초에 생성한 객체를 반환한다.

DBCP(DataBase Connection Pool)DB와 커넥션을 맺고 있는 객체를 관리하는 역할을 담당합니다. 일반적으로 DB는 기타외부와 접속을 할 때마다 connection을 생성해야 하는데 connection을 맺는 작업은 매우 많은 시간과 자원을 소모하게 됩니다. 때문에 다수의 사용자들이 동시다발적으로 connection을 요청할 경우 즉 공통된 객체를 여러개 생성해서 사용해야하는 상황에서 많이 사용

2. 다음은 디자인 패턴 중 하나의 기본 구조도 입니다. 어떤 패턴의 구조도인지 설명하고 작동원리를 설명하세요



## Product

- 팩토리 메서드가 생성하는 객체의 인터페이스 정의한다.

## ConcreteProduct

- Product 클래스에 정의된 인터페이스를 실제로 구현하는 클래스

## Creator

- Product 타입의 객체를 반환하는 팩토리 메서드를 선언하는 클래스
- 팩토리 메서드를 기본적으로 구현, ConcreteProduct 객체를 반환한다.
- Product 객체의 생성을 위해 팩토리 메서드를 호출한다.

## ConcreteCreator

- 팩토리 메서드를 재정의하여 ConcreteProduct의 인스턴스를 반환

Ans.

**product의 세부 종류인 concreteProduct 객체의 생성을 creator의 자식 클래스인 concreteCreator에서 담당하는 것. 객체 생성 인터페이스(creator), 생성할 객체 종류 (concrete product) 선정 및 실제 생성을 담당하는 자식 클래스(concrete creator)로 구성된다.**

**크리에이터(Creator)**클래스는 새로운 제품 객체들을 반환하는 팩토리 메서드를 선언합니다  
아이스크림 공장 .**구상 크리에이터(concrete creator)**들은 기초 팩토리 메서드를 오버라이드(재정의)하여 다른 유형의 제품을 반환하게 하도록 합니다 다양한 맛의 아이스크림 공장 .  
**제품(product)**은 인터페이스를 선언합니다(아이스크림) . 인터페이스는 생성자와 자식 클래스들이 생성할 수 있는 모든 객체에 공통입니다.**구상 제품들(concrete product)**은 제품 인터페이스의 다양한 구현들.