

Builder Pattern

결론 및 정리

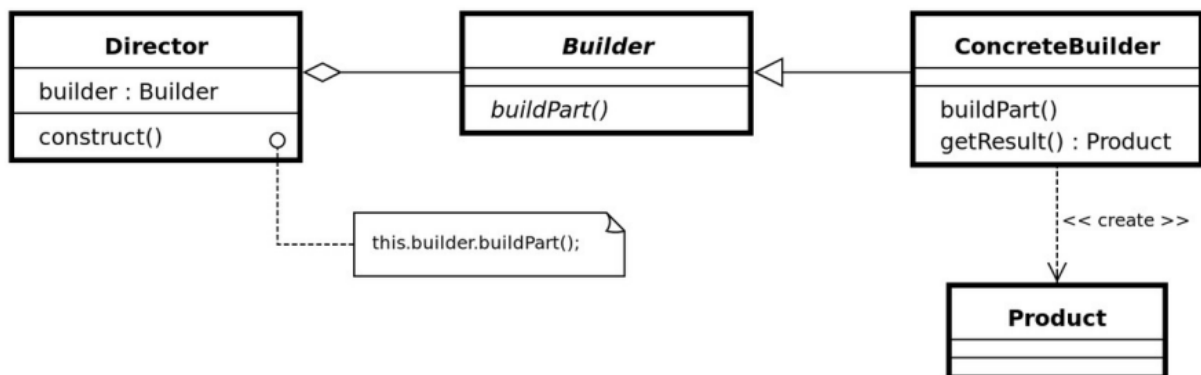
정의

Builder Pattern은 Creational Pattern 중 하나로, **복잡한 객체의 생성 과정과 표현 방법을 분리하여 동일한 생성 절차에서 서로 다른 표현 결과를 만들 수 있게 하는 패턴**.

- **Director**: 객체의 생성 과정을 캡슐화하는 클래스입니다. Builder를 사용하여 객체를 생성한다.
- **Builder**: 객체의 생성 방법과 표현 방법을 정의하는 인터페이스나 추상 클래스.
- **ConcreteBuilder**: Builder의 메서드를 구체적으로 구현하여, 실제로 객체의 부분들을 생성하고 조립하는 클래스.
- **Product**: 생성될 복잡한 객체.

Product 가 있다 하자. Builder 는 그 Product 에 관한 세부적인 정보를 포함하며, 세부적인 정보에 대한 Setup 및, Setup된 정보를 기반으로 Product 를 만들어주는 Method 를 포함하고 있다.

Director 는 Builder 를 이용한 생성을 지시하는 Class 로, 생성과정을 Wrap 하고 있다.



구현

1. Product 의 생성과정을 정의하는 Builder 인터페이스/추상 클래스를 만듭니다.
2. Builder를 구현/상속하는 ConcreteBuilder 클래스를 만들어, 객체의 각 부분을 생성하고 조립하는 로직을 구현.
3. Director 클래스를 사용하여, Builder를 통해 객체를 생성.
4. 클라이언트는 Director와 ConcreteBuilder를 사용하여 필요한 Product 객체를 생성.

Context & Problem & Merit

- 복잡한 객체의 생성 과정이 여러 단계로 나뉘거나, 다양한 표현 방법이 필요할 때 Builder Pattern을 사용.
 - 객체 생성 절차는 동일하되, 생성 결과는 다를 수 있습니다.
1. **유연성**: 복잡한 객체의 생성 과정과 표현 방법을 분리하여, 다양한 표현 결과를 얻을 수 있음.
 2. **명확성**: 복잡한 객체의 생성 과정을 단계별로 나누어, 코드의 가독성과 유지 보수성을 향상 시킴.
 3. **불변성**: Builder를 사용하여 완성된 객체는 변경할 수 없게 만들 수 있습니다. 이를 통해 객체의 불변성을 보장할 수 있음.
-

◆ [빌더\(Builder\) 패턴 - 완벽 마스터하기 \(tistory.com\)](http://tistory.com) : Builder Pattern 찾아보면서 발견한 사이트인데, 꽤 괜찮은 듯 하여 남겨둠.