Builder Pattern
복잡한 인스턴스를 조립한다.

01. Builder 패턴

- 구조를 가진 인스턴스를 쌓아 올리는 패턴

- 같은 구성 요소(part)를 가지지만, 서로 다른 방법으로 구축되는 경우에 적용함
  - 예: 구조물 '문서'가 title, string, item라는 구성요소로 가진다.
    - 이 구성요소를 만들기 위한 메소드가 필요하다
      - makeTitle( ), makeString( ), makeItem( )
      - 이러한 메소드들을 Builder 인터페이스에 선언
    - 각 구성요소를 만드는 방식은 빌더마다 다르다.
      - 각각의 구체적인 빌더는 Builder 인터페이스에 선언된 메소드들을 구현함
      - 즉, 빌더마다 부품을 만드는 방법은 다르다.

02. 예제 프로그램

- Builder 패턴을 사용해서, '문서'를 작성하는 프로그램

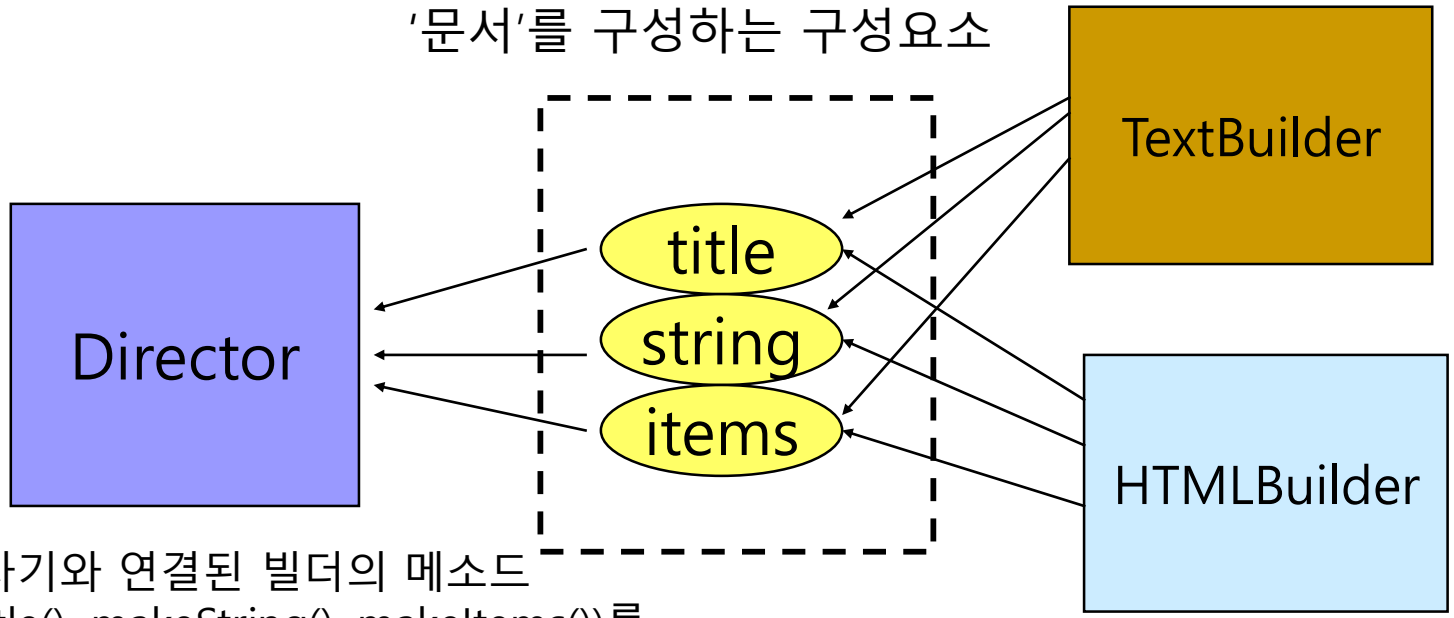
- '문서'의 구조
  - 타이틀을 하나 포함한다.
  - 문자열을 여러 개 포함한다.
  - 항목을 여러 개 포함한다.

02. 예제 프로그램

- Builder 클래스는 문서를 구성하는데 필요한 구성 요소를 만들어주는 메소드를 정의한 추상클래스이다.
  - Director 클래스가 이 메소드를 이용해서 구체적인 하나의 문서를 만든다
  - 문서를 작성하기 위한 구체적인 처리는 Builder 클래스의 하위 클래스가 결정한다.
    - TextBuilder 클래스: 일반 텍스트를 이용해서 문서를 만든다
    - HTMLBuilder 클래스: HTML을 사용해서 문서를 만든다

- Director가 TextBuilder를 사용하면, 일반텍스트의 문서가 생기고, HTMLBuilder를 사용하면 HTML 문서가 생긴다.

02. 예제 프로그램



자기와 연결된 빌더의 메소드
(makeTitle(), makeString(), makeItems())를
이용해서 title, string, items을
생산하는 명령을 내린다.
(빌더에게 일을 시킬 때 재료를 제공한다)

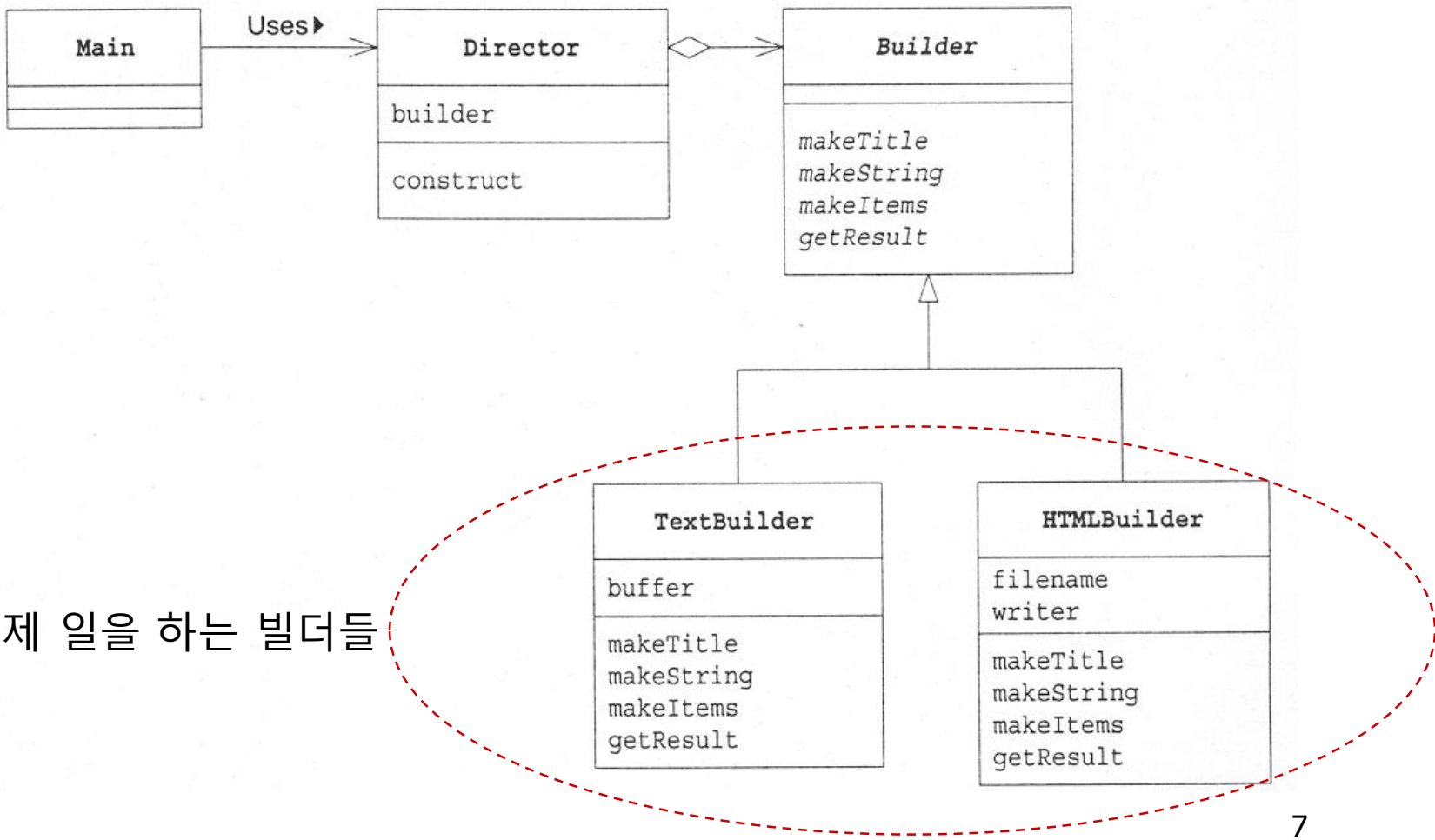
두 빌더는 Title, string, items를 만들어내는
메소드를 다 같이 제공한다.
(그러나, title, string, items를 만들어내는
방식은 다르다)

02. 예제 프로그램

이름	해설
Builder	문서를 구성하기 위한 메소드를 결정하는 추상 클래스
Director	하나의 문서를 만드는 클래스
TextBuilder	일반 텍스트(보통의 문자열)를 사용해서 문서를 만드는 클래스
HTMLBuilder	HTML 파일을 사용해서 문서를 만드는 클래스
Main	동작 테스트용 클래스

02. 예제 프로그램

- 클래스 다이어그램



실제 일을 하는 빌더들

02. 예제 프로그램

- Builder 클래스
  - '문서'를 만들 때 사용되는 구성요소를 생산하는 메소드들을 선언한 추상 클래스
    - makeTitle( ), makeString( ), makeItems( )
  - getResult( )
    - 완성된 문서를 얻을 때 호출되는 메소드

02. 예제 프로그램

- Director 클래스
  - Builder 클래스에 선언되어 있는 메소드를 이용해서 문서를 만듬
  - Director 클래스의 생성자의 인자는 Builder 형입니다.
    - 실제로는, Builder의 하위 클래스가 전달된다.
    - Director 객체가 생성될 때, 자신과 관련있는 Builder가 연결된다.
  - construct( ): 문서를 만드는 메소드
    - Builder에 선언된 메소드들을 이용하여 문서를 만든다.

02. 예제 프로그램

- TextBuilder 클래스
  - Builder 클래스의 하위 클래스
  - StringBuffer를 이용하여, 일반 텍스트 문서를 만든다.

- HTMLBuilder 클래스
  - Builder 클래스의 하위 클래스
  - PrintWriter를 이용하여, HTML 태그가 포함된 HTML 문서 파일을 만든다.

02. 예제 프로그램

- Main 클래스
  - Main은 Director의 생성자를 이용해서 필요한 구체적인 Builder를 포함하는 Director를 생성한다.
  - Director는 Builder 메소드만을 사용해서 문서를 만든다.
    - Director는 실제로 작동하고 있는 것이 TextBuilder인지, HTMLBuilder 인지 의식하지 못 한다.

03. 등장 역할

- Builder(건축자)의 역할
  - 인스턴스 구축을 위한 인터페이스(API)를 결정한다
  - 예제에서는, Buider 클래스가 해당됨

- ConcreteBuilder(구체적 건축자)의 역할
  - Builder 인터페이스를 구현한 구체적인 클래스
  - 예제에서는, TextBuilder나 HTMLBuilder가 해당됨

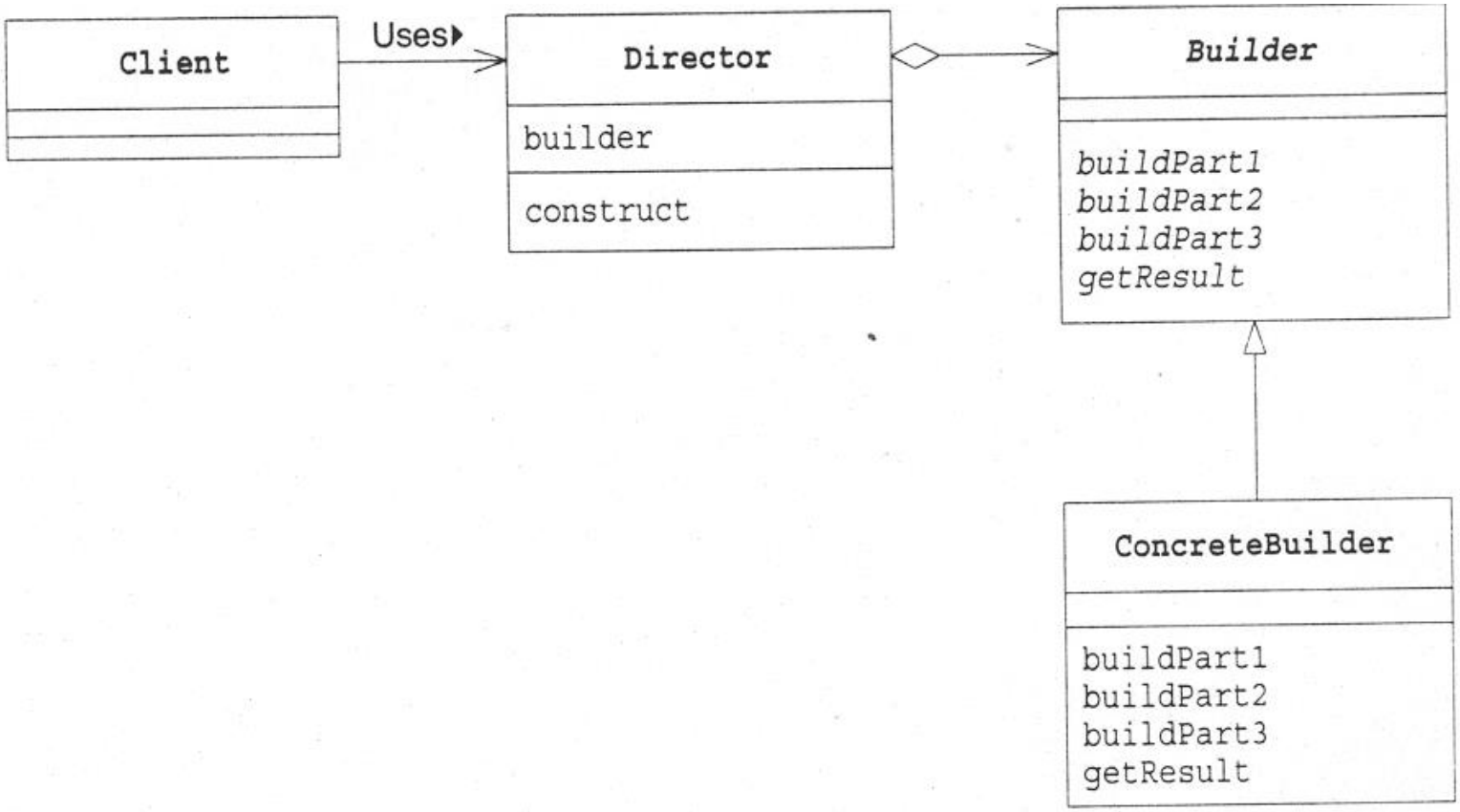
03. 등장 역할

- Director(감독자)의 역할
  - Builder의 인터페이스 만을 사용해서, 인스턴스를 생성한다
  - ConcreteBuilder에 의존한 프로그래밍은 하지 않는다.

- Client(의뢰인)의 역할
  - Builder 패턴을 이용하는 역할
  - 예제에서, Main 클래스가 해당됨

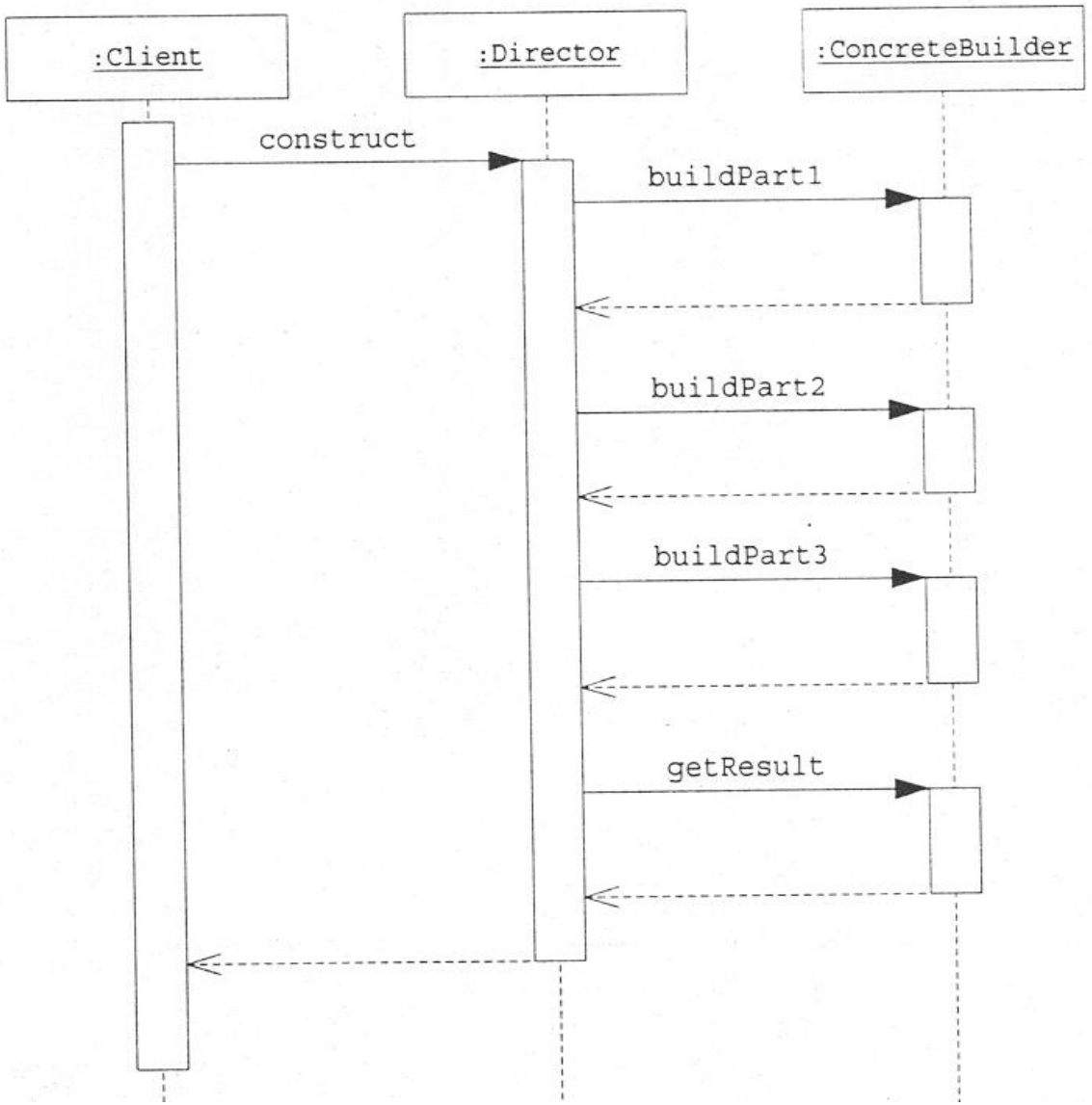
03. 등장 역할

- 클래스 다이어그램



03. 등장 역할

- 시컨스 다이어그램



04. 독자의 사고를 넓혀주는 힌트

- 클래스가 알고 있는 것은 무엇인가?
  - Director 클래스는 자신이 이용하고 있는 Builder 클래스의 하위 클래스가 무엇인지 모른다.
    - 이유: Director 클래스의 소스 안에서는, TextBuilder나 HTMLBuilder 클래스는 전혀 사용하지 않고, Builder 만을 이용하기 때문에
  - 모르기 때문에, '부품 교환'이 용이하다.
    - 예1: 새로운 구체적인 Builder를 추가하기가 쉽다.
      - XMLBuilder를 쉽게 여기에 추가할 수 있다.
    - 예2: 기존의 TextBuilder나 HTMLBuilder의 수정이 용이하다.

05. 관련 패턴

- Template Method(3장)
  - Builder 패턴에서는 Director가 Builder를 조정한다.
  - Template Method 패턴에서는 상위 클래스가 하위 클래스 사용 방식를 조정한다.
- Composite(11장)
- Facade 패턴(15장)

06. 요약

- 구조를 가진 인스턴스를 쌓아올리는 Builder 패턴