

디자인 패턴

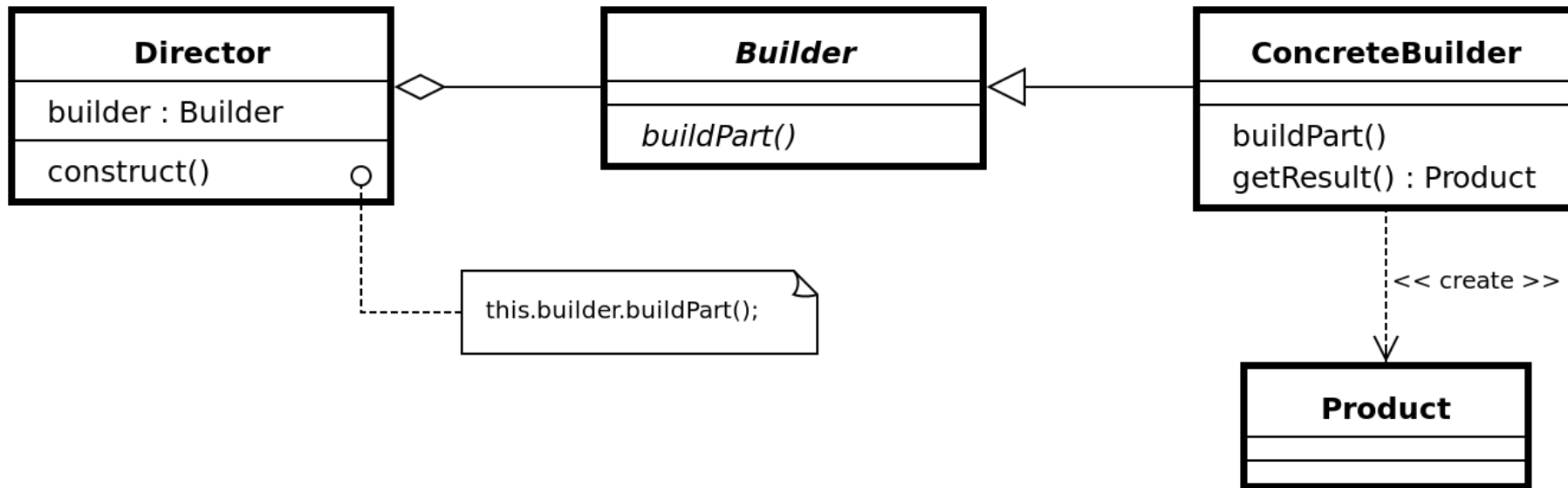
빌더(BUILDER)

INDEX

1. Definition
2. Advantages
3. Disadvantages
4. Examples

Definition

- Q) 빌더 패턴이란?
- A) construction(생성)과 representation(표현)의 분리
다른 표현이라도 동일한 생성 절차를 제공



Definition

- 기타연주
- 시니피앙 vs 시니피에
- 표현 vs 생성
- Director vs Builder
- 같은 설계도, 다른 제작자

Advantages

- 제품 내부 표현의 다양화
- 생성과 표현 코드 분리(encapsulation)
- 객체 생성 절차 제어권(control) 제공

Disadvantages

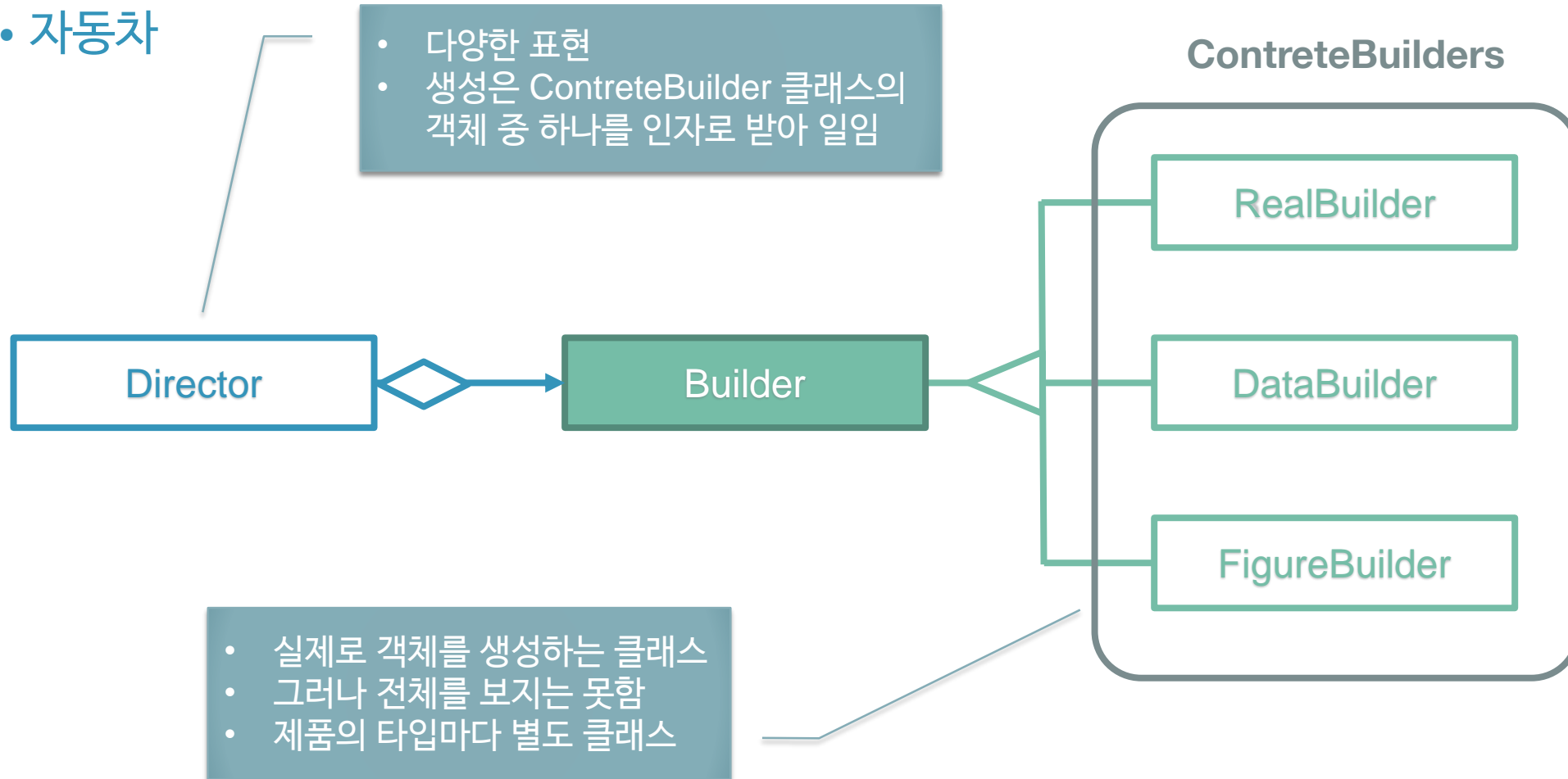
- 제품의 타입마다 새로운 ConcreteBuilder 필요

Examples

예시	표현(Director)	생성(Builder)된 제품(Product)
드라마	같은 대사	다른 장르의 드라마
자동차	같은 설계도	진짜 자동차 / 3D model / 장난감 드론
문자 인코딩	보기엔 같은 문자열	실제로는 다른 16진수 값 데이터
파일 확장자	뷰어로 보면 다 똑같음	실제로는 다른 16진수 값 데이터
레이싱 코스	같은 구조의 코스	실제 인게임 땅바닥 / UI에 표기될 미니맵
멀티플랫폼 빌더	같은 코드	각종 플랫폼 상에서의 2진 데이터

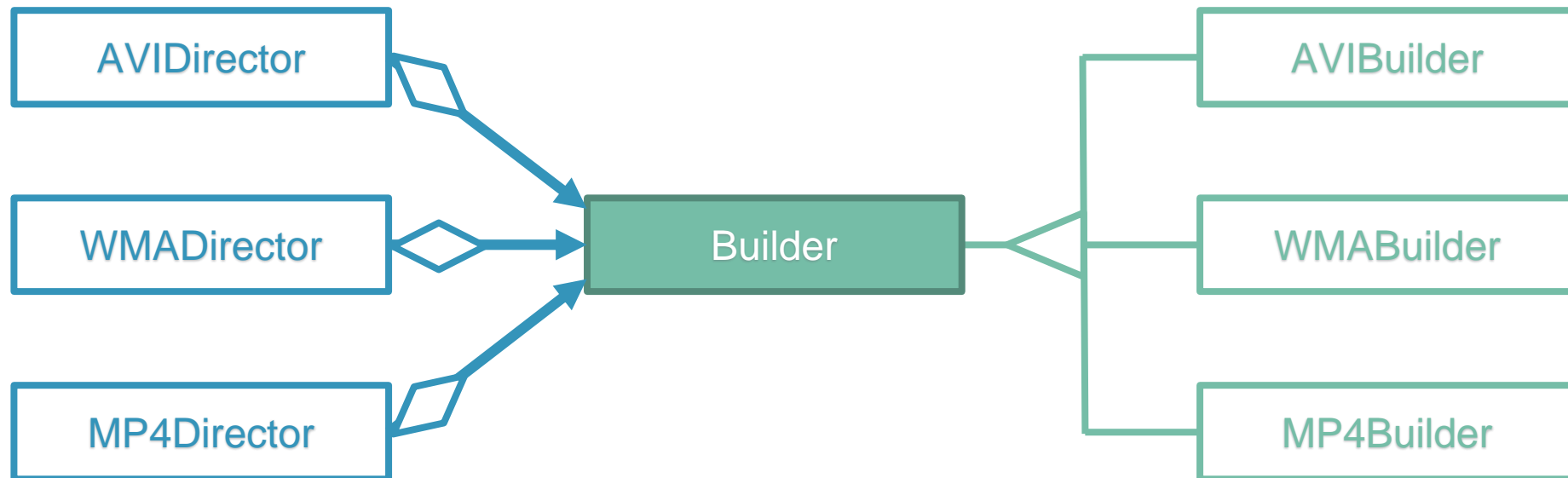
Examples

- 자동차



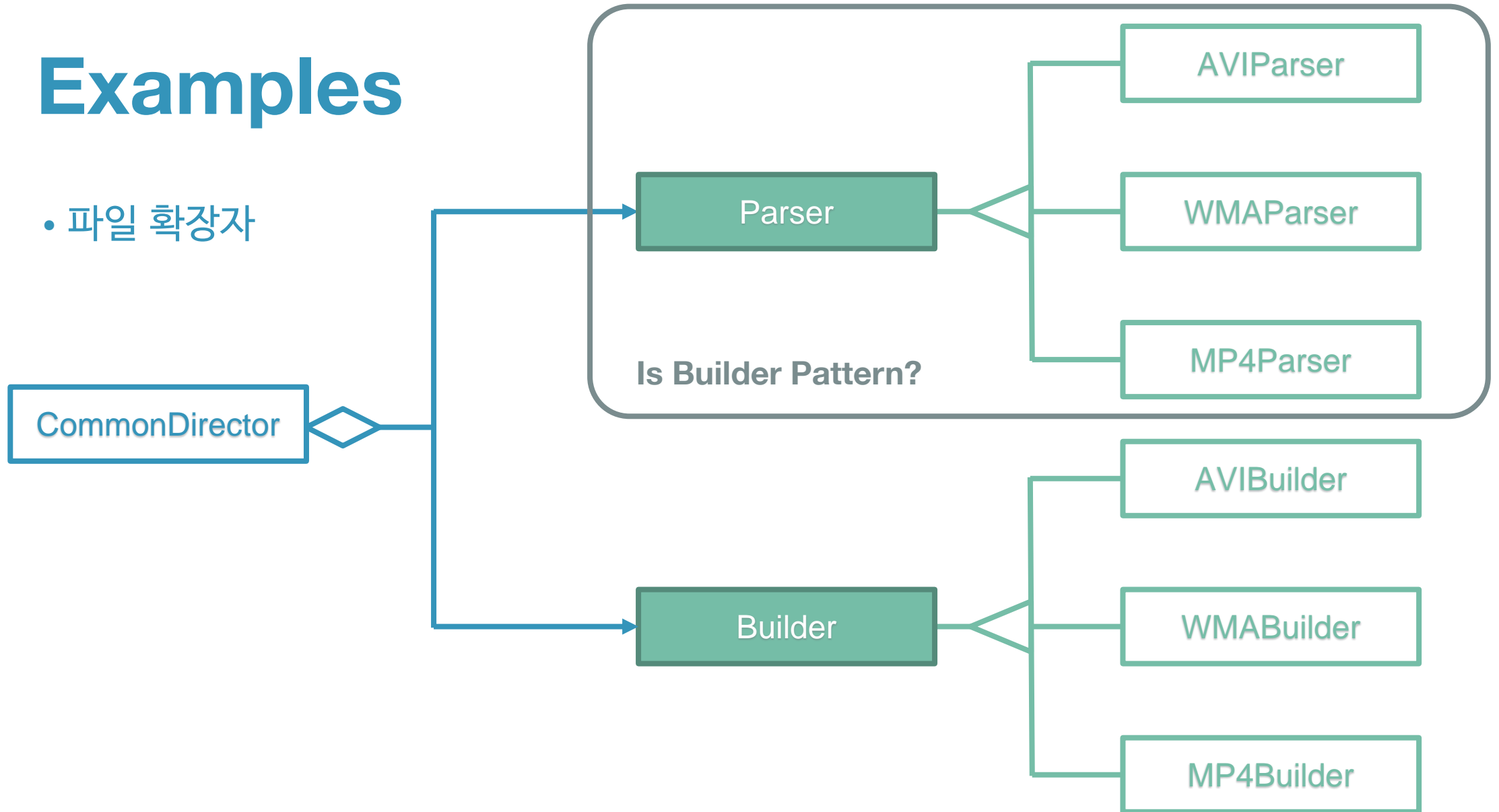
Examples

- 파일 확장자



Examples

- 파일 확장자



Examples

```
AVIDirector::Convert(  
    char* data,  
    Builder& builder){  
    if (...) { // Parsing  
        builder.Build...();  
    }  
    ...  
    return builder.GetResult();  
}
```

```
CommonDirector::ConvertAnything(  
    char* data,  
    Parser& parser,  
    Builder& builder){  
    char* buff = parser.GetResult(data);  
    return builder.GetResult(buff);  
}
```

Thank you
