

# 의존성

## 의존성 정의

- 코드에서 두 모듈간의 연결을 의존성이라고 합니다.
- 객체지향언어에서 두 클래스 간의 관계를 말하기도 합니다.

## 의존성 종류

- 의존관계
- 연관관계
- 집합관계
- 합성관계

## 의존관계

- 클래스 필드로 다른 클래스의 객체를 가지고 있는 관계

```
public class B {  
    private int numB;  
  
    public int getNumB() {  
        return this.numB;  
    }  
}  
  
public class A {  
    private int numA;  
  
    // add 메소드가 반환한 이후에는 B 클래스의 b 객체는 제거된다.  
    public int add(B b) {  
        return numA + b.getNumB();  
    }  
}
```

## 연관관계



- 클래스 필드로 다른 클래스의 객체를 가지고 있는 관계

```
public class B {  
    private int numB;  
  
    public int getNumB() {
```

```

        return this.numB;
    }
}

public class A {
    private int numA;
    private B b;

    // add 메소드가 반환한 이후에도 B 클래스의 b 객체는 여전히 남아 있다.
    public int add() {
        return numA + this.b.getNumB();
    }
}

```

## 집합관계

- Association(연관관계)의 특수한 형태
- 클래스 A와 클래스 B의 생명주기는 반드시 일치하지 않는다



```

public class B {
    private int numB;

    public int getNumB() {
        return this.numB;
    }
}

public class A {
    private int numA;
    private B b;

    public A(B externalB) {
        this.b = externalB;
    }
}

```

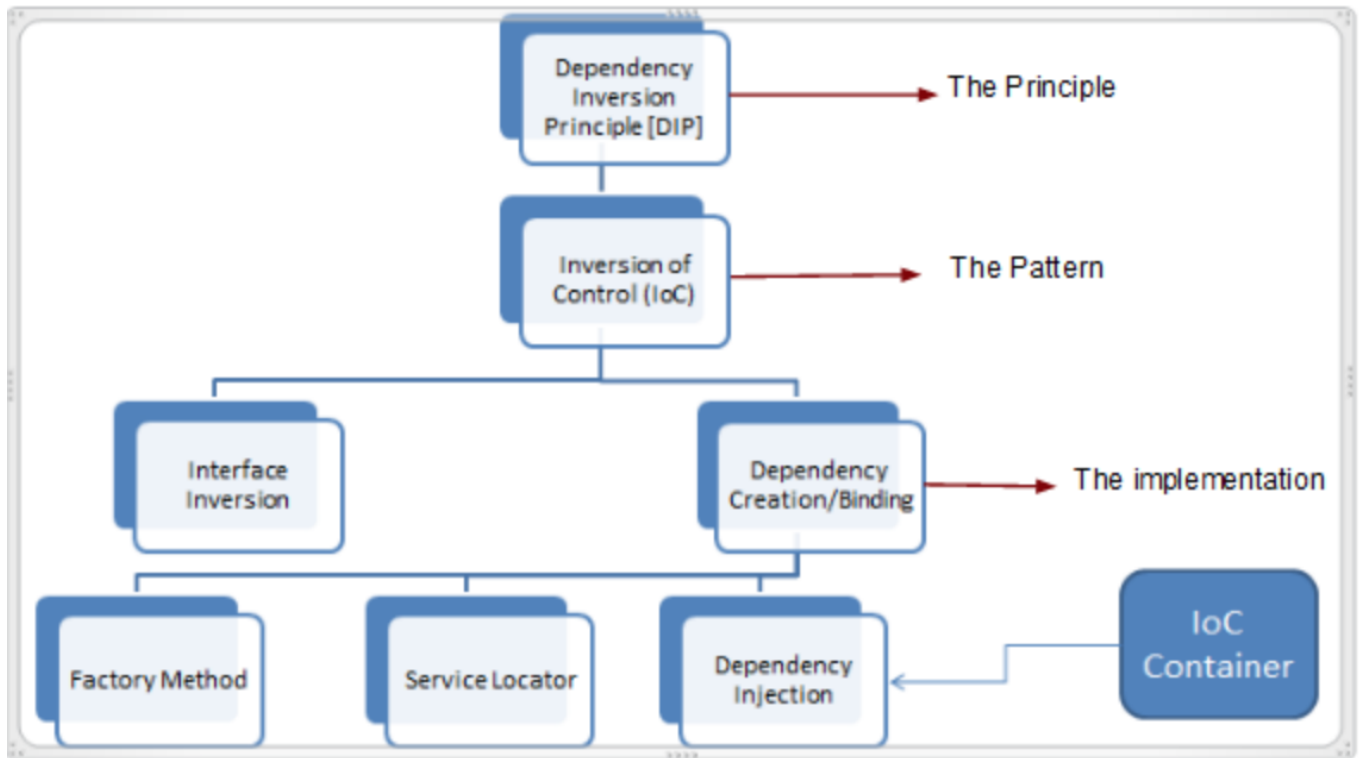
## 합성관계

## 의존성 주입

### Dependency Injection (DI)

- IoC 의 패턴 중에 하나
- DI = Dependency Injection
- Object 간의 의존성을 낮춘다.
- 외부에서 객체를 생성하고 전달한다.

## Dependency Inversion Principle



- 상위 모듈이 하위 모듈에 의존관계를 가지지 않도록 구현해야 합니다.
- 추상클래스는 그 구현체의 내용에 의존관계를 가지지 않는다.
- 구현체가 추상클래스에 의존관계를 가질 수 있다.

## DI 의 예 - Factory Method 패턴

### FactoryMethod

## Factory Method

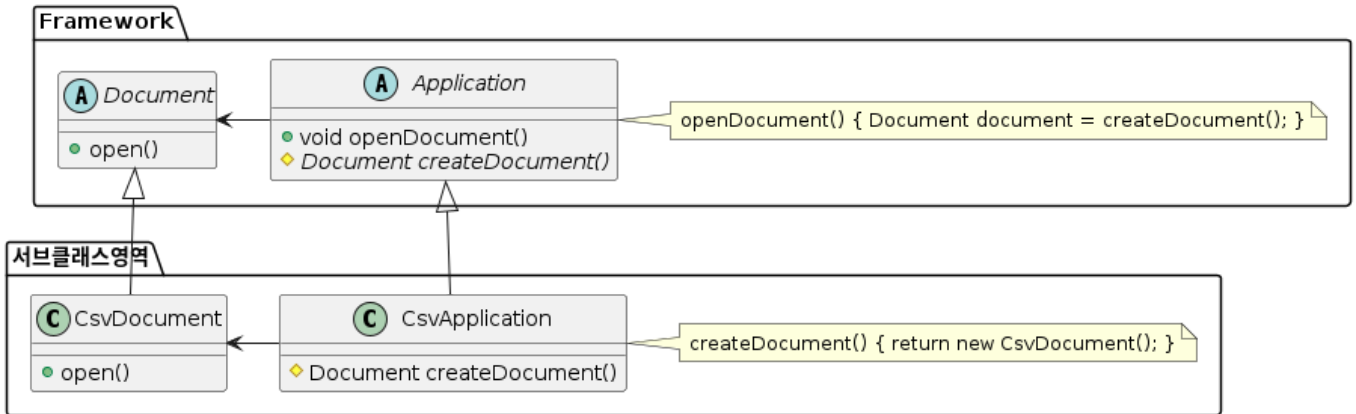
### 문제

- 사용자에게 다양한 문서를 읽어서 객체로 결과를 반환하는 프레임워크 개발을 해야 합니다.
- Application 클래스와 Document 클래스로 추상화 할수 있는데, 이 두 클래스는 모두 추상 클래스이고 이 클래스들을 상속해서 문서의 종류에 따른 대응을 할 수 있습니다.
- Application 클래스는 언제 Document 클래스를 생성하고 사용해야 하는지는 알 수 있지만 프레임워크에서 구체적으로 어떤 문서를 처리할 Document 를 생성해야 하는지는 결정 할 수 없습니다.

### 해법

- Document 의 서브클래스 중 어느 클래스를 생성하는 지는 Application 클래스의 서브클래스가 결정하도록 설계합니다.

- Application 클래스의 서브클래스는 추상화된 createDocument()메소드를 정의하여 적당한 Document 클래스의 서브클래스를 반환하도록 합니다.
- createDocument() 메소드를 factory method 라고 부릅니다.



## 구현 코드

- feature/pattern\_factory\_method
- Application.java
  - 어떤 Document 를 처리할지 Application 추상 객체에서 결정하지 않습니다.

상위 모듈이 하위 모듈에 의존관계를 가지지 않도록 구현해야함.  
순환 참조하면 안되는 이유