



# IoC (Inversion of Control)

강 사 : 임 정 훈



# 목차

1

IoC 소개

2

바람직한 Class 설계방법

3

클래스 간의 강 결합 예제

4

인터페이스를 이용한 약 결합 예제

5

스프링 Bean 을 적용한 IoC 실현

6

UML 클래스 모델링



# IoC 소개

\* IoC(Inversion of Control)

1. 제어의 역전
2. 애플리케이션의 제어 흐름구조를 바꾸는 기법
3. Object 가 자신이 사용할 Object 를 선택하거나 객체를 생성하지 않는다 .
4. 프레임워크 (spring) 에 의해서 애플리케이션의 제어 흐름 구조를 바꾼다 .



# IoC 소개

## Spring IoC (Inversion of Control : 제어의 역전 )

기존 제어 방식 :



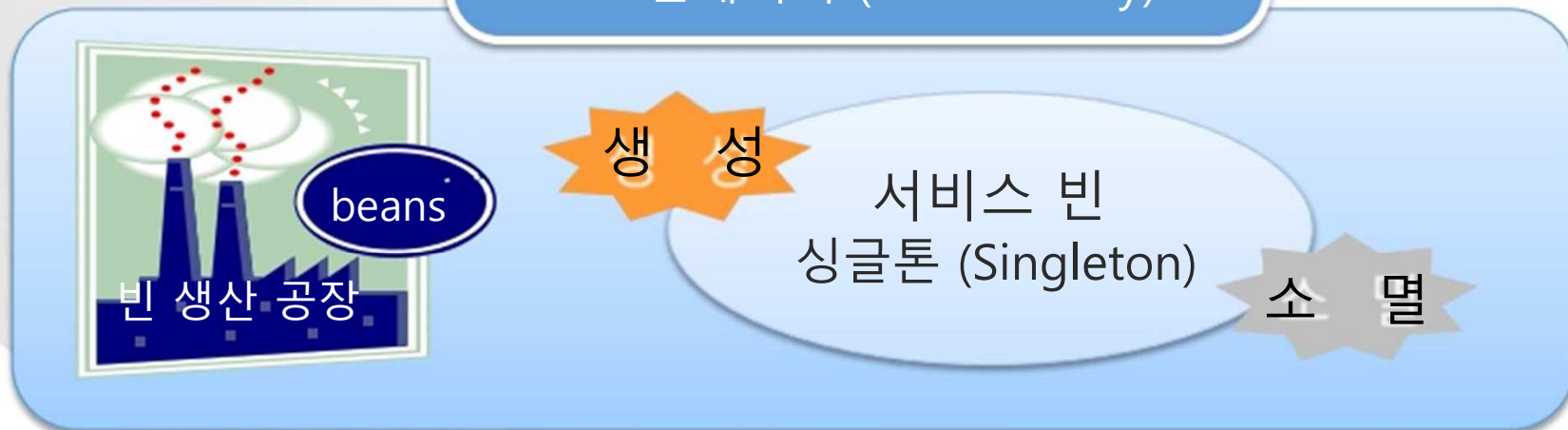
Spring IoC 방식 :



Bean : Spring 이 만들고 관계를 설정하는 객체 (POJO)

개발에 있어서 확장성과 재 사용성 극대화 추구

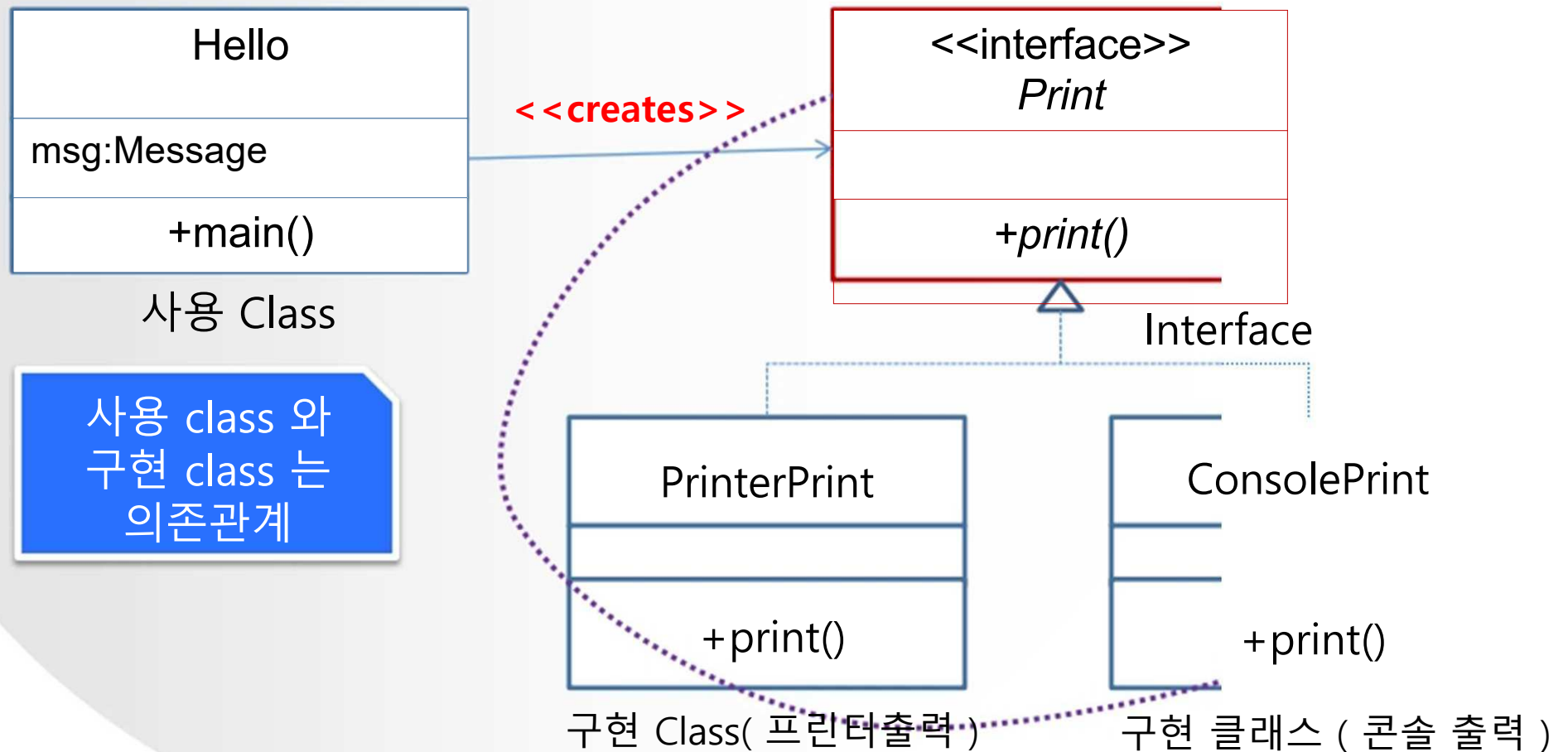
### IoC 컨테이너 (BeanFactory)





# 기존 제어 방식 예

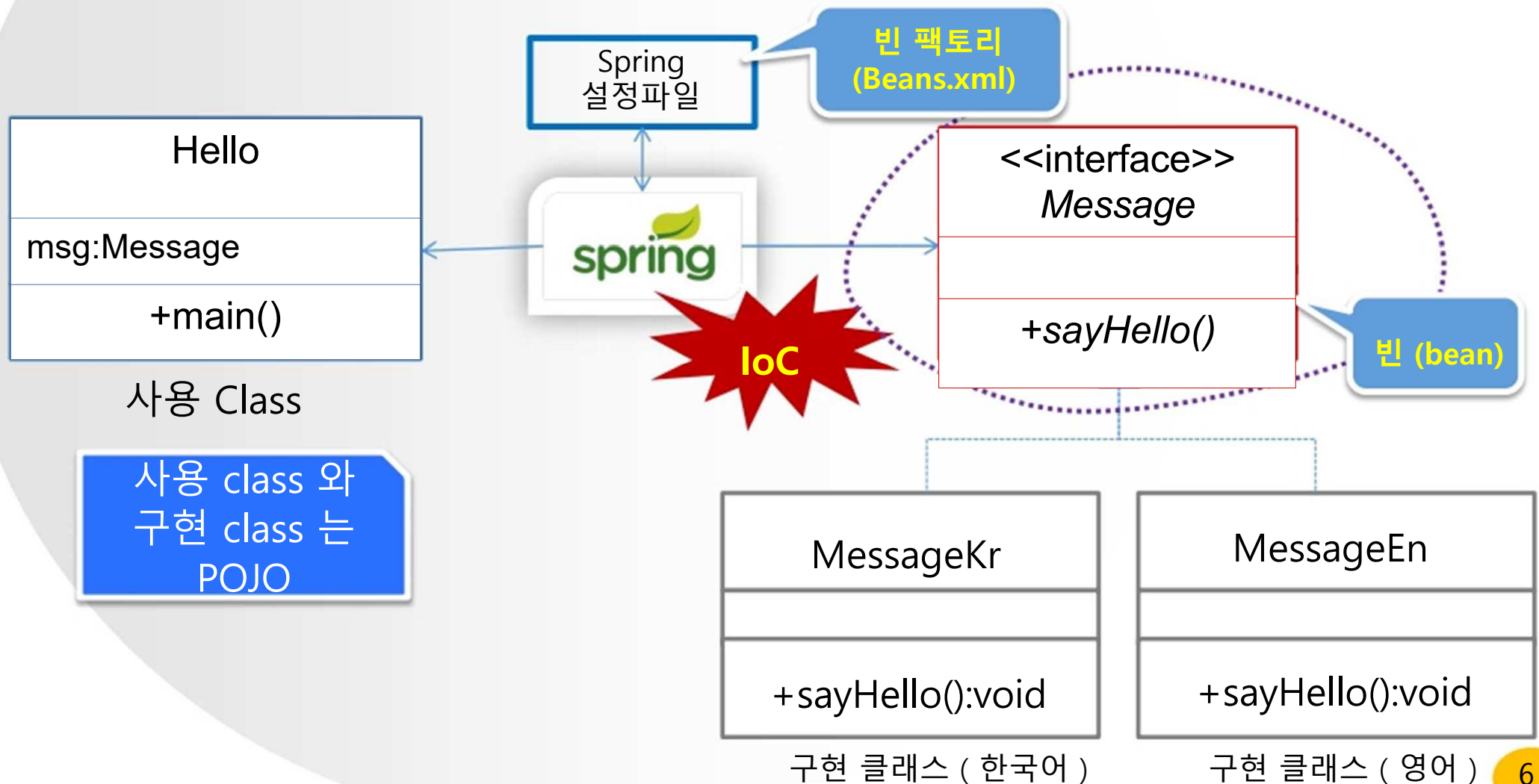
## Print 인터페이스를 적용한 기존 클래스 의존 방식





# Spring IoC 방식 예

Spring 프레임워크를 적용한 IoC( 제어의 역전 )





# Singleton 클래스 예

단 한 개의 인스턴스만 생성하도록 설계된 Class

Singleton
-instance:Singleton
+getInstance():Singleton

클래스 모델링

한 개의  
인스턴스만  
생성

```
class Singleton{  
    private static Singleton instance;  
    private Singleton( ){ } // 생성자  
    public static Singleton getInstance(){  
        if(instance == null){  
            instance = new Singleton();  
        }  
        return instance;  
    }  
}
```

스프링 객  
체 ( 빈 ) 는  
싱글톤방식





# Singleton 클래스 예

단 한 개의 인스턴스만 생성하도록 설계된 Class

Singleton
-instance:Singleton
+getInstance():Singleton

클래스 모델링

한 개의  
인스턴스만  
생성

```
class Singleton{  
    private static Singleton instance;  
    private Singleton( ){ } // 생성자  
    public static Singleton getInstance(){  
        if(instance == null){  
            instance = new Singleton();  
        }  
        return instance;  
    }  
}
```

스프링 객  
체 ( 빈 ) 는  
싱글톤방식





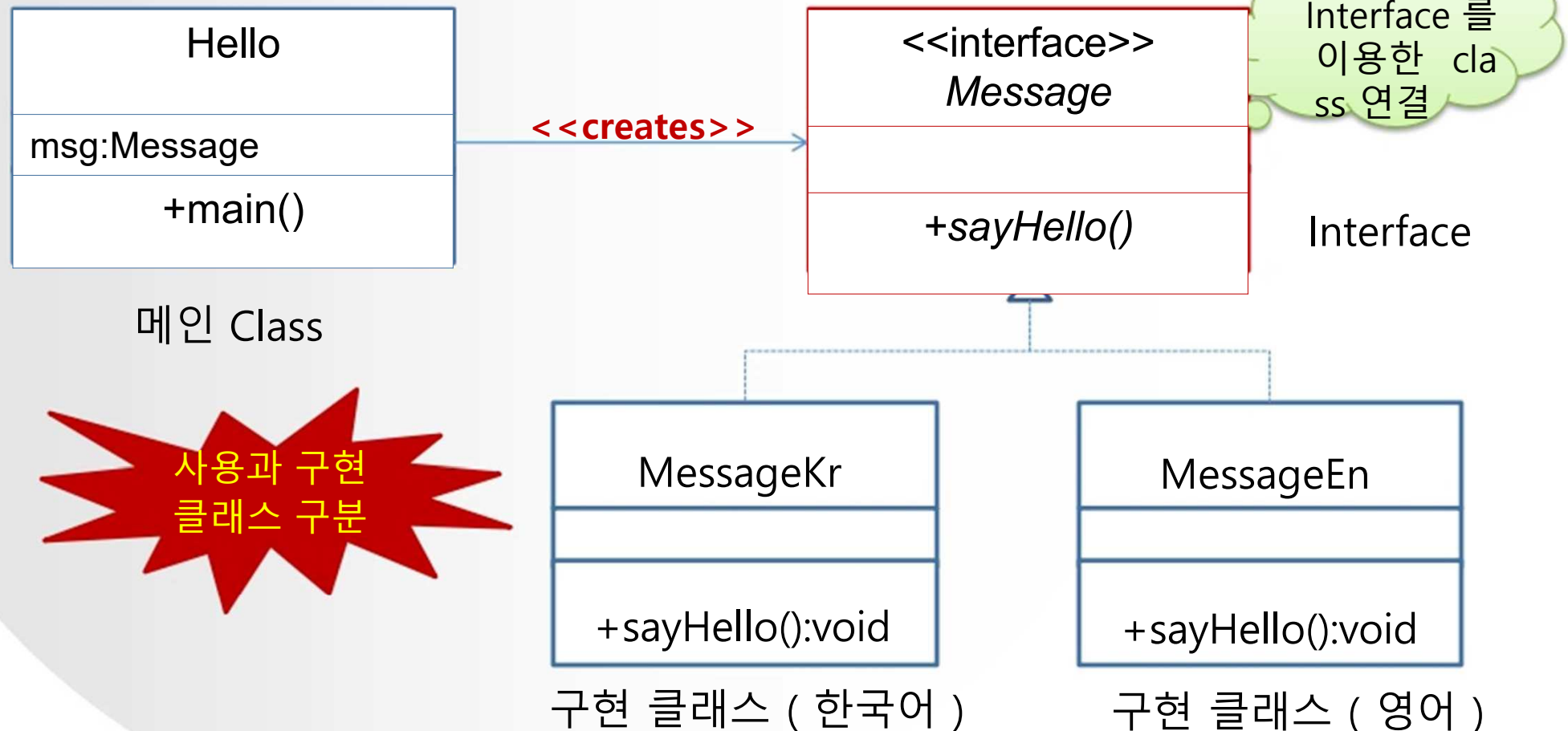
## 바람직한 class 설계 방법

- \* 결합도 (Coupling) 낮게 , 응집도 (Cohesion) 높게 설계
- \* 결합도 : 클래스 ( 모듈 ) 와 클래스 간의 상호작용 척도
  - 1) 강결합 (Tightly Coupled) : 특정 클래스에서 코딩 변경 발생하는 경우 수정범위가 넓다 . -> 유지보수가 어려움  
클래스와 클래스 간의 결합 정도가 높다는 의미
  - 2) 약결합 (Loosely Coupled) : 클래스 간의 관계를 가능한 약하게 설계하는 방식으로 특정 클래스의 코드 변경에 따른 영향을 최소화하는 한다 .(interface 적용 )
- \* 응집도 : 클래스를 구성하는 내부기능들의 연관성 척도 응집도가 높을 수록 클래스의 독립성은 높아진다 .



# 인터페이스를 이용한 약 결합 예제

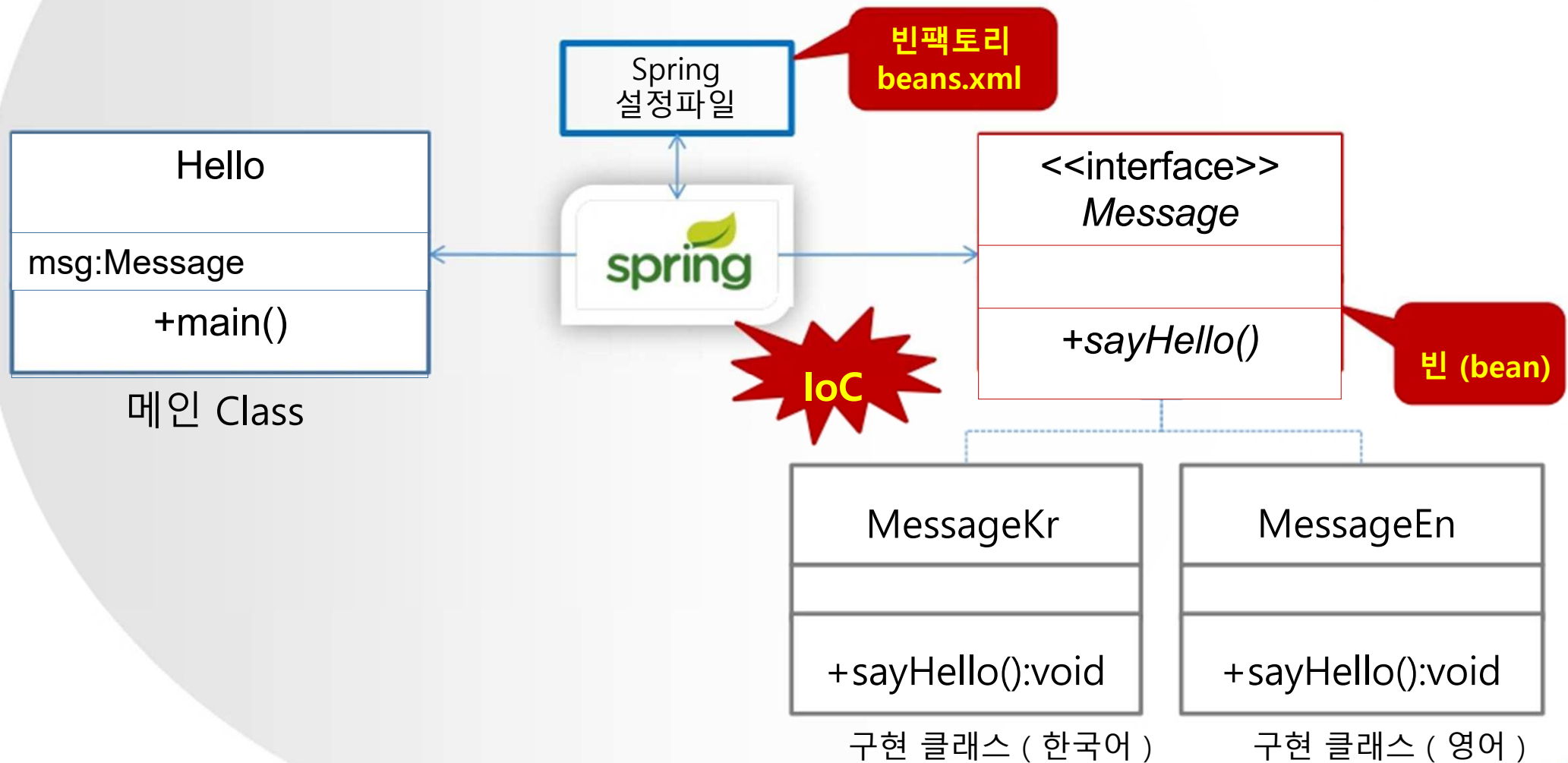
## Sample02. 인터페이스를 적용한 약한 결합 관계





# 스프링 Bean 을 적용한 IoC 실현

Sample03. SPRING 프레임워크를 적용 ( 유연성과 확장성 증대 )





# 스프링 Bean 을 적용한 IoC 과제

