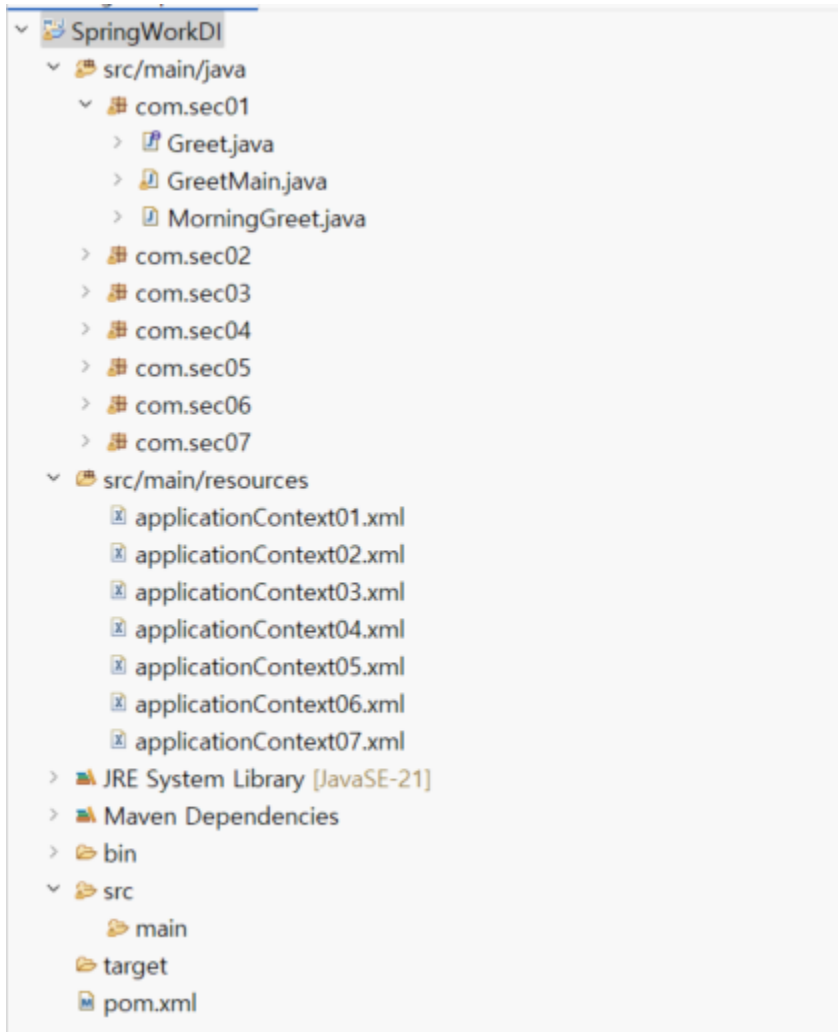

SpringWorkDI



문제 번호	패키지명	주요 주제
1 번	com.sec01	기본 인터페이스 DI (MorningGreet)
2 번	com.sec02	EveningGreet 클래스 DI
3 번	com.sec03	GreetService 에 Setter 주입
4 번	com.sec04	생성자 주입 방식 적용
5 번	com.sec05	GreetManager 중첩 DI 구조
6 번	com.sec06	동일 구조에서 ref 변경
7 번	com.sec07	모든 객체 외부 설정으로 구성

[문제 1] Greet 인터페이스와 MorningGreet 구현체를 정의하고 XML 로 주입하여 아침 인사를 출력하시오.

1. 클래스/메서드 요약

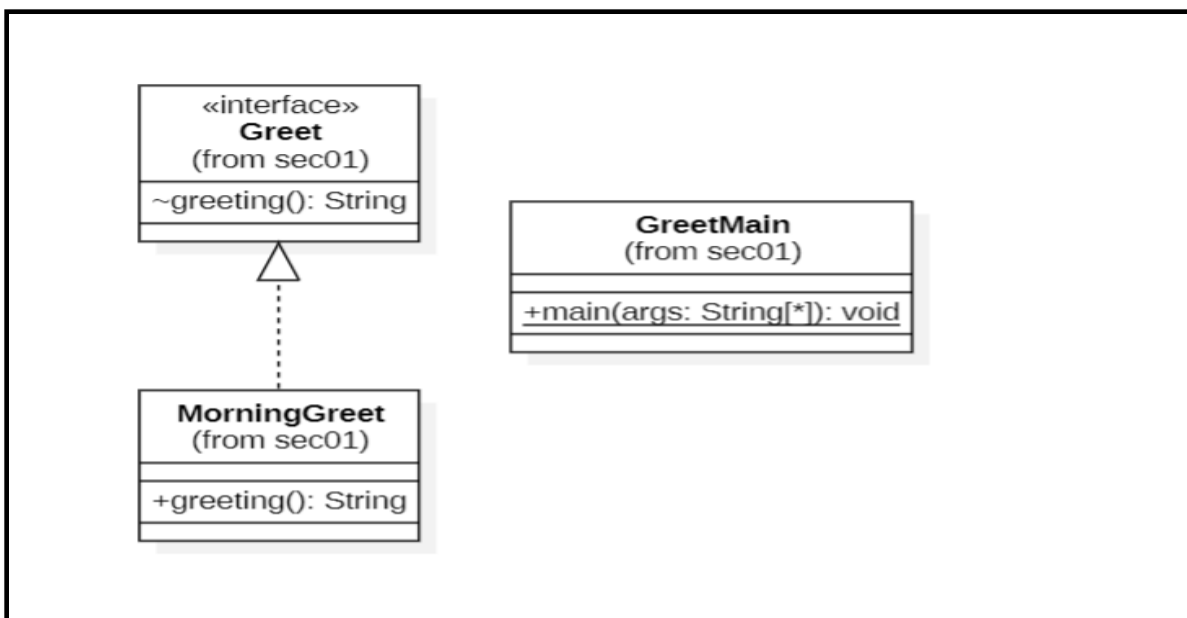
클래스명	메서드	설명
Greet	+greeting(): String	인사 인터페이스
MorningGreet	+greeting(): String	아침 인사 구현
GreetMain	+main(String[] args): void	ApplicationContext 로 Bean 주입

2. 실행 결과

좋은 아침입니다

3. applicationContext.xml

```
<bean id="morningGreet" class="com.used.MorningGreet"/>
```



[문제 2] EveningGreet 클래스를 추가하고 XML 에서 주입받아 출력하도록 수정하시오.

1. 클래스/메서드 요약

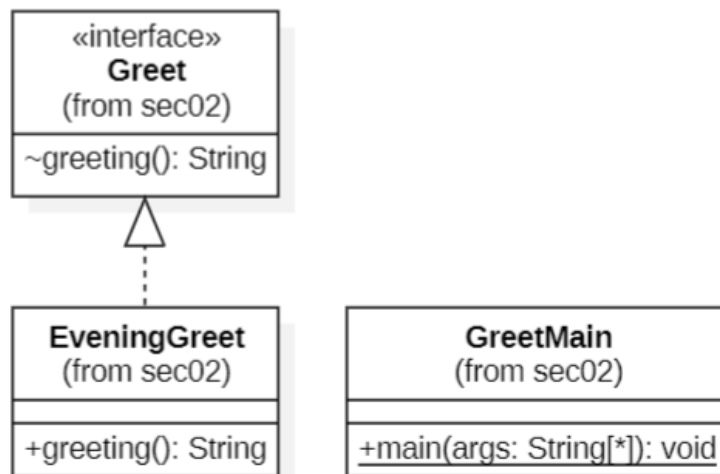
클래스명	메서드	설명
EveningGreet	+greeting(): String	저녁 인사 구현
GreetMain	+main(String[] args): void	빈 변경으로 출력 전환

2. 실행 결과

편안한 저녁입니다.

3. applicationContext.xml

```
<bean id="eveningGreet" class="com.used.EveningGreet"/>
```



[문제 3] GreetService 클래스를 새로 만들고, 내부에 Greet 을 갖도록 하여 Setter 로 주입하시오.

1. 클래스/메서드 요약

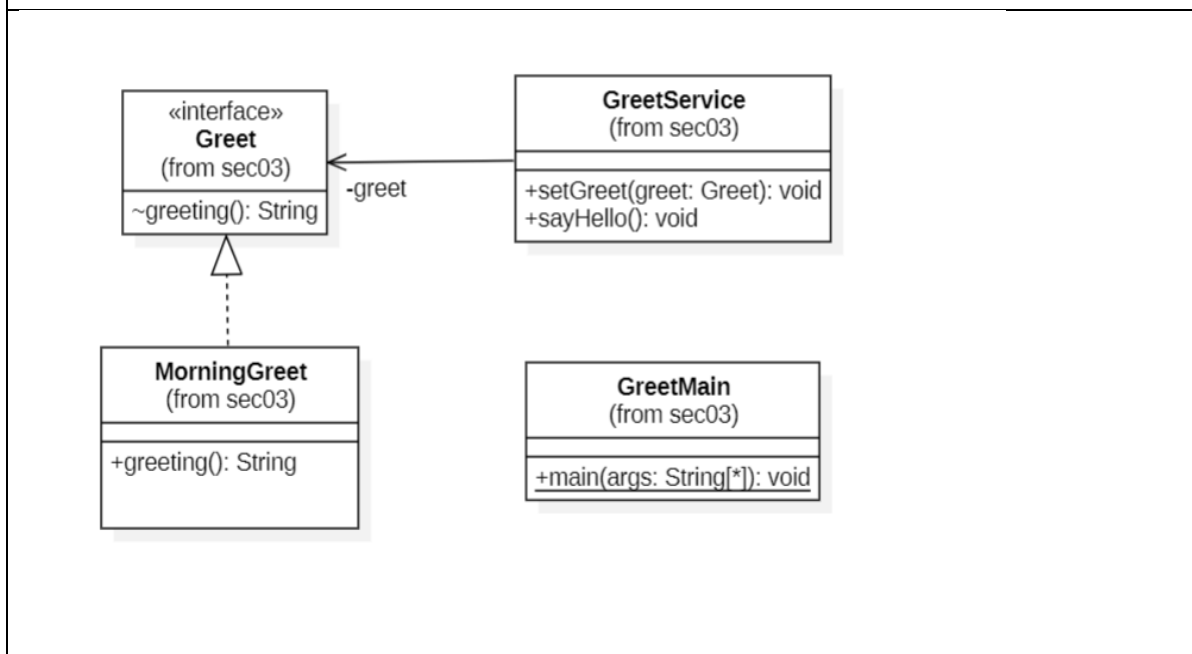
클래스명	메서드	설명
GreetService	+setGreet(Greet), +sayHello()	Setter 로 Greet 주입
applicationContext.xml		property 를 통한 Greet 주입 설정

2. 실행 결과

좋은 아침입니다

3. applicationContext.xml

```
<bean id="greetService" class="com.used.GreetService">
<property name="greet" ref="morningGreet"/>
</bean>
```



[문제 4] GreetService 를 생성자 방식으로 Greet 을 주입받는 구조로 바꾸시오.

1. 클래스/메서드 요약

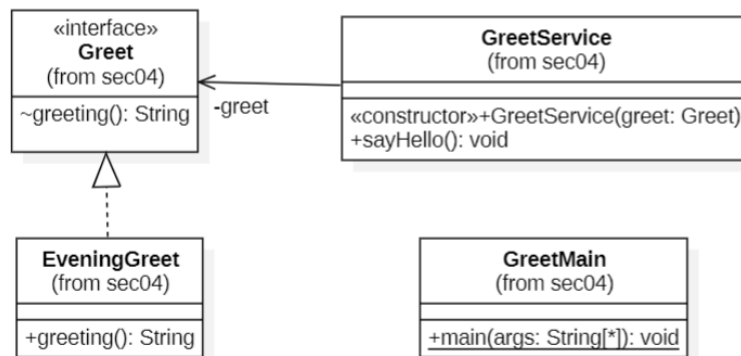
클래스명	메서드	설명
GreetService	+GreetService(Greet), +sayHello()	생성자 주입 구조

2. 실행 결과

안녕하세요 (저녁 인사)

3. applicationContext.xml

```
<bean id="greetService" class="com.used.GreetService">
  <constructor-arg ref="eveningGreet"/>
</bean>
```



[문제 5] GreetManager 클래스에서 GreetService 를 포함하고, 그 안에 Greet 이 포함된 구조로 중첩 DI 를 구성하시오.

1. 클래스/메서드 요약

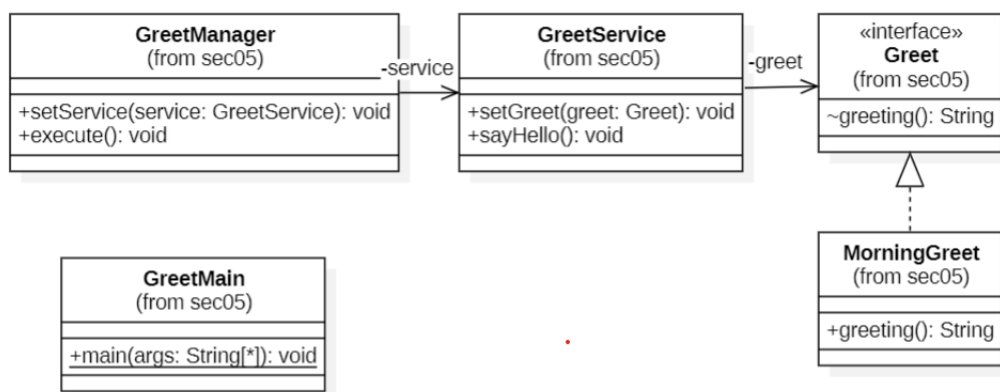
클래스명	메서드	설명
GreetManager	+setService(GreetService)	Has-A 구조
applicationContext.xml		greetService 를 GreetManager 에 주입

2. 실행 결과

좋은 아침입니다

3. applicationContext.xml

```
<bean id="greetManager" class="com.used.GreetManager">
  <property name="service" ref="greetService"/>
</bean>
```



[문제 6] GreetService 가 Greet 을 멤버로 가지도록 하고, XML 에서 여러 Greet 구현체를 조건에 따라 주입해보시오.

1. 클래스/메서드 요약

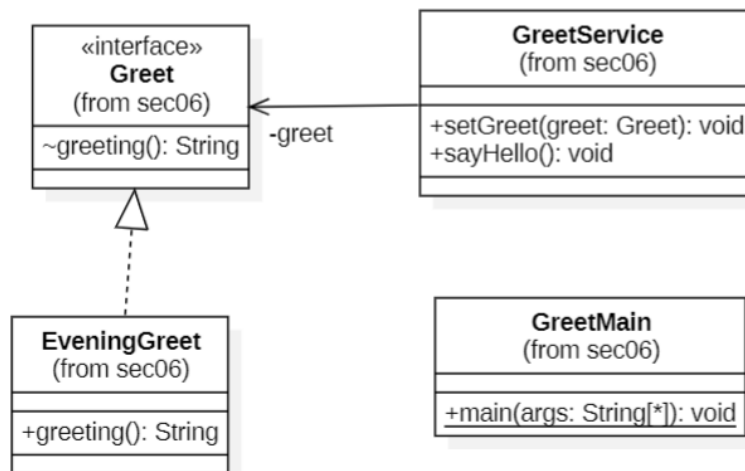
클래스명	메서드	설명
GreetService	+setGreet(Greet)	Setter 로 동적 주입
applicationContext.xml		동일 구조에서 다른 ref 주입

2. 실행 결과

안녕하세요 (저녁 인사)

3. applicationContext.xml

```
<bean id="greetService" class="com.used.GreetService">
  <property name="greet" ref="eveningGreet"/>
</bean>
```



[문제 7] 별도의 설정 파일 없이 Java 코드 내에서 직접 객체 생성 없이 XML 을 통해 객체 생성 및 주입되는 구조를 이해하고 구현하시오.

1. 클래스/메서드 요약

클래스명	메서드	설명
GreetMain	ApplicationContext 사용	XML 만으로 모든 객체 구성

2. 실행 결과

좋은 아침입니다

3. 예시 코드 GreetMain.java

```
ApplicationContext ctx = new
ClassPathXmlApplicationContext("applicationContext.xml");
Greet g = (Greet) ctx.getBean("morningGreet");
System.out.println(g.greeting());
```

