## JAVA기초입문과정

CHAPTER8 인터페이스

### Contents

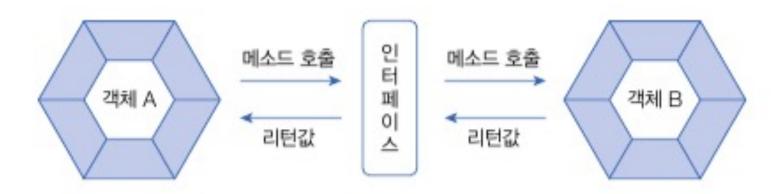


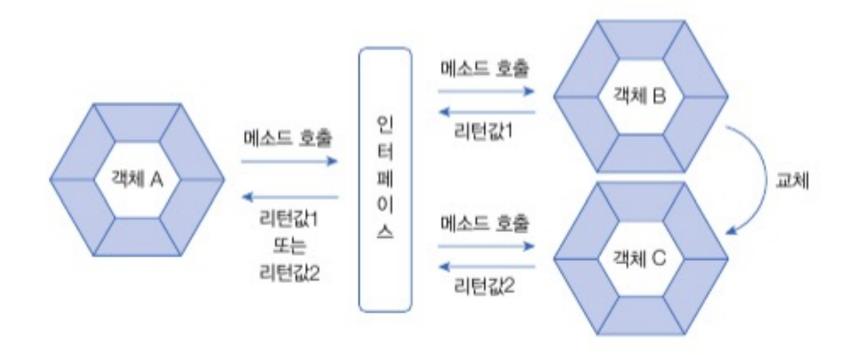




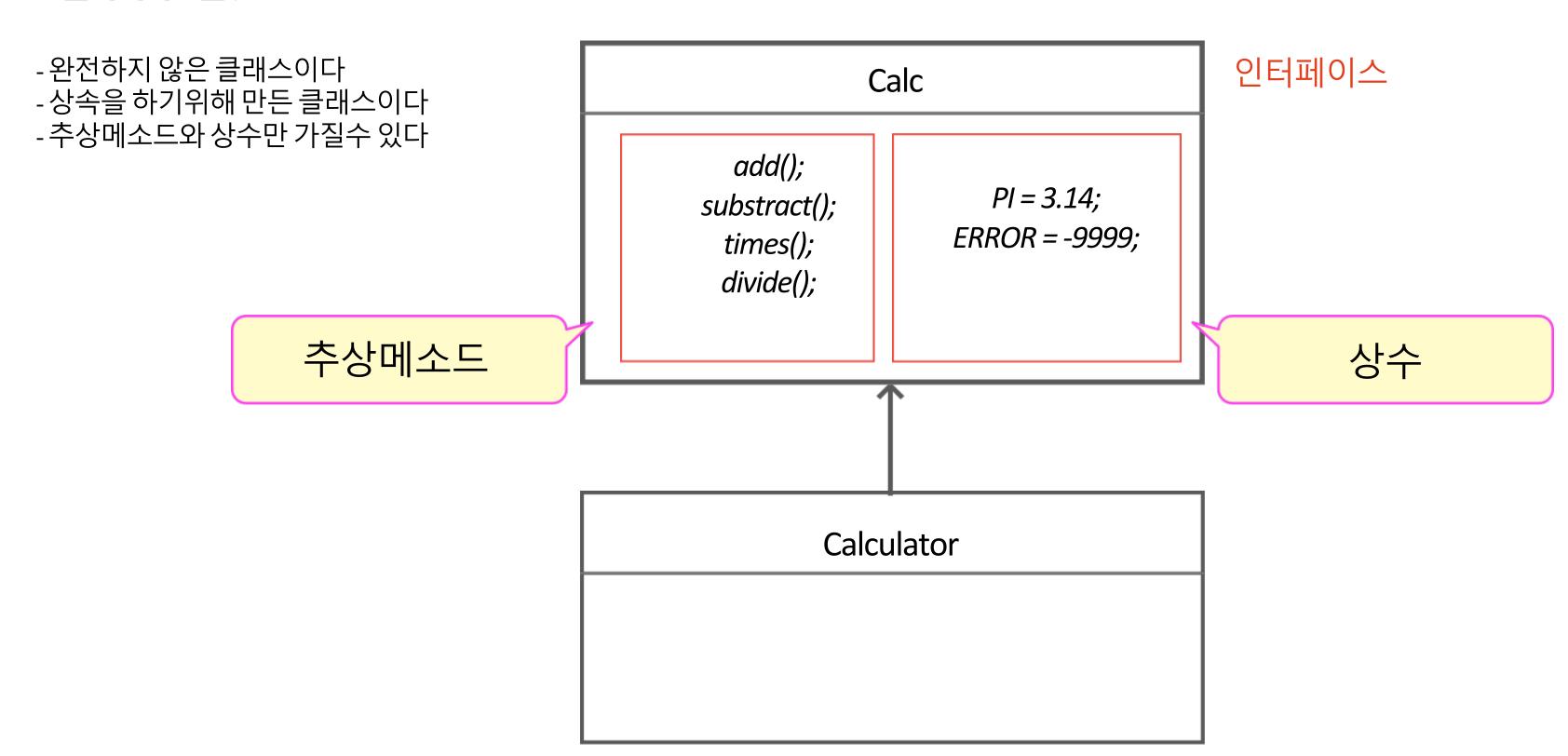


#### 1. 인터페이스란?



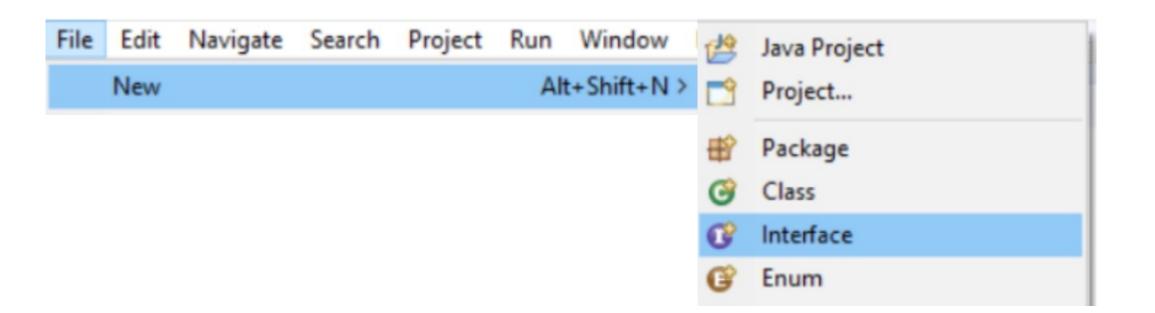


#### 1. 인터페이스란?

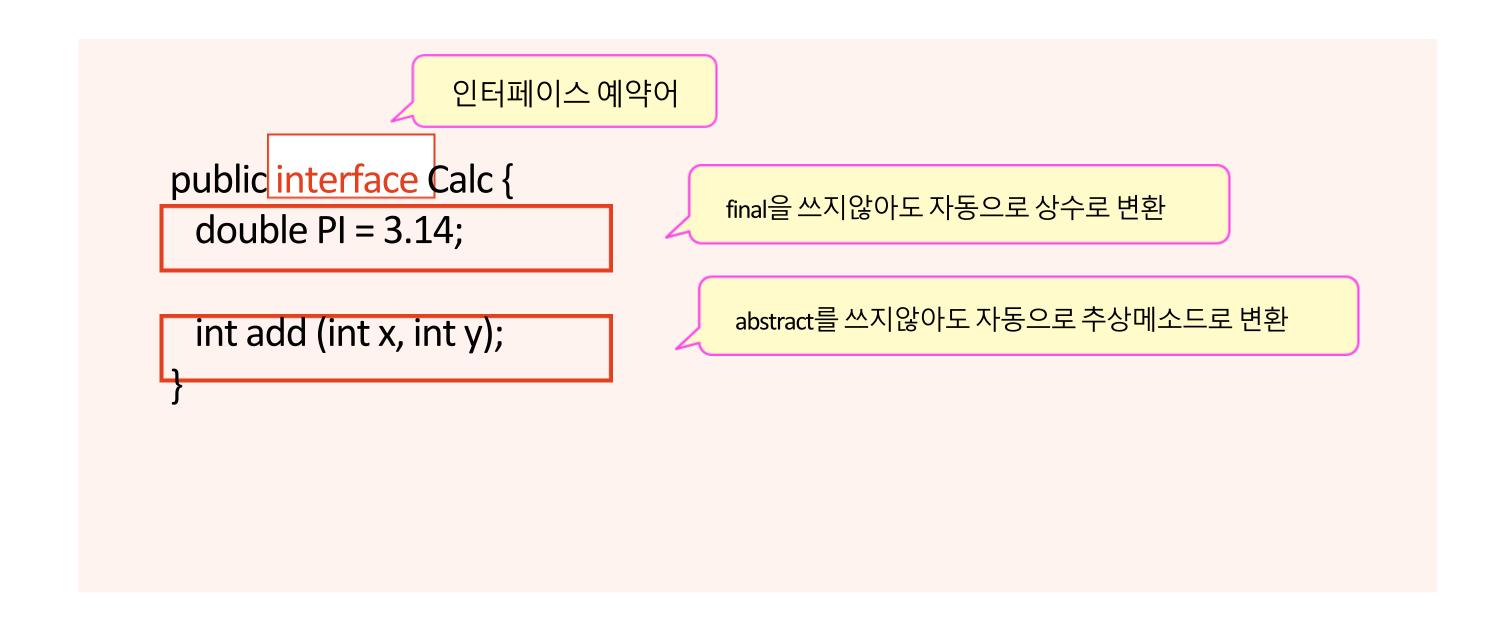


#### 2. 인터페이스 만들기

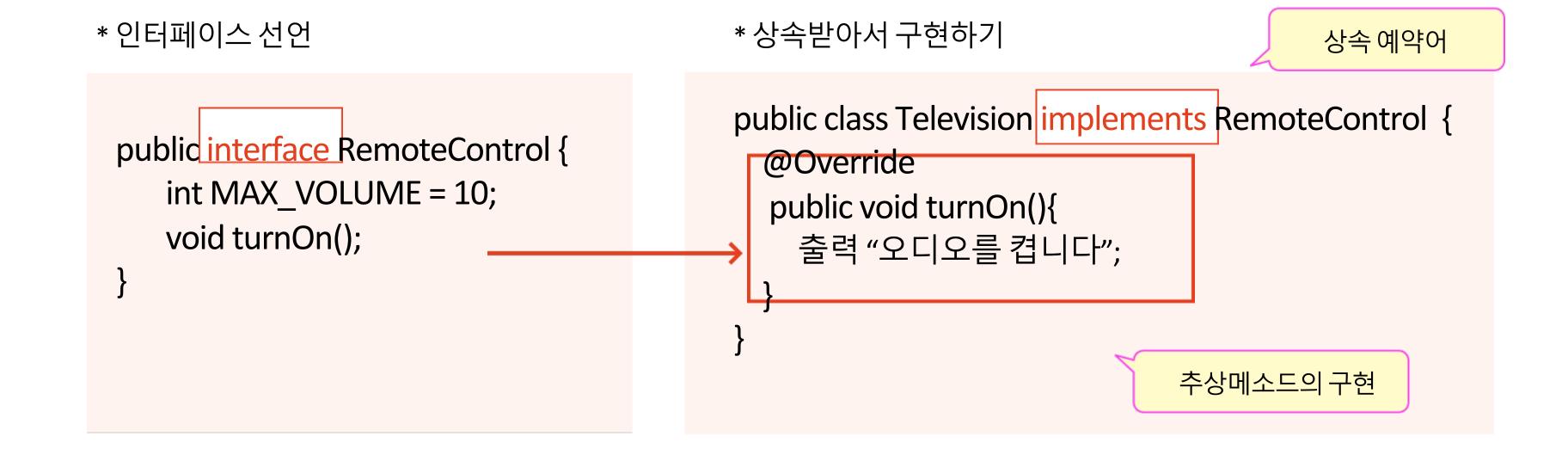
\* 이클립스에서 인터페이스 만들기 [ New -> Interface ] 클릭

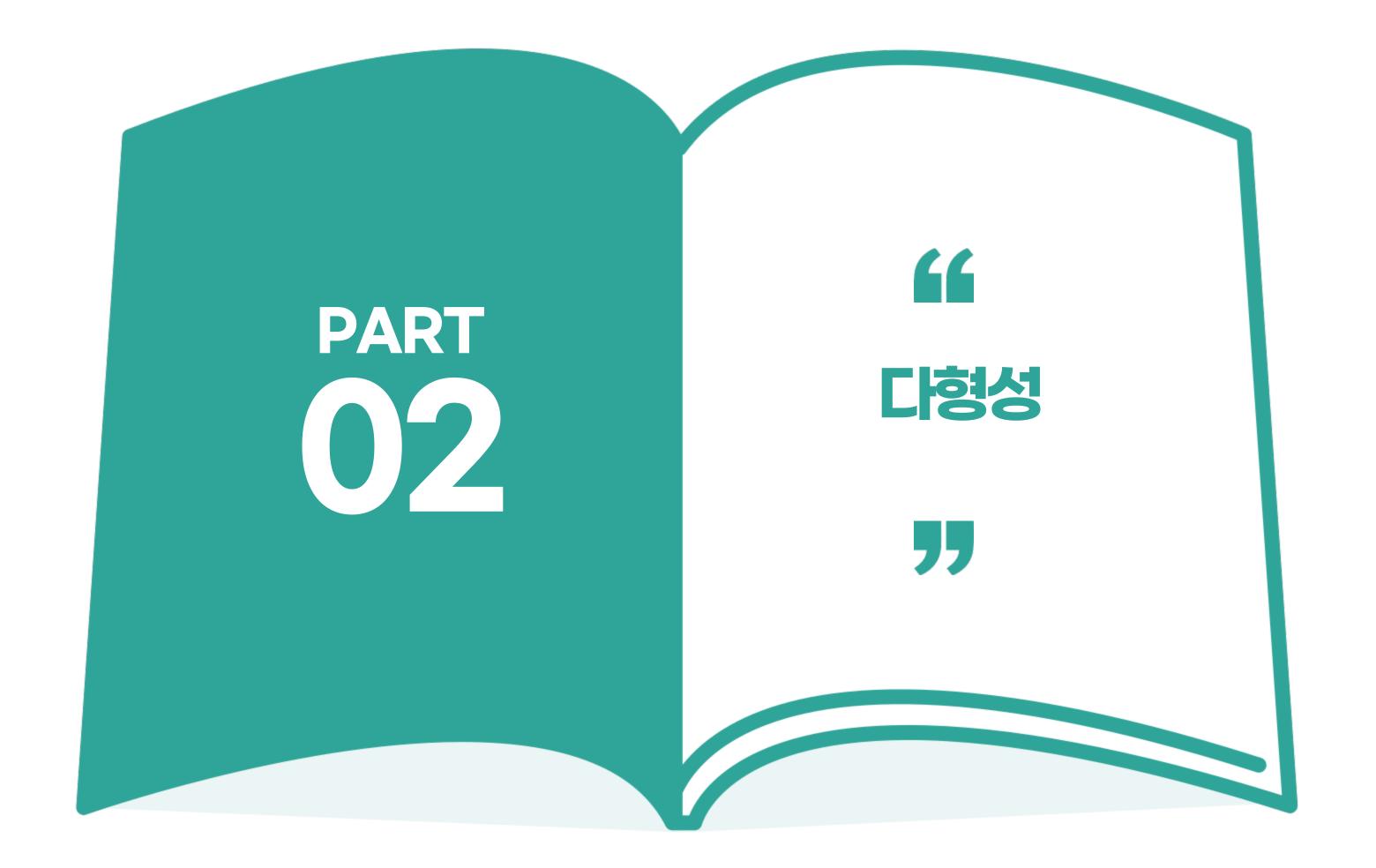


#### 3. 인터페이스 구현하기



#### 4. 클래스에서 인터페이스를 상속받아서 구현하기

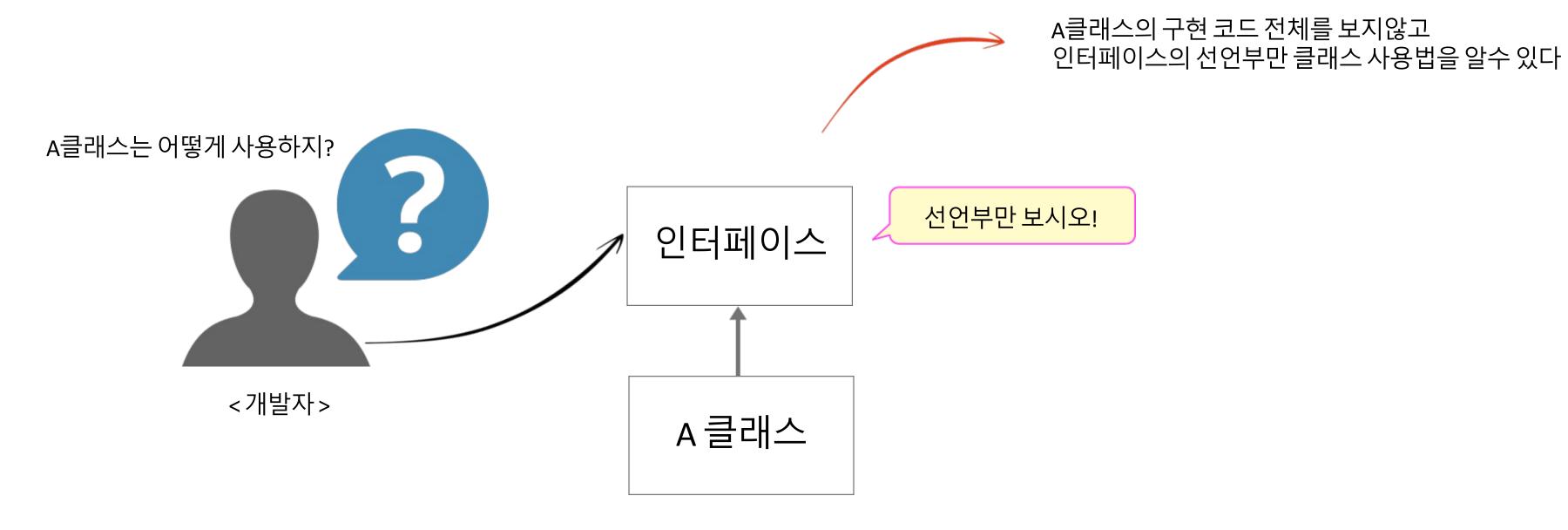




#### 1. 인터페이스 역할

껍데기만 있는 인터페이스는 왜 사용하는걸까?

인터페이스는 외부에 메소드 사용법을 미리 알려주는 역할을 한다

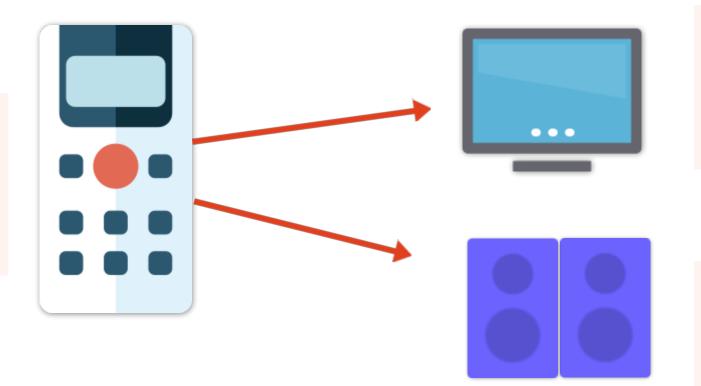


#### 2. 인터페이스와 다형성

기능사용법이 모두 동일하다

추상메소드

turnOn(); turnOff();



<인터페이스>

<구현클래스>

turnOn(); -> TV를 켯습니다 turnOff(); -> TV를 것습니다

turnOn(); -> Audio 를 켯습니다 turnOff(); -> Audio 를 것습니다

#### \* 예시1

상황 - 고객센터에서 전화상담하는 상담원들이 있다 고객센터에 전화가 오면 일단 대기열에 저장된다 상담원에게 전화를 배분하는 정책은 다음과 같이 여러 방식으로 구현할수있다



정책1. 순서대로 배분하기

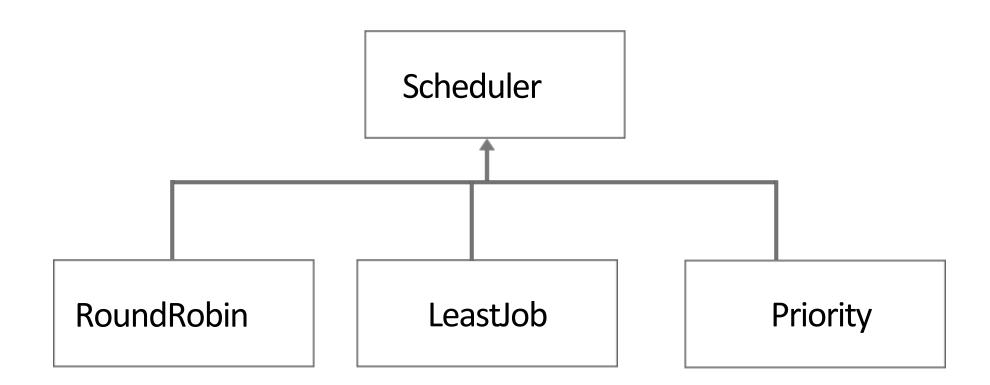


정책2. 짧은 대기열 찾아 배분하기



정책3. 우선순위에따라 배분하기

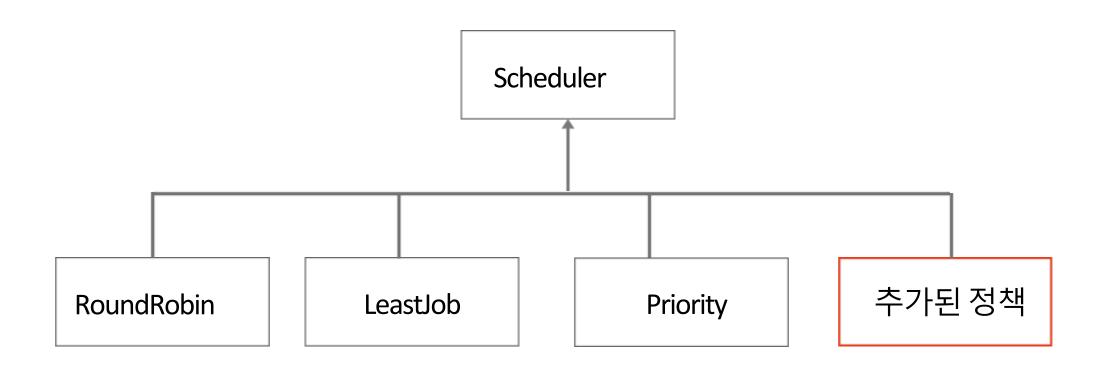
- 1. 정책 인터페이스 구현하기
- 2. 인터페이스를 상속받아 1번 정책 클래스 구현하기
- 3. 인터페이스를 상속받아 2번 정책 클래스 구현하기
- 4. 인터페이스를 상속받아 3번 정책 클래스 구현하기

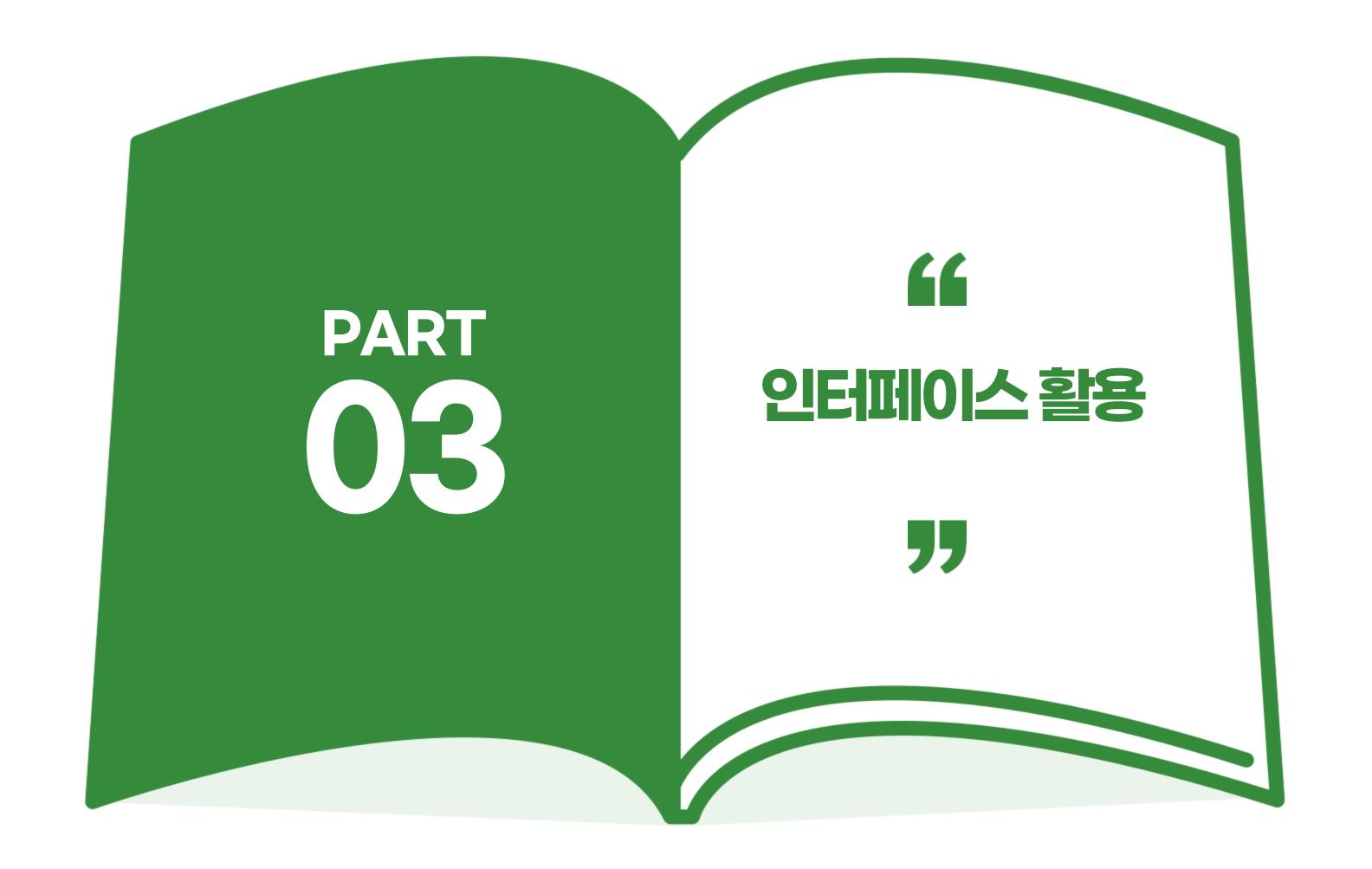


#### \*정리

- 상담 정화 배분 정책은 언제든지 바뀔 수 있다
- 이런경우에 인터페이스와 다형성을 사용하면 클래스를 사용하는 쪽에서는 쉽게 바꿀수 있다

```
-i i = new RoundRobin();
                     인스턴스만 교체
li = new LeastJob();
i.method1();
                 메소드 사용은 수정이 필요없음
i.method2();
```

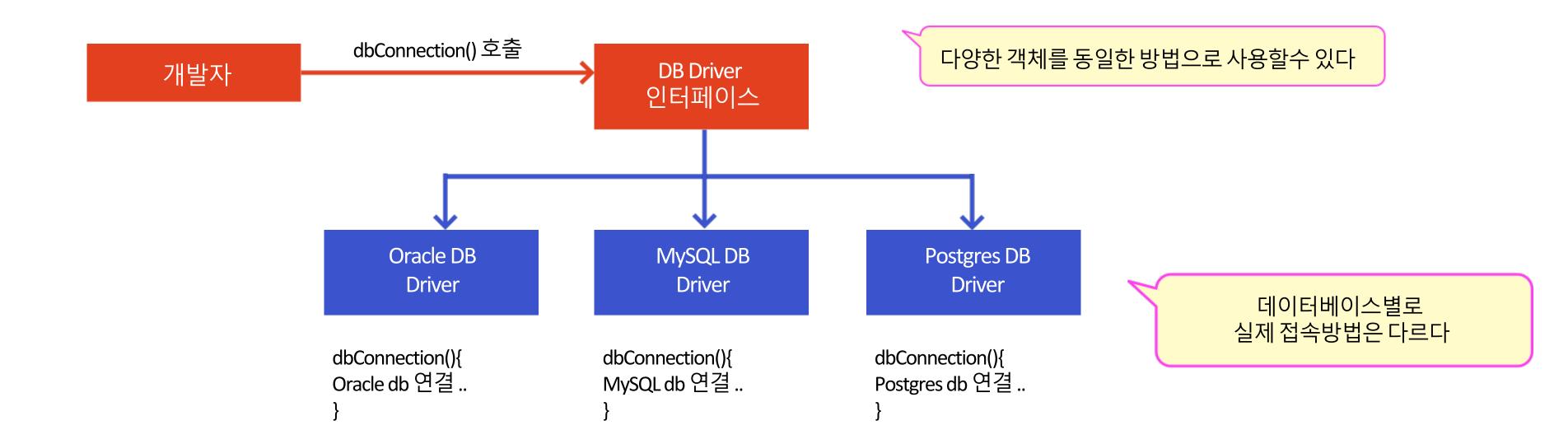




#### 인터페이스 활용

#### \* 예시2

상황 - 어느 회사에서 시스템을 개발했다 이 시스템은 자료를 저장하기 위해 데이터베이스를 사용한다 처음에는 MySQL을 사용하다가, 다른회사에 가서 시스템을 설치하려고하니 Oracle 데이터베이스에 저장해달라고 요청을 받았다



# Thank You!