

# 프로그래밍 언어 응용

chapter10

## 인터페이스

제공된 자료는 훈련생의 수업을 돕기 위한 것으로, 타인과 공유하시면 안됩니다.

# Contents

part.1

인터페이스

part.2

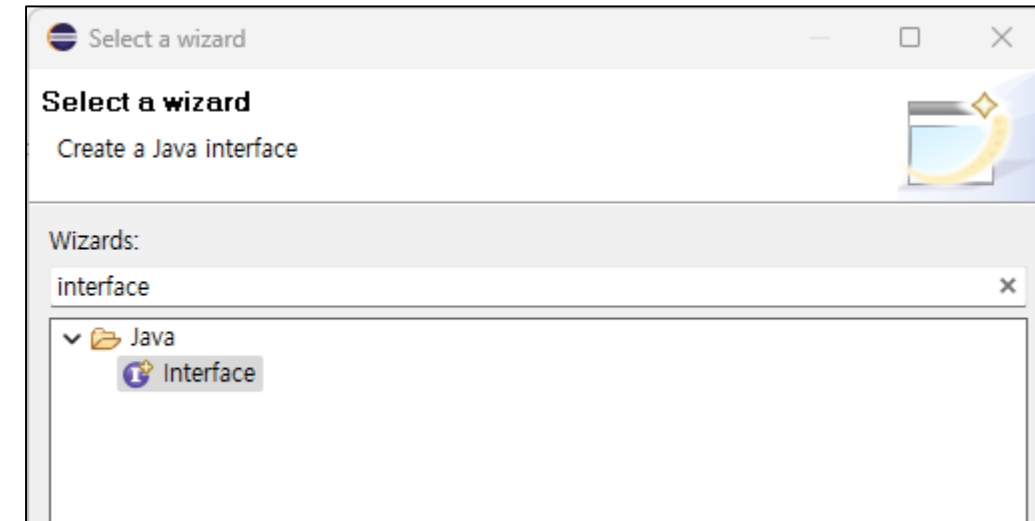
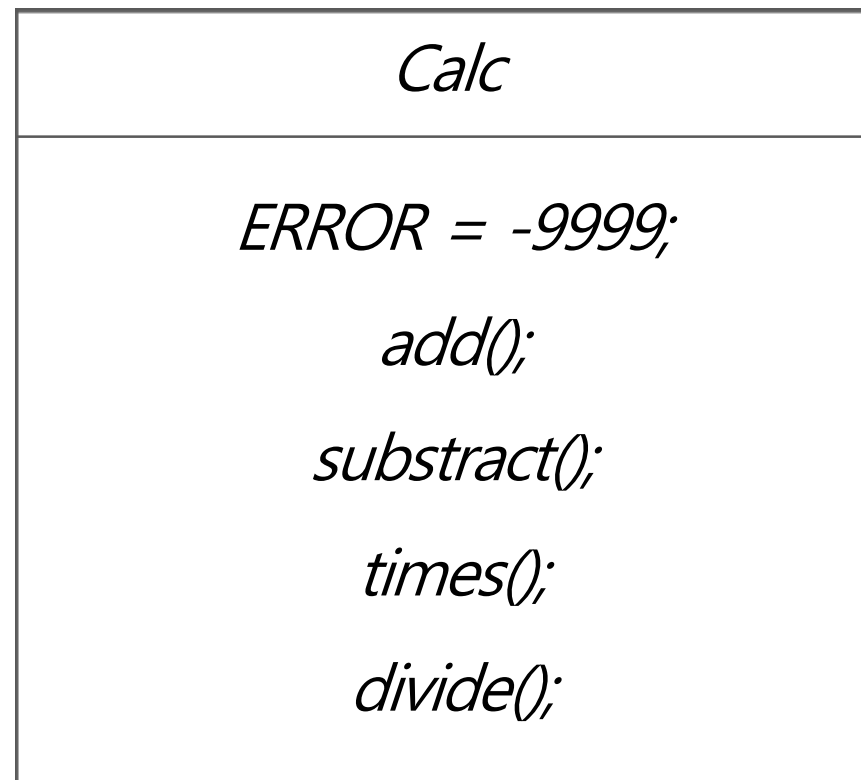
다형성

part.1

인터페이스 활용하기

## 인터페이스란?

- 추상메소드와 상수로 이루어진 불완전한 클래스이다.



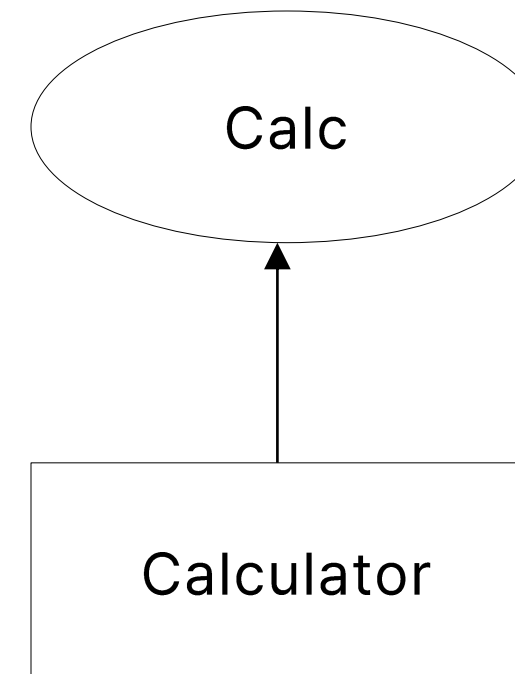
### 인터페이스의 특징

- 인터페이스를 선언할 때 "interface" 키워드를 사용한다.
- 인터페이스에서 변수를 선언하면 final이 없어도 자동으로 상수가 된다.
- 인터페이스에서 메소드를 선언하면 abstract이 없어도 자동으로 추상메소드가 된다.

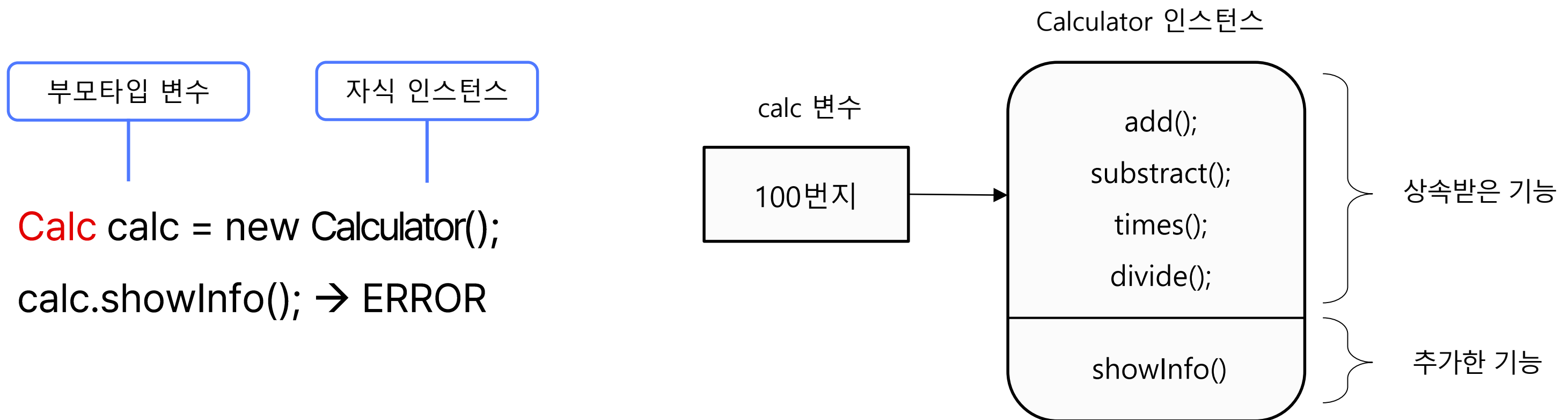
```
interface Calc {  
    int ERROR = -9999;           //final int ERROR;  
    int add (int n1, int n2);     //abstract int add();  
    int subtract(int n1, int n2);  
    int times(int n1, int n2);  
    int divide(int n1, int n2);  
}
```

인터페이스를 상속받을 때 “implements” 키워드를 사용한다.  
자식클래스는 상속받은 모든 추상메소드를 구현해야 한다.

```
class Calculator implements Calc {  
    @Override  
    int add(int num1, int num2) {  
        return num1 + num2;  
    }  
    ...  
}
```

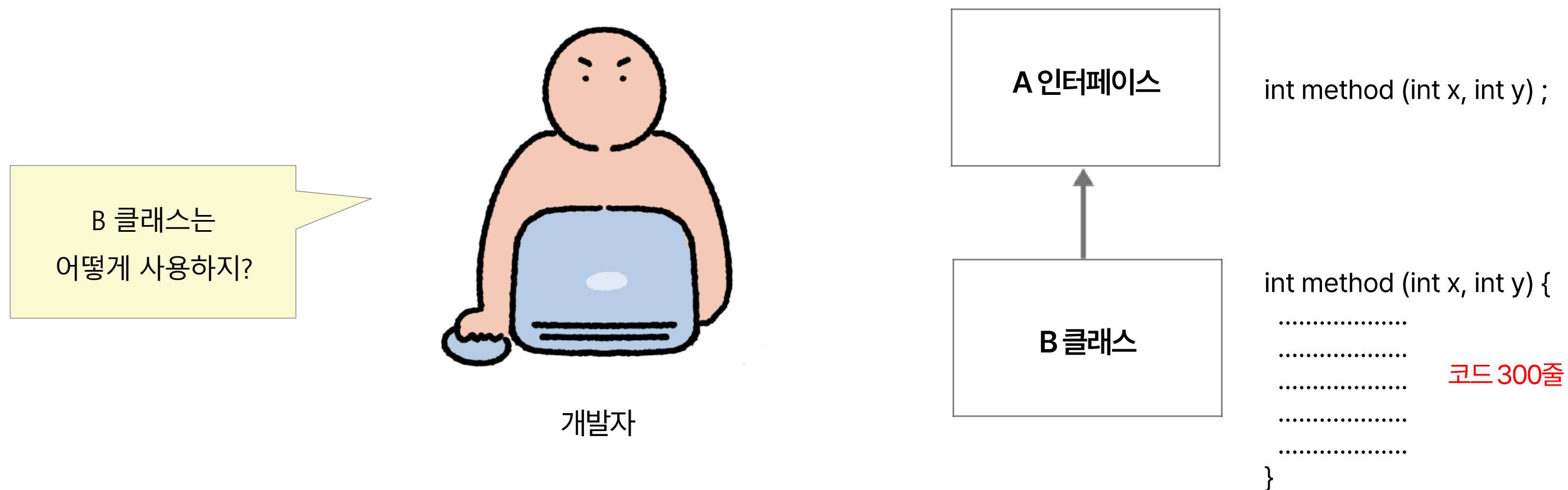


부모 인터페이스 변수에 자식 객체를 할당하여 형변환 할 수 있다.  
자식클래스에서 추가한 속성이나 기능은 사용할 수 없다.

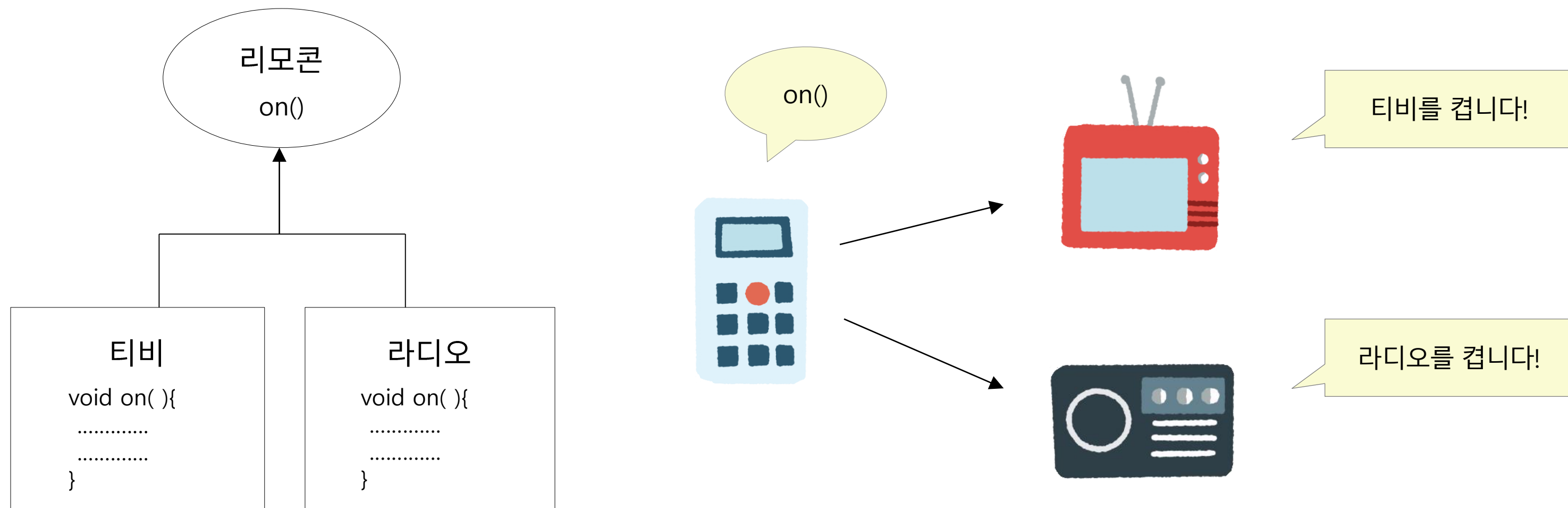


인터페이스는 다른 클래스를 만들기 위해 작성한다.

인터페이스의 추상 메소드를 보면, 인터페이스를 구현한 클래스의 사용법을 알 수 있다.

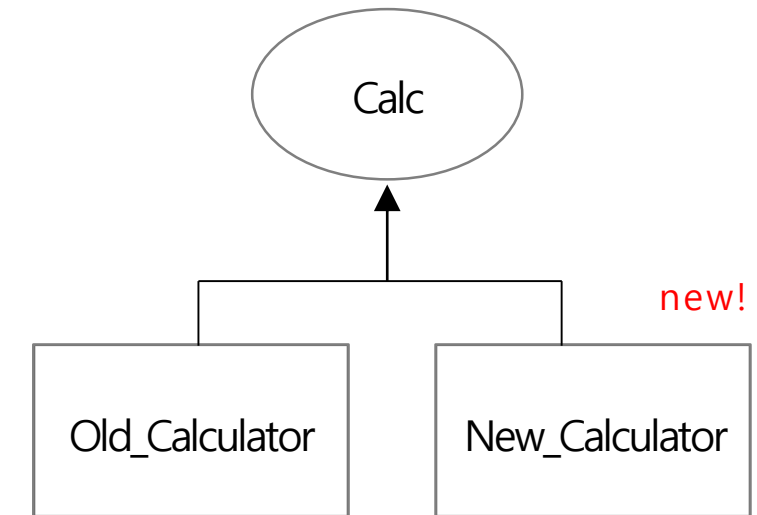


인터페이스는 여러 클래스를 동일한 방식으로 사용할 수 있도록 도와주는 중간 다리 역할을 한다.  
같은 인터페이스로 구현한 자식 객체들은 같은 추상 메소드를 구현했기 때문에, 사용방법이 모두 같다.



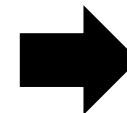


인터페이스를 사용하지 않은 경우에는, 특정 클래스에 직접 의존을 하기 때문에, 코드를 변경하거나 다른 클래스로 교체하기가 어렵다.  
예를 들어 계산기 클래스의 객체를 생성하여 연산을 수행하면, OldCalculator 클래스에 직접 의존하고 있어서, 다른 계산기 클래스로 교체하려면 코드를 많이 수정해야 한다.



```
OldCalculator oldCalculator = new OldCalculator();
```

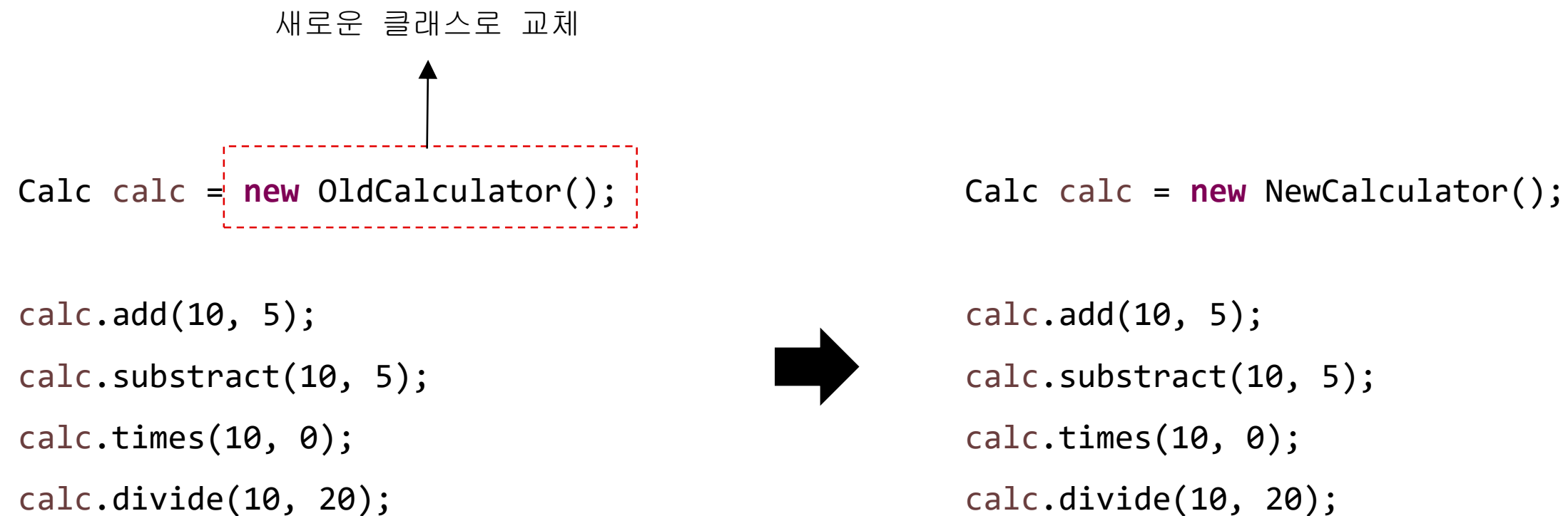
```
oldCalculator.addmethod(10, 5);  
oldCalculator.subtractmethod(10, 5);  
oldCalculator.timesmethod(10, 0);  
oldCalculator.dividemethod(10, 20);
```



```
NewCalculator newCalculator = new NewCalculator();
```

```
newCalculator.addfunction(10, 5);  
newCalculator.subtractfunction(10, 5);  
newCalculator.timesfunction(10, 0);  
newCalculator.dividefunction(10, 20);
```

클래스 대신 인터페이스에 의존하도록 설계하면, 코드를 변경하거나 다른 클래스로 교체하기 쉽다.  
예를 들어 계산기 인터페이스를 사용하여 연산을 수행하면, 계산기의 인스턴스를 생성하는 부분만 변경하면 된다.



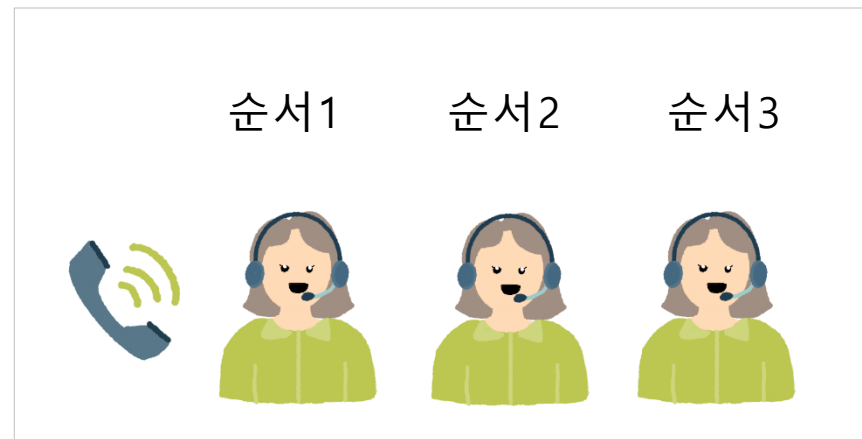
# 인터페이스 활용하기    고객 전화 상담 프로그램 구현하기

고객센터에는 전화 상담을 하는 상담원들이 있습니다.

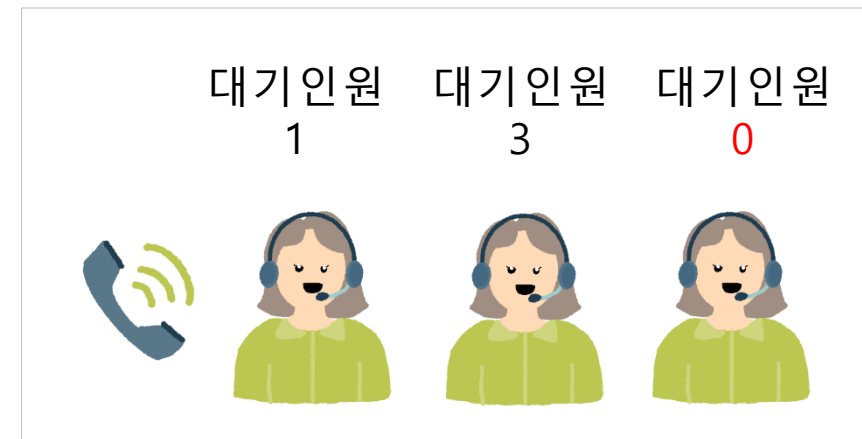
고객센터로 전화가 오면 일단 대기열에 저장하고, 전화를 상담원에게 전달합니다.

전화를 상담원에게 나눠주는 정책은 다음과 같이 3가지가 있습니다.

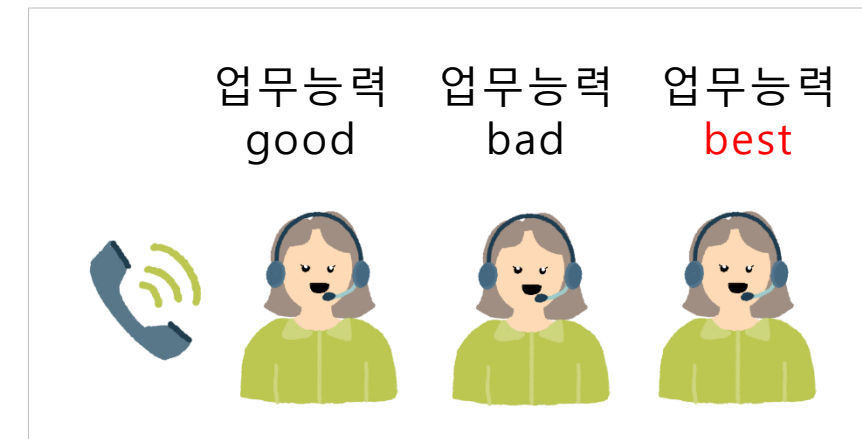
1. 전화 순서대로 상담원에게 하나씩 전달한다.
2. 상담 대기가 짧은 상담원에게 전화를 전달한다.
3. 등급이 높은 고객의 전화를 업무능력이 좋은 상담원에게 전달한다.



순서대로 전달

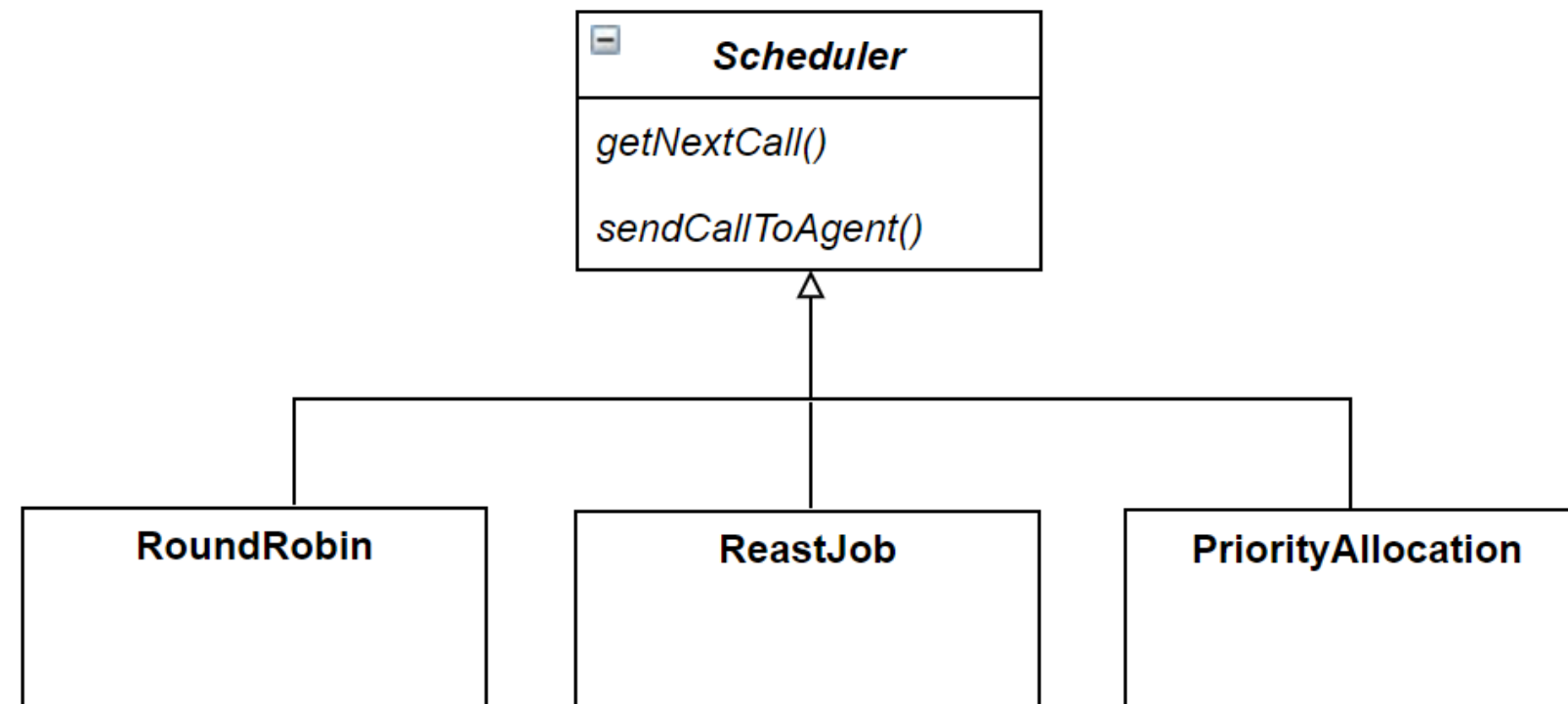


짧은 대기열 찾아서 전달



우선순위에 따라 전달

1. 상담원에게 전화 업무를 배분하는 기능을 구현하기 위해 Scheduler 인터페이스를 만든다.
2. Scheduler 인터페이스를 상속받는 3개의 클래스를 만든다.
3. 3개의 클래스를 정책에 맞게 구현한다.

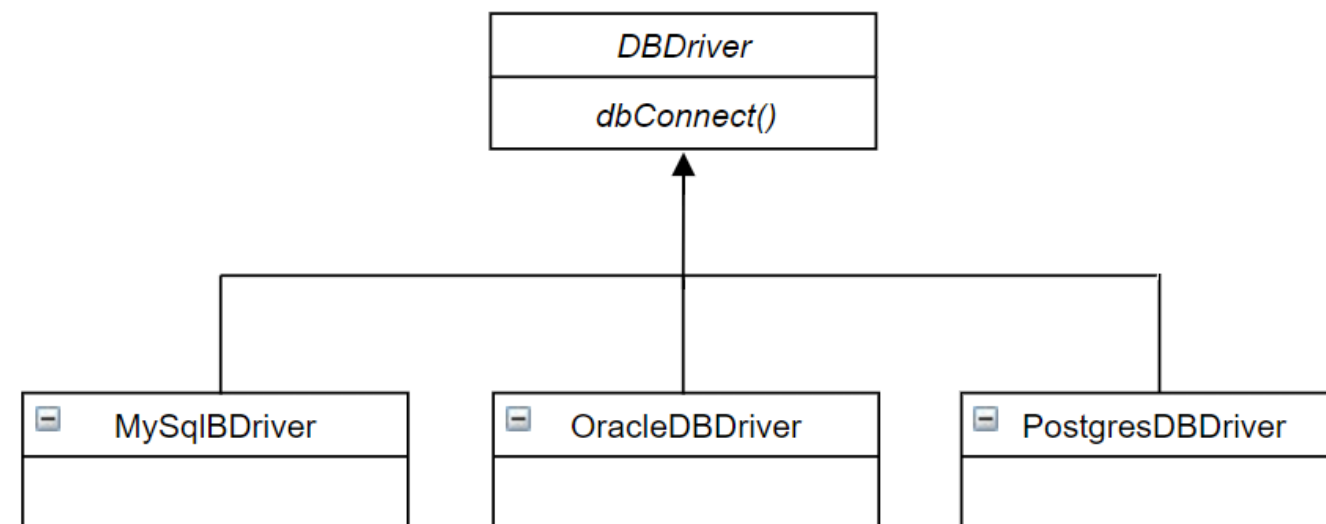


## Q. 2번 문제

아래 그림은 DBDriver 인터페이스와 하위 클래스의 상속관계를 표현한 것입니다.

세클래스에는 데이터베이스 연결하는 기능이 정의되어 있습니다.

1을 입력하면 MySQL, 2를 입력하면 Oracle, 3을 입력하면 Postgres가 연결되도록 클래스를 구현하세요.



드라이버를 선택하세요.

1 : mysql  
2 : oracle  
3 : postgres

1

MySql DB에 연결합니다.

## 1을 입력한 경우