

JAVA Programming

Practice 4: Abstract class and interface

BeomSeok Kim

Department of Computer Engineering
KyungHee University
passion0822@khu.ac.kr

[실습 4-1] 추상클래스와 상속



```
package javabook.ch05;

public class RoadBike extends AbstractBicycle {

    void prtInfo( ) {
        System.out.println("ID : " + id);
        System.out.println("Brand : " + brand);
    }

    public static void main(String[] args) {
        RoadBike rb = new RoadBike( );
        rb.id = 300;
        rb.brand = "LOOK";
        rb.prtInfo( );
    }
}
```

```
public abstract class AbstractBicycle {
    // 멤버변수 선언
    int id;
    String brand;

    // 추상 메소드 선언
    abstract void prtInfo();

    // 일반 메소드 구현
    public String getBrand() {
        return brand;
    }
}
```

```
Console X
ID : 300
Brand : LOOK
```

[실습 4-2] 인터페이스



```
package javabook.ch05;

public interface IBicycle {

    int FrameSize_L = 18;
    int FrameSize_M = 16;
    int FrameSize_S = 15;

    abstract void prtInfo( );

}
```

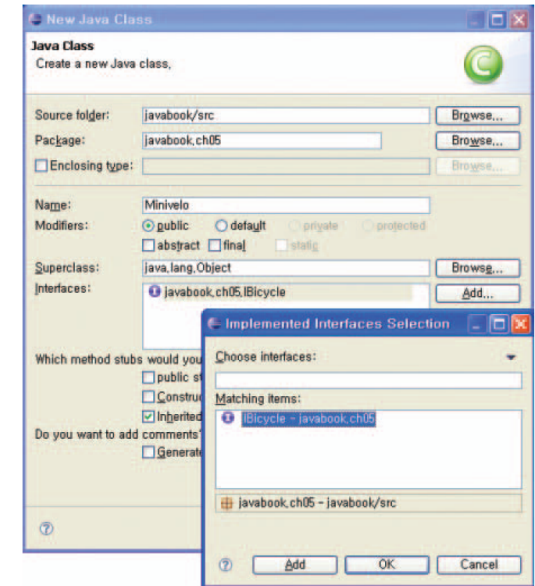
[실습 4-2] 인터페이스

```
package javabook.ch05;

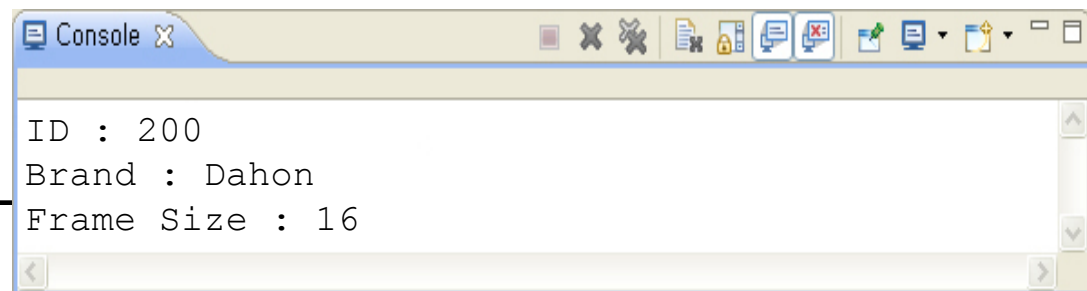
public class Minivelo implements IBicycle {
    int id;
    String brand;

    public void prtInfo( ) {
        System.out.println("ID : " + id);
        System.out.println("Brand : " + brand);
        System.out.println("Frame Size : " + IBicycle.FrameSize_M);
    }

    public static void main(String[] args) {
        Minivelo mv = new Minivelo( );
        mv.id = 200;
        mv.brand = "Dahon";
        mv.prtInfo( );
    }
}
```



[그림 5-1] 인터페이스 구현 추가



[실습 4-3] 다중인터페이스 구현



```
package javabook.ch05;

public class MotorBicycle implements IBicycle, IMotor {

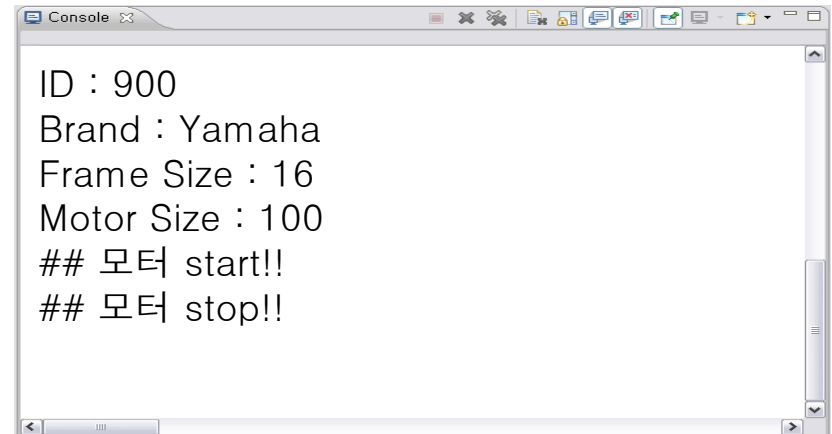
    int id;
    String brand;

    public void prtInfo( ) {
        System.out.println("ID : " + id);
        System.out.println("Brand : " + brand);
        System.out.println("Frame Size : " + IBicycle.FrameSize_M);
        System.out.println("Motor Size : " + IMotor.Motor_Size_Large);
    }

    public void start( ) {
        System.out.println("## 모터 start!!");
    }

    public void stop( ) {
        System.out.println("## 모터 stop!!");
    }

    public static void main(String[ ] args) {
        MotorBicycle mb = new MotorBicycle( );
        mb.id = 900;
        mb.brand = "Yamaha";
        mb.prtInfo( );
        mb.start( );
        mb.stop( );
    }
}
```



[과제] 악기연주프로그램

- 악기를 연주하는 프로그램을 만들어 피아노, 플룻 등을 연주하는 프로그램을 작성한다.
 - ✓ Instrument 클래스를 추상 클래스로 정의
 - play, volumeUp, volumeDown 메소드를 추상메소드로 정의
 - ✓ Instrument 클래스를 상속받는 Piano 클래스와 Flute 클래스 정의
 - 위에 명시된 추상 메소드를 구현
 - System.out.println()을 이용해 간단한 메시지를 아래 예제와 같이 출력하도록 구현

```
Console X
Piano : volumeUp( ) 호출됨
Piano : volumeDown( ) 호출됨
Piano : play( ) 호출됨
```

```
Console X
Flute : volumeUp( ) 호출됨
Flute : volumeDown( ) 호출됨
Flute : play( ) 호출됨
```

Thank You!
Q&A