

JAVA Programming

Practice 3: JAVA and object-oriented

BeomSeok Kim

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

[실습 3-1] 참조변수와 객체

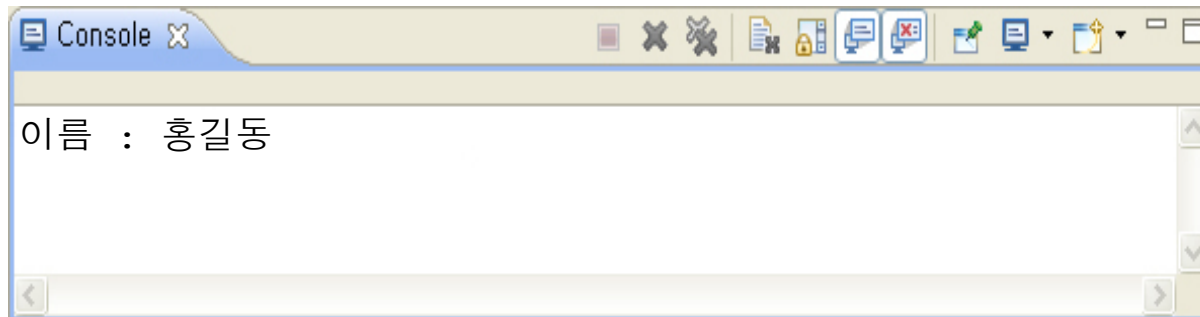
```
package javabook.ch04;

public class ObjExam1 {

    String name;

    public void prtName( ) {
        System.out.println("이름 : " + name);
    }

    public static void main(String[ ] args) {
        ObjExam1 obj1 = new ObjExam1( );
        obj1.name = "홍길동";
        obj1.prtName( );
    }
}
```



[실습 3-2] 생성자

```
package javabook.ch04;

public class ObjExam2 {
    String name;
    int num;

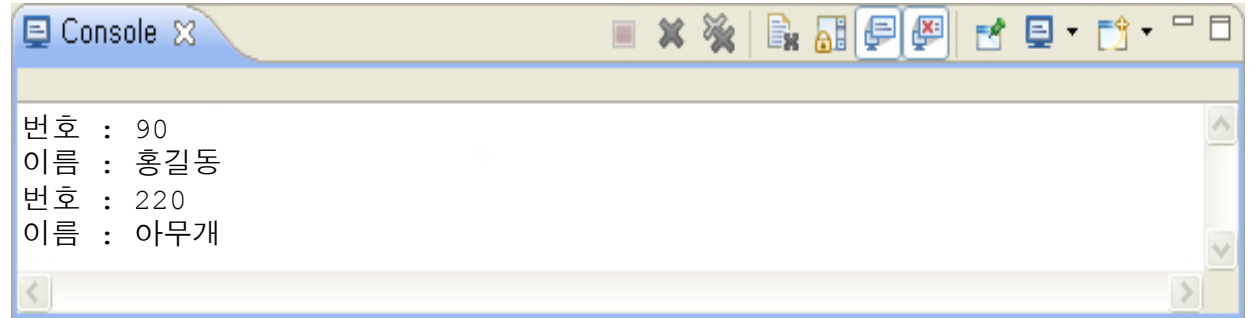
    public ObjExam2( ) {
        name = "홍길동";
        num = 90;
    }

    public ObjExam2(int num, String name) {
        this.num = num;
        this.name = name;
    }

    public void prtVars( ) {
        System.out.println("번호 : " + num);
        System.out.println("이름 : " + name);
    }

    public static void main(String[ ] args) {
        ObjExam2 obj1 = new ObjExam2( );
        ObjExam2 obj2 = new ObjExam2(220, "아무개");

        obj1.prtVars( );
        obj2.prtVars( );
    }
}
```



[실습 3-3] 생성자-응용

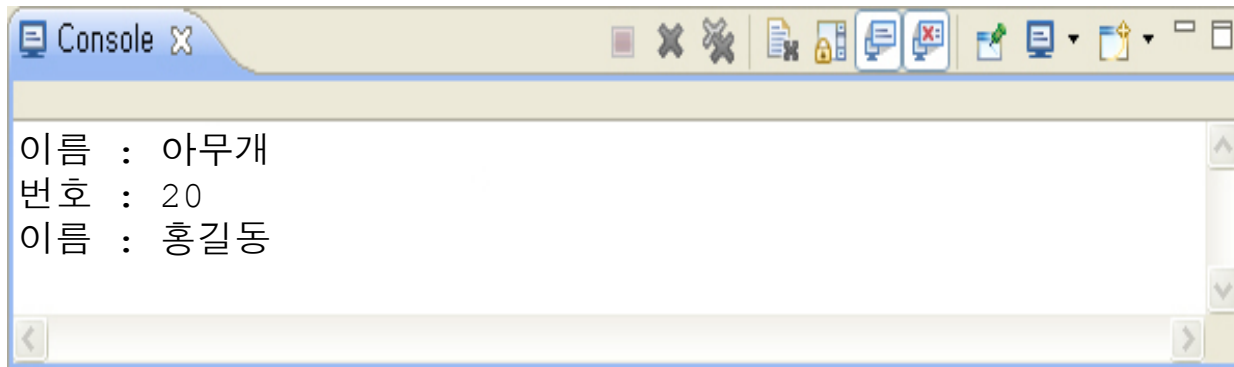
```
package javabook.ch04;

public class Extra4_1 {

    public static void main(String[] args) {
        ObjExam1 obj1 = ( );
        ObjExam2 obj2 = new ObjExam2(20, "홍길동");

        obj1.( ) = "아무개";

        obj1.prtName( );
        ( ).prtVars( );
    }
}
```



[실습 3-4] 상속

```
package javabook.ch04;

public class Bicycle {
    int id;
    String brand;

    public Bicycle(int id, String brand) {
        this.id = id;
        this.brand = brand;
    }
}
```

```
package javabook.ch04;

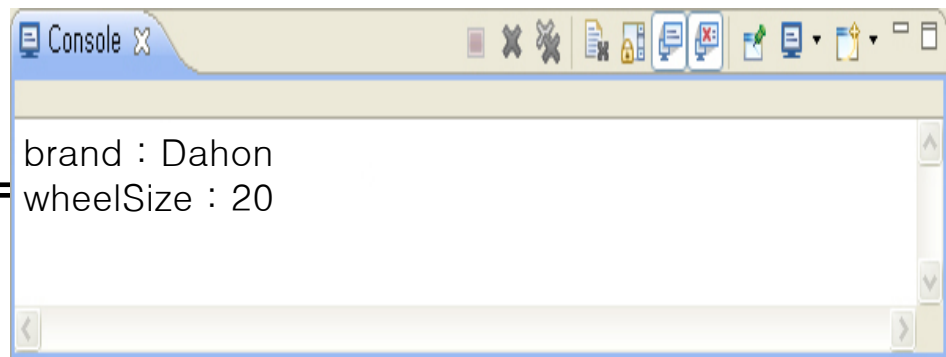
public class Extra4_2 extends Bicycle {
    (redacted);

    public void prtInfo( ) {
        System.out.println("brand : " + brand);
        System.out.println("wheelSize : " + wheelSize);
    }

    public Extra4_2(int id, String brand) {
        super(id, brand);
    }

    public static void main(String[ ] args) {
        Extra4_2 mv = new Extra4_2(20091098, "Dahon");
        (redacted) = 20;

        mv.prtInfo( );
    }
}
```



[실습 3-5] 메소드 오버로딩

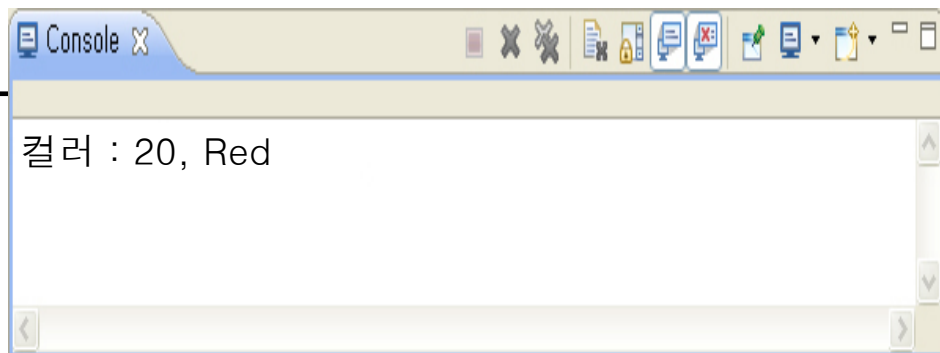
```
package javabook.ch04;
public class OverLoadingExam {

    int color;
    String strColor;

    void setColor(int color) {
        this.color = color;
    }

    void setColor(String color) {
        this.strColor = color;
    }

    public static void main(String[ ] args) {
        OverLoadingExam ole = new OverLoadingExam( );
        ole.setColor(20);
        ole.setColor("Red");
        System.out.println("컬러 : " + ole.color + ", " + ole.strColor);
    }
}
```



[실습 3-6] 메소드 오버로딩-응용

```
package javabook.ch04;
```

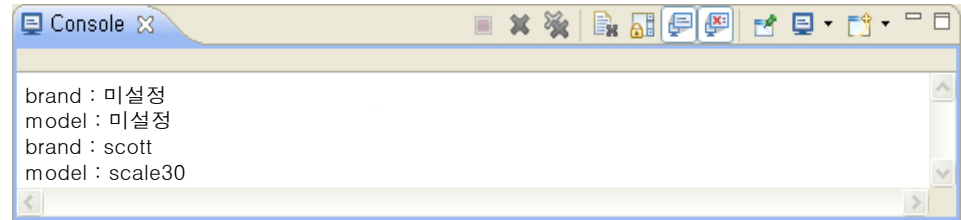
```
public class Extra4_3 {  
    String brand;  
    String model;
```

```
    public void prtInfo( ) {  
        System.out.println("brand : " + brand);  
        System.out.println("model : " + model);  
    }
```

```
    public Extra4_3( ) {  
        brand = "미설정";  
        model = "미설정";  
    }
```

```
    public Extra4_3(String brand, String model) {  
        this.brand = brand;  
        this.model = model;  
    }
```

```
    public static void main(String[ ] args) {  
        Extra4_3 e1 = (                  );  
        Extra4_3 e2 = (                                  );  
  
        e1.prtInfo( );  
        e2.prtInfo( );  
    }  
}
```

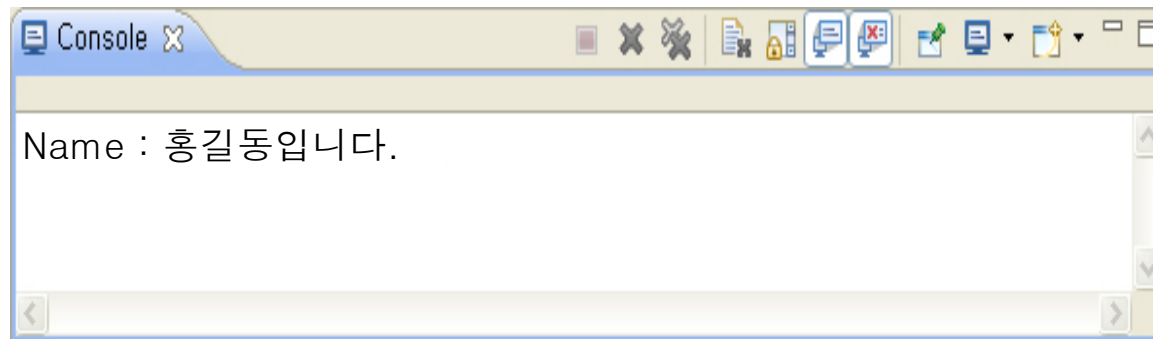


[실습 3-5] 메소드 오버라이딩

```
package javabook.ch04;

public class Extra4_4 extends ObjExam1 {
    ( ) {
        System.out.println("Name : " + name + "입니다.");
    }

    public static void main(String[ ] args) {
        Extra4_4 e = ( );
        ( ) = "홍길동";
        e.prtName( );
    }
}
```



[실습 3-6] 래퍼클래스

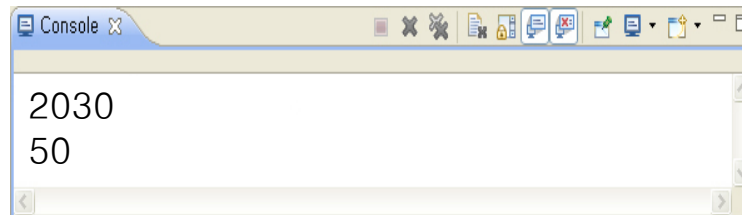
```
package javabook.ch04;

public class WrapperExam {
    int num1;
    String num2;

    public void go( ) {
        num1 = 20;
        num2 = "30";

        System.out.println(num1 + num2);
        int num3 = Integer.parseInt(num2);
        System.out.println(num1 + num3);
    }

    public static void main(String[ ] args) {
        WrapperExam we = new WrapperExam( );
        we.go( );
    }
}
```



[과제] 조건에 맞는 타원과 사각형 그리기



- 슈퍼클래스: Shape
 - ✓ `calcArea()`: 화면에 'Shape:calcArea() 실행' 메시지만 출력
 - ✓ `double d1, d2`을 선언하고, 각각 20, 30을 할당
- 서브클래스: Rectangle, Circle
 - ✓ `main()` 메소드를 갖도록 구현
 - ✓ `calcArea()` 메소드를 오버라이딩하여 사각형 및 원의 넓이를 구할 수 있도록 정의
 - 변수 `d1, d2`를 사용하며 공식은 아래와 같음
 - 사각형의 넓이: $d1 * d2$
 - 원의 넓이: $d1 * d2 * 3.14$
 - Circle 클래스에서 `calcArea(double p1, double p2)` 메소드를 오버로딩하여 구현해 원의 반지름을 입력값으로 취할 수 있도록 함
- 참고
 - ✓ 인스턴스 변수 및 클래스 변수의 차이점의 구체적인 사안은 주석으로 설명할 것
 - ✓ 메소드 오버로딩과 메소드 오버라이딩을 각각 주석으로 설명할 것

[과제] 조건에 맞는 타원과 사각형 그리기



```
Console X
== Rectangle : calcArea( ) ==
사각형의 넓이 : 600.0
```

```
Console X
== Circle : calcArea( ) ==
원의 넓이 : 1884.0
== Circle : calcArea(double d1, double d2) ==
원의 넓이 : 3140.0
```

Thank You!
Q&A