

# 예제 6-1 : Object 클래스로 객체 속성 알아내기

1

객체 레퍼런스만으로 객체의 클래스명, 해시코드 값, 객체의 문자열을 출력해보자  
→ Point 클래스를 util 패키지에 만든 후, 이 프로그램 동일한 기능을 하도록 수정하시오

```
class Point {  
    int x, y;  
    public Point(int x, int y) {  
        this.x = x;  
        this.y = y;  
    }  
}  
  
public class ObjectPropertyEx {  
    public static void print(Object obj) {  
        System.out.println(obj.getClass().getName()); // 클래스 이름  
        System.out.println(obj.hashCode()); // 해시 코드 값  
        System.out.println(obj.toString()); // 객체를 문자열로 만들어 출력  
        System.out.println(obj); // 객체 출력  
    }  
    public static void main(String [] args) {  
        Point p = new Point(2,3);  
        print(p);  
    }  
}
```

# 예제 6-2 : Point 클래스에 toString() 작성

2

Point 클래스에 Point 객체를 문자열로 리턴하는 toString() 메소드를 작성하라.

-> toString 메소드에서 Point(2, 3)을 출력하는 대신 Point (x:2, y:3) 형태로 출력하도록 수정하시오

```
class Point {  
    int x, y;  
    public Point(int x, int y) {  
        this.x = x;  
        this.y = y;  
    }  
    public String toString() {  
        return "Point(" + x + "," + y + ")";  
    }  
}
```

Point 객체를 문자열로 리턴하는 toString() 작성

```
public class ToStringEx {  
    public static void main(String [] args) {  
        Point p = new Point(2,3);  
        System.out.println(p.toString());  
        System.out.println(p); // p는 p.toString()으로 자동 변환  
        System.out.println(p + "입니다."); // p.toString() + "입니다"로 자동 변환  
    }  
}
```

# 예제 6-3 : Point 클래스에 equals() 작성

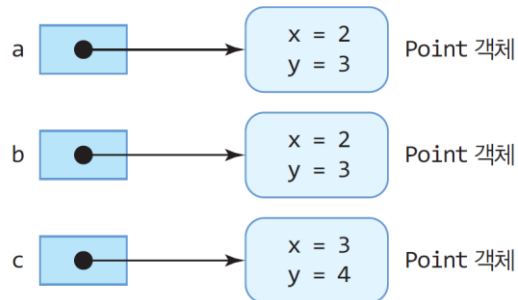
3

Point 클래스에 두 점의 좌표가 같으면 true를 리턴하는 equals()를 작성하라.

➔ 이 예제에서 전달된 obj가 Point 클래스 객체인지 검사하여 아니면 false를 반환하도록 수정

```
class Point {  
    int x, y;  
    public Point(int x, int y) {  
        this.x = x; this.y = y;  
    }  
    public boolean equals(Object obj) {  
        Point p = (Point)obj;  
        if(x == p.x && y == p.y) return true;  
        else return false;  
    }  
}
```

```
public class EqualsEx {  
    public static void main(String[] args) {  
        Point a = new Point(2,3);  
        Point b = new Point(2,3);  
        Point c = new Point(3,4);  
  
        if(a == b) // false  
            System.out.println("a==b");  
        if(a.equals(b)) // true  
            System.out.println("a is equal to b");  
        if(a.equals(c)) // false  
            System.out.println("a is equal to c");  
    }  
}
```



## 예제 6-4 : Rect 클래스와 equals() 만들기 연습

4

int 타입의 width(너비)와 height(높이) 필드를 가지는 Rect 클래스를 작성하고, 면적이 같으면 두 Rect 객체가 같은 것으로 판별하는 equals()를 작성하라. 생성자에서 너비와 높이를 받아 width, height 필드를 초기화하라.

```
class Rect {
    int width;
    int height;
    public Rect(int width, int height) {
        this.width = width;
        this.height = height;
    }
    public boolean equals(Object obj) {
        Rect p = (Rect)obj;
        if (width*height == p.width*p.height)
            return true;
        else
            return false;
    }
}
```

```
public class EqualsEx {
    public static void main(String[] args) {
        Rect a = new Rect(2,3);
        Rect b = new Rect(3,2);
        Rect c = new Rect(3,4);
        if(a.equals(b))
            System.out.println("a is equal to b");
        if(a.equals(c))
            System.out.println("a is equal to c");
        if(b.equals(c))
            System.out.println("b is equal to c");
    }
}
```

➔ 위 프로그램에 대해 equals() 메소드를 width, height 각각의 값이 같은지 비교하는 것으로 수정하시오. 또한, main에서 세 개의 사각형이 모두 다른 경우 "동일한 사각형이 없습니다" 라는 메시지를 출력하시오