

LAB. 상자를 나타내는 Box클래스를 작성하자

1. 먼저 상자의 속성을 나열하여 보라.

-> 다각형으로 이루어진 다면체이며, 사각형으로 이루어진 상자는 가로, 세로, 높이로 설명할 수 있다.

2. 상자의 속성 중에서 가로, 세로, 높이를 나타내는 속성만을 선택하고 width, length, height라는 이름을 부여하여 필드로 만들어라.

```
int width;  
int length;  
int height;
```

3. 위의 필드들을 포함하는 Box클래스를 정의하라.

```
class Box  
{  
    int _____;  
    int _____;  
    int _____;  
}
```

```
1 package Box;  
2  
3 class Box {  
4     int width;  
5     int length;  
6     int height;  
7 }
```

4. 각 필드에 대하여 설정자와 접근자 메소드를 작성하여 Box클래스에 추가하라.

setWidth(****)

getWidth()

```
3  class Box {
4      int width;
5      int length;
6      int height;
7
8  public void setWidth(int width)
9      {
10         this.width = width;
11     }
12 public int getWidth()
13     {
14         return width;
15     }
16 public void setLength(int length)
17     {
18         this.length = length;
19     }
20 public int getLength()
21     {
22         return length;
23     }
24 public void setHeight(int height)
25     {
26         this.height = height;
27     }
28 public int getHeight()
29     {
30         return height;
31     }
```

5. 상자의 부피를 계산하는 메소드 `getVolume()`을 작성하여 `Box`클래스에 추가하라.

```
4      int width;  
5      int length;  
6      int height;  
7      int volume;  
  
34  public void getVolume()  
35  {  
36      volume = getWidth() * getLength() * getHeight();  
37      System.out.println("상자의 부피 ==> " + volume);  
38  }
```

6. 현재 상자의 속성값을 문자열로 만들어서 반환하는 `toString()` 메소드를 작성하여 `Box`클래스에 추가하라.

```
40  public String toString()  
41  {  
42      return volume + "";  
43  }
```

7. `BoxTest`라는 클래스를 작성하고 `main()` 메소드를 추가한다. `main()` 메소드 안에서 상자 객체를 가리킬 수 있는 참조 변수 `box1`을 정의하라.

```
3  class BoxTest {  
4  
5  public static void main(String[] args)  
6  {  
7      Box box1;  
8  
9  }  
10 }
```

8. 상자 객체를 생성하여서 참조 변수 box1이 가리키게 하라.

```
3  class BoxTest {
4
5  public static void main(String[] args)
6  {
7      Box box1;
8      box1 = new Box();
9
10 }
11 }
```

9. 설정자 메소드를 호출하여서 상자 box1의 가로, 세로, 높이를 100, 100, 100으로 설정하여 보자.

```
10      box1.setWidth(100);
11      box1.setLength(100);
12      box1.setHeight(100);
```

10. 참조 변수 box1을 통하여 getVolume() 메소드를 호출하여 보자. 어떤 값이 반환되는가?

```
3  class BoxTest {
4
5  public static void main(String[] args)
6  {
7      Box box1;
8      box1 = new Box();
9
10     box1.setWidth(100);
11     box1.setLength(100);
12     box1.setHeight(100);
13     box1.getVolume();
14
15 }
16 }
```

Problems @ Javadoc Declaration Console X
<terminated> BoxTest [Java Application] /Library/Java/JavaVirtualM
상자의 부피 ==> 1000000

-> 가로, 세로, 높이를 모두 곱한 값인 100,000 이 나온다.

11. System.out.println(box1); 하면 무엇이 출력되는가?

```
3  class BoxTest {
4
5  public static void main(String[] args)
6  {
7      Box box1;
8      box1 = new Box();
9
10     box1.setWidth(100);
11     box1.setLength(100);
12     box1.setHeight(100);
13     box1.getVolume();
14     System.out.println(box1);
15
16 }
```

Problems Javadoc Declaration Console X

<terminated> BoxTest [Java Application] /Library/Java/JavaVirtualMachines/box1
상자의 부피 ==> 1000000
Box.Box@7ad041f3

-> 문자열로 반환된 값이 나온다.

12. 두 번째 객체인 box2를 생성하고 box2의 가로, 세로, 높이를 200, 200, 200으로 설정하여 보자.

```
16     Box box2 = new Box();
17
18     box2.setWidth(200);
19     box2.setLength(200);
20     box2.setHeight(200);
21
```

13. 참조 변수 box2의 값을 box1에 대입한 후에 접근자를 통하여 box1의 속성값을 출력하여 보자. 무엇을 알 수 있는가?

```
16      Box box2 = new Box();
17
18      box2.setWidth(200);
19      box2.setLength(200);
20      box2.setHeight(200);
21
22      box1.setWidth(box2.getWidth());
23      box1.setLength(box2.getLength());
24      box1.setHeight(box2.getHeight());
25
26      System.out.println("\n대입된 box1");
27
28      System.out.println(box1.getWidth());
29      System.out.println(box1.getLength());
30      System.out.println(box1.getHeight());
31  }
```

Problems @ Javadoc Declaration Console X

<terminated> BoxTest [Java Application] /Library/Java/JavaVirtualMach
상자의 부피 ==> 1000000
Box.Box@7ad041f3

대입된 box1
200
200
200

- > 위의 내용은 남겨둔 채 새롭게 box1에 box2의 값을 넣어 출력했을 때, 출력된 box1의 값은 box2의 값과 같음을 알 수 있다.