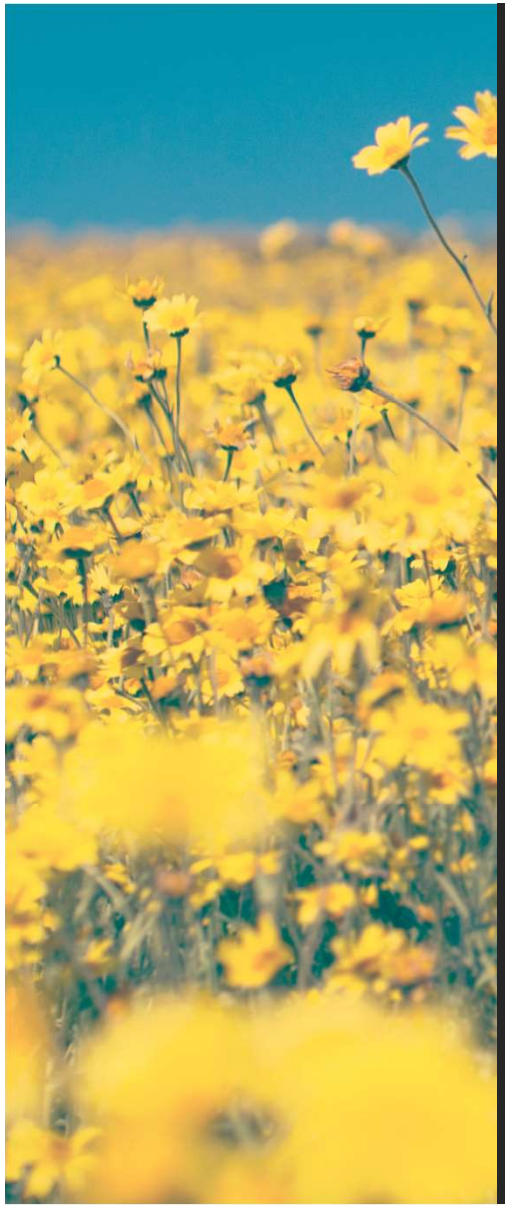


JAVA

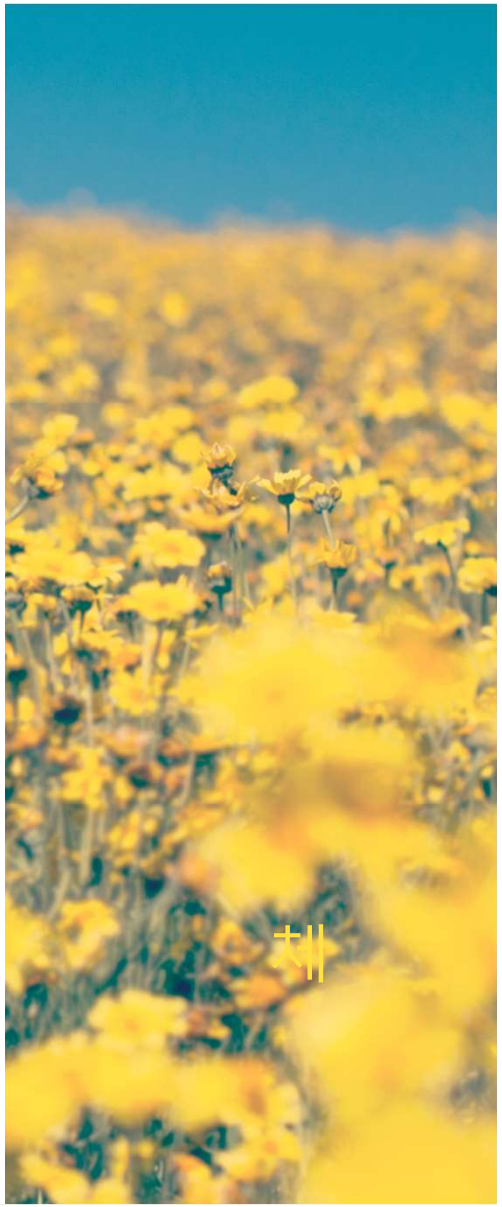
GU, DA HAE

JAVA

접근 제어 지시자

public 클래스

객체



JAVA

접근 제어 지시자 Access Control Specifiers

클래스 멤버의 **접근 허용 범위를 제한**한다.

키워드	클래스 내부	동일 패키지	상속받은 클래스	이외의 영역
private	O	X	X	X
default	O	O	X	X
protected	O	O	O	X
public	O	O	O	O

default는 **접근 제어 지시자를 설정하지 않은 상태**를 가리킨다.

JAVA

예제

```java
public class A{
    int x;
    int y;
    private void printInfo(){
        System.out.println(x + ", " + y);
    }
}
```

```java
public class Ex01{
    public static void main(String[] args){
        A inst = new A();
        inst.x = 10;
    }
}
```

JAVA

예제

```java
package AAA;


public class A{
	int x;
	int y;
	void printInfo(){
		System.out.println(x + ", " + y);
	}
}
```

```java
import AAA.A;


public class Ex01{
	public static void main(String[] args){
		A inst = new A();
		inst.x = 10;
	}
}
```

JAVA

예제

```java
package AAA;


public class A{
	public int x;
	public int y;
	public void printInfo(){
		System.out.println(x + ", " + y);
	}
}
```

```java
import AAA.A;


public class Ex01{
	public static void main(String[] args){
		A inst = new A();
		inst.x = 10;
		inst.printInfo();
	}
}
```

JAVA

정보 은닉

Imformation Hiding

클래스의 멤버 변수를 private으로 선언하여 외부에 보이지 않도록 숨기는 것을 정보 은닉이라 한다.

특별한 경우가 아니라면 **멤버 변수는 반드시 private**으로 선언한다.

멤버 변수를 초기화하는 **생성자는 반드시 public**으로 선언한다.

외부에서 private 변수에 간접접근을 허용하기 위해 **Access 메서드**를 사용한다.

- 변수의 이름이 xxx면 변수의 값을 변경하는 메서드는 setXxx, 변수의 값을 반환하는 메서드는 getXxx로 정의한다.
- 모든 변수가 Access 메서드를 정의해야하는 것은 아니다.

JAVA

예제

```java
package AAA;

public class Point{
    private int x;
    private int y;

    public Point(int _x, int _y){
        x = _x;
        y = _y;
    }
    public void setX(int _x){ x = _x;}
    public void setY(int _y){ y = _y; }
    public int getX(){ return x; }
    public int getY(){ return y; }
    public void printInfo(){
        System.out.println(x + ", " + y);
    }
}
```

```java
import AAA.Point;

public class Ex01{
     public static void main(String[] args){
          Point p = new Point(10, 20);
          p.printInfo();
          p.setX(-10);
          p.setY(-20);
          p.printInfo();
     }
}
```

JAVA

해보기

Circle 클래스는 아무런 문제가 없지만 아래와 같은 제약을 둔다면 문제점을 발견할 수 있다.

- 좌표의 범위는 x : 0~100, y : 0~100이다.
- x는 오른쪽으로 이동할 수록 증가한다. y는 아래로 이동할수록 증가한다.

Circle 클래스는 정보 은닉이 되어있지 않기 때문에, 좌표 범위가 벗어난 값이 저장될 수도 있다. 이런 문제가 발생하지 않도록 정보은닉을 해보자. 이 과정에서 추가로 다양한 메서드를 정의할 수 있다.

```
public class Circle{
	public int x;
	public int y;
	public double radius;

	public void showArea(){
		System.out.println("원 넓이 : " + (3.141592 * radius * radius));
	}
	public void printInfo(){
		System.out.println("중심 좌표 : " + x + ", " + y);
		showArea();
	}
}
```

JAVA

해보기

Rectangle 클래스는 좌표상의 직사각형을 표현한 것이다.

- 좌표의 범위는 x : 0~100, y : 0~100이다.
- x는 오른쪽으로 이동할 수록 증가한다. y는 아래로 이동할수록 증가한다.
- 왼쪽 위의 좌표는 오른쪽 아래 좌표 보다 작아야 한다.

Rectangle 클래스를 정보 은닉하여 위의 제약사항을 지킬 수 있도록 수정해보자.

```
public class Rectangle{
     public int ulx, uly; // 좌 상단 좌표
     public int lrx, lry; // 우 하단 좌표

     public void showArea(){
          int xLen = lrx - ulx;
          int yLen = lry - uly;
          System.out.println("넓이 : " + (xLen * yLen));
     }
}
```

JAVA

접근 제어 지시자와 클래스

접근 제어 지시자 중 default와 public은 클래스 정의에도 사용한다.

키워드	동일 패키지	이외의 영역
default	O	X
public	O	O

클래스의 접근 지시자와 메서드의 접근 지시자를 동일하게 설정한다.

JAVA

예제

```java
package one;


class A{

     ...

}
```

```java
package two;


class B{

     public void test(){

          one.A inst = new one.A();

          ...

     }

}
```

JAVA

예제

```java
package one;


public class A{

     ...

}
```

```java
package two;


class B{

     public void test(){

          one.A inst = new one.A();

          ...

     }

}

// public 클래스는 어디에서나 사용할 수 있다.
```

JAVA

예제

```java
package one;


public class A{

     private A() {...}

     // 클래스는 public 생성자는 private??

     // 인스턴스 생성은 어떻게...?

}
```

```java
package two;


class B{

     public void test(){

          one.A inst = new one.A();

          ...

     }

}
```

JAVA

예제

```java
public class A{

    public A(){} // 자동 삽입되는 디폴트 생성자

}
```

```java
class A{

    A(){} // 자동 삽입되는 디폴트 생성자

}


// 클래스의 접근 지시자 종류에 따라 디폴트 생성자의 접근 지시자도 달라진다.
```

JAVA

public 클래스

public 클래스를 정의하면, 클래스명이 반드시 소스 파일 이름과 같아야 하고, 다른 public 클래스와 함께 정의 할 수 없는 등의 제약이 존재한다. 제약 사항을 지키기 귀찮다는 이유로 default 클래스를 정의하는 경우가 많다.

언제 public 클래스를 정의해야 할까?

JAVA

라이브러리

라이브러리는 이미 만들어져 있는 클래스의 모음을 뜻한다. 필요하면 언제든지 사용할 수 있게 미리 정의해 놓은 클래스라는 뜻이다.

> 라이브러리는 언제나, 누구나 제한 없이 사용할 수 있어야 하기 때문에 public 클래스로 정의한다.

키워드	동일 패키지	이외의 영역
default	O	X
public	O	O

JAVA

예제

```
package Figure;

public class FigureArea{
    private Circle c;
    private Rectangle r;
    public FigureArea(){
        c = new Circle();
        r = new Rectangle();
    }
    public double getCircleArea(double r){
        c.setRadius(r);
        return c.getArea();
    }
    public double getRectangleArea(double w, double
h){
        r.setWdith(w);
        r.setHeight(h);
        return r.getArea();
    }

}
```

```
class Circle{
    private double radius;
    Circle(){ radius = 0; }
    void setRadius(double r){ radius = r;}
    double getArea(){ return 3.141592 * radius *
radius; }
}

class Rectangle{
    private double width;
    private double height;
    Rectangle(){ width = height = 0; }
    void setWdith(double w){ width = w; }
    void setHeight(double h){ height = h; }
    double getArea(){ return width * height; }
}
```

JAVA

예제

```java
import Figure.FigureArea;


public class Ex01{
     public static void main(String[] args){
          Cal.FigureArea area = new FigureArea();


          System.out.println("반지름이 1.2인 원 넓이 : " + area.getCircleArea(1.2));
          System.out.println("3.7 x 8.7 크기의 사각형 넓이 : " + area.getRectangleArea(3.7, 8.7));
     }
}
```

JAVA

해보기

좌표상의 원을 표현하는 Circle 클래스를 정의하여 사용자에게 라이브러리로 제공해야 한다. 아래의 Point 클래스를 기반으로 Circle 클래스를 정의해 보자.

Circle 클래스는 원의 중심좌표와 반지름 길이를 저장할 수 있어야 한다.

```
class Point{
	private int x;
	private int y;

	Point(int _x, int _y){
		x = _x;
		y = _y;
	}
	void showPointInfo(){
		System.out.println("[" + x +", " + y + "]");
	}
}
```

JAVA

해보기

좌표상의 링 형태의 도형을 표현하는 클래스를 정의해서 사용자에게 라이브러리 형태로 제공해야 한다. 링은 2개의 원을 이용하여 표현할 수 있다. 앞의 해보기에서 정의한 Circle 클래스를 활용하여 Ring 클래스를 정의해 보자.

안쪽 원과 바깥 쪽 원의 중심좌표가 같다면 두께가 일정한 링을 표현한 것이며, 중심 좌표가 다르다면 두께가 일정하지 않은 링을 표현하는 것이다.

```
public static void main(String[] args){
     Ring r = new Ring(1,1,4,2,2,9);
     r.showRing();
}
```

```
[안쪽 원]
반지름 : 4
중심좌표 : [1, 1]

[바깥쪽 원]
반지름 : 9
중심좌표 : [2, 2]
```