



JAVA 내부클래스

클래스가 여러클래스와 관계를 맺는 경우에는 독립적으로 선언하는 것이 좋지만,
특정 클래스와 관계를 맺는 경우에는 클래스 내부에 선언하는 것이 좋다.

내부클래스란 클래스 내부에 선언한 클래스이다.

내부클래스 장점

1. 내부클래스에서 외부클래스의 멤버를 쉽게 접근할 수 있다.
2. 코드의 복잡성을 줄일 수 있다.



내부클래스 선언위치

내부클래스는 선언되는 위치에 따라서 두 가지로 분류된다.

클래스의 멤버로 선언되는 클래스를 **멤버클래스**라고 한다.

메소드 내부에서 선언되는 클래스를 **로컬클래스**라고 한다.

멤버클래스는 클래스나 객체가 사용 중이라면 재사용이 가능하다.

로컬클래스는 메소드를 실행할 때만 사용되고 메소드가 종료되면 없어진다.



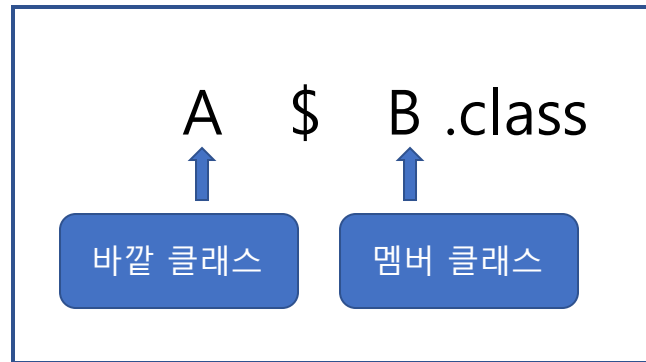
내부클래스 선언위치

선언 위치에 따른분류		선언위치	설명
멤버 클래스	인스턴스 멤버 클래스	<pre>class A { class B {...} }</pre>	A 객체를 생성해야만 사용할 수 있는 B클래스
	정적 멤버 클래스	<pre>class A { static class B {...} }</pre>	A 클래스로 바로 접근할 수 있는 B 클래스
로컬 클래스		<pre>class A { void method() { class B {...} } }</pre>	method()가 실행할 때만 사용할 수 있는 B클래스

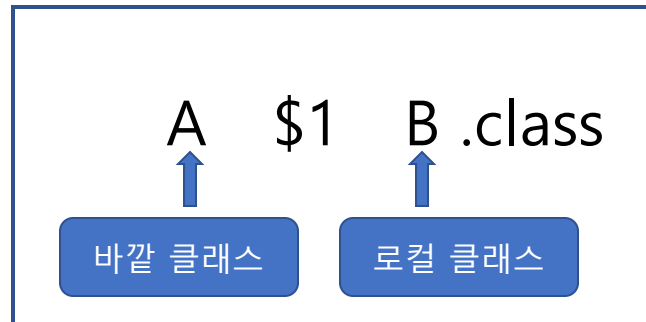


내부클래스 파일이름

내부클래스도 컴파일하면 .class 파일이 생성된다.



로컬클래스일 경우에는 \$1이 포함된 바이트코드파일이 생성된다.





인스턴스 멤버 클래스

인스턴스 멤버 클래스는 인스턴스 필드와 인스턴스 메소드만 선언이 가능하고
정적 필드와 메소드는 선언할 수 없다.

```
public class A {  
  
    class B {  
        int x = 100;  
        public void xPrint() {  
            System.out.println(x);  
        }  
    }  
}
```



정적 멤버 클래스

정적 멤버클래스는 static 키워드로 선언된 클래스를 말한다.
정적 멤버클래스는 모든 종류의 필드와 메소드를 선언할 수 있다.

```
static class B {  
    int x = 100;  
    static int y = 200;  
  
    public void xPrint() {  
        System.out.println(x);  
    }  
    public static void yPrint() {  
        System.out.println(y);  
    }  
}
```



로컬 클래스

메소드 내에서도 선언하는 클래스를 로컬 클래스라고 한다.
로컬클래스는 접근제한자(public, private) 및 static을 붙일 수 없다.

```
public void method() {  
    class B {  
        int x;  
        void xPrint() {  
            System.out.println(x);  
        }  
    }  
    B b = new B();  
    b.xPrint();  
}
```



익명 객체 / 익명 내부클래스

익명객체는 이름이 없는 객체를 말한다. 익명객체를 만들려면 조건이 있다.
어떤 클래스를 상속하거나 인터페이스를 구현해야만 한다.

이 경우 클래스 변수는 이름이 없는 객체를 참조하고, 인터페이스 변수는 이름이 없는 구현객체를 참조하게 된다. 이름이 없기 때문에 생성자도 가질 수 없다

익명 클래스는 클래스의 선언과 동시에 객체생성을 하기 때문에 단 한 번만 사용될 수 있고 오직 하나의 객체만을 생성할 수 있는 일회용 클래스이다.



익명 내부클래스

```
class A {  
    public void hello() {  
        System.out.println("안녕하세요");  
    }  
}
```

```
A a = new A() {  
    public void hello() {  
        System.out.println("좋은하루 되세요~!");  
    }  
};  
a.hello();
```

← 하나의 실행문이므로
세미콜론(;)을 붙여야 한다.