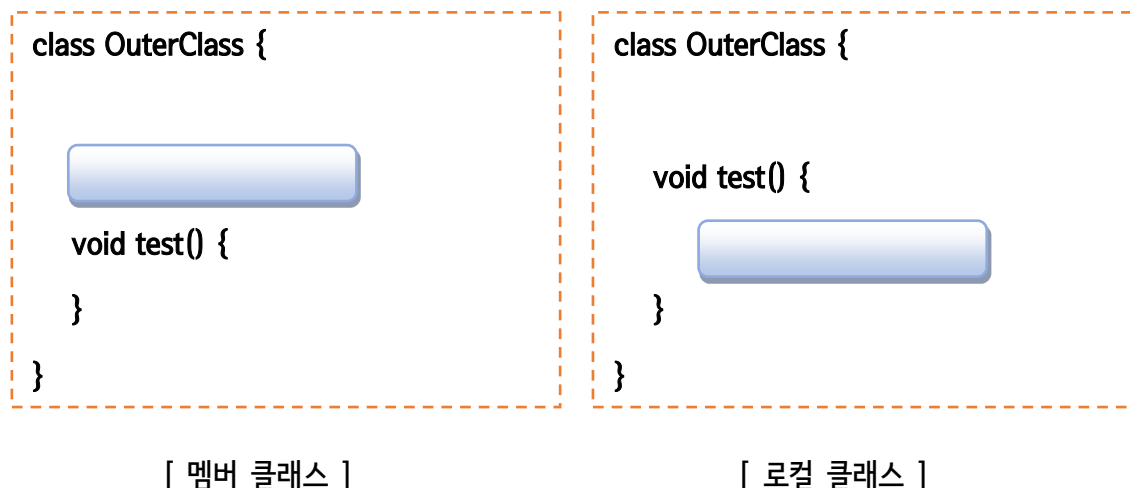


# 내부클래스

내부 클래스란 클래스의 내부에 정의되는 클래스로서 특정 클래스의 내에서만 주로 사용되는 클래스를 내부 클래스로 정의한다. 내부 클래스에서는 외부 클래스의 멤버들을 접근할 수 있으며 캡슐화를 통해 코드의 복잡성을 줄일 수 있다.

## ■ 내부 클래스의 종류

내부 클래스는 정의되는 위치에 따라 멤버 클래스와 로컬 클래스로 나뉜다. 멤버 변수와 지역 변수로 나뉘는 변수와 같이 내부 클래스도 클래스의 멤버로 정의되는 멤버 클래스와 메서드 내에 정의되는 로컬 클래스로 나뉘며 각각 변수와 비슷한 유효범위와 성격을 지원한다.



멤버 클래스는 인스턴스 클래스와 스태틱 클래스로 구성되며 로컬 클래스는 이름이 있는 로컬 클래스와 이름이 없는 로컬 클래스(anonymous 클래스)로 나뉜다.

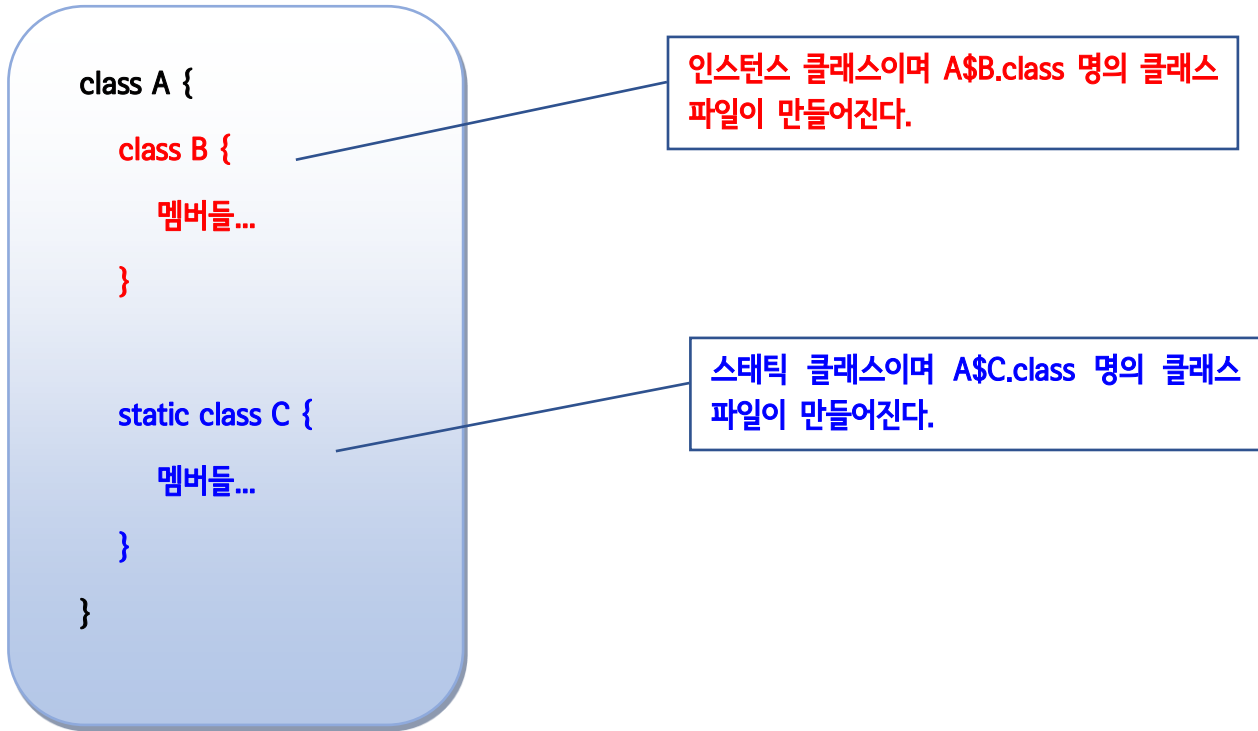
내부 클래스는 다른 클래스내에 포함되어 정의되지만 클래스 파일은 독립적으로 만들어 다음과 같은 규칙으로 파일 명이 정해진다.

내부 클래스의 종류	생성되는 클래스명의 규칙
인스턴스 클래스	외부클래스\$내부클래스.class
스태틱 클래스	외부클래스\$내부클래스.class
이름있는 로컬 클래스	외부클래스\$N내부클래스.class
이름없는 로컬 클래스	외부클래스\$N.class

로컬 클래스의 경우에는 Java 소스에 정의되는 순서에 따라서 클래스명 중간 또는 뒤에 1부터 시작하는 숫자자 자동으로 부여된다.

## ■ 멤버 클래스

동일 클래스에서는 물론이고 다른 클래스에서도 이 클래스들을 사용할 수 있다. 멤버 변수와 비슷한 성격을 갖는다. 인스턴스 클래스와 스테틱 클래스로 나뉜다.



인스턴스 클래스는 외부 클래스의 인스턴스 멤버처럼 다루어 지며 주로 외부 클래스의 인스턴스 멤버들과 관련된 작업에 사용될 목적으로 정의된다. 인스턴스 클래스 내에서는 static 멤버를 정의할 수 없다.

스테틱 클래스는 외부 클래스의 클래스 멤버(static 멤버)처럼 다루어 지며 주로 외부 클래스의 클래스 메서드내에서 사용될 목적으로 정의된다.

멤버 클래스들도 멤버 변수들과 비슷한 방법으로 접근 제어자 지정이 가능하며 동일 클래스에서 뿐만 아니라 외부 클래스의 밖에서도 접근하고자 하는 경우에는 각각 다음과 같은 방법으로 사용할 수 있다.

인스턴스 클래스	A a = new A(); A.B b = a.new B(); b.멤버
스테틱 클래스	A.C.멤버

## ■ 로컬 클래스

메서드 내에 정의되는 클래스로서 로컬 변수와 비슷한 성격을 갖고며, 활용 범위가 정의되어 있는 메서드 블록 내부로 제한된다. Interface 는 로컬로 정의될 수 없다. 포함하는 클래스의 멤버 변수와 포함하는 메서드의 final 로컬 변수, final 매개변수사용이 가능하다.

로컬 클래스는 이름이 있는 로컬 클래스와 이름이 없는 로컬 클래스로 나뉘며 이름이 있는 로컬 클래스를 **로컬 클래스**, 이름이 없는 로컬 클래스를 **익명 클래스**라고 부른다.

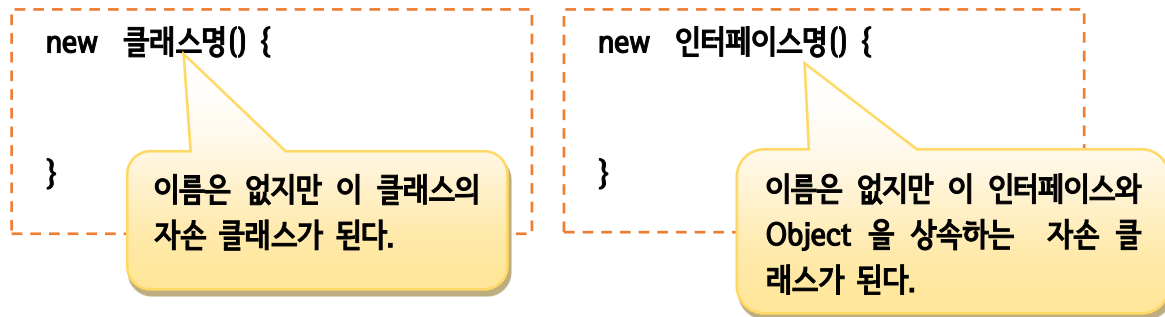
```
class X {  
    iny num;  
    void sam(final int i) {  
        int total = 20;  
        final String s="test";  
        class Y {  
            멤버들...  
        }  
        Y y = new Y();  
        y.멤버들...;  
    }  
}
```

로컬 클래스이며 X\$1\$Y.class 명의 클래스 파일이 만들어진다. Y 클래스 내에서는 X 클래스의 멤버 변수 num, sam() 메서드의 final 지역변수 s 를 사용할 수 있다. Y 클래스는 sam() 메서드 내에서만 사용 가능하다.

```
class N {  
    void pr(Test t) {  
        ...  
    }  
    void sam() {  
        pr(new Test() {  
            멤버들...  
        });  
        ..  
    }  
}
```

익명 클래스이며 N\$1.class 명의 클래스 파일이 만들어진다. 정의된 위치에서 한 번만 객체 생성이 가능하다. 클래스의 정의와 객체 생성을 동시에 하는 1회용 클래스라고 할 수 있다.

익명 클래스의 경우 new 키워드 뒤의 생성자 메서드의 명칭이 기존 클래스 명인 경우에는, 자동적으로 이 클래스의 자손 클래스가 되며, 인터페이스 명인 경우에는 이 인터페이스를 구현하여 추가 상속하는 클래스로서 부모 클래스는 Object 이 된다.



메서드 호출 시 매개변수의 타입이 추상 클래스 형이거나 인터페이스 형이어서 가볍게 구현(자손) 클래스를 만들고 객체를 생성해서 전달하려는 경우 유용하게 사용될 수 있는 구문이다.