

JAVA 프로그래밍

Chapter 14 중첩 클래스

목차

- ❖ 중첩 클래스(Member Inner Class)
- ❖ 정적 중첩 클래스(Static Inner Class)
- ❖ 지역 중첩 클래스(Local Inner Class)
- ❖ 익명 중첩 클래스(Anonymous Inner Class)

중첩(Nested) 클래스

❖ 중첩(Nested=Inner) 클래스의 특징

- 클래스 내부에 또 다른 클래스를 가짐으로 클래스 관리의 효율을 높인 것.
- 내부 클래스는 외부 클래스의 모든 멤버들을 마치 자신의 멤버처럼 사용할 수 있다.
- 서로 관련 있는 클래스를 논리적으로 묶어서 표현함으로써, 코드의 캡슐화를 증가시킨다.
- static 내부 클래스를 제외하고는 다른 내부 클래스는 항상 외부 클래스를 통해서 생성할 수 있다.

❖ 중첩(Nested=Inner) 클래스의 종류

종류	설명
Member	멤버 변수나 멤버 메소드와 같이 클래스가 정의된 경우에 사용한다.
Static	static 변수(클래스 변수)와 같이 클래스가 static으로 선언된 경우에 사용한다.
Local	특정한 메소드 내에 클래스가 정의된 경우에 사용한다.
Anonymous	참조할 수 있는 이름이 없는 경우에 사용한다.

중첩(Nested) 클래스

❖ 중첩(Nested=Inner) 클래스의 종류

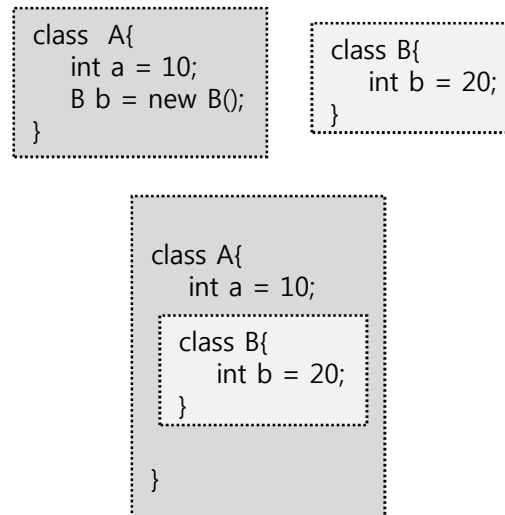
선언 위치에 따른 분류		선언 위치	설명
멤버 클래스	인스턴스 멤버 클래스	<pre>class A { class B {...} }</pre>	A 객체를 생성해야만 사용할 수 있는 B 중첩 클래스
	정적 멤버 클래스	<pre>class A { static class B {...} }</pre>	A 객체 생성없이 클래스로 바로 접근할 수 있는 B 중첩 클래스
로컬 클래스		<pre>class A { void method(){ class B {...} } }</pre>	method()가 실행할 때만 사용할 수 있는 B 중첩 클래스

중첩 클래스

❖ 인스턴스 멤버 클래스

- 인스턴스 멤버 클래스는 static 키워드 없이 선언된 클래스를 말한다.
- 인스턴스 멤버 클래스는 인스턴스 필드와 메서드만 선언이 가능하고 정적 필드와 메서드는 선언할 수 없다.
- 인스턴스 클래스는 주로 외부 클래스의 인스턴스 변수나 인스턴스 메서드에 사용될 목적으로 선언된다.
- 객체 생성은 Outer클래스의 객체를 생성하고 Inner객체를 생성해야 한다.

```
class Outer{
    class Inner{
        Inner(){} // 생성자
        int field1; // 인스턴스 필드
        static int field2; // 에러, 정적필드(X)
        void method1(){} // 인스턴스 메서드
        static void method2(){} // 에러, 정적 메서드(X)
    }
}
// 생성파일
Outer.class, Outer$Inner.class
// 객체 생성
Outer outer = new Outer();
Outer.Inner inner = outer.new Inner();
```



정적 중첩 클래스

❖ 정적 멤버 클래스

- 정적 멤버 클래스는 static 키워드로 선언된 클래스를 말한다.
- 정적 멤버 클래스는 모든 종류의 필드와 메서드를 선언할 수 있다. 중첩 클래스 내부에 static 멤버를 포함할 수 있는 형태.
- 단, Outer의 non-static 멤버 포함 불가하다.
- 정적 멤버 클래스는 정적 필드와 같은 특성을 갖는다.
- 객체 생성없이 Outer 클래스 이름으로 직접 접근이 가능하다.

```
class Outer{
    static class Inner{
        Inner(){}           // 생성자
        int field1;         // 인스턴스 필드
        static int field2;  // 정적필드(0)
        void method1(){}    // 인스턴스 메서드
        static void method2(){} // 정적 메서드(0)
    }
}
// 생성파일
Outer.class, Outer$Inner.class
// 객체 생성
Outer outer = new Outer();
Outer.Inner inner = new Outer.Inner();
```

지역 중첩 클래스

❖ 지역 중첩 클래스(Local Inner Class)

- 지역 클래스는 외부 클래스의 메서드나 초기화 블록에서 선언된 클래스를 말한다.
- 메서드 실행 시에 사용되는 클래스를 정의한 형식으로 접근제한자(public, private)나 static을 가질 수 없다.
- 인스턴스 필드와 메소드만 선언이 가능하고 정적필드와 메소드는 선언할 수 없다.
- 외부 클래스의 멤버변수들은 접근할 수 있다.(상수포함)
- 메소드 내에 있는 지역변수들은 접근할 수 없다.(상수는 제외)
- 객체 생성은 외부에서 할 수 없다.

```
class Outer{
    public void method(){
        class Inner{
            Inner(){           // 생성자
                int field1;    // 인스턴스 필드
                static int field2; // 에러, 정적필드(X)
                void method1(){ // 인스턴스 메서드
                    static void method2(){} // 에러, 정적 메서드(X)
                }
                Inner inner = new Inner(); // 객체 생성
            }
        }
    }
}
// 생성파일
Outer.class, Outer$숫자Inner.class
```

익명 중첩 클래스

❖ 익명 중첩 클래스(Anonymous Inner Class)

- 익명(anonymous) 객체는 이름이 없는 객체를 말한다.
- 익명 클래스는 클래스의 선언과 동시에 객체를 생성하므로, 단 하나의 객체만을 생성하는 일회용 클래스이다.
- 기존 클래스(인터페이스)의 특정 메서드를 오버라이딩하여 원하는 형태로 재정의하여 사용하는 방식
- 기존 클래스 또는 인터페이스가 존재해야 한다.
- 외부 멤버 중 final만 포함할 수 있다.
- new Inner(){...} 자체가 객체 생성임.
- UI 이벤트 처리 객체나 스레드 객체를 간편하게 생성할 목적으로 많이 활용됨.

```
class Inner {...} // 기존 클래스

class Outer{
    public void method(){
        Inner inner = new Inner(){...}; // 객체 생성
    }
}
// 생성파일
Outer.class, Outer$숫자.class
```