

Chapter 09 중첩 선언과 익명 객체

9.1 중첩 클래스

9.2 인스턴스 멤버 클래스

9.3 정적 멤버 클래스

9.4 로컬 클래스

9.5 바깥 멤버 접근

9.6 중첩 인터페이스

9.7 익명 객체

중첩 클래스

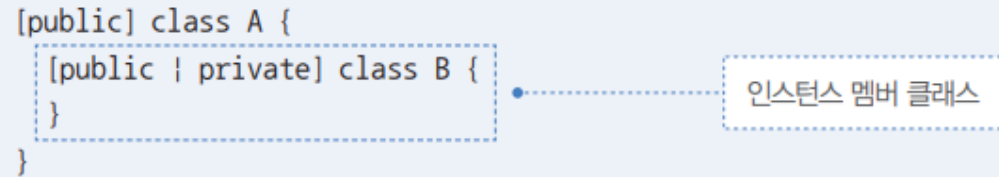
- 클래스 내부에 선언한 클래스. 클래스의 멤버를 쉽게 사용할 수 있고 외부에는 중첩 관계 클래스를 감춤으로써 코드의 복잡성을 줄일 수 있음
- 멤버 클래스: 클래스의 멤버로서 선언되는 중첩 클래스
- 로컬 클래스: 메소드 내부에서 선언되는 중첩 클래스

선언 위치에 따른 분류		선언 위치	객체 생성 조건
멤버 클래스	인스턴스 멤버 클래스	<pre>class A { class B { ... } }</pre>	A 객체를 생성해야만 B 객체를 생성할 수 있음
	정적 멤버 클래스	<pre>class A { static class B { ... } }</pre>	A 객체를 생성하지 않아도 B 객체를 생성할 수 있음
로컬 클래스		<pre>class A { void method() { class B { ... } } }</pre>	method가 실행할 때만 B 객체를 생성할 수 있음

인스턴스 멤버 클래스

- A 클래스의 멤버로 선언된 B 클래스

```
[public] class A {  
    [public | private] class B {  
    }  
}
```




구분	접근 범위
public class B { }	다른 패키지에서 B 클래스를 사용할 수 있다.
class B { }	같은 패키지에서만 B 클래스를 사용할 수 있다.
private class B { }	A 클래스 내부에서만 B 클래스를 사용할 수 있다.

- 인스턴스 멤버 클래스 B는 주로 A 클래스 내부에서 사용되므로 private 접근 제한을 갖는 것이 일반적

정적 멤버 클래스

- static 키워드와 함께 A 클래스의 멤버로 선언된 B 클래스

```
[public] class A {  
    [public | private] static class B {  
    }  
}
```



정적 멤버 클래스

구분	접근 범위
public static class B { }	다른 패키지에서 B 클래스를 사용할 수 있다.
static class B { }	같은 패키지에서만 B 클래스를 사용할 수 있다.
private static class B { }	A 클래스 내부에서만 B 클래스를 사용할 수 있다.

- 정적 멤버 클래스는 주로 default 또는 public 접근 제한을 가진다.

로컬 클래스

- 생성자 또는 메소드 내부에서 다음과 같이 선언된 클래스
- 생성자와 메소드가 실행될 동안에만 객체를 생성할 수 있음

```
[public] class A {
```

```
    //생성자
```

```
    public A() {
```

```
        class B { }
```

```
    }
```

```
    //메소드
```

```
    public void method() {
```

```
        class B { }
```

```
    }
```

```
}
```

로컬 클래스

바깥 클래스의 멤버 접근 제한

- 정적 멤버 클래스 내부에서는 바깥 클래스의 필드와 메소드를 사용할 때 제한이 따름

구분	바깥 클래스의 사용 가능한 멤버
인스턴스 멤버 클래스	바깥 클래스의 모든 필드와 메소드
정적 멤버 클래스	바깥 클래스의 정적 필드와 정적 메소드

- 정적 멤버 클래스는 바깥 객체가 없어도 사용 가능해야 하므로 바깥 클래스의 인스턴스 필드와 인스턴스 메소드는 사용하지 못함

바깥 클래스의 객체 접근

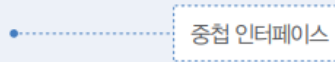
- 중첩 클래스 내부에서 바깥 클래스의 객체를 얻으려면 바깥 클래스 이름에 `this`를 붙임

바깥클래스이름.this → 바깥객체

중첩 인터페이스


- 해당 클래스와 긴밀한 관계를 맺는 구현 객체를 만들기 위해 클래스의 멤버로 선언된 인터페이스

```
class A {  
    [public | private] [static] interface B {  
        //상수 필드  
        //추상 메소드  
        //디폴트 메소드  
        //정적 메소드  
    }  
}
```



- 안드로이드와 같은 UI 프로그램에서 이벤트를 처리할 목적으로 많이 활용

```
>>> Button.java  
  
1  package ch09.sec06.exam01;  
2  
3  public class Button {  
4      //정적 중첩 인터페이스  
5      public static interface ClickListener {  
6          //추상 메소드  
7          void onClick();  
8      }  
9  }
```



익명 객체

- 이름이 없는 객체. 명시적으로 클래스를 선언하지 않기 때문에 쉽게 객체를 생성할 수 있음
- 필드값, 로컬 변수값, 매개변수값으로 주로 사용

익명 자식 객체

- 부모 클래스를 상속받아 생성되는 객체
- 부모 타입의 필드, 로컬 변수, 매개변수의 값으로 대입할 수 있음

```
new 부모생성자(매개값, ...) {  
    //필드  
    //메소드  
}
```

익명 구현 객체

- 인터페이스를 구현해서 생성되는 객체
- 인터페이스 타입의 필드, 로컬 변수, 매개변수의 값으로 대입할 수 있음
- 안드로이드와 같은 UI 프로그램에서 이벤트를 처리하는 객체로 많이 사용

```
new 인터페이스() {  
    //필드  
    //메소드  
}
```

Thank you!