

중첩(Nested) 클래스

중첩(Nested) 클래스

❖ 중첩 클래스와 중첩 인터페이스란?

- 중첩 클래스: 클래스 멤버로 선언된 클래스

```
class ClassName {  
    class NestedClassName {  
    }  
}
```

중첩 클래스

- 중첩 인터페이스: 클래스 멤버로 선언된 인터페이스
 - UI 컴포넌트 내부 이벤트 처리에 많이 활용

```
class ClassName {  
    interface NestedInterfaceName {  
    }  
}
```

중첩 인터페이스

중첩(Nested) 클래스

❖ 중첩 클래스의 분류

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 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

- 클래스 생성시 바이트 코드 따로 생성

A \$ B .class
바깥 클래스 멤버 클래스

A \$1 B .class
바깥 클래스 로컬 클래스

중첩(Nested) 클래스

❖ 인스턴스 멤버 클래스

```
class A {  
    /**인스턴스 멤버 클래스**/  
    class B {  
        B() { }                -----생성자  
        int field1;            -----인스턴스 필드  
        //static int field2;    -----정적 필드 (x)  
        void method1() { }     -----인스턴스 메소드  
        //static void method2() { } -----정적 메소드 (x)  
    }  
}
```

```
A    a = new A();  
A.B  b = a.new B();  
b.field1 = 3;  
b.method1();
```

중첩(Nested) 클래스

❖ 정적 멤버 클래스

- `static` 키워드로 선언된 클래스, 모든 종류의 필드, 메소드 선언 가능

```
class A {  
    /**정적 멤버 클래스**/  
    static class C {  
        C() {}                -----생성자  
        int field1;            -----인스턴스 필드  
        static int field2;     -----정적 필드  
        void method1() {}      -----인스턴스 메소드  
        static void method2() {} -----정적 메소드  
    }  
}
```

```
A.C c = new A.C();  
c.field1 = 3;        //인스턴스 필드 사용  
c.method1();         //인스턴스 메소드 호출  
A.C.field2 = 3;      //정적 필드 사용  
A.C.method2();       //정적 메소드 호출
```

중첩(Nested) 클래스

❖ 로컬 클래스 - 메소드 내에서만 사용

```
void method() {  
    /**로컬 클래스**/  
    class D {  
        D() { }                -----생성자  
        int field1;            -----인스턴스 필드  
        //static int field2;    -----정적 필드(x)  
        void method1() { }     -----인스턴스 메소드  
        //static void method2() { } -----정적 메소드(x)  
    }  
    D d = new D();  
    d.field1 = 3;  
    d.method1();  
}
```

```
void method() {  
    class DownloadThread extends Thread { ... }  
    DownloadThread thread = new DownloadThread();  
    thread.start();  
}
```

중첩(Nested) 클래스

❖ 중첩 클래스: A.java

```
/** 바깥 클래스 */  
class A {  
    A() {  
        System.out.println("A 객체가 생성됨");  
    }  
  
    /** 인스턴스 멤버 클래스 */  
    public class B {  
        B() {  
            System.out.println("B 객체가 생성됨");  
        }  
  
        int field1;  
  
        // static int field2;  
        void method1() {  
        }  
        // static void method2() { }  
    }  
}
```

중첩(Nested) 클래스

❖ 중첩 클래스: A.java

```
/** 정적 멤버 클래스 */  
static class C {  
    C() {  
        System.out.println("C 객체가 생성됨");  
    }  
  
    int field1;  
    static int field2;  
  
    void method1() {  
    }  
  
    static void method2() {  
    }  
}
```


중첩(Nested) 클래스

❖ 중첩 클래스: A.java

```
void method() {
    /** 로컬 클래스 */
    class D {
        D() {
            System.out.println("D 객체가 생성됨");
        }

        int field1;

        // static int field2;
        void method1() {
        }
        // static void method2() { }
    }
    D d = new D();
    d.field1 = 3;
    d.method1();
}
```

중첩(Nested) 클래스

❖ 중첩 클래스 객체 생성: Main.java

```
public class Main {  
    public static void main(String[] args) {  
        A a = new A();  
  
        // 인스턴스 멤버 클래스 객체 생성  
        A.B b = a.new B();  
        b.field1 = 3;  
        b.method1();  
  
        // 정적 멤버 클래스 객체 생성  
        A.C c = new A.C();  
        c.field1 = 3;  
        c.method1();  
        A.C.field2 = 3;  
        A.C.method2();  
  
        // 로컬 클래스 객체 생성을 위한 메소드 호출  
        a.method();  
    }  
}
```

중첩(Nested) 클래스

❖ 바깥 필드와 메소드에서 사용 제한

```
public class A {  
    //인스턴스 멤버 클래스  
    class B {}  
  
    //정적 멤버 클래스  
    static class C {}  
}
```

```
public class A {  
    //인스턴스 필드  
    B field1 = new B();          ----- (o)  
    C field2 = new C();          ----- (o)  
  
    //인스턴스 메소드  
    void method1() {  
        B var1 = new B();        ----- (o)  
        C var2 = new C();        ----- (o)  
    }
```

```
    //정적 필드 초기화  
    //static B field3 = new B();    ----- (x)  
    static C field4 = new C();      ----- (o)  
  
    //정적 메소드  
    static void method2() {  
        //B var1 = new B();        ----- (x)  
        C var2 = new C();          ----- (o)  
    }
```

중첩(Nested) 클래스

❖ 바깥 필드와 메서드에서 사용 제한: A.java

```
public class A {  
    // 인스턴스 필드  
    B field1 = new B();  
    C field2 = new C();  
  
    // 인스턴스 메소드  
    void method1() {  
        B var1 = new B();  
        C var2 = new C();  
    }  
  
    // 정적 필드 초기화  
    // static B field3 = new B();  
    static C field4 = new C();  
  
    // 정적 메소드  
    static void method2() {  
        // B var1 = new B();  
        C var2 = new C();  
    }  
}
```

중첩(Nested) 클래스

중첩(Nested) 클래스

중첩(Nested) 클래스

중첩(Nested) 클래스

중첩(Nested) 클래스

중첩(Nested) 클래스

중첩(Nested) 클래스
