

2025년 상반기 K-디지털 트레이닝

# 중첩 선언과 익명 객체

---

[KB] IT's Your Life

### ✓ 다음 클래스를 정의하세요.

```
package ch09.sec02.exam01;

public class A {
    //인스턴스 멤버 클래스
    class B {}

    //인스턴스 필드 값으로 B 객체 대입
    B field = new B();

    //생성자
    A() {
        B b = new B();
    }

    //인스턴스 메소드
    void method() {
        B b = new B();
    }
}
```

- ✓ 앞에서 정의한 클래스를 이용해서 클래스 A와 내부 클래스 B를 인스턴스화 시켜보세요.

```
package ch09.sec02.exam01;

public class AExample {
    public static void main(String[] args) {
        //A 객체 생성

        //B 객체 생성

    }
}
```

### ✓ 다음 클래스를 정의하세요.

```
package ch09.sec02.exam02;

public class A {
    //인스턴스 멤버 클래스
    class B {
        //인스턴스 필드
        int field1 = 1;

        //정적 필드(Java 17부터 허용)
        static int field2 = 2;

        //생성자
        B() {
            System.out.println("B-생성자
실행");
        }

        //인스턴스 메소드
        void method1() {
            System.out.println("B-method1
실행");
        }
    }
}

//정적 메소드(Java 17부터 허용)
static void method2() {
    System.out.println("B-method2
실행");
}

//인스턴스 메소드
void useB() {
    //B 객체 생성 및 인스턴스 필드 및 메소드 사용
    B b = new B();
    System.out.println(b.field1);
    b.method1();

    //B 클래스의 정적 필드 및 메소드 사용
    System.out.println(B.field2);
    B.method2();
}
```

## 중첩 선언과 익명 객체

✓ 다음 클래스를 실행했을 때 결과가 어떻게 나올지 적어보고, 실제 확인하세요.

```
package ch09.sec02.exam02;

public class AExample {
    public static void main(String[] args) {
        //A 객체 생성
        A a = new A();

        //A 인스턴스 메소드 호출
        a.useB();
    }
}
```

### ☑ 다음처럼 클래스를 정의하세요.

```
package ch09.sec03.exam01;

public class A {
    //static 멤버 클래스
    static class B {}

    //인스턴스 필드 값으로 B 객체 대입
    B field1 = new B();

    //정적 필드 값으로 B 객체 대입
    static B field2 = new B();

    //생성자
    A() {
        B b = new B();
    }

    //인스턴스 메소드
    void method1() {
        B b = new B();
    }

    //정적 메소드
    static void method2() {
        B b = new B();
    }
}
```

- ✓ 정적 내부 멤버 클래스 B의 인스턴스를 생성하세요.

```
package ch09.sec03.exam01;

public class AExample {
    public static void main(String[] args) {
        //B 객체 생성

    }
}
```

### ☑ 다음처럼 클래스를 정의하세요.

```
package ch09.sec03.exam02;

public class A {
    //정적 멤버 클래스
    static class B {
        //인스턴스 필드
        int field1 = 1;

        static int field2 = 2;

        //생성자
        B() {
            System.out.println("B-생성자
실행");
        }

        //인스턴스 메소드
        void method1() {
            System.out.println("B-method1
실행");
        }
    }
}

static void method2() {
    System.out.println("B-method2
실행");
}
```



## 중첩 선언과 익명 객체

- ✓ 앞의 클래스를 이용하여 다음과 같이 결과가 나오도록 코드를 완성하세요.

```
package ch09.sec03.exam02;

public class AExample {
    public static void main(String[] args) {
        //B 객체 생성 및 인스턴스 필드 및 메소드 사용

        //B 클래스의 정적 필드 및 메소드 사용

    }
}
```

B-생성자 실행  
1  
B-method1 실행  
2  
B-method2 실행

## 중첩 선언과 익명 객체

- ✓ 다음 클래스 내부에 ClickListener 인터페이스를 정의하세요.
  - 인터페이스의 메서드는 void onClick() 하나 추가

```
package ch09.sec06.exam01;  
  
public class Button {  
  
}
```

- ✓ 앞에서 만든 Button 클래스를 이용하여 ClickListener 구현 클래스를 로컬 클래스로 정의하세요.

```
package ch09.sec06.exam03;

public class ButtonExample {
    public static void main(String[] args) {

        //Ok 버튼 클릭 이벤트를 처리할 ClickListener 구현 클래스(로컬 클래스)

    }
}
```

✓ 다음처럼 RemoteControl 인터페이스를 정의하세요.

```
package ch09.sec07.exam02;

public interface RemoteControl {
    //추상 메소드
    void turnOn();
    void turnOff();
}
```

## 중첩 선언과 익명 객체

✓ 다음 클래스를 주석에 맞게 익명 구현 객체를 생성하여 필드에 대입하는 코드를 정의하세요.

- use1() 메서드를 호출 했을 때 출력 내용은 다음과 같다.

```
TV를 켭니다.  
TV를 끕니다.
```

```
package ch09.sec07.exam02;  
  
public class Home {  
    //필드에 익명 구현 객체 대입  
  
    //메소드(필드 이용)  
    public void use1() {  
        rc.turnOn();  
        rc.turnOff();  
    }  
}
```

- ✓ 앞에서 만든 인터페이스가 올바르게 동작하는지 확인할 수 있도록 완성하세요.

```
package ch09.sec07.exam02;  
  
public class HomeExample {  
    public static void main(String[] args) {  
  
    }  
}
```

TV를 켭니다.  
TV를 끕니다.