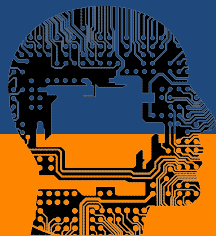




한국IT진흥부설

정보보호교육학원 아이섹



자바 프로그래밍 기초

static, final

By SoonGu Hong(Kokono)

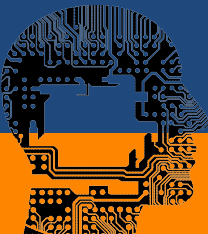


한국IT진흥부설

정보보호교육학원 아이섹

자바 프로그래밍 기초

1. static



1-1. static

- static이란 직역하면 '고정된'이라는 의미를 갖고 있습니다.
- 자바에서는 static을 사용 제한자로 이용하며 필드, 메서드에 붙여서 정적 멤버로 만들 때 사용합니다.
- 정적 멤버란 객체에 속한 멤버가 아닌 **클래스에 고정된 멤버**로서 **객체를 생성하지 않고** 사용할 수 있는 필드와 메서드를 말합니다.

```
class A {  
  
    int x;    //인스턴스 필드  
  
    //인스턴스 메서드  
    void instanceMet() {}  
}
```

```
class B {  
    //정적 필드  
    static int y;  
  
    //정적 메서드  
    static void staticMet() {}  
}
```

1-2. 정적 멤버와 인스턴스 멤버의 차이

- 인스턴스 멤버는 객체에 소속된 멤버로서 객체당 각각 다른 값을 가질 수 있으나 정적 멤버는 클래스에 소속된 멤버이기 때문에 **객체 간 모든 값을 공유**하는 공유데이터가 됩니다.

```
class A {  
  
    int x;    //인스턴스 필드  
  
    //인스턴스 메서드  
    void instanceMet() {  
    }  
}
```

```
A a = new A();  
  
a.x = 1;  
a.instanceMet();
```

반드시
객체 생성 후
멤버에 접근!

```
class B {  
  
    //정적 필드  
    static int y;  
  
    //정적 메서드  
    static void staticMet() {  
    }  
}
```

```
//B b = new B(); //필요없음  
  
B.y = 10;  
B.staticMethod();
```

객체 생성 없이
클래스 이름만으로
접근!

1-3. 정적 초기화 블록

- 인스턴스 필드는 생성자를 통해 초기화할 수 있지만, 정적 필드는 객체 생성 없이 사용하기 때문에 생성자에서 초기화 작업을 하는 것이 올바르지 않습니다.
- 따라서 자바에서는 정적 필드 초기화 작업을 위해 정적 블록을 제공합니다. 정적블록은 클래스가 로딩될 때 자동으로 실행됩니다.

```
class B {  
    //정적 필드  
    static int y;  
  
    //정적 블록  
    static {  
        y = 100;  
    }  
}
```

1-4. 정적 메서드와 정적 블록 사용시 주의사항

- 정적 메서드 블록과 정적 초기화 블록 내에서는 인스턴스 필드와 메서드를 사용할 수 없고 정적 필드와 정적 메서드만 사용할 수 있습니다.

```
class C {  
  
    int a;  
    void met() {}  
  
    void s_met() {  
  
        a = 10; (x)  
        met(); (x)  
  
    }  
}
```

```
class C {  
  
    int a;  
    void met() {}  
  
    void s_met() {  
        C c = new C();  
        c.a = 10; (o)  
        c.met(); (o)  
  
    }  
}
```

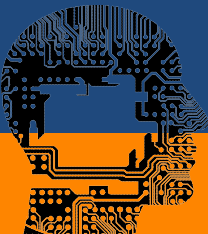


한국IT진흥부설

정보보호교육학원 아이섹

자바 프로그래밍 기초

2. final



2-1. final class

- final은 “마지막의” 라는 뜻을 가진 단어로서, 자바에서는 클래스, 필드, 메서드에 붙여서 사용할 수 있습니다.
- 클래스에 final이 붙으면 마지막 클래스라는 의미가 되므로 이것을 상속해서 확장시키지 마라! 라는 의미로 사용됩니다. 즉, final클래스는 **상속이 불가능**해집니다.

```
final class A {
```

```
}
```

```
class B extends A {  
    //불가능!!
```

```
}
```


2-2. final method

- 메서드에 final이 붙으면 이것은 최종의 메서드야! 라는 의미가 되므로 최종본은 건드리면 안되겠죠?
- 즉, final메서드는 상속시 자식클래스가 **재정의(오버라이딩)**할 수 없게 됩니다.

```
class A {  
  
    final void fMethod() {}  
  
}
```

```
class B extends A {  
  
    @Override  
    void fMethod() {}  
    //불가능!!  
}
```

2-3. final field

- 필드에 final이 붙으면 최종의 필드라는 뜻이 되므로, 한번 값이 저장되면 그것이 최종본이 된다는 의미가 됩니다.
- 즉, 한번 값을 저장하면 이후로는 절대로 **변경이 불가능**해집니다.

```
class A {  
    final String nation = "한국";  
}
```

```
main() {  
    A a = new A();  
    a.nation = "중국"; (X)  
    //값 변경 불가능!  
}
```

2-4. 상수(static final)

- 상수는 변수와 달리 불변의 값을 의미합니다.
- 불변의 값이란 원주율 파이나 지구의 반지름과 같은 것을 말합니다. 그렇다면 final필드를 상수라고 볼 수 있을까요??
- 아닙니다. 불변의 값이란 값이 변하지 않는 것과 동시에 유일해야 합니다. final필드는 값 변경만 안될 뿐, 객체별로 다른 값을 가질 수 있기 때문에 상수 조건에 어긋납니다.
- 그래서 자바에서는 객체별 유일한 값(static), 값 변경불가(final)을 만족시키기 위해 필드에 static과 final을 함께 붙여 상수로 취급합니다.

감사합니다
THANK YOU