

# 자바 제어자 (modifier) #2

**final, abstract**

# 자바 제어자 (modifier) #2 - final

final / abstract

# 자바 제어자 (modifier) #2 - final

## 1 🖱️ final 필드, final 지역변수 - 처음 지정된 값을 바꿀 수 없음

- final 필드

2

```
class A1 {  
    int a = 3;  
    final int b = 5;  
    A1(){  
  
    }  
}
```

3

```
class A2 {  
    int a;  
    final int b;  
    A2(){  
        a=3;  
        b=5;  
    }  
}
```

4

```
class A3 {  
    int a=3;  
    final int b=5;  
    A3(){  
        a=7;  
        b=9; (불가능)  
    }  
}
```

- final 지역변수

6

```
class B {  
    void bcd() {  
        int a=3;  
        final int b=5;  
        a=7;  
        //b=9; //(불가능)  
    }  
}
```

필드값에 바로 값을  
대입한 경우



생성자에서 필드 값을  
초기화한 경우

5

```
A1 a1 = new A1();  
a1.a = 7;  
//a1.b = 9; //(불가능)  
  
A2 a2 = new A2();  
a2.a = 7;  
//a2.b = 9; //(불가능)
```

7

### CHECK

- final 필드/지역변수는 값의 변경 뿐만 아니라 **추가적인 값의 대입 자체가 불가능**

ex.

```
final int a=3;  
a=3; //(불가능)
```

# 자바 제어자 (modifier) #2 - final

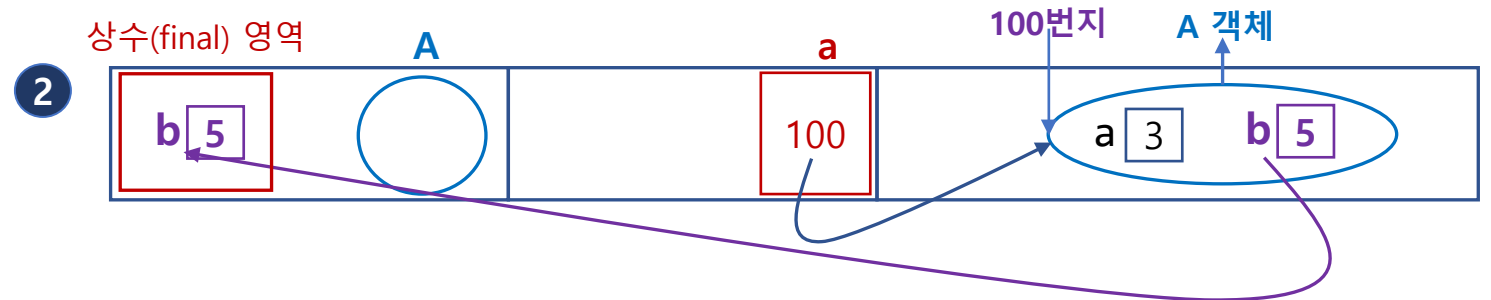
☞ final 필드, final 지역변수 - 처음 지정된 값을 바꿀 수 없음

- final 필드/지역변수의 메모리 구조

1

```
class A {  
    int a;  
    final int b;  
    A () {  
        a=3; b=5;  
    }  
}
```

A a = new A();

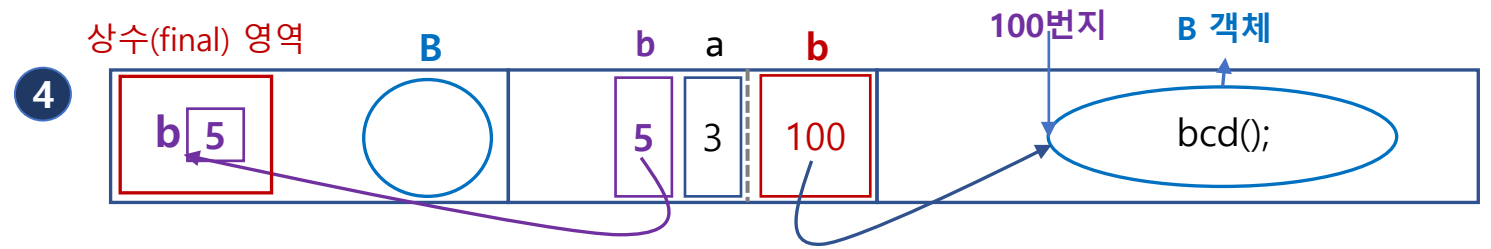


객체 생성시 final 필드는 final 영역에 복사

3

```
class B {  
    void bcd() {  
        int a=3;  
        final int b=5;  
    }  
}
```

B b = new B();  
b.bcd();



Stack 메모리에 final 지역변수 초기화시 final 영역에 복사

# 자바 제어자 (modifier) #2 - final

☞ final 메서드, final 클래스

3

**Class String**

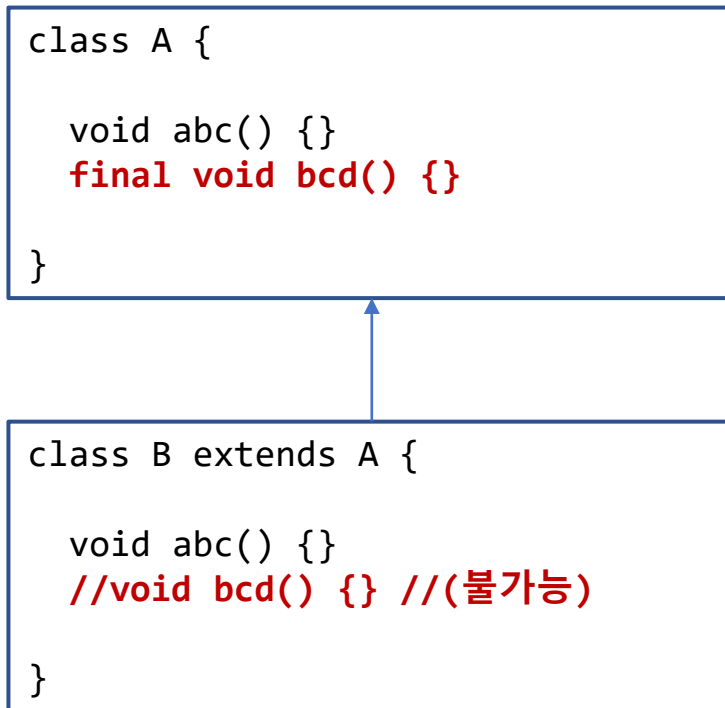
java.lang.Object  
java.lang.String

**All Implemented Interfaces:**  
Serializable, CharSequence, Comparable<String>

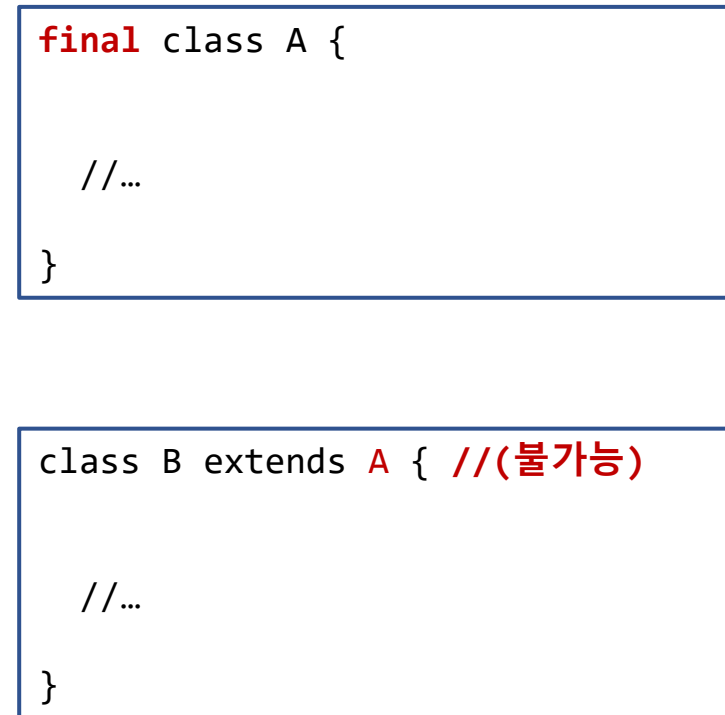
public final class String  
extends Object  
implements Serializable, Comparable<String>, CharSequence

The String class represents character strings. All string literals in Java  
Strings are constant: their values cannot be changed after they are created.

1 - final 메서드 - 상속시 Override 불가

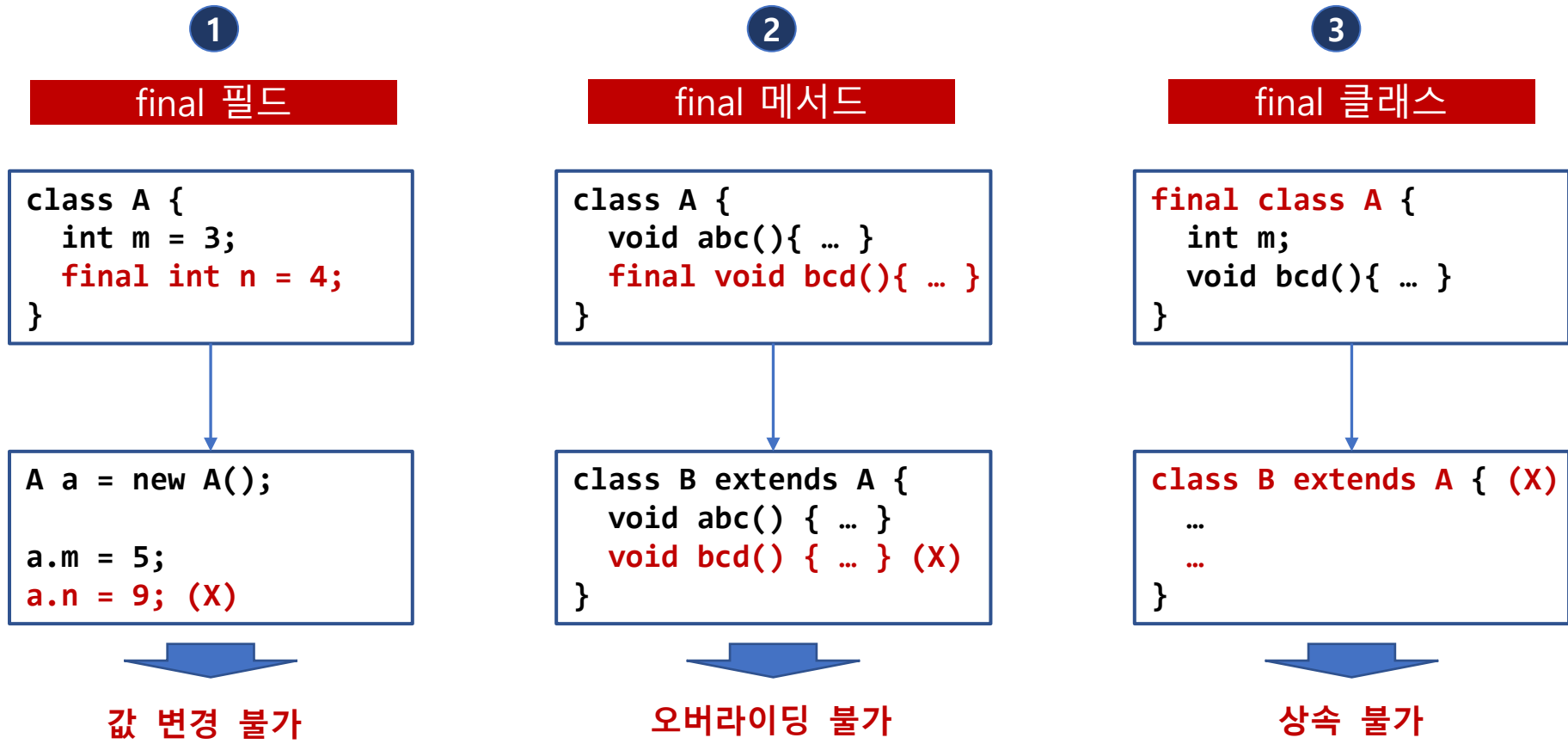


2 - final 클래스 - 상속 자체 불가



# 자바 제어자 (modifier) #2 - final

☞ final 필드/메서드/클래스 메서드 최종정리



# 자바 제어자 (modifier) #2 - abstract

**abstract** 미국식  명사 ['æbstrækt] 동사 [æb'strækt]  형용사 ['æbstrækt] 영국식

[다른 뜻\(1건\)](#) | [예문보기](#)

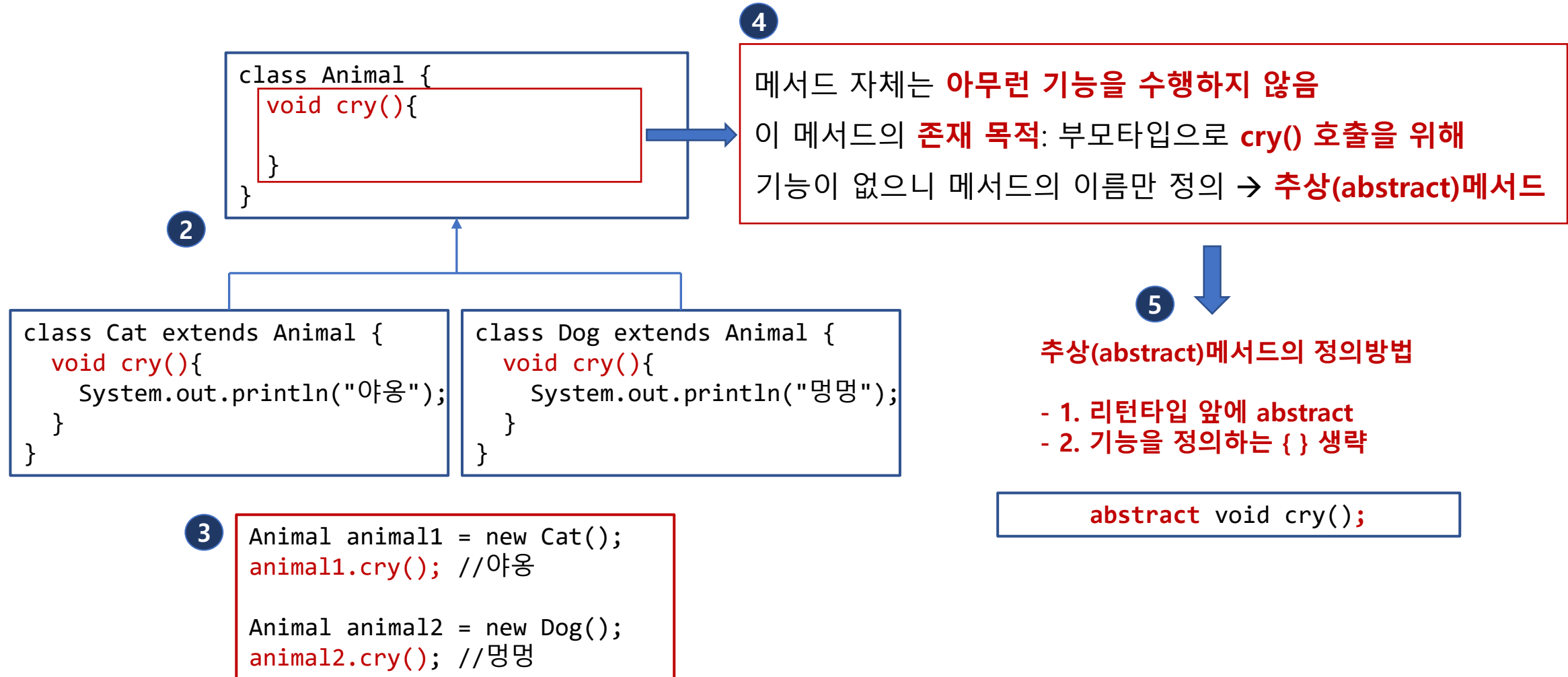
1, 추상적인 2, 관념적인, 추상적인 3, 추상적인

# 자바 제어자 (modifier) #2 - abstract

6

```
abstract class Animal {  
    abstract void cry();  
}
```

☞ abstact(추상) 메서드 ① - 완성되지 않은 미완성 메서드





# 자바 제어자 (modifier) #2 - abstract

abstract class는 자식 클래스 없이 객체 생성 불가

☞ abstact(추상) 클래스

1 - 내부에 추상메서드(미완성메서드)를 하나이상 포함하는 클래스

3

```
class Animal {  
    void cry(){  
    }  
}
```

추상클래스로 대체

4

```
abstract class Animal {  
    abstract void cry();  
}
```

2

추상(abstract)클래스의 정의방법

- 클래스 앞에 abstract

```
abstract class Animal {  
    abstract void cry();  
}
```

```
class Cat extends Animal {  
    void cry(){  
        System.out.println("야옹");  
    }  
}
```

```
class Dog extends Animal {  
    void cry(){  
        System.out.println("멍멍");  
    }  
}
```

5

```
Animal animal1 = new Cat();  
animal1.cry(); //야옹  
  
Animal animal2 = new Dog();  
animal2.cry(); //멍멍
```

# 자바 제어자 (modifier) #2 - abstract

☞ abstact(추상) 클래스

- 내부에 추상메서드(미완성메서드)를 하나이상 포함하는 클래스

1

일반클래스로 정의한 경우

```
class Animal {  
    void cry(){  
    }  
}
```

```
class Cat extends Animal {  
    void cRy(){  
        System.out.println("야옹");  
    }  
}
```

2

```
Animal animal1 = new Cat();  
animal1.cry(); //출력없음
```

→ 엉뚱한 결과

3

추상(abstract) 클래스로 정의한 경우

```
abstract class Animal {  
    abstract void cry();  
}
```

```
class Cat extends Animal {  
    void cRy(){  
        System.out.println("야옹");  
    }  
}
```

→ 오타발생

4

오류 발생

Q. 오류가 발생하는 이유는 무엇일까?

# The End