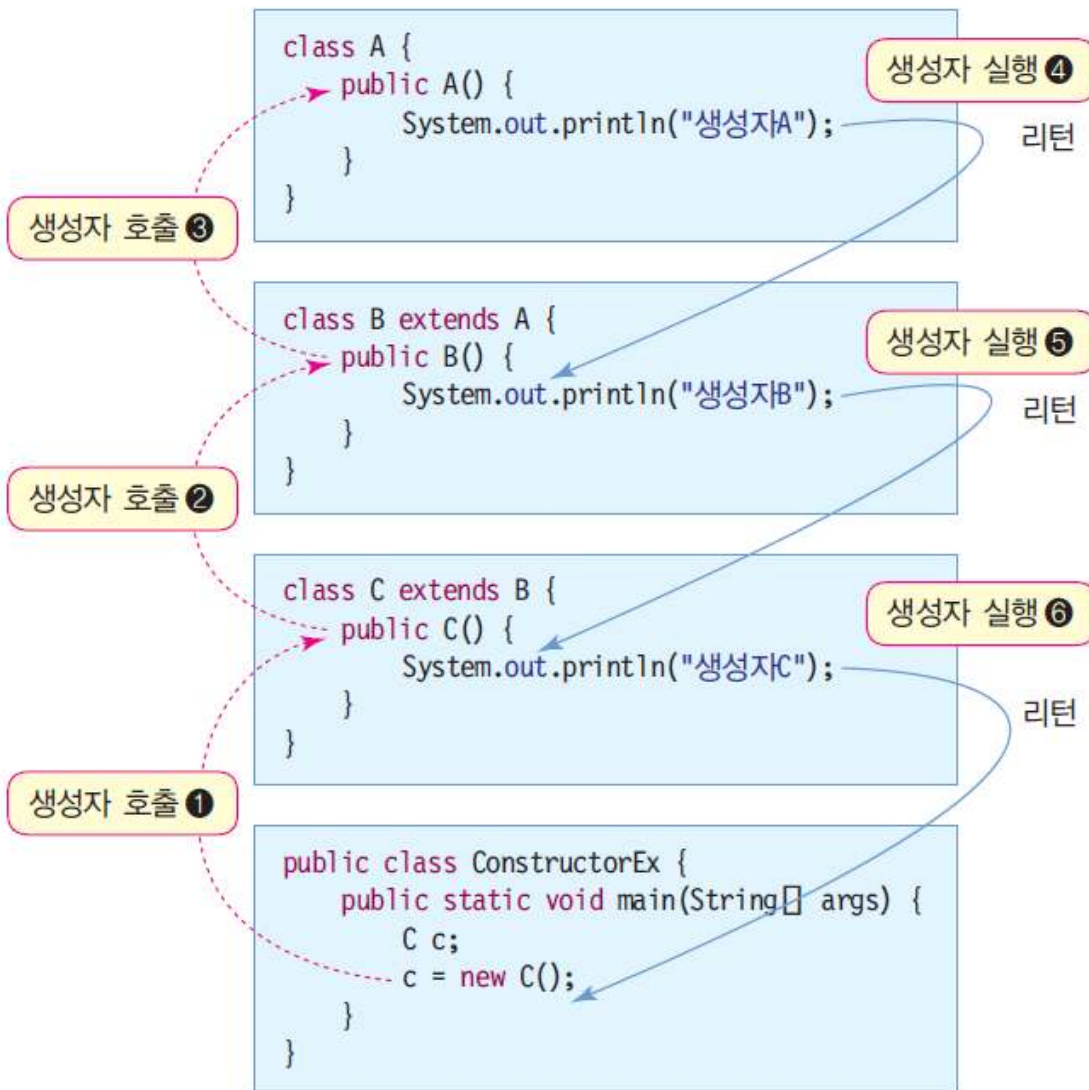


슈퍼클래스와 서브 클래스의 생성자간의 호출 및 실행 관계

17



예상 실행 결과는 ?

생성자A
생성자B
생성자C

서브 클래스와 슈퍼 클래스의 생성자 짝 맞추기

18

- 슈퍼 클래스와 서브 클래스
 - ▣ 각각 여러 개의 생성자 작성 가능
- 슈퍼 클래스와 서브 클래스의 생성자 사이의 짝 맞추기
 - ▣ 서브 클래스와 슈퍼 클래스의 생성자 조합 4 가지

경우	1	2	3	4
서브 클래스	기본 생성자	기본 생성자	매개 변수를 가진 생성자	매개 변수를 가진 생성자
슈퍼 클래스	기본 생성자	매개 변수를 가진 생성자	기본 생성자	매개 변수를 가진 생성자

- ▣ 서브 클래스에서 슈퍼 클래스의 생성자를 선택하지 않는 경우
 - 컴파일러가 자동으로 슈퍼 클래스의 기본 생성자 선택
- ▣ 서브 클래스 개발자가 슈퍼 클래스의 생성자를 명시적으로 선택하는 경우
 - `super()` 키워드를 이용하여 선택

슈퍼 클래스의 기본 생성자 자동 호출 - 서브 클래스의 기본 생성자 경우

19

서브 클래스의 생성자가
슈퍼 클래스의 생성자를
선택하지 않은 경우

```
class A {  
    public A() {  
        System.out.println("생성자A");  
    }  
    public A(int x) {  
        .....  
    }  
}
```

서브클래스의 생성자가 기본 생성자인 경우, 컴파일러는 자동으로 슈퍼클래스의 기본 생성자와 짝을 맺음

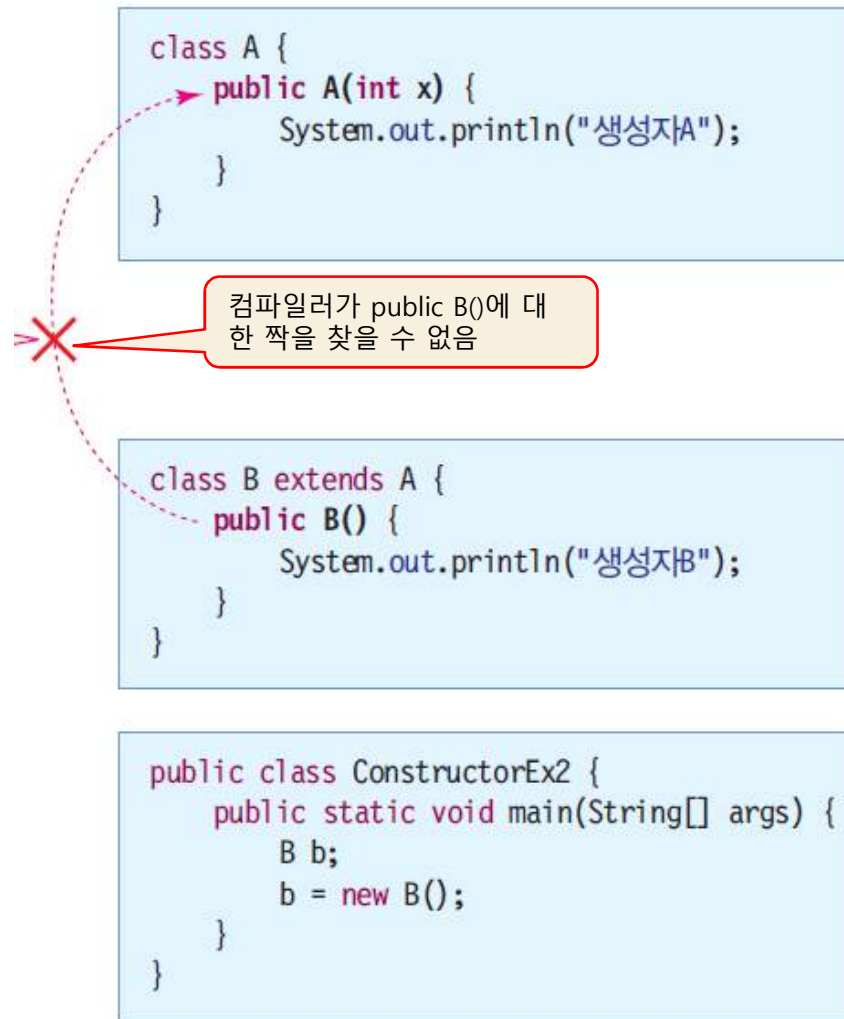
```
class B extends A {  
    public B() {  
        System.out.println("생성자B");  
    }  
}
```

```
public class ConstructorEx2 {  
    public static void main(String[] args) {  
        B b;  
        b = new B(); // 생성자 호출  
    }  
}
```

생성자A
생성자B

슈퍼 클래스에 기본 생성자가 없어 오류 난 경우

20



컴파일러에 의해 "Implicit super constructor A() is undefined.
Must explicitly invoke another constructor" 오류 발생

슈퍼 클래스의 기본 생성자 자동 호출 - 서브 클래스의 매개 변수를 가진 생성자 경우

21

서브 클래스의 생성자가
슈퍼 클래스의 생성자를
선택하지 않은 경우

```
class A {  
    ➔ public A() {  
        System.out.println("생성자A");  
    }  
    public A(int x) {  
        System.out.println("매개변수생성자A");  
    }  
}
```

```
class B extends A {  
    public B() {  
        System.out.println("생성자B");  
    }  
    ➔ public B(int x) {  
        System.out.println("매개변수생성자B");  
    }  
}
```

```
public class ConstructorEx3 {  
    public static void main(String[] args) {  
        B b;  
        b = new B(5);  
    }  
}
```

생성자A
매개변수생성자B

super()를 이용하여 슈퍼 클래스 생성자 선택

22

□ super()

- ▣ 서브 클래스에서 명시적으로 슈퍼 클래스의 생성자를 선택 호출할 때 사용
- ▣ 사용 방식
 - `super(parameter);`
 - 인자를 이용하여 슈퍼 클래스의 적당한 생성자 호출
 - 반드시 서브 클래스 생성자 코드의 제일 첫 라인에 와야 함

super()를 이용한 사례

23

```
class A {  
    public A() {  
        System.out.println("생성자A");  
    }  
    public A(int x) {  
        System.out.println("매개변수생성자A" + x);  
    }  
}
```

```
class B extends A {  
    public B() {  
        System.out.println("생성자B");  
    }  
    public B(int x) {  
        super(x);  
        System.out.println("매개변수생성자B" + x);  
    }  
}
```

super(); 라고 하면 A() 호출

```
public class ConstructorEx4 {  
    public static void main(String[] args) {  
        B b;  
        b = new B(5);  
    }  
}
```

매개변수생성자A5
매개변수생성자B5