

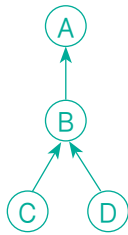
실력 상승!

연습 문제

Q1 생성자의 2가지 문법적 조건은 무엇인가?

- a. 클래스의 이름과 동일
- b. 리턴타입이 없다

Q2 다음과 같은 상속 구조도가 있을 때 이를 코드로 작성하시오(클래스의 내용은 작성하지 않음).

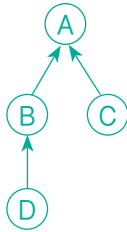


```
class A {}
```

```
class B extends A{}
class C extends B{}
class D extends B{}

```

Q3 다음과 같은 상속 관계에서 다형적 표현을 사용한 객체 생성 코드 중 올바른 것은 ○, 잘못된 것은 X에 표시하시오.



// 다형적 표현

A a1 = new A(); ○ (○ / X)

A a2 = new B(); ○ (○ / X)

A a3 = new C(); ○ (○ / X)

A a4 = new D(); ○ (○ / X)

B b1 = new A(); X (○ / X)

B b2 = new B(); ○ (○ / X)

B b3 = new C(); X (○ / X)

B b4 = new D(); ○ (○ / X)

C c1 = new A(); X (○ / X)

C c2 = new B(); X (○ / X)

C c3 = new C(); ○ (○ / X)

C c4 = new D(); X (○ / X)

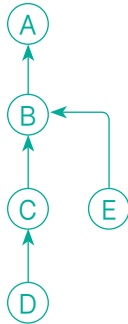
D d1 = new A(); X (○ / X)

D d2 = new B(); X (○ / X)

D d3 = new C(); X (○ / X)

D d4 = new D(); ○ (○ / X)

Q4 상속 구조가 다음과 같다. 다음 중 타입 변환이 올바른 것은 ○, 잘못된 것은 X에 표시하시오.



```

A a1 = new A();
B b1 = (B)a1; X (○ / X)
C c1 = (C)a1; X (○ / X)
D d1 = (D)a1; X (○ / X)
E e1 = (E)a1; X (○ / X)
  
```

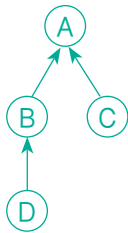
```

A a2 = new B();
B b2 = (B)a2; ○ (○ / X)
C c2 = (C)a2; X (○ / X)
D d2 = (D)a2; X (○ / X)
E e2 = (E)a2; X (○ / X)
  
```

```

A a3 = new C();
B b3 = (B)a3; ○ (○ / X)
C c3 = (C)a3; ○ (○ / X)
D d3 = (D)a3; X (○ / X)
E e3 = (E)a3; X (○ / X)
  
```

Q5 상속 구조가 다음과 같을 때 알맞은 출력 결과를 고르시오.



```

A aa = new A();
System.out.println(aa instanceof A); (true / false) true
System.out.println(aa instanceof B); (true / false) false
System.out.println(aa instanceof C); (true / false) false
System.out.println(aa instanceof D); (true / false) false
  
```

```

A ad = new D();
System.out.println(ad instanceof A); (true / false) true
System.out.println(ad instanceof B); (true / false) true
System.out.println(ad instanceof C); (true / false) false
System.out.println(ad instanceof D); (true / false) true
  
```

Q6 클래스 A와 B가 다음과 같다.

```
class A {  
    void hello() {  
        System.out.println("안녕하세요");  
    }  
}  
class B extends A {  
    void hello() {  
        System.out.println("반갑습니다");  
    }  
}
```

다음 실행 코드의 결과를 쓰시오.

```
A aa = new A();  
aa.hello();  
  
B bb = new B();  
bb.hello();  
  
A ab = new B();  
ab.hello();
```

실행 결과

안녕하세요
반갑습니다
반갑습니다

Q7 다음과 같이 클래스 A를 상속받아 B, C, D, E 클래스를 생성하고자 한다. 다음 중 오류를 포함하고 있는 클래스는 무엇이고, 오류가 발생한 이유는 무엇인지 쓰시오.

```
class A {
    void method() {
    }
}
class B extends A {
    public void method() {
    }
}
class C extends A {
    protected void method() {
    }
}
class D extends A {
    void method() {
    }
}
class E extends A {
    private void method() {
    }
}
```

오류를 포함하고 있는 클래스명	오류가 발생한 이유
E	오버라이딩시 접근지정자의 범위는 좁아질 수 없다.

Q8 다음과 같이 인스턴스 멤버와 정적 멤버를 포함하고 있는 클래스 A와 B가 있다.

```
class A {
    int m = 2;
    static int n = 4;
    void method1() {
        System.out.println("A 클래스 instance method");
    }
    static void method2() {
        System.out.println("A 클래스 static method");
    }
}
class B extends A {
    int m = 6;
    static int n = 8;
    void method1() {
        System.out.println("B 클래스 instance method");
    }
    static void method2() {
        System.out.println("B 클래스 static method");
    }
}
```

다음 실행 코드의 결과를 쓰시오.

```
A ab = new B();
System.out.println(ab.m);
System.out.println(ab.n);
ab.method1();
ab.method2();
```

실행 결과

```
2
4
B 클래스 instance method
A 클래스 static method
```

Q9 다음과 같이 클래스 B는 클래스 A를 상속한 후 내부에 아무것도 추가하지 않은 상태다. 이때 오류가 발생하는데, 오류가 발생한 이유를 설명하시오.

```
class A {
    A(int a) {
        System.out.println("A 생성자");
    }
}
class B extends A {    // 오류 발생
```

- 컴파일러가 자동으로 추가해주는 생성자의 첫 줄에 `super()` 메서드가 호출되며 이는 부모클래스의 기본 생성자를 호출함.
- 그런데 부모클래스에 기본생성자가 없어서 오류 발생

Q10 클래스 A와 B의 구조는 다음과 같다.

```
class A {
    A() {
        System.out.println("A 생성자1");
    }
    A(int a) {
        this();
        System.out.println("A 생성자2");
    }
}
class B extends A {
    B() {
        System.out.println("B 생성자1");
    }
    B(int a) {
        super(a);
        System.out.println("B 생성자2");
    }
}
```

다음 코드의 실행 결과를 쓰시오.

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

A 생성자1
A 생성자2
B 생성자2

Q11 다음과 같은 클래스 A가 있다.

```
class A {  
    int data;  
    A(int data) {  
        this.data = data;  
    }  
}
```

다음 코드의 실행 결과로 false가 출력되는 이유를 설명하시오.

```
A a1 = new A(3);  
A a2 = new A(3);  
System.out.println(a1.equals(a2));    // false
```

- A 클래스 내부의 equals() 메서드는 오버라이딩을
하지 않아 Object 클래스의 equals() 메서드
- 이는 스택메모리의 값 비교 즉, 객체의 위치를 비교
- 따라서 두 객체의 생성위치가 서로 다르기 때문에
false값이 출력

Q12 **Q11**에서 출력값이 true가 나오도록 클래스 A를 수정하시오.

```
class A {  
    int data;  
    A(int data) {  
        this.data = data;  
    }  
  
    @Override  
    public boolean equals(Object obj) {  
        if(obj instanceof A) {  
            return this.data == ((A)obj).data;  
        }  
        return false;  
    }  
}
```

```
A a1 = new A(3);  
A a2 = new A(3);  
System.out.println(a1.equals(a2));    // true
```