

-- 1. 다음 중 생성자에 대한 설명으로 옳지 않은 것은? (\*\*)

- A. 모든 생성자의 이름은 클래스의 이름과 동일하다.
- B. 생성자는 객체를 생성하기 위한 것이다.
- ☒ C. 클래스에는 기본 생성자가 반드시 하나 이상 있어야 한다.
- D. 생성자가 없는 클래스는 컴파일러가 기본 생성자를 추가한다.
- E. 생성자는 오버로딩 할 수 없다.

-- 2. 다음 중 오버로딩 조건으로 옳지 않은 것은? (\*\*)

- A. 메서드의 이름이 같아야 한다.
- B. 매개변수의 개수나 타입이 달라야 한다.
- ☒ C. 리턴타입이 달라야 한다.
- D. 매개변수의 이름이 달라야 한다.

-- 3. 다음 중 오버라이딩 조건으로 옳지 않은 것은?

- A. 조상의 메서드와 이름이 같아야 한다.
- B. 매개변수의 수와 타입, 리턴타입이 모두 같아야 한다.
- C. 접근 제어자는 조상의 메서드보다 좁은 범위로만 변경할 수 있다.
- ☒ D. 조상의 메서드보다 더 많은 수의 예외를 선언할 수 있다.

-- 4. 다음 중 아래 add메서드를 잘못 오버로딩 한 것은?

- `long add(int a, int b) { return a+b;}`
- ☒ `long add(int x, int y) { return x+y;}`
- B. `long add(long a, long b) { return a+b;}`
- C. `int add(byte a, byte b) { return a+b;}`
- D. `int add(long a, int b) { return (int)(a+b);}`

-- 5. 다음 중 초기화에 대한 설명으로 옳지 않은 것은?

- A. String 멤버변수는 ""로 자동 초기화된다.
- B. 지역변수는 반드시 초기화해야 한다.
- ☒ C. 생성자보다 초기화 블록이 먼저 수행된다.
- D. 인스턴스변수보다 클래스변수가 먼저 초기화된다.

-- 6. 다음 중 지역변수에 대한 설명으로 옳지 않은 것은?

- ☒ A. 자동 초기화되므로 별도의 초기화가 필요없다.
- B. 지역변수가 선언된 메서드가 종료되면 지역변수도 함께 소멸된다.
- C. 매서드의 매개변수로 선언된 변수도 지역변수이다.
- D. 스택(stack)영역에 생성되며 가비지 컬렉터에 의해 소멸된다.

-- 7. 다음 중 접근 제어자에 대한 설명으로 옳지 않은 것은?

- A. public은 접근제한이 전혀 없는 접근 제어자이다.
- B. (default)가 붙으면, 같은 패키지 내에서만 접근이 가능하다.
- ☒ C. 지역변수에도 접근 제어자를 사용할 수 있다.
- D. protected가 붙으면, 같은 패키지 내에서 또는 다른 패키지의 자손 클래스에서 접근이 가능하다.

-- 8. 다음 중 인터페이스 설명으로 옳지 않은 것은?

- A. 표준화를 가능하게 해준다.
- B. 서로 관계없는 클래스들에게 관계를 맺어줄 수 있다.
- ☒ C. 독립적인 프로그래밍이 가능하다.
- D. 다중상속을 가능하게 해준다.
- E. 패키지간의 연결을 도와준다.

-- 9. 다음 설명 중 맞는 것을 2개 고르시오.

- A. 클래스에서 인터페이스는 단일 implements만 가능하다.
- ☒ B. 같은 이름의 메소드를 여러개 만드는 것을 오버로딩 이라한다.
- ☒ C. 상위 클래스의 메소드를 하위 클래스에서 재정의 하는 것을 오버라이딩이라 한다.
- D. 클래스의 상속시 생성자도 상속이 된다.

-- 10. 자바의 상속 특징 중에서 틀린 것은?

- A. 자바에서는 클래스의 다중 상속을 지원하지 않는다.
- B. 자바에서는 상속의 횟수에 제한이 없다.
- ☒ C. 자바에서 모든 클래스의 최상위 슈퍼 클래스는 java.lang.System 클래스이다.
- D. 자바에서 상속을 표현하는 키워드는 extends이다.

-- 11. 다음 A 인터페이스를 상속받아 B 인터페이스를 만들기 위한 코드는?

```
public interface A {  
    String DEFAULT_GREETING = "Hello World";  
    void method1();  
}
```

A

- A. public interface B extends A { }
- B. public interface B implements A { }
- C. public class B extends A { }
- D. public class B implements A { }

-- 12. 다음이 설명하는 접근 지정자는 무엇인지 쓰시오.

- A. 모든 클래스에서 접근 가능 public
- B. 같은 클래스 내부멤버에서만 접근 가능 private
- C. 같은 패키지내의 모든 클래스에서만 접근 가능 default
- D. 같은 패키지과 상속받은 하위 클래스에서만 접근 가능 protected

-- 13. 다음 중 1), 2), 3), 4) 메소드가 주석 위치에 왔을 때, 에러가 나는 메소드는?

```
class Super {  
    public float getNum() { return 3.0f; }  
}  
  
public class Sub extends Super {  
    /* */  
}
```

- A. public void getNum(double d) { }
- B. public float getNum() { return 4.0f; }
- C. public void getNum() { }
- D. public double getNum(float d) { return 4.0d; }

-- 14. 다음 코드의 실행 결과는?

```
class MyClass{  
    int x= 12;  
    public void method(int x) {  
        x+=x;  
        System.out.println(x);  
    }  
}  
  
public class Test {  
    public static void main(String[] args) {  
        Test t = new Test();  
        t.method(5);  
    }  
}
```

10

- A. 5
- B. 10
- C. 12
- D. 17
- E. compile error

--15. 다음 코드의 실행 결과는?

```
class Base {  
    Base() { System.out.println("콜"); }  
}
```

콜  
콜

```
public class Alpha extends Base {  
    public static void main( String[] args ) {  
        Alpha aa = new Alpha();  
        Base bb = new Base();  
    }  
}
```

-- 16. 다음 코드의 실행 결과는?

```
class Change {  
    static String str = "ABC123";  
    public static void change(String str) {  
        Change.str += "456";  
    }  
    public static void main(String[] args) {  
        String str = "ABC123";  
        System.out.println(str);  
        change(str);  
        System.out.println("After:"+Change.str);  
    }  
}
```

ABC123  
ABC123456

-- 17. 다음 코드의 실행 결과는?

```
public class Test {  
    public int aMethod() {  
        int i = 0;  
        i++;  
        return i;  
    }  
    public static void main (String args[]) {  
        Test test = new Test();  
        test.aMethod();  
        int j = test.aMethod();  
        System.out.println(j);  
    }  
}
```

1

-- 19. 다음 코드의 실행 결과는?

```
class Super {
    public Integer getLenght() { return new Integer(4); }
}

public class Sub extends Super {
    public Long GetLenght() { return new Long(5); }

    public static void main(String[] args) {
        Super sooper = new Super();
        Sub sub = new Sub();

        System.out.println(sooper.getLenght().toString()+
            sub.getLenght().toString() );
    }
}
```

- A. 4,4
- B. 4,5
- C. 5,4
- D. 5,5

A

-- 20. 다음 코드의 실행 결과는?

```
class Super {
    public int i = 0;
    public Super(String text) {
        i = 1;
    }
}

public class Sub extends Super {
    public Sub(String text) {
        i = 2;
    }

    public static void main(String args[]) {
        Sub sub = new Sub("Hello");
        Super("dddd");
        System.out.println(sub.i);
    }
}
```

- A. 0
- B. 1
- C. 2
- D. Compilation fails.

D

-- (문 21~22) 다음 코드를 보고 물음에 답하십시오.

```
class Shape {
    public void draw() {
        System.out.println(" Shape ");
    }
}

class Circle extends Shape {
    public void draw() {
        System.out.println(" Circle ");
    }

    public void sayCircle() {
        System.out.println(" Hello Circle ");
    }
}

public class ClassTest {
    public static void main(String[] args) {
        Shape s = new Circle();
        Circle c = new Circle();
        s.draw();    -- (1)
        s.sayCircle();    -- (2)
        c.draw();    -- (3)
        c.sayCircle();    -- (4)
    }
}
```

-- 21. (1)~(4) 중 호출 시 에러가 발생하는 라인은?

- A. (1)
- B. (2)
- C. (3)
- D. (4)

2

-- 22. 위 21번 문항의 에러가 발생하지 않을 경우 (1)과 (3)라인의 코드 실행 결과는?

- A. Shape Circle
- B. Circle Shape
- C. Shape Shape
- D. Circle Circle

D