클래스 / 인터페이스 개념 문제

클래스 개념 문제

1. 자바의 상속에 대한 설명 중 틀린 것은 무엇입니까?

❶ 자바는 다중 상속을 허용한다.

❷ 부모의 메소드를 자식 클래스에서 재정의(오버라이딩)할 수 있다.

❸ 부모의 private 접근 제한을 갖는 필드와 메소드는 상속의 대상이 아니다.

❹ final 클래스는 상속할 수 없고, final 메소드는 오버라이딩할 수 없다.

2. 클래스 타입 변환에 대한 설명 중 틀린 것은 무엇입니까?

❶ 자식 객체는 부모 타입으로 자동 타입 변환된다.

❷ 부모 객체는 어떤 자식 타입으로도 강제 타입 변환된다.

❸ 자동 타입 변환을 이용해서 필드와 매개변수의 다형성을 구현한다.

❹ 강제 타입 변환 전에 instanceof 연산자로 변환 가능한지 검사하는 것이 좋다.

3. final 키워드에 대한 설명으로 틀린 것은 무엇입니까?

❶ final 클래스는 부모 클래스로 사용할 수 있다.

❷ final 필드는 초기화된 후에는 변경할 수 없다.

❸ final 메소드는 재정의(오버라이딩)할 수 없다.

❹ static final 필드는 상수를 말한다.

4. 오버라이딩(Overriding)에 대한 설명으로 틀린 것은 무엇입니까?

❶ 부모 메소드의 시그너처(리턴 타입, 메소드명, 매개변수)와 동일해야 한다.

❷ 부모 메소드보다 좁은 접근 제한자를 붙일 수 없다. (예: public (부모) ➜ private (자식)).

❸ @Override 어노테이션을 사용하면 재정의가 확실한지 컴파일러가 검증한다.

❹ protected 접근 제한을 갖는 메소드는 다른 패키지의 자식 클래스에서 재정의할 수 없다.

5. 추상 클래스에 대한 설명으로 틀린 것은 무엇입니까?

❶ 직접 객체를 생성할 수 없고, 상속만 할 수 있다.

❷ 추상 메소드를 반드시 가져야 한다.

❸ 추상 메소드는 자식 클래스에서 재정의(오버라이딩)할 수 있다.

❹ 추상 메소드를 재정의하지 않으면 자식 클래스도 추상 클래스가 되어야 한다.

6. Parent 클래스를 상속해서 Child 클래스를 다음과 같이 작성했는데, Child 생성자에서 컴파일 에러가 발생했습니다. 그 이유와 해결 방법을 설명해보세요.

```
public class Parent {
  public String name;

  public Parent(String name) {
    this.name = name;
  }
}
```

```
public class Child extends Parent {
  public int studentNo;

  public Child(String name, int studentNo) {
    this.name = name;
    this.studentNo = studentNo;
  }
}
```

7. Parent 클래스를 상속받아 Child 클래스를 다음과 같이 작성했습니다. ChildExample 클래스를 실행했을 때 호출되는 각 클래스의 생성자의 순서를 생각하면서 출력 결과를 작성해보세요.

```
public class Parent {
  public String nation;

  public Parent() {
    this("대한민국");
    System.out.println("Parent() call");
  }

  public Parent(String nation) {
    this.nation = nation;
    System.out.println("Parent(String nation) call");
  }
}
```

```
public class Child extends Parent {
  public String name;

  public Child() {
    this("홍길동");
    System.out.println("Child() call");
  }

  public Child(String name) {
    this.name = name;
    System.out.println("Child(String name) call");
  }
}
```

```
public class ChildExample {
  public static void main(String[] args) {
    Child child = new Child();
  }
}
```

8. Tire 클래스를 상속받아 SnowTire 클래스를 다음과 같이 작성했습니다. SnowTireExample 클래스를 실행했을 때 출력 결과를 작성해보세요.

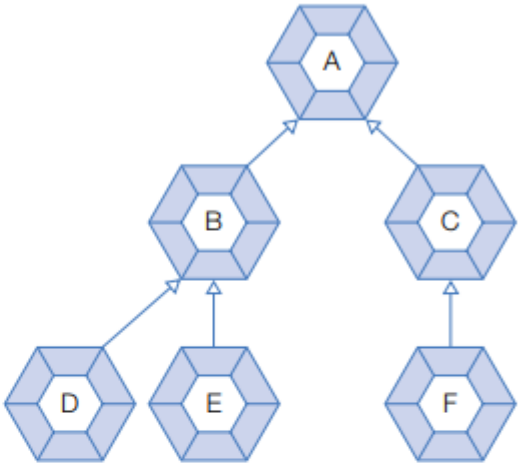
```
public class Tire {
  public void run() {
    System.out.println("일반 타이어가 굴러갑니다.");
  }
}
```

```
public class SnowTire extends Tire {
  @Override
  public void run() {
    System.out.println("스노우 타이어가 굴러갑니다.");
  }
}
```

```
public class SnowTireExample {
  public static void main(String[] args) {
    SnowTire snowTire = new SnowTire();
    Tire tire = snowTire;

    snowTire.run();
    tire.run();
  }
}
```

9. A, B, C, D, E, F 클래스가 다음과 같이 상속 관계에 있을 때 다음 빈칸에 들어올 수 없는 코드를 선택하세요.

```
//변수 대입
B b = [          ];

---------------------------------

//메소드 선언
void method(B b) { … }

//메소드 호출
method([          ])
```



❶ new B()
❷ (B) new A()
❸ new D()
❹ new E()

10. 다음과 같이 작성한 Computer 클래스에서 컴파일 에러가 발생했습니다. 그 이유를 설명해보세요.

```
public abstract class Machine {
  public void powerOn() { }
  public void powerOff() { }
  public abstract void work();
}
```

```
public class Computer extends Machine {
}
```

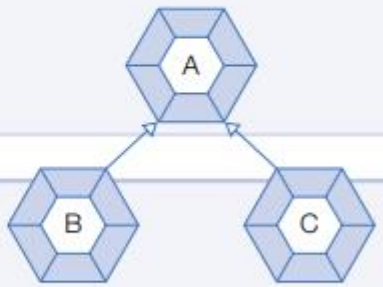
11. MainActivity의 onCreate()를 실행할 때 Activity의 onCreate()도 실행시키고 싶습니다. 밑줄에 들어갈 코드를 작성해보세요.

```
public class Activity {
  public void onCreate() {
    System.out.println("기본적인 실행 내용");
  }
}
```

```
public class MainActivity extends Activity {
    @Override
    public void onCreate() {
         _________.onCreate();
        System.out.println("추가적인 실행 내용");
    }
}
```

12. 다음과 같은 Example 클래스에서 action() 메소드를 호출할 때 매개값이 C 객체일 경우에만 method2()가 호출되도록 밑줄에 들어갈 코드를 작성해보세요.

```
public class A {
  public void method1() {
    System.out.println("A-method1()");
  }
}
```




```
public class B extends A {
  public void method1() {
    System.out.println("B-method1()");
  }
}
```

```
public class C extends A {
  public void method1() {
    System.out.println("C-method1()");
  }
  public void method2() {
    System.out.println("C-method2()");
  }
}
```

```
public class Example {
  public static void action(A a) {
    a.method1();
    if( ____________________________ ) {
          c.method2();
    }
  }

  public static void main(String[] args) {
    action(new A());
    action(new B());
    action(new C());
  }
}
```

인터페이스 개념 문제

1. 인터페이스에 대한 설명으로 틀린 것은 무엇입니까?

❶ 인터페이스로 객체(인스턴스)를 생성할 수 있다.
❷ 인터페이스는 다형성의 주된 기술로 사용된다.
❸ 인터페이스를 구현한 객체는 인터페이스로 동일하게 사용할 수 있다.
❹ 인터페이스를 사용함으로써 객체 교체가 쉬워진다.

2. 인터페이스의 구성 멤버에 대한 설명으로 틀린 것은 무엇입니까?

❶ 인터페이스는 인스턴스 필드가 없고 상수를 멤버로 가진다.
❷ 추상 메소드는 구현 클래스가 재정의해야 하는 멤버이다.
❸ 디폴트 메소드는 구현 클래스에서 재정의할 수 없다.
❹ 정적 멤버는 구현 객체가 없어도 사용할 수 있는 멤버이다.

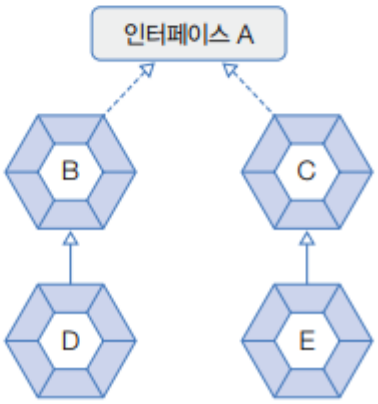
3. 인터페이스 다형성에 대한 설명으로 틀린 것은 무엇입니까?

❶ 필드가 인터페이스 타입일 경우 다양한 구현 객체를 대입할 수 있다.
❷ 매개변수가 인터페이스 타입일 경우 다양한 구현 객체를 대입할 수 있다.
❸ 배열이 인터페이스 타입일 경우 다양한 구현 객체를 저장할 수 있다.
❹ 구현 객체를 인터페이스 타입으로 변환하려면 강제 타입 변환을 해야 한다.

4. 인터페이스 A를 B와 C가 구현하고 B를 상속해서 D 클래스를, C를 상속해서 E 클래스를 만들었습니다. 다음 빈칸에 들어올 수 있는 것을 모두 선택하세요.

```
//메소드 선언
void method(A a) { … }

//메소드 호출
method(                    );
```

❶ new B()
❷ new C()
❸ new D()
❹ new E()



5. TV 클래스를 실행했을 때 "TV를 켰습니다."라고 출력되도록 밑줄과 박스에 들어갈 코드를 작성해보세요.

```
public interface Remocon {
  public void powerOn();
}
```

```
public class TV ____________________ {
  
  public static void main(String[] args) {
    Remocon r = new TV();
    r.powerOn();
  }
}
```

6. Soundable 인터페이스는 다음과 같은 sound() 추상 메소드를 가지고 있습니다. SoundableExample 클래스의 printSound() 메소드는 매개변수 타입으로 Soundable 인터페이스를 가집니다. printSound()를 호출할 때 Cat과 Dog 객체를 주고 실행하면 각각 "야옹"과 "멍멍"이 출력되도록 Cat과 Dog 클래스를 작성해보세요.

```
public interface Soundable {
  public String sound();
}
```

```
public class SoundableExample {
  public static void printSound(Soundable soundable) {
    System.out.println(soundable.sound());
  }

  public static void main(String[] args) {
    printSound(new Cat());
    printSound(new Dog());
  }
}
```

7. DaoExample 클래스의 main() 메소드에서 dbWork() 메소드를 호출할 때 OracleDao와 MySqlDao 객체를 매개값으로 주고 호출했습니다. dbWork() 메소드는 두 객체를 모두 매개값으로 받기 위해 DataAccessObject 타입의 매개변수를 가지고 있습니다. 실행 결과를 보고 DataAccessObject 인터페이스와 OracleDao, MySqlDao 구현 클래스를 각각 작성해보세요.

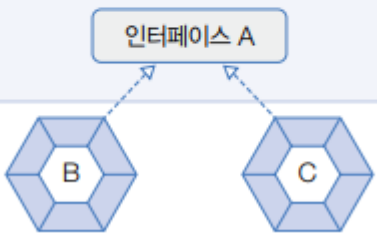
```
public class DaoExample {
  public static void dbWork(DataAccessObject dao) {
    dao.select();
    dao.insert();
    dao.update();
    dao.delete();
  }

  public static void main(String[] args) {
    dbWork(new OracleDao());
    dbWork(new MySqlDao());
  }
}
```

실행 결과

```
Oracle DB에서 검색
Oracle DB에 삽입
Oracle DB를 수정
Oracle DB에서 삭제
MySql DB에서 검색
MySql DB에 삽입
MySql DB를 수정
MySql DB에서 삭제
```

8. 다음과 같이 인터페이스와 클래스가 선언되어 있습니다. action() 메소드를 호출할 때 매개값이 C 객체일 경우에만 method2()가 호출되도록 밑줄에 들어갈 코드를 작성해보세요.

```
public interface A {
  public void method1();
}
```



```
public class B implements A {
  @Override
  public void method1() {
    System.out.println("B - method1()");
  }
}
```

```
public class C implements A {
  @Override
  public void method1() {
    System.out.println("C - method1()");
  }

  public void method2() {
    System.out.println("C - method2()");
  }
}
```

```
public class Example {
  public static void action(A a) {
    a.method1();
    if ( ____________________________ ) {
      c.method2();
    }
  }

  public static void main(String[] args) {
    action(new B());
    action(new C());
  }
}
```