

11. 추상클래스와 추상메서드

추상메서드와 추상클래스는 키워드 `abstract`로 선언한다.

- 추상메서드 : 메서드를 선언만 하고, 그 내용은 기술하지 않은 메서드이다.
- 추상클래스 : 클래스 내에 추상메서드가 하나라도 있으면 추상클래스이다.

```

abstract class ADT //추상클래스(내부에 추상메서드가 있으므로)
{
    abstract int compute(int x, int y); //추상메서드(내용은 없고, 선언만 있다)
    int compute(int x){ return ++x; } //일반메서드(내용이 정의되어 있다)
}
class Hap extends ADT //추상메서드는 자식클래스에서 정의(메서드 오버라이딩)
{
    int compute(int a, int b){ return a + b; }
}
class Gob extends ADT //추상메서드는 자식클래스에서 정의(메서드 오버라이딩)
{
    int compute(int a, int b){ return a * b; }
}
public class Test{
    public static void main(String args[]){
        Hap hap = new Hap();
        Gob gob = new Gob();
        System.out.println(hap.compute(2, 3)); // 5
        System.out.println(gob.compute(2, 3)); // 6
    }
}
    
```

- ① 추상클래스에 선언된 추상메서드는 자식클래스에서 정의하여 사용하게 된다.
 - 자식클래스에서 어떻게 정의하느냐? 에 따라 추상메서드 기능은 다르게 된다.
 - 즉, 객체지향언어에서 다형성을 지원하는 요인이 된다.
- ② 추상클래스는 객체(인스턴스)를 생성할 수 없다. 완성된 구조가 아니므로
 - ADT t = new ADT(); → 인스턴스가 생성되지 않고 오류 발생
- ③ 하지만, 추상클래스 형의 객체 변수는 자신의 자식클래스의 객체를 참조할 수 있다.
 - ADT t = new Hap(); → 인스턴스(t)가 정상적으로 생성된다.
- ④ 추상클래스의 자식클래스는 일반 또는 추상클래스가 될 수 있다.

일반클래스	상속받는 추상메서드를 재정의해야 함(재정의하지 않으면 오류 발생)
추상클래스	상속받는 추상메서드를 반드시 재정의할 필요는 없음

2. 다음 자바코드에서 오류가 발생하는 문장은? [2021년 국가 7급]

```

abstract class Shape {
    public void paint() { draw(); }
    abstract public void draw();
}
abstract class Circle extends Shape {
    public int radius;
    public Circle(int radius) { this.radius = radius; }
    double getArea() { return (3.14 * radius * radius); }
}
public class NamedCircle extends Circle {
    String name;
    public NamedCircle(int radius, String name) {
        super(radius);
        this.name = name;
    }
    public void draw() { System.out.println("반지름이 " + radius + "인 원을 그린다."); }
    public static void main(String[] args) {
        Shape s;
        s = new Shape();
        NamedCircle w = new NamedCircle(5, "Ring");
        System.out.println(w.getArea());
        w.draw();
    }
}

```

- ① super(radius);
- ② Shape s;
- ③ s = new Shape();
- ④ System.out.println(w.getArea());

☞ 객체 생성

•s = new Shape(); //Shape은 추상클래스이다. 추상클래스는 객체를 생성할 수 없다. (오류 발생)

정답 : ③

4 <http://cafe.daum.net/pass365>(홍재연)

3. 다음 C++ 코드에서 표현되지 않고 있는 것은? [2010년 국가 7급]

```
-----  
class Shape : public Item, public object  
{  
    public :  
        virtual void group() = 0;  
        virtual void ungroup() = 0;  
        virtual move(Position x, Position y) = 0;  
}
```

-
- ① 다중 상속클래스
 - ② 추상클래스
 - ③ 인터페이스 클래스
 - ④ 멤버함수 원형(prototype)

☞ C++ 코드

// C++ 코드 분석

```
class Shape : public Item, public object           //자식클래스 : 부모클래스(다중상속)  
{  
    public :  
        virtual void group() = 0;                 //순가상함수(자바의 추상메서드)  
        virtual void ungroup() = 0;               //순가상함수(자바의 추상메서드)  
        virtual move(Position x, Position y) = 0; //순가상함수(자바의 추상메서드)  
}
```

- C++에서 순가상함수는 자바의 추상메서드와 같다.(C++에서는 0을 대입하는 형식)
- 즉, 선언만 있고 내용은 없다.(멤버함수 원형)
- 따라서, 순가상함수는 자식클래스에서 재정의하여 사용하게 된다.

- 클래스 Shape는 함수 선언만 있는 추상클래스이다.

정답 : ③

4. C 언어의 추상 데이터 타입(abstract data type)에 대한 설명으로 가장 옳지 않은 것은? [2022년 군무원 7급]

- ① 사용자들은 추상 데이터 타입 내부의 데이터에 직접 접근하여 사용하고 변경 가능하다.
- ② 사용자들은 추상 데이터 타입이 제공하는 연산만을 사용할 수 있다.
- ③ 만약 다른 사람이 추상 데이터 타입의 구현을 변경하더라도 인터페이스가 변경되지 않는다면 사용자들은 추상 데이터 타입을 같은 방식으로 사용할 수 있다.
- ④ 추상 데이터 타입은 데이터 타입 정의가 그 데이터 타입의 구현으로부터 분리된 데이터 타입을 말한다.

♣ 추상 데이터 타입 - 추상자료형

// 먼저, 추상 데이터 타입에 대해 C 코드를 이용하여 예를 들어 설명한다.

```
int sum(int a, int b); // 함수 선언(두 정수의 합) - 추상 데이터 타입(ADT)
void main()
{
    printf("%d\n", add(1, 2));
}
int sum(int a, int b) // 함수 구현
{
    return a + b; // 연산
}
```

- 위의 코드에서 추상적이라는 것은 함수의 기능만 기술한 것을 말한다.(내용은 없는 것)
- 함수 내부의 구현은 기술하지 않은 것을 추상 데이터 타입(ADT)이라 한다.
- 함수 내부의 자세한 구현은 별도로 기술하는 것이다.(함수 선언과 구현을 따로 한 것)
- 이런 것을 'ADT는 구현에 독립적이다' 라고 한다. ADT는 사용설명서와 같은 것이다.
- ADT를 통해서 데이터나 연산이 무엇(what)인지는 알 수 있다.
- ADT를 통해서 데이터나 연산이 어떻게(how) 동작하는지는 알 수 없다.
- 사용자는 ADT에 접근하여 사용할 수는 있지만, 직접 변경은 불가능하다.
- 여기서, '사용자는 직접 변경은 불가능하다' 라는 것을 잘 이해해야 한다.
- 예 : C에서 사용자는 함수 printf()를 사용할 수는 있지만, 직접 변경은 불가능하다는 것

주의할 핵심 내용 데이터 타입(자료형) = 객체 + 객체에 동작하는 연산의 집합

- 객체(object)는 1, 2, 3, 4, ... 같은 값을 나타내는 것이다.
- 연산(operation)은 객체끼리 더하고(+), 빼고(-) 등을 처리하는 것이다.
- 여기서, 주의할 것은 자료형에는 객체만 있는 것이 아니라 그에 대한 연산이 포함되어 있다.