● 추상클래스(abstract class): 미완성 설계도로 미완성 메서드(추상 메서드)를 포함하고 있다. 추상클래스로는 인스턴스를 생성할 수 없다. 추상클래스는 상속을 통해서 자손클래스에 의해서만 완성될 수 있다. 추상클래스에도 생성자가 있으며, 멤버변수와 메서드도 가질 수 있다. 추상메서드를 포함하고 있지 않은 클래스에도 키워드 'abstract'을 붙여서 추상클래스로 지정할 수 있다. 추상메서드가 없는 완성된 클래스라 할지라도 추상클래스로 지정되면 클래스의 인스턴스를 생성할 수 없다.

● 추상메서드(abstract method): 선언부(반환타입, 메서드 이름, 매개변수)만 작성하고 구현부는 작성하지 않은 채로 남겨둔 메서드다. 메서드의 내용이 상속받는 클래스에따라 달라질 수 있기 때문에, 조상 클래스에서는 선언부만을 작성하고 주석을 덧붙여 어떤 기능을 수행할 목적으로 작성되어있는지 알려주고, 실제 내용은 상속받은 클래스에서그 클래스에 알맞게 구현하도록 비워두는 것이다. 추상메서드는 구현부가 없으므로 괄호{}대신 끝을 알리는 ';'을 적어준다.

```
/* 주석을 통해 어떤 기능을 수행할 목적으로 작성하였는지 설명한다. */
abstract 리턴타입 메서드이름();
```

추상클래스로부터 상속받은 자손클래스는 오버라이딩을 통해 조상인 추상클래스의 추상메서드를 모두 구현해주어야 한다. 만일 조상으로부터 상속받은 추상메서드 중 하나라도 구현하지 않는다면 자손클래스 역시 추상클래스로 지정해주어야 한다.

▶ 추상클래스 작성: 추상화는 기존의 클래스의 공통부분을 뽑아내서 조상 클래스를 만드는 것이다 (이와 반대되는 구체화는 상속을 통해 클래스를 구현, 확장하는 작업이다). 사실 메서드 구현부에 아무런 내용도 없이 괄호{}만 있어도 일반 메서드로 간주되어 자손 클래스에서 오버라이딩할 수 있지만, 굳이 abstract을 붙여서 추상메서드로 선언하는 이유는 자손 클래스에서 추성메서드를 반드시 구현하도록 강요하기 위해서이다.

```
abstract class Unit {
    int x, y;
    abstract void move(int x, int y);   // 추상메서드
    void stop() {
        /* 현재 위치에 정지*/
    }
}

class Marine extends Unit {
    void move(int x, int y) {
        /* Marine 클래스에 알맞도록 지정된 위치로 이동하는 코드*/
    }
    void stimPack() {
        /* Marine 클래스의 메서드 */
    }
}

class Tank extends Unit {
    void move(int x, int y) {
        /* Tank 클래스에 알맞도록 지정된 위치로 이동하는 코드*/
    }
    void changeMode() {
        /* Tank 클래스의 메서드 */
    }
}

class Dropship extends Unit {
    void move(int x, int y) {
        /* Dropship 클래스에 알맞도록 지정된 위치로 이동하는 코드*/
    }
    void load() {
        /* Dropship 클래스의 메서드 */
    }
    void unload() {
        /* Dropship 클래스의 메서드 */
    }
}
```

```
class Ex {
    public static void main(String[] args) {
        Unit[] group = new Unit[4];
        group[0] = new Marine();
        group[1] = new Tank();
        group[2] = new Marine();
        group[3] = new Dropship();

        for (int i = 0; i < group.length; i++) {
            group[i].move(100, 200);
        }
    }
}
```

다형성, 조상 클래스타입의 참조변수로 자손 클래스의 인스턴스를 참조할 수 있는 성질, 덕분에, Unit클래스에 move메서드가 추상메서드로 정의되어 있다 하더라도 이처럼 Unit클래스 타입의 참조변수로 move메서드를 호출하는 것이 가능하다. 메서드는 참조변수의 타입에 관계없이 실제 인스턴스에 구현된 것이 호출되기 때문이다. group[i].move(100, 200)과 같이 호출하는 것이 Unit클래스의 추상메서드인 move를 호출하는 것처럼 보이지만

사실 이 추상메서드가 구현된 Marine, Tank, Dropship 인스턴스의 메서드가 호출되는 것이다. 모든 클래스의 조상인 Object클래스 타입의 배열로도 서로 다른 종류의 인스턴스를 하나의 묶음으로 다룰 수 있지만, Object클래스에서는 move메서드가 정의되어 있지 않기 때문에 move메서드를 호출하는 부분에서 에러가 발생한다.

Resource: 자바의 정석