

제 26 강

추상메서드와 추상 클래스

교재:p138~142

목차

1. 추상 메서드와 추상 클래스

1. 추상 메서드
2. 추상 클래스

1. 추상 메서드

추상 메서드

: 추상 메서드는 선언 부만 정의하고 구체적인 내용은 비워 놓은 메서드

주의할 점

추상메서드가 있는 클래스를 상속받는 자식클래스는 반드시 부모 클래스의 추상 메서드를 구현해야한다!

1. 추상 메서드

추상 메서드 구현 방법

<구조>

`abstract` 리턴타입 메서드명();

- 구현하지 않으므로 { } 생략 하고 ; 으로 마무리

2. 추상 클래스

추상 클래스

: 추상 클래스는 추상 메서드를 멤버(필드)로 갖는 클래스

주의할 점

- 추상 클래스는 일반 메서드 필드에 존재 가능
- 단, 추상 메서드를 하나라도 포함해야 함
- 객체를 생성할 수 없다

2. 추상 클래스

추상 클래스 구현 방법

<구조>

```
abstract class 클래스명{  
    ...  
    abstract 리턴타입 메서드명( );  
}
```

- 구현하지 않으므로 { } 생략 하고 ; 으로 마무리

2. 추상 클래스

추상 클래스 사용 목적?

구현은 자식 클래스가 하도록 메서드의 기능을 비워놓고 싶을 때 사용!

ex)

도형 class - draw():그리다

사각형 class- draw():사각형을 그리다

원 class - draw(): 원을 그리다

선 class - draw(): 선을 그리다

2. 추상 클래스

추상 클래스 사용 시 주의할 점

```
abstract class Aclass{  
    abstract void f1();  
}  
class B extends Aclass{  
}
```

!x The type B must implement the inherited abstract method Aclass.f1()
2 quick fixes available:
➔ [Add unimplemented methods](#)
➔ [Make type 'B' abstract](#)

자식 클래스에서
추상 메서드를 구현하지 않으면 오류!

```
abstract class Aclass{  
    void f1() {  
        System.out.println("f1()");  
    }  
    abstract void f2();  
}
```

일반 메서드와 함께 사용 가능!
추상 메서드는 **abstract** 키워드 붙이기

2. 추상 클래스

<실습> ShapeMain.java

도형 클래스를 생성하고 draw() 메서드를 만들고, 도형 클래스를 상속받는 사각형, 원, 선 클래스를 생성한다.

각 클래스에 맞게 draw() 오버 라이딩 하기

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

어떤 도형을 그리는지 정확히 모르기 때문에
사실상 불필요한 기능

```
class Rect extends Shape{  
    void draw() {  
        System.out.println("사각형을 그리다.");  
    }  
}  
class Circle extends Shape{  
    void draw() {  
        System.out.println("원을 그리다.");  
    }  
}  
class Line extends Shape{  
    void draw() {  
        System.out.println("선을 그리다.");  
    }  
}
```

2. 추상 클래스

<실습> ShapeMain.java

도형 클래스를 생성하고 draw() 메서드를 만들고, 도형 클래스를 상속받는 사각형, 원, 선 클래스를 생성한다.

각 클래스에 맞게 draw() 오버 라이딩 하기

```
abstract class Shape{  
  
    abstract void draw();  
}
```

따라서 추상 메서드와 추상 클래스로 정의 후
자식 클래스에서 오버 라이딩!

```
class Rect extends Shape{  
    void draw() {  
        System.out.println("사각형을 그리다.");  
    }  
}  
class Circle extends Shape{  
    void draw() {  
        System.out.println("원을 그리다.");  
    }  
}  
class Line extends Shape{  
    void draw() {  
        System.out.println("선을 그리다.");  
    }  
}
```

2. 추상 클래스

<실습> ShapeMain.java

도형 클래스를 생성하고 draw() 메서드를 만들고, 도형 클래스를 상속받는 사각형, 원, 선 클래스를 생성한다.

각 클래스에 맞게 draw() 오버 라이딩 하기

```
public class ShapeMain {  
    public static void main(String[] args) {  
        Shape shapes[]=new Shape[3];  
        shapes[0]=new Rect();  
        shapes[1]=new Circle();  
        shapes[2]=new Line();  
  
        for(int i=0;i<3;i++) {  
            shapes[i].draw();  
        }  
    }  
}
```

객체 배열의 각 요소를
업 캐스팅을 통한 자식 값으로 채움

오버 라이딩 한 자식 메서드가
동적 바인딩에 의해 실행

<출력 결과>

사각형을 그린다.
원을 그린다.
선을 그린다.

2. 추상 클래스

<실습> Exam-54.java

포켓몬이라는 추상 클래스를 만들고, 구체적인 피카츄, 꼬부기 클래스를 구현하는 예제

```
abstract class Pokemon{
    String name;
    abstract void attack();
    abstract void sound();

    public String getName() {
        return this.name;
    }
}
```

```
class Pikachu extends Pokemon{
    Pikachu(){
        this.name="피카츄";
    }
    void attack() {
        System.out.println("전기 공격");
    }
    void sound() {
        System.out.println("피카 피카!");
    }
}

//-----
class Squirtle extends Pokemon{
    Squirtle(){
        this.name="꼬부기";
    }
    void attack() {
        System.out.println("물 공격");
    }
    void sound() {
        System.out.println("꼬북 꼬북!");
    }
}
```

2. 추상 클래스

<실습> Exam-54.java

포켓몬이라는 추상 클래스를 만들고, 구체적인 피카츄, 꼬부기 클래스를 구현하는 예제

```
public class AbstractClass2 {  
    public static void main(String[] args) {  
        Pikachu pikachu1= new Pikachu();  
        System.out.println("이 포켓몬은 "+pikachu1.getName()+"입니다.");  
        pikachu1.attack();  
        pikachu1.sound();  
  
        Squirtle squirtle1= new Squirtle();  
        System.out.println("이 포켓몬은 "+squirtle1.getName()+"입니다.");  
        squirtle1.attack();  
        squirtle1.sound();  
    }  
}
```

<출력 결과>

```
이 포켓몬은 피카츄입니다.  
전기 공격  
피카 피카!  
이 포켓몬은 꼬부기입니다.  
물 공격  
꼬북 꼬북!
```