



제13장

추상클래스와 인터페이스

1. 추상 클래스(abstract class)란?

■ 추상 클래스의 개념

- 추상 클래스는 완성되지 않은 설계도와 같다.
- 하여, 인스턴스를 생성할 수가 없다.
- 추상 메서드(선언부만 존재)를 최소 1개 이상 포함하고 있는 클래스이다.
- 일반 메서드가 추상 메서드를 호출할 수 있다.(호출할 때 필요한 건 선언부임)
- 다른 클래스를 작성하는데 도움을 줄 목적으로 이용된다.

```
public abstract class ContentSender {  
    String title;  
    String nm;  
  
    public ContentSender(String title,String nm) {  
        this.title = title;  
        this.nm = nm;  
    }  
    abstract void sendMsg(String content);  
}
```

* 추상 클래스가 인스턴스를 생성하기 위해서는 상속을 통하여 자손클래스에서 추상 메서드를 다 구현해야 비로소 인스턴스를 생성할 수 있다.

2. 추상 메서드란(abstract method)?

▣ 메서드의 선언부만 있고, 구현부(정의부, 몸통)이 없는 메서드를 말한다.

```
//추상 메서드  
//해당 클래스에 맞게 power메서드를 작성한다.  
abstract void power(String name);
```

- 필요하다면, **자손마다 다르게 구현될 것이라고 예상될 때** 사용한다.
- 추상 클래스를 상속받는 자손클래스는 조상의 추상 메서드의 구현부를 완성해야 인스턴스를 생성할 수가 있다.
- 하지만, 일부만 구현하는 경우도 빈번하다.

* 추상클래스의 일부 메서드만 구현할 때, 역시 abstract 키워드를 붙여야 된다.

```
abstract class ContentSender {  
    abstract void sendMsg(String content);  
    abstract void recieveMsg(String content);  
}
```

```
abstract class kakaoSender extends ContentSender {  
    @Override  
    public void sendMsg(String recipient) {  
        System.out.println("받는사람 =" + recipient);  
    }  
}
```

3. 추상클래스의 작성

- 상속 받을 자손클래스에서 **공통적으로 사용될 것이라고 예상되는 것을 모아서** 하나의 추상클래스로 만들거나, 이미 기존에 만들어 놓은 클래스라면 **공통부분을 뽑아서** 추상클래스로 만든다.

```
public class Cat {  
    public void sound(){  
        /* 소리를 내는 부분 */  
    }  
}
```

```
public class Dog {  
    public void sound() {  
        /* 소리를 내는 부분 */  
    }  
}
```

```
Animal[] ani = new Animal[2];  
ani[0] = new Cat();  
ani[1] = new Dog();
```

```
public abstract class Animal {  
    //추상메서드  
    public abstract void sound();  
}  
  
public class Cat extends Animal {  
    @Override  
    public void sound() {  
        System.out.println("야옹~");  
    }  
}  
  
public class Dog extends Animal {  
    @Override  
    public void sound() {  
        System.out.println("멍멍!");  
    }  
}
```

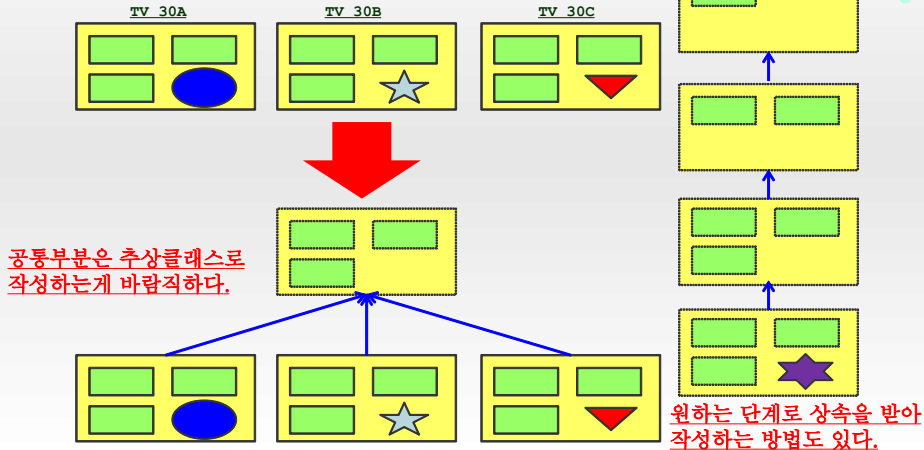
다형성에서 학습했다.

1) 코드중복 제거된다.

2) 한곳에서 관리가 된다

3) 오류가 줄어든다.

4. 추상클래스의 이해



감사합니다.

