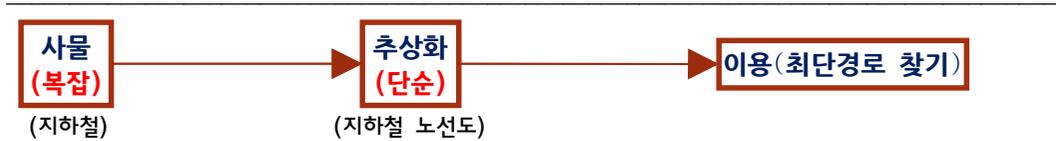


# 추상화(abstraction)

## 1. 개요



- 컴퓨터 과학에서 추상화는 어떤 종류의 대상들에 대해 그것이 가져야 할 핵심적인 특징들을 가지는 모델을 만드는 것이다.
- 여러 가지 사물이나 개념에서 공통되는 특성이나 속성 따위를 추출하여 파악하는 작용

- 추상화는 어떤 사물에서 핵심 부분만을 추출하여 이해하기 쉽도록 표현하는 기술이다.
- 추상화 : 복잡 → 단순 (지하철노선도, 달력, 시계 등)
- 추상화는 문제가 발생했을 때, 해결책을 쉽게 찾을 수 있도록 한다.
- 추상화는 복잡한 문제나 시스템을 이해하거나 설계하는 데 중요한 요소이다.
- 예 : 컴퓨터 시스템을 구성하는 같은 종류의 하드웨어 장치는 다양하게 설계할 수 있다.
- 프로그래머가 하드웨어에 직접 접근하는 것은 문제를 매우 복잡하게 만들 수 있다.
- 추상화는 프로그래머가 각종 장치에서 작동하는 프로그램 개발을 돕는다.
- 하드웨어 추상화는 같은 종류의 장치(예.프린터)에 대한 공통 명령어 집합을 제공한다.

// 전산에서 추상화



↓ 전산 시험에서 추상화 쓰임새

- 추상화된 소프트웨어
- 추상화 기법을 통해 객체의 설계비용과 시간을 줄일 수 있다.
- 추상화 메커니즘을 적절히 이용한다면 구조적이고 단계적인 설계를 할 수 있다.
- 프로그래머는 추상화에 의존해야지, 구체화에 의존하면 안 된다.(의존성 역전의 원칙)
- 확장에 열려 있으려면, 추상화에 의존해야 한다.
- 모델링 작업은 시스템의 기능, 동작 및 구성을 추상화 형태로 표현해 가는 과정이다.
- 일반화 : 유사한 클래스들 사이의 공유되는 속성과 동작을 묶어주는 추상화 기법이다.

## 2. 클래스(class)

다음은 추상화한 '계좌'를 Java code 형태로 구현한 것이다.

```
public class 계좌
{
    private String 계좌번호 = "389-02-123456"; //속성
    private String 비밀번호 = "1234";
    private long 잔고 = 1000;
    private float 이율 = 6.02F;

    public void 입금하다(long 금액) { //메서드
        잔고 = 잔고 + 금액;
    }
    public void 출금하다(long 금액) {
        잔고 = 잔고 - 금액;
    }
    public long 조회하다() {
        return 잔고;
    }
    public void 이율변경(float 새이율) {
        이율 = 새이율;
    }
}
```

클래스는 같은 형태의 속성과 메서드를 갖는 객체를 생성할 수 있는 **틀(template)**이다. 즉, 클래스는 사용자가 정의한 새로운 자료형으로 객체 타입(object type)이다.

```
계좌 하나 = new 계좌(계좌번호, 비밀번호, .....)
```

여기서, 계좌 : 클래스 이름  
하나 : 생성되는 객체 이름  
new : 객체를 생성하기 위한 명령어

결론적으로 객체지향언어에서 객체를 생성하여 사용하려면 먼저 클래스를 만들어야 한다. 모든 객체는 클래스로부터 생성되기 때문이다.

객체가 생성되는 과정을 실체화(instantiation)라 하고, 생성된 객체를 **인스턴스(instance, 사례)**라 한다. 객체와 인스턴스는 같은 뜻으로 사용된다.

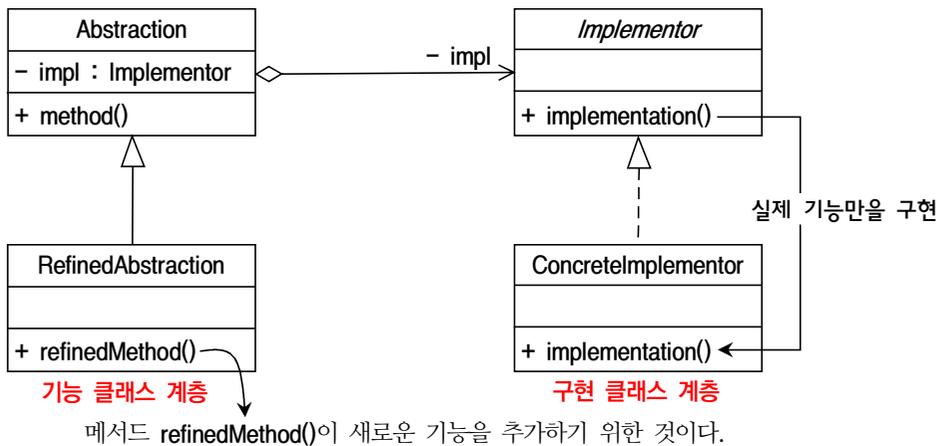
**기출문제 분석**

1. 추상화와 구현을 분리하여, 실행시간에 동적으로 인터페이스의 다른 구현들을 사용할 수 있는 디자인 패턴은? [2022년 서울 7급]

- ① adapter 패턴      ② bridge 패턴
- ③ proxy 패턴        ④ facade 패턴

☞ bridge 패턴

• 브리지 패턴은 추상과 구현을 분리하는 것이다. 키워드는 분리이다.



기능 클래스 계층 (추상층)	<ul style="list-style-type: none"> <li>• 부모클래스를 상속받은 자식클래스에서 새로운 기능을 추가하는 계층이다.</li> <li>• 새로운 기능 추가는 새로운 메서드를 정의하는 것이다.</li> <li>• 부모클래스는 기본적인 기능을 가지고 있고</li> <li>• 자식클래스에서는 새로운 기능을 추가할 수 있도록 한다.</li> </ul>
구현 클래스 계층	<ul style="list-style-type: none"> <li>• 새로운 기능을 추가하기 위한 계층이 아니다.</li> <li>• 즉, 새로운 메서드를 추가하기 위한 것이 아니다.</li> <li>• 단지, 부모클래스에 정의된 추상메서드의 실제 기능만을 구현한다.</li> </ul>

—(브리지 패턴 정의)—

브리지 패턴은 구현(implementation)으로부터 추상(abstraction) 층을 분리하여 이 둘이 서로 독립적으로 변화할 수 있도록 한다.

구현부에서 추상층을 분리하여 각각 독립적으로 변경이 가능하고 확장도 가능하도록 한다.

2. 컴퓨팅 사고(computational thinking)에서 주어진 문제의 중요한 특징만으로 문제를 간결하게 재정의함으로써 문제 해결을 쉽게 하는 과정은? [2021년 국가 9급]

- ① 분해
- ② 알고리즘
- ③ 추상화
- ④ 패턴 인식

♣ 추상화(abstraction)

---

- 추상화 : 복잡 → 단순
  - 추상화는 복잡한 시스템, 자료, 모듈 등으로부터 핵심 개념 또는 기능을 추출하는 것을 말한다.
  - 추상화는 사물의 주된 특징은 부각시키고, 나머지는 과감하게 생략하는 방식으로 진행된다.
  - 추상화는 문제를 중요 특징만으로 간결하게 재정의함으로써 문제 해결을 쉽게 하는 과정이다.
  - 예 : 같은 종류의 기기(프린트 등)에 대해서 공통명령어 집합을 사용한다.
- 

정답 : ③

3. 소프트웨어 설계의 원칙으로 옳지 않은 것은? [2015년 서울 9급]

- ① 상세설계로 갈수록 추상화 수준은 증가한다.
- ② 계층적 조직이 제시되며, 모듈적이어야 한다.
- ③ 설계는 분석 모델까지 추적이 가능하도록 한다.
- ④ 요구사항 분석에서 얻은 정보를 이용하여 반복적 방법을 통해 이루어져야 한다.

♣ 소프트웨어 설계의 원칙

---

- 상세설계로 갈수록 추상화 수준은 증가한다.(×)  
→ 상세설계로 갈수록 추상화 수준은 낮아진다. 즉, 감소한다. 기계에 가깝다.
- 

정답 : ①