

Java로 배우는 디자인패턴 입문  
Chapter 3. Template Method  
구체적인 처리를 하위 클래스에게 위임

교재: 자바언어로배우는디자인패턴입문(개정판)/YukiHiroshi저/김윤정역/영진닷컴

덕성여자대학교 컴퓨터학과

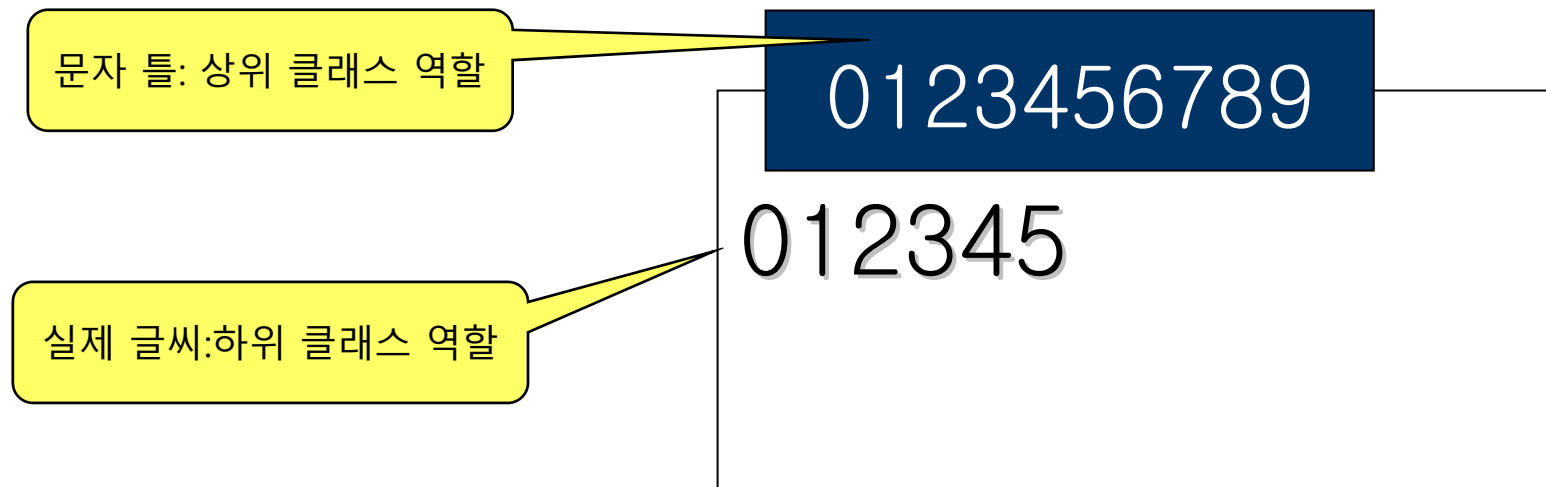
최 승 훈

# 01. Template Method 패턴

---

## □ 템플릿이란?

- 문자 모양을 따라 구멍이 뚫려있는 얇은 플라스틱 판
- 필기도구의 종류(싸인펜, 연필, 색연필)에 따라 실제로 쓰여지는 문자의 인스턴스가 결정된다.



# 01. Template Method 패턴

---

## □ Template Method 패턴

- 상위 클래스에, 템플릿 역할을 하는 메소드가 정의됨.
  - 그 메소드에서는 추상 메소드들을 호출한다.
- 상위 클래스의 이 메소드만 보면, 추상 메소드들이 어떻게 호출되는지는 알 수 있다. 하지만, 최종적으로 어떤 처리를 하는지는 모른다.

# 01. Template Method 패턴

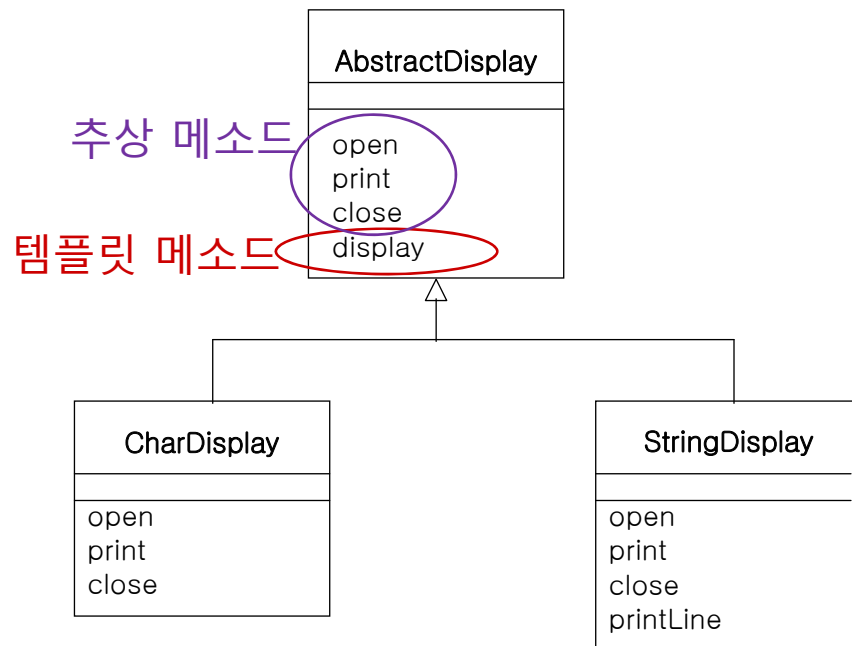
---

## □ Template Method 패턴

- 하위 클래스가 추상 메소드를 실제로 구현한다.
  - 다른 하위 클래스가 다른 구현을 하면, 다른 처리가 실행된다.
  - 하위 클래스에서 어떤 구현을 하더라도, **처리의 큰 흐름은 상위 클래스가 결정한대로 이루어진다.**
- 상위 클래스에서 처리의 흐름을 결정하고, 하위 클래스에서 구체적인 내용을 결정하는 디자인 패턴

## 02. 예제 프로그램

### ❑ 클래스 다이어그램



#### ❑ AbstractDisplay

- display( ) 메소드:
  - 이 안에서 open, print, close 메소드가 사용(호출)된다.
- open, print, close는 추상 메소드임
- display( ) 메소드를 Template Method라고 한다.

- ❑ open, print, close 메소드를 실제로 구현하고 있는 것은, CharDisplay와 StringDisplay 클래스이다.

## 02. 예제 프로그램

---

이름	해설
AbstractDisplay	Display 메소드만 구현되어 있는 추상 클래스
CharDisplay	Open, print, close 메소드를 구현하고 있는 클래스
StringDisplay	Open, print, close 메소드를 구현하고 있는 클래스
Main	동작 테스트용 클래스

## 02. 예제 프로그램

---

- ❑ `AbstractDisplay` 클래스 (`sample/AbstractDisplay.java`)
  - `display()`
    - `open` 메소드를 호출한다.
    - `print` 메소드를 5번 호출한다.
    - `close` 메소드를 호출한다.
  - `open`, `print`, `close` 추상 메소드가 실제로 하는 일은, 하위 클래스를 살펴보아야 한다.

## 02. 예제 프로그램

---

- CharDisplay 클래스 (sample/CharDisplay.java)
  - 상위 클래스인 AbstractDisplay 클래스의 추상 메소드인 open, print, close를 구현함

메소드명	처리
open	문자열 '<<'를 표시합니다.
Print	생성자에서 제공된 문자 한 개를 표시합니다.
Close	문자열 '>>'를 표시합니다.



## 02. 예제 프로그램

---

- ❑ StringDisplay 클래스 (sample/StringDisplay.java)
  - 상위 클래스인 AbstractDisplay 클래스의 추상 메소드인 open, print, close를 구현함
  - string.getBytes().length 문장
    - 문자열의 바이트 개수를 얻는다.
    - string.length() 와 어떻게 다른가?
      - string = "안녕하세요." 인 경우
        - string.length() : 6을 반환
        - string.getBytes().length : 11을 반환

메소드명	처리
open	문자열 '+-----+'를 표시합니다.
Print	생성자에서 제공된 문자열을 ' '와 ' '사이에 두어 표시합니다.
Close	문자열 '+-----+'를 표시합니다.

## 02. 예제 프로그램

---

- ❑ Main 클래스 (sample/Main.java)
  - CharDisplay와 StringDisplay의 인스턴스를 생성한 후, display( ) 를 호출한다.
  - d1.display( ) / d2.display( ) / d3.display( ) 의 실제 동작은,
    - d1, d2, d3 객체가 속한 클래스에서 결정한다.

## 03. Template Method 패턴에 등장하는 역할

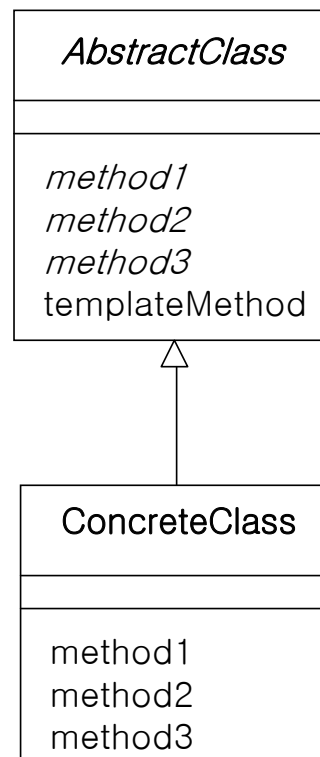
---

- ❑ AbstractClass(추상 클래스)의 역할
  - 템플릿 메소드 구현
  - 템플릿 메소드 안에서 사용하는 추상 메소드 선언
  - 예제에서, AbstractDisplay 클래스가 해당됨
  
- ❑ ConcreteClass(구체적 클래스)의 역할
  - AbstractClass 역할에서 선언되어 있는 추상 메소드를 구현
  - 예제에서, CharDisplay와 StringDisplay 클래스가 해당됨

## 03. Template Method 패턴에 등장하는 역할

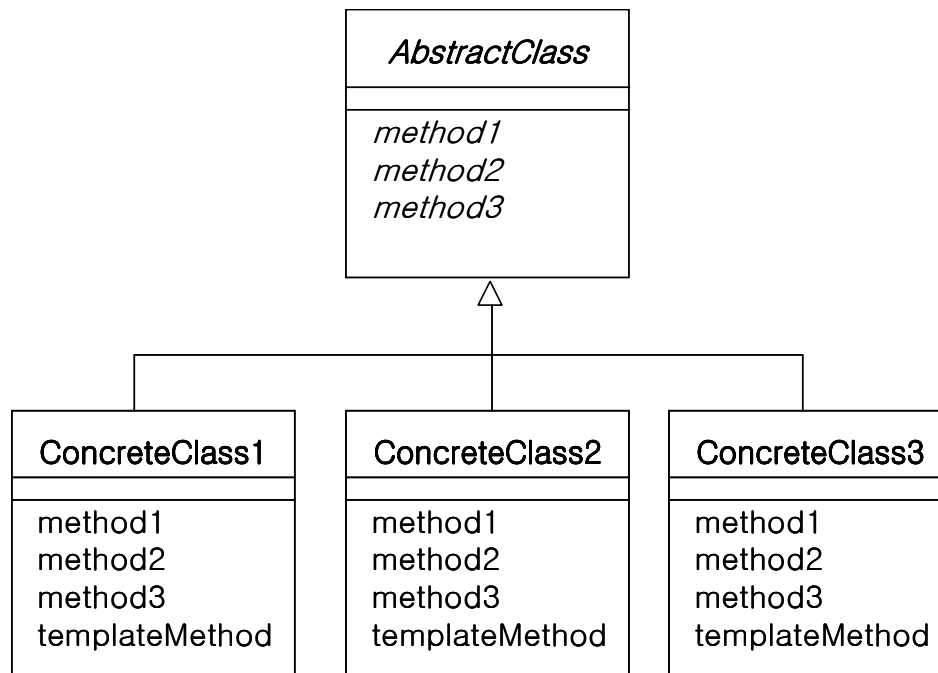
---

- 일반적인 클래스 다이어그램



## 04. 독자의 사고를 넓혀주는 힌트

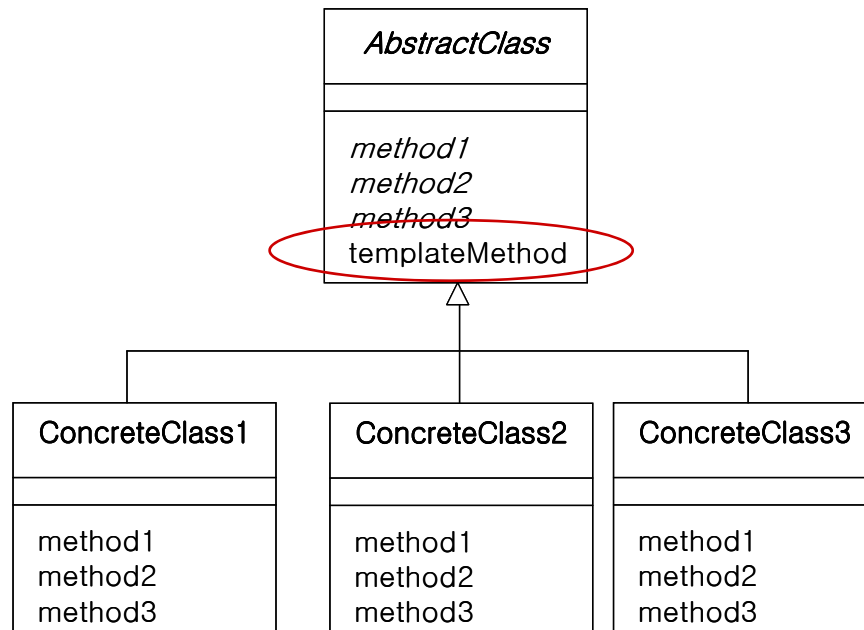
- ❑ 로직을 공통화할 수 있다
  - TemplateMethod 패턴을 이용하지 않은 경우



templateMethod( )가  
method1, 2, 3를  
같은 방식(로직)으로  
사용하는 경우,  
같은 코드가  
ConcreteClass1, 2, 3에 반복해서  
나타난다  
=> 로직을 바꾸어야 하는 경우,  
ConcreteClass1, 2, 3의  
모든 templateMethod( )를  
수정해야 한다.

## 04. 독자의 사고를 넓혀주는 힌트

- ❑ 로직을 공통화할 수 있다
  - TemplateMethod 패턴을 이용한 경우



templateMethod( )가  
부모 클래스인 AbstractClass에서  
구현됨으로써,  
공통된 로직이 한 곳에  
집중되어 있다.

## 05. 관련 패턴

---

- ❑ Factory 패턴(4장)
- ❑ Strategy 패턴 (10장)

## 06. 보강: 클래스 계층과 추상 클래스

---

- ❑ 상위 클래스가 하위 클래스에게 요청
  - 상위 클래스에 선언된 추상 클래스의 구현을, 하위 클래스에게 요청한다.
  
- ❑ 추상 클래스의 의의
  - 추상 클래스는 인스턴스를 만들 수 없다.
  - 메소드의 이름은 추상 클래스인 상위 클래스에서 결정하고, 실제 하는 일은 하위 클래스에서 결정한다.



## 07. 요약

---

- ❑ 상위 클래스에서 처리의 흐름을 규정하고,
- ❑ 하위 클래스에서 처리의 내용을 구체화하는 패턴

## 연습문제 (각자 공부할 것)

---

- ❑ 3-1
  - java.io.InputStream 클래스에서 사용되는 TemplateMethod 패턴
- ❑ 3-2
  - final 메소드의 의미
- ❑ 3-3
  - 메소드 accessibility 문제
    - public / private / **protected** 등
- ❑ 3-4
  - 추상 클래스와 인터페이스 클래스의 차이점