

Chapter. 08

추상 클래스

# 추상 클래스 응용 - 템플릿 메서드

FAST CAMPUS  
ONLINE  
자바 기초

강사. 박은종

Chapter. 08

02 추상 클래스 응용  
- 템플릿 메서드

# I 템플릿 메서드

템플릿 : 틀이나 견본을 의미

템플릿 메서드 : 추상 메서드나 구현된 메서드를 활용하여 전체의 흐름(시나리오)를 정의 해 놓은 메서드  
final로 선언하여 재정의 할 수 없게 함

템플릿 메서드 패턴: 디자인 패턴의 일종

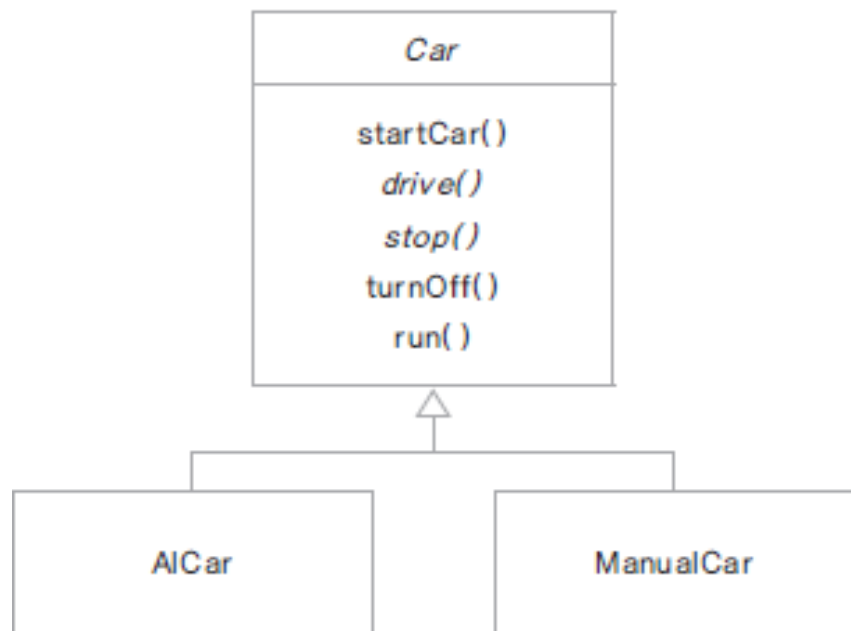
프레임 워크에서 많이 사용되는 설계 패턴

추상 클래스로 선언된 상의 클래스에서 추상 메서드를 이용하여 전체 구현의 흐름을 정의하고 구체적인 각 메서드 구현은

하위 클래스에 위임함

하위 클래스가 다른 구현을 했다고 해서 템플릿 메서드에 정의된 시나리오 대로 수행됨

# I 템플릿 메서드 구현하기 예제



```
public abstract class Car {
    public abstract void drive( );
    public abstract void stop( );

    public void startCar( ) {
        System.out.println("시동을 켭니다");
    }

    public void turnOff( ) {
        System.out.println("시동을 끕니다.");
    }

    final public void run( ) {
        startCar( );
        drive( );
        stop( );
        turnOff( );
    }
}
```

템플릿 메서드

# I final 예약어

final 변수는 값이 변경될 수 없는 상수임  
public static final double PI = 3.14;  
오직 한 번만 값을 할 당할 수 있음

final 메서드는 하위 클래스에서 재정의 (overriding) 할 수 없음

final 클래스는 더 이상 상속되지 않음  
예) java의 String 클래스

# I public static final 상수 값 정의 하여 사용하기

프로젝트 구현 시 여러 파일에서 공유해야 하는 상수 값은 하나의 파일에 선언하여 사용하면 편리 함

```
public class Define {
    public static final int MIN = 1;
    public static final int MAX = 99999;
    public static final int ENG = 1001;
    public static final int MATH = 2001;
    public static final double PI = 3.14;
    public static final String GOOD_MORNING = "Good Morning!";
}
```

```
public class UsingDefine {
    public static void main(String[] args) {
        System.out.println(Define.GOOD_MORNING);
        System.out.println("최솟값은 " + Define.MIN + "입니다.");
        System.out.println("최댓값은 " + Define.MAX + "입니다.");
        System.out.println("수학 과목 코드 값은 " + Define.MATH + "입니다.");
        System.out.println("영어 과목 코드 값은 " + Define.ENG + "입니다.");
    }
}
```

static으로 선언했으므로 인스턴스를 생성  
하지 않고 클래스 이름으로 참조 가능