

# 차량 조립 프로젝트

CONTENTS

# 목차

CHAPTER 1

공통과제 소개

CHAPTER 2

공통과제 미션

Chapter 01

# 공통과제 소개

차량 조립 프로젝트

# 차량 제조 순서

## 1) 자동차 타입을 선택한다.

세단, SUV(에스-유-브이), 트럭

## 2) 자동차에 들어갈 부품을 선택한다.

엔진, 제동장치, 조향장치

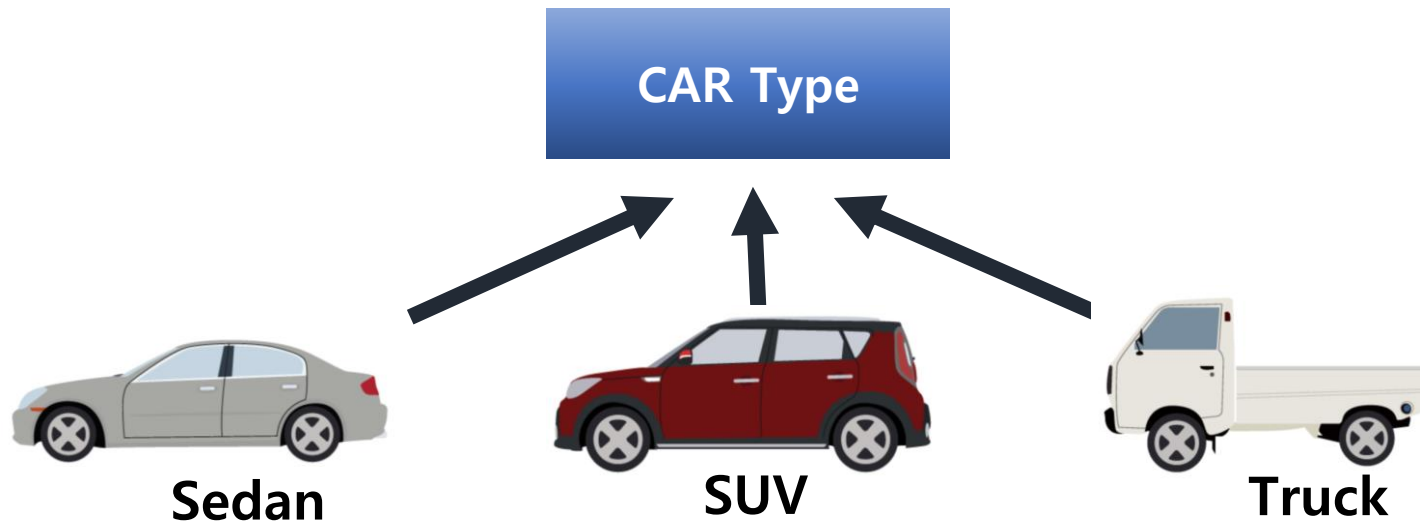
## 3) 완성된 차량을 테스트한다.

선택한 부품이  
자동차 타입에 사용 가능한지 검사

## 제조 순서 1) 자동차 타입 선택

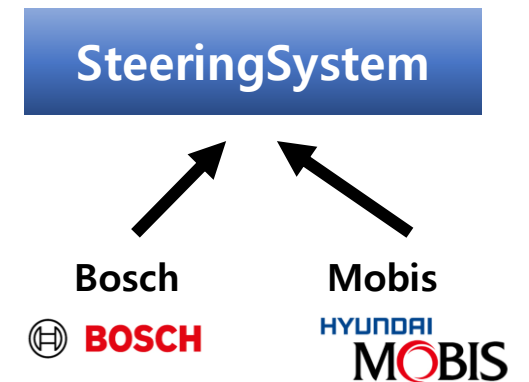
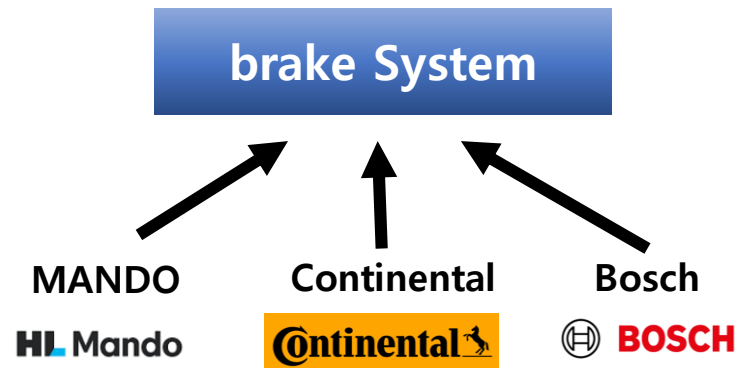
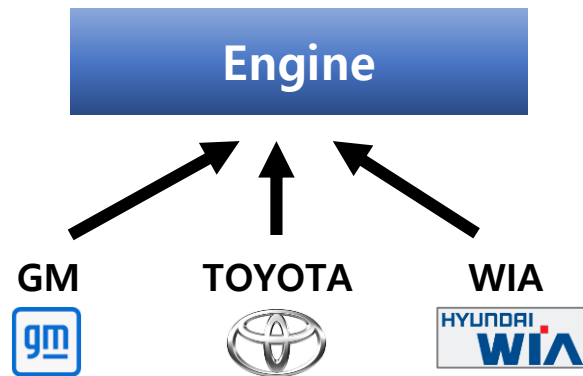
차량의 종류를 선택한다.

총 세가지 타입을 제작할 수 있으며,  
향후에 타입이 더 추가될 수 있다.



## 제조 순서 2) 자동차 조립

엔진, 제동장치, 조향장치를 각각 선택한다.



# 제조 순서 3) 자동차 완성 가능조합 확인

## 제한조건 1

- 제동장치에 Bosch 제품을 사용했다면, 조향장치도 Bosch 제품을 사용해야한다.  
(타사 제품과 호환되지 않는다.)

## 제한조건 2

- Continental은 Sedan용 제동장치를 만들지 않는다.  
(-> 세단에 Continental 제품 사용 불가)
- 도요타는 SUV용 엔진을 만들지 않는다.
- WIA는 Truck용 엔진을 만들지 않는다.
- Mando는 Truck용 제동장치(brake System)을 만들지 않는다.

## 기존에 사용중인 시스템의 아쉬운 점

- 절차지향식 코드로, 유지보수가 어려운 구조
- 안전하지 않은 문법들이 사용
- 확장성이 고려되지 않음
- 유닛테스트가 없음



Chapter 02

# 공통과제 미션

차량 조립 프로젝트

## [미션 1] 함수 레벨 리팩토링

### 함수 레벨로 클린코드 만들기

#### 요구사항

가독성 좋은 코드

안전성이 좋은 코드

- 안전한 문법 사용
- 순수함수 권장

## [미션 2] 클래스 레벨 리팩토링

모듈화가 잘 되어있는 구조로 리팩토링 한다.

### 요구사항

- Unit Test
- 확장성을 고려한 설계
- 디자인 패턴 적용
- 코드 커버리지 100%

## 요구사항 정리 (D1 ~ D5)

### 1. 함수 레벨 리팩토링

D1 - 가독성있는 코드로 함수 분리, 가독성 좋은 네이밍 사용

### 2. 메서드 레벨 리팩토링

D2 - Regression Test 를 위한 Unit Test 개발

D3 - 확장성을 고려한 설계, 부품이 추가되더라도 Client Code에 변경이 없도록 한다.

### 3. 디자인 패턴 사용하기

D4 - 리팩토링에 디자인 패턴을 적용한다.

### 4. 코드 커버리지 100%

D5 - 리팩토링이 끝난 코드에, 코드 커버리지가 100% 되어야 한다.

# 제출 방법

개인 Repository 를 Github에 생성한다.  
Repository에 다음과 같이 제출한다.

**mission1** 폴더에는 메서드 레벨 리팩토링 후 코드가 포함 되어야한다.  
**mission2** 폴더에는 클래스 레벨 리팩토링 후 코드가 포함 되어야한다.

**README.md 파일에는 다음과 같은 내용이 적혀야 한다.**

1. D1 ~ D5 까지 적용 여부 체크리스트
2. Code Coverage 최종 측정결과 캡처화면

요구사항 체크리스트	
D1	O
D2	O
D3	X
D4	X
D5	O

체크리스트 예시

## 평가방법

- D1 ~ D5 까지 체크리스트에 체크가 된 항목은, 수동으로 진행 수준 평가 예정
- D1 ~ D5 까지 체크리스트에 체크가 안된 항목은, 오답처리
- Coverage 캡처화면이 없는 경우, 오답처리

**감사합니다.**