

**[ 문제 1 ] 다음과 같은 조건을 만족하는 프로그램을 작성 하시오**

각 부서 별 평가 점수를 입력 받아 부서 별 평가를 하고자 한다.

A, B, C, D, E 와 같이 5개의 평가 기준으로 구성 되며 각각의 점수를 이용하여 구한다.

(A+B)/2 : 60%

(C+D)/2 : 20%

E : 20%

90 이상이면: Gold Class

80 이상이면: Silver Class

70 이상이면: Bronze Class

70 미만이면: Normal Class

**1. 사용 데이터**

5개의 점수를 입력 받는다.

5개가 아니면 "다시 입력 하세요"라고 메시지를 출력 한다.

10부터 99까지의 값을 입력 받는다.

입력된 10부터 99까지의 정수가 아닌 경우 "다시 입력 하세요"라고 메시지를 출력 한다.

**2. 구현 클래스**

| Package명 | 클래스명   | 메소드                               | 설명                         |
|----------|--------|-----------------------------------|----------------------------|
| workshop | Test01 | <u>+main(String args[]): void</u> | main 함수 안에서 모든 코드 작업<br>진행 |

**3. 실행 결과**

모두 double로 선언 하고 계산한다.

10 20 40 50 80을 입력 하였을 때 아래와 같이 출력 된다.

평가점수: 34.0점

Class: Normal Class

## [ 문제 2 ] 아래 조건에 맞는 프로그램을 작성 하시오.

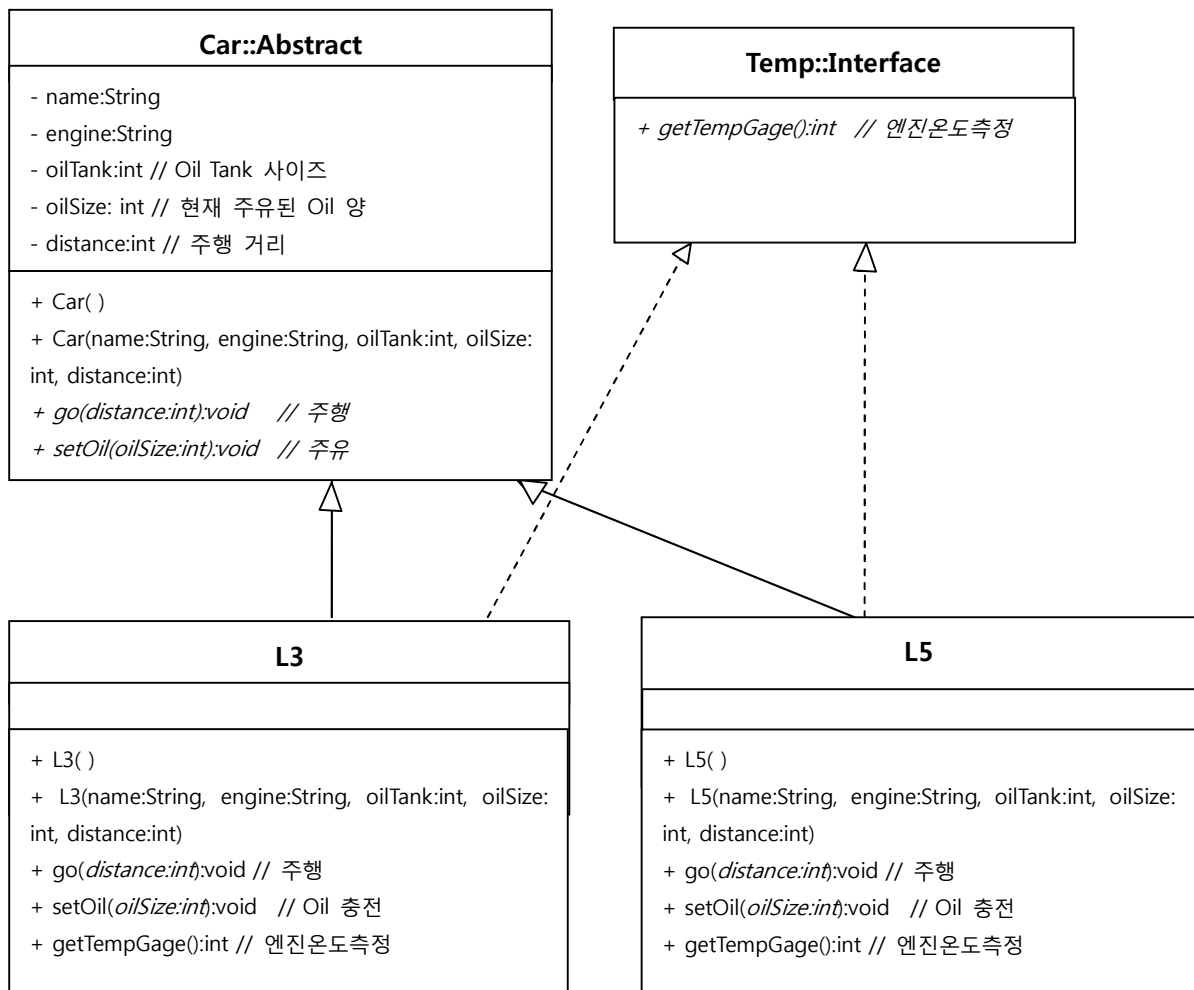
Abstract 클래스에서 상속받고 Interface를 구현한 두 개의 클래스를 구현 하여 실행 결과가 나오도록 프로그램을 작성 한다

### 1. 사용 데이터

| vehicleName | engineSize | oilTank | oilSize | distance |
|-------------|------------|---------|---------|----------|
| L3          | 1500       | 50      | 25      | 0        |
| L5          | 2000       | 70      | 35      | 0        |

- oilTank는 자동차의 기본적인 주유 가능 사이즈 이다
- oilSize는 자동차에 현재 주유 되어진 양을 의미한다, 최초에는 반만 주유 되어 있다
- distance는 자동차의 주행 거리 이며 초기 값은 0이다

### 2. 클래스 다이어그램



## 3. 구현 클래스

| Package명 | 클래스명    | 메소드   | 설명                              |
|----------|---------|---|---------------------------------|
| com.car  | Car     | +Car()  | 기본생성자                           |
|          |         | +Car(name:String, engine:String, oilTank:int, oilSize: int, distance:int) | 5개의 클래스변수를 받는 생성자               |
|          |         | +go(distance:int):void  | distance 만큼 주행 하고 현재 주유량을 감소 한다 |
|          |         | +setOil(oilSize:int):void   | 특정 양을 주유 한다                     |
|          | Temp    | +getTempGage():int  | 엔진 온도를 리턴 한다                    |
|          | L3      | +L3()   | 기본생성자                           |
|          |         | +L3(name:String, engine:String, oilTank:int, oilSize: int, distance:int)  | 5개의 클래스변수를 받는 생성자               |
|          |         | +go(distance:int):void  | 10주행 시 현재 주유량 1 감소              |
|          |         | +setOil(int oilSize):void   | 현재 주유량 증가                       |
|          |         | +getTempGage():int  | 엔진 온도를 리턴 한다<br>10주행 시 엔진온도 1증가 |
|          | L5      | +L5()   | 기본생성자                           |
|          |         | +L5(name:String, engine:String, oilTank:int, oilSize: int, distance:int)  | 5개의 클래스변수를 받는 생성자               |
|          |         | +go(distance:int):void  | 8주행 시 현재 주유량 1 감소               |
|          |         | +setOil(int oilSize):void   | 현재 주유량 증가                       |
|          |         | +getTempGage():int  | 엔진 온도를 리턴 한다<br>5주행 시 엔진 온도 1증가 |
|          | CarTest | +main(args:String[]):void   | main 함수를 이용 하여 동작               |

\* class 명과 method 명은 변경 하지 않는다

\* getXXX와 setXXX는 직접 구현 한다

## 4. CarTest 클래스 구조

```

public class CarTest {

    public static void main(String args[]) {
        // Car type의 객체 배열을 2개 만든다
        // 배열에 사용데이터에서 제공된 2개의 Car 객체를 넣는다.
        // 생성된 자동차의 기본 정보 출력
        // 각각의 자동차에 25씩 주유 한다
        // 25씩 주유한 자동차의 정보를 출력 한다
        // 각각의 자동차에 80씩 주행 한다
        // 80씩 주행한 정보를 출력 하며 엔진온도 정보를 출력 한다
    }
}

```

### 4. 실행 결과 예

| vehicleName | engineSize | oilTank | oilSize | distance | temperature |
|-------------|------------|---------|---------|----------|-------------|
| L3          | 1500       | 50      | 25      | 0        | 0           |
| L5          | 2000       | 75      | 35      | 0        | 0           |
| 25 주유       |            |         |         |          |             |
| vehicleName | engineSize | oilTank | oilSize | distance | temperature |
| L3          | 1500       | 50      | 50      | 0        | 0           |
| L5          | 2000       | 75      | 60      | 0        | 0           |
| 80 주행       |            |         |         |          |             |
| vehicleName | engineSize | oilTank | oilSize | distance | temperature |
| L3          | 1500       | 50      | 42      | 80       | 8           |
| L5          | 2000       | 75      | 50      | 80       | 16          |