

OOP 기반 중고차 재고관리 프로그램 작성

SORENTO

- ID : 1
- 차종 : **SUV**
- 차명: SORENTO
- 가격: 20,000
- 색깔: BLUE

GV80

- ID : 2
- 차종 : **SUV**
- 차명: GV80
- 가격: 50,000
- 색깔: RED

EV6

- ID : 3
- 차종 : **SEDAN**
- 차명: EV6
- 가격: 30,000
- 색깔: BLACK

G90

- ID : 4
- 차종 : **SEDAN**
- 차명: G90
- 가격: 70,000
- 색깔: YELLOW

- 위의 4가지 차종에 대한 중고차 재고 관리 프로그램을 작성하라
- 중고차를 저장 및 출력하기 위한 클래스로 “StockMngr”를 작성하고, 모든 자동차 객체를 관리하기 위해 “Stock” 인터페이스를 사용한다.
- 아래 예제코드를 참고하여, 위 4가지 차종에 대한 클래스 설계 및 구현

- 아래 추상메소드 외 추상 메서드와 상수 추가 가능

```
4 interface Stock {  
5     // 차량의 ID값 출력, 차량이 리스트에 저장될 때 1씩 증가  
6     int getId();  
7  
8     // SEDAN, SUV 구분, 예) 자동차가 GV80일 경우 "SUV" 리턴  
9     String getType();  
10  
11     // 제품명 리턴, 예) EV6  
12     String getProductName();  
13  
14     // 제품 색상 값 리턴, 예) BLACK  
15     String getProductColor();  
16  
17     // 제품 가격 리턴, 예) EV6 -> 30,000  
18     String getProductPrice();  
19 }
```

- 아래 메서드는 반드시 구현, 추가 메서드 및 멤버변수 선언 및 구현 가능

```
23 class StockMngr {
24     // 중고차 저장 리스트
25     Stock stock_list[];
26
27     public StockMngr(int argStockBucketSize) {
28         stock_list = new Stock[argStockBucketSize];
29     }
30
31     // 중고차 리스트에 저장, 반환값은 현재 저장된 중고차 대수
32     int insertStock(Stock argStock) {
33         return 0;
34     }
35
36     // 차량 리스트 출력: 아래 양식으로 출력
37     void printStockList() {
38         // 1. ID : 1 차종 : SUV,      차명 : SORENTO, 가격 : 20,000, 색깔 : BLUE
39         // 2. ID : 2 차종 : SEDAN,   차명 : G90,      가격 : 70,000, 색깔 : YELLOW
40     }
41
42     // 조건에 맞는 차량 리스트 출력
43     void printStockList(String argType) {
44         // argType은 SUV 또는 SEDAN -> SUV일 경우 SUV만 출력
45     }
46 }
```

- 아래 메서드는 반드시 구현, 추가 메서드 및 멤버변수 선언 및 구현 가능

```

48 public class CarStockManagementProg {
49     StockMngr myStock;
50
51     public CarStockManagementProg () {
52         myStock = new StockMngr(10);
53
54         // 아래 메뉴 출력 후 입력 값에 따라 동작
55         // 1. 자동차 입력
56         // 차종 : SUV
57         // 차명 : GV80
58         // 가격 : 50,000
59         // 색깔 : BLUE
60         // 2. 전체 자동차 목록 출력 : 입력 순서로 출력
61         // 1. ID : 1 차종 : SUV, 차명 : SORENTO, 가격 : 20,000, 색깔 : BLUE
62         // 2. ID : 2 차종 : SEDAN, 차명 : G90, 가격 : 70,000, 색깔 : YELLOW
63         // 3. SUV 자동차 목록 출력
64         // 1. ID : 1 차종 : SUV, 차명 : SORENTO, 가격 : 20,000, 색깔 : BLUE
65         // 2. ID : 4 차종 : SUV, 차명 : SORENTO, 가격 : 10,000, 색깔 : YELLOW
66         // 4. SEDAN 자동차 목록 출력
67         // 1. ID : 1 차종 : SEDAN, 차명 : EV60, 가격 : 40,000, 색깔 : BLUE
68         // 2. ID : 2 차종 : SEDAN, 차명 : G90, 가격 : 70,000, 색깔 : YELLOW
69         // 5. 프로그램 종료
70
71         // 1~4번 항목 실행 후 다시 메뉴 출력
72
73     }

```

메뉴 선택 시 실행 예제

- 메뉴 1번 선택 시 실행

```
static int insertCar(String argType, String argName,  
                    int argPrice, String argColor) {  
    // 입력 값을 기반으로 해당 차량 객체 생성 후 입력  
    // 반환 값은 현재 myStock에 저장된 차량 대수  
    // 예) SUV, GV80, 50000, RED  
    // 1. GV80 객체 생성 : 차량 ID는 객체 생성 시 마다 1씩 증가  
    // 2. myStock.insert(GV80)  
    return 0;  
}  
}
```