

1주차 과제

과제	<p>1. 차량 정보 저장을 위한 Car클래스를 작성하라. Car클래스는 브랜드, 모델명, 가격을 저장하고 있다.</p> <p>2. Car클래스 관리를 위한 CarManager클래스를 작성하라. CarManager의 carList는 vector에 Car 인스턴스를 저장해 가지고 있다. 입력은 파일로부터 받는다.</p>
파일	common.hpp, class.hpp, class.cpp, main.cpp로 파일분할하고 클래스 선언과 정의 분리할것.
클래스 다이어그램	<pre> classDiagram class Car { -brand -model -price +Car() +GetBrand() +GetModel() +GetPrice() +SetPrice() } class CarManager { -carList -fileName +~CarManager() +CarManager() +ChangePrice() +FindCarInfo() +Menu() +PrintCarList() +SaveChanges() +SortCarList() +SortCmp() } </pre>
클래스 정보 (설명 필요한 정보만 기술)	<pre> - class Car string brand //브랜드명 string model //모델명 int price //가격 Car(string, string, int) //생성자 void SetPrice(int price) //가격 재설정을 위한 setter - class CarManager vector<Car> carList //Car의 인스턴스를 가지는 벡터 string fileName; //파일 이름 CarManager(string) //생성자, 파일이름 인자로 받아 그 파일에서 정보 읽어 carList에 저장할 것 void Menu() //메뉴 출력 함수 </pre>

	void PrintCarList() //carList에 저장된거 출력하는 함수 void FindCarInfo(string brand, string model) //brand와 model을 넘기면 차량 정보를 찾아 출력해주는 함수 void ChangePrice(string brand, string model, int price) //brand, model, price를 넘기면 brand와 model이 일치하는 Car인스턴스를 찾아 price로 변경해주는 함수 void SortCarList() //carList 정렬해주는 함수, 브랜드 -> 모델 순으로 정렬할 것 void SaveChanges() //carList에 있는 정보 파일에 다시 저장
요구조건	1. 모든 멤버 변수 접근 지정자 private으로 둘 것 2. Car생성시 생성자에서 멤버 이니셜라이저 사용할 것 3. vector 사용해 볼 것 4. 파일 입출력시 예외처리 구문 사용해볼 것 5. SortCarList() 내부적으로 <algorithm>에 정의된 sort 함수 사용할 것, 사용시SortCmp() 정의해서 compare함수로 쓸 것(람다식 사용할줄 알면 그거 써도 됨). 정렬 알고리즘 직접 구현하지 말 것(quick sort 구현하면 인정)
주의사항	
예시용 데이터	CarListFile.txt
목적	파일 입출력, 예외처리, OOP, STL

실행 예

```
Car Manager Menu
1. Print Car List
2. Find Car Info
3. Change Price
4. Sort Car List
5. Save Changes
Input : 1

hyundai sonata 3000
hyundai avante 2200
cadillac cts 5500
toyota camry 3000
nissan maxima 4000
volkswagen passat 3500
```

```
Car Manager Menu
1. Print Car List
2. Find Car Info
3. Change Price
4. Sort Car List
5. Save Changes
Input : 2
Brand : toyota
Model : camry
```

```
toyota camry 3000
```

```
Car Manager Menu
1. Print Car List
2. Find Car Info
3. Change Price
4. Sort Car List
5. Save Changes
Input : 3
Brand : hyundai
Model : sonata
Price : 2800
```

```
Price Changed
```

```
Car Manager Menu
1. Print Car List
2. Find Car Info
3. Change Price
4. Sort Car List
5. Save Changes
Input : 4
```

```
Car Manager Menu
1. Print Car List
2. Find Car Info
3. Change Price
4. Sort Car List
5. Save Changes
Input : 1
```

```
cadillac cts 5500
hyundai avante 2200
hyundai sonata 2800
nissan maxima 4000
toyota camry 3000
volkswagen passat 3500
```

```
Car Manager Menu
1. Print Car List
2. Find Car Info
3. Change Price
4. Sort Car List
5. Save Changes
Input : █
```

아래 두 함수 그대로 써도 됨

main()

```
int main()
{
    CarManager carManager("CarListFile.txt");
    carManager.Menu();
}
```

CarManager::Menu()

```
void CarManager::Menu()
{
    enum menu { printCarList = 1, findCarInfo = 2, changePrice = 3, sortCarList = 4, saveChanges = 5 };
    while (true)
    {
        int sel;
        cout << "Car Manager Menu" << endl;
        cout << "1. Print Car List" << endl;
        cout << "2. Find Car Info" << endl;
        cout << "3. Change Price" << endl;
        cout << "4. Sort Car List" << endl;
        cout << "5. Save Changes" << endl;
        cout << "Input : ";
        cin >> sel;

        string brand, model;
        int price;
        switch (sel)
        {
            case printCarList:
                cout << endl;
                PrintCarList();
                break;
            case findCarInfo:
                cout << "Brand : ";
                cin >> brand;
                cout << "Model : ";
                cin >> model;
                cout << endl;
                FindCarInfo(brand, model);
                break;
            case changePrice:
                cout << "Brand : ";
                cin >> brand;
                cout << "Model : ";
                cin >> model;
                cout << "Price : ";
                cin >> price;
                cout << endl;
                ChangePrice(brand, model, price);
                break;
            case sortCarList:
                SortCarList();
                break;
            case saveChanges:
                SaveChanges();
                break;
            default:
                cout << "Input Val Error" << endl;
                break;
        }
        cout << endl << endl;
    }
}
```