

[문제] 다음과 같은 조건을 만족하는 프로그램을 작성 하시오

MVC패턴을 이용한 ArrayList 문제이다. 해당 구현 클래스 다이어그램과 클래스 구조를 참고하여 프로젝트를 완성하시오

### 1. 구현 클래스 다이어그램 (Class Diagram)

Car
- parkingNum : int // 주차 번호 - carNum: int // 차량 번호 - carType : int // 차량 종류 - owner : String // 차주 이름
+ Car() + Car(carNum : int, carType : int, owner : String) + setter() / getter() + toString() : String

ParkingTowerMain
+ <u>main(args:String[]) : void</u>

ParkingTowerMenu
~ sc : Scanner // 생성 - ptm : ParkingTowerManager // 생성
+ ParkingTowerMenu() + mainMenu() : void + insertCar() : void // 주차 + selectList() : void // 전체 조회

ParkingTowerManager
- carList : ArrayList<Car> // 생성
+ ParkingTowerManager() + insertCar(car:Car) : void + selectList() : ArrayList<Car>

### 3. 구현 클래스 설명

Package명	Class명	Method	설명
run	Run	<u>+main(args:String[]) : void</u>	ParkingTowerMenu 클래스 객체 생성하고 mainMenu() 메소드 실행
model	Car	+ Car()	기본생성자
		+ Car(carNum:int, carType:int, owner:String)	매개변수 3개짜리 생성자 (주차번호는 매개 변수로 받지 않아요)
		+ toString() : String	필드 값 문자열 합친 후 리턴 category 분류 별로 출력 1 : 경차 / 2 : 세단 3 : SUV / 4 : 트럭
view	ParkingTower Menu	+ ParkingTowerMenu()	기본 생성자
		+ mainMenu() : void	주차타워 관리 프로그램에 해당하는 메인 메뉴 출력, 각 메뉴에 해당하는 메소드 실행 → 반복 출력되게 함
		+ insertCar() : void	차량번호, 차량종류, 차주이름을 키보드로 입력 받고 입력 받은 값을 가지고 Car객체 생성 생성한 Car 객체를 ParkingTowerManager의 insertCar 메소드로 전달
		+ selectList() : void	ParkingTowerMangaer의 selectList 메소드 호출하여 리턴 받은 리스트가 비어있는 경 우 "주차된 차량이 없습니다." 출력 비어있지 않은 경우 전체 리스트 목록 출력

\* 위와 같이 추가, 삭제, 검색에 필요한 정보는 키보드로 입력 받도록 각각의 메소드 따로 구현

Package명	Class명	Method	설명
controller	ParkingTower Manager	+ ParkingTowerManager()	기본 생성자
		+ insertCar(car:Car) : void	전달받은 Car객체의 차량번호를 setter메소드를 통해 설정 한 후 에 carList에 추가
		+ selectList() : ArrayList<Car>	→ carList 리턴

```
public class ParkingTowerMenu{

    // Scanner 객체 생성

    // ParkingTowerManager 객체 생성 : ptm


    public void mainMenu() {

        === *주차타워 관리* ===

        1. 차량 주차                → insertCar()
        2. 전체 출력                 → selectList()
        0. 끝내기                   → "프로그램 종료" 출력 후 main()으로 리턴
        메뉴 선택 :                  >> 입력 받음
                                    >> 메뉴 화면 반복 실행 처리

    }



    public void insertCar(){

        // "차량 번호 :"              >> 입력 받음 (carNum)
        // "차량량 타입 (1:경차 / 2:세단 / 3:SUV / 4:트럭) : "      >> 입력 받음 (carType)
        // "차주 :"                    >> 입력 받음 (owner)


        // 위에서 입력 받은 carNum, carType, owner를 매개변수로 한 Car 객체 생성 (car)
        // ParkingTowerManager의 insertCar 메소드로 car 전달

    }




    public void selectList(){

        // ParkingTowerManager의 selectList() 메소드 호출
        // → 리턴 값 전달 받음 (carList)


        // carList가 비어 있을 경우          >> "주차된 차량이 없습니다."출력
        // carList가 비어있지 않을 경우       >> "for문을 이용하여 carList 출력
                                            >> 또는 for each 문 이용하여 출력

    }

}
```

```
public class CarManager{  
    // ArrayList 객체 생성 (carList)  
  
    public void insertCar(Car car) {  
        // 전달 받은 car는 현재 주차번호가 null인 채로 들어오는데  
        // 새로운 차량이 추가될 때마다 추가되는 차량의 주차번호는  
        // 리스트 마지막 주차 번호의 다음번호로 부여 해야 됨  
        int lastNo = 0; // 우선 변수 생성 및 초기화  
        lastNo = carList.get(carList.size()-1).getParkingNum() + 1; // 마지막 차량량 번호 + 1  
        // 하지만 리스트에 차량이 없는 경우, 즉 첫 차량 등록일 경우 위의 문장에서 예외 발생  
        ➔ 어떤 예외처리가 발생하는지 알아보고 try/catch문을 이용하여 오류 해결  
        // 해당 예외 발생 시 lastNo = 1; 로 초기화  
        // setter를 이용하여 car 주차 번호를 lastNo로 적용  
        // carList의 car 추가  
    }  
  
    public ArrayList<Car> selectList(){  
        // carList 리턴  
    }  
}
```