

**[문제 2] 다음과 같은 조건을 만족하는 프로그램을 작성 하시오**

[문제 1]는 MVC패턴을 이용한 ArrayList 문제였다. 이 문제는 [문제 1]를 HashMap을 이용하여 푸는 문제이다. [문제 2] 프로젝트를 복사하여 다음 클래스 다이어그램과 클래스 구조를 참고하여 수정해 가면서 어떤 차이가 있는지 확인해보자. 오히려 헷갈린다면 다시 처음부터 구현하자.

**1. 구현 클래스 다이어그램 (Class Diagram)**

Car
- <del>parkingNum : int</del> // <del>주차 번호</del>
- 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
+ <del>selectMap()</del> : void // 전체 조회
+ deleteCar() : void // 출차
+ <del>searchCar()</del> : void // 차량 검색

ParkingTowerManager
- <del>carMap</del> : HashMap<Integer, Car> // 생성
- <del>parkingNo</del> : int //생성, 0으로 초기화
+ ParkingTowerManager()
+ insertCar(car:Car) : void
+ <del>selectMap()</del> : Map<Integer, Car>
+ deleteCar(carNum:int) : Car
+ searchCar(owner:String) : <del>HashMap&lt;Integer, Car&gt;</del>

### 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
		+ selectMap() : void	ParkingTowerMangaer의 <b>selectMap</b> 메소드 호출하여 리턴 받은 리스트가 비어있는 경 우 "주차된 차량이 없습니다." 출력 비어있지 않은 경우 전체 맵목록 출력
		+ deleteCar() : void	출차할 차량번호를 키보드로 입력 받아 ParkingManager의 deleteCar 메소드로 전달 → 리턴 받은 결과 값을 가지고 출력
		+ searchCar() : void	검색할 차주 이름을 키보드로 입력 받아 ParkingTowerManager의 searchCar 메소드 로 전달 리턴 받은 맵이 비어있는 경우 "검색 결과 없음" 출력 리턴 받은 맵이 비어있지 않을 경우 검색 결과 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
		+ selectMap() : Map<Integer,Car>	→ carMap 리턴
		+ deleteCar(carNum:int) : Car	carMap 의 차들 중 전달 받은 carNum값이 일치하는 차가 있을 경우 삭제 → del 리턴
		+ searchCar(owner:String) : Map<Integer, Car>	carMap의 차량들 중 전달 받은 owner값을 포함(contains)한 차량 을 searchMap에 추가 → searchMap 리턴

#### 4. class 구조

```
public class ParkingTowerMenu{
    // Scanner 객체 생성
    // ParkingTowerManager 객체 생성 : ptm

    public void mainMenu() {
        === *주차타워 관리* ===
        1. 차량 주차                → insertCar()
        2. 전체 출력                → selectMap()
        3. 차량 출차                → deleteCar()
        4. 주차 차량 검색            → searchCar()
        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 selectMap (){
        // ParkingTowerManager의 selectMap() 메소드 호출
        // → 리턴 값 전달 받음 (carMap)

        // carMap가 비어 있을 경우                >> "주차된 차량이 없습니다."출력
        // carMap가 비어있지 않을 경우            >> Iterator 이용하여 출력
    }
```

```

public void deleteCar(){
    // "차량 번호 : "                                >> 입력 받음 (carNum)
    // ParkingTowerManager의 deleteCar 메소드로 carNum 전달
    // → 리턴 값 전달 받음 (res)

    // res가 null이 아닐 경우                >> res 출력, "차량 출차 완료!" 출력
    // res가 null일 경우                    >> "입력하신 차량이 주차되어 있지 않습니다."출력
}

```

```

public void searchCar() {
    // "주차 차량 검색(차주입력) : "                >> 입력 받음 (owner)
    // ParkingTowerManager의 searchCar 메소드로 owner 전달
    // → 리턴 값 전달 받음 (searchMap)

    // searchMap가 비어 있을 경우                >> "검색한 차량이 주차 되어 있지 않습니다."출력
    // searchMap가 비어있지 않을 경우            >> Iterator 이용하여 출력

}
}

```

```
public class CarManager{
    // HashMap<Integer, Car> 객체 생성 (carMap)
    // int 형 parkingNo=0 선언
    public void insertCar(Car car) {
        //차량이 추가될 때 Map의 key값을 주차 번호로하고 value값을 Car객체로한다.
        //이때 키값은 parkingNO를 1씩 추가하면 사용

        //carMap에 put을 사용하여 차량 추가
    }

    public int deleteCar(int carNum){
        // carMap을 entrySet()을 통해 Set타입으로 바꿔주고
        // for문을 통해 내가 전달받은 carNum값을을 getValue()와 비교후 존재 하는 경우 삭제한다.
        // // 성공적으로 삭제 할 경우 1 리턴
        // 삭제 되지 않은 경우 즉, 존재하는 도서 번호가 Key값에 없는 경우 0 리턴
    }

    public Map<Integer, Car> searchCar(String owner){
        Map<Integer, Car> searchMap = new HashMap<Integer, Car> ();

        // map의 entrySet(), Iterator를 통해 반복문 실행
        // → entry의 getValue() 중 전달받은 title을 포함한 도서가 있을 경우
        // searchMap에 entry의 getKey()값과 entry의 getValue()값을 가지고 저장
        // searchMap 리턴
    }

    public Map<Integer, Car> selectMap(){
        // carMap 리턴
    }
}
```