



# ParkingLot Program System



# 팀 구성원 소개



신선호

PPT 제작

안위재

프로그램구현

이상목

의견조율  
프로그램구현  
오류수정

이종운

프로그램구현  
발표

최재희

PPT 제작



WHAT  
WE DO ?





## 목표

학습 내용을 다양하게  
활용할 수 있는 방안을  
고려하여  
주차관리시스템을  
구현하고자 함



## 구성

입출차 관리  
회원 관리  
주차장 관리



## 프로그램구현

추상메소드 활용  
ArrayList 활용  
Try-Catch 활용

# 프로젝트일정

1219

프로젝트 주제 설정  
역할 분담

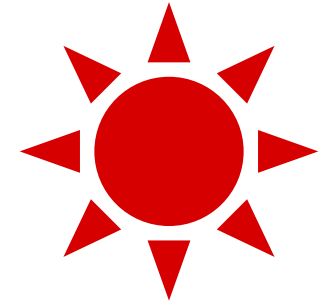
1220

프로그램 코드 분담

1221

1222

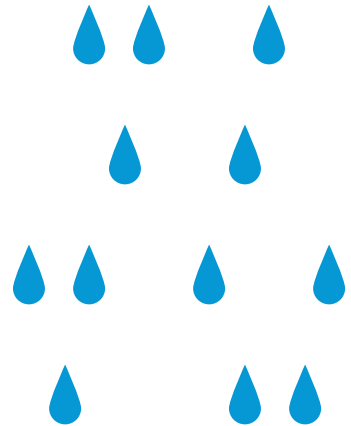
1226



프로그램 구현

프로그램 오류 수정

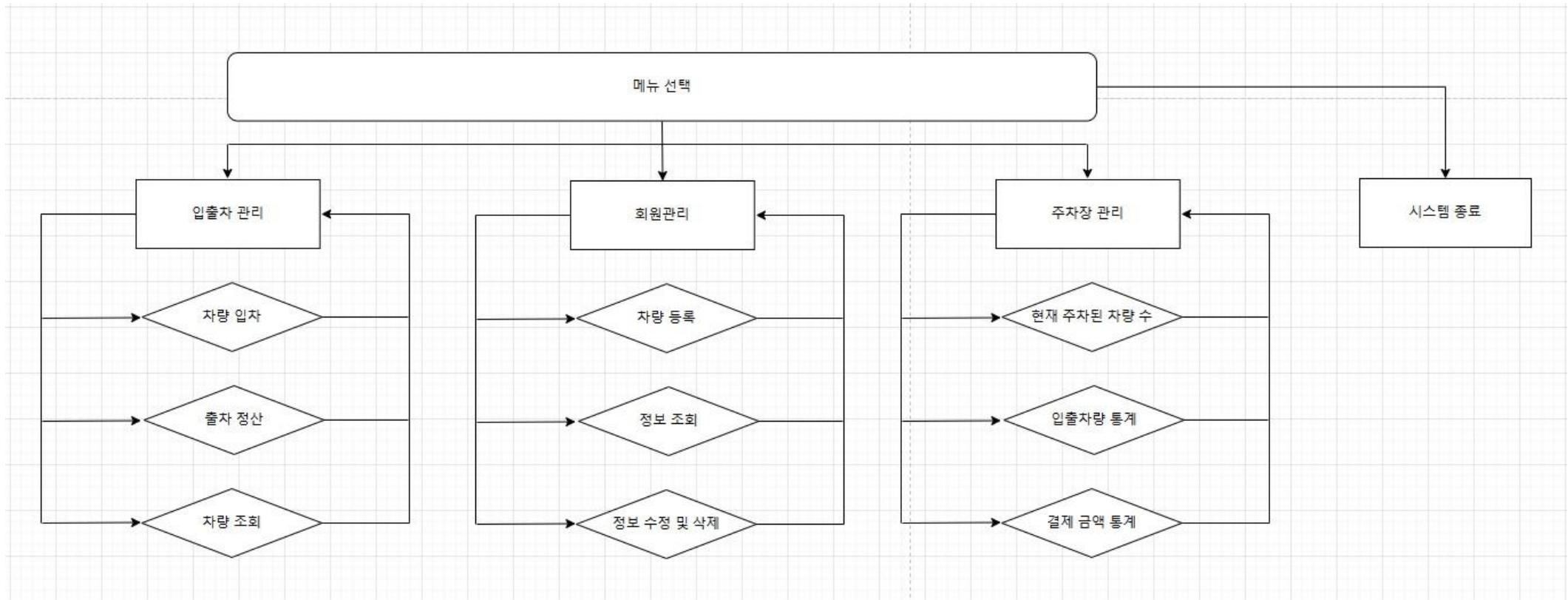
PPT 제작



- 1 알고리즘
- 2 주요코드설명
- 3 실행결과
- 4 자체평가



# 일고리즘



# 주요코드설명





# Main 주요코드

```
package com.java.parking;

import java.util.Arrays;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int selectUMenu; // 전체 메뉴
        int selectLMenu1; // 입출차 관리
        int selectLMenu2; // 회원 관리
        int selectLMenu3; // 주차장 관리
        int selectLMenu2_1; // 회원 정보 조회(전체 or 특정)
        int selectLMenu2_2; // 회원 정보 수정 및 삭제

        boolean runUMenu = true; // 전체 메뉴
        boolean runLMenu1 = true; // 입출차 관리
        boolean runLMenu2 = true; // 회원 관리
        boolean runLMenu3 = true; // 주차장 관리
        boolean runLMenu2_1 = true; // 회원 정보 조회(전체 or 특정)
        boolean runLMenu2_2 = true; // 회원 정보 수정 및 삭제

        ParkingLot parkingLot = new ParkingLot(); // 주차장 객체 생성
        parkingLot.setTest(); // 테스트용 자료 추가
        MemberList memberList = new MemberList(); // 멤버 리스트 객체 생성

        while(runUMenu) {
            Utils.showUI("메뉴 선택");
            List<String> mainMenu = Arrays.asList("입출차 관리", "회원 관리", "주차장 관리", "
            selectUMenu = Utils.showMenu(mainMenu, sc);
```

```
package com.java.parking;

import java.time.Instant;

public class Utils {
    // 메뉴 출력
    public static int showMenu(List<String> menu, Scanner sc) {
        while(true) {
            try { //예외처리
                int i = 1;
                for(String item : menu) {
                    System.out.printf("%d. %-15s", i, item);
                    i++;
                }
                System.out.println();
                System.out.print("> ");
                int selectedMenu = sc.nextInt();
                sc.nextLine();
                if (selectedMenu > menu.size() || selectedMenu < 1) {
                    throw new IllegalArgumentException("범위 내 숫자 값을 입력해 주세요.");
                }
                return selectedMenu;
            } catch (IllegalArgumentException e) {
                System.out.println(e.getMessage());
            } catch (Exception e) {
                System.out.println("잘못된 입력입니다.");
                sc.nextLine();
            }
        }
    }
}
```

# 입출차관리 주요코드

```
case 1:
    runLMenu1 = true;
    while(runLMenu1) {
        Utils.showUI("입출차 관리 메뉴 선택");
        List<String> subMenu1 = Arrays.asList("차량 입차", "출차 정산", "입차 차량 조회", "이전 메뉴");
        selectLMenu1 = Utils.showMenu(subMenu1, sc);

        switch(selectLMenu1) {
            // 차량 입차
            case 1:
                Utils.showUI("차량 입차");
                parkingLot.carIn();
                break;
            // 출차 정산
            case 2:
                Utils.showUI("출차 정산");
                if(parkingLot.getSpace() > 0) { // 출차 차량 존재 여부 확인
                    System.out.print("출차 차량 번호 입력 > ");
                    String outCarNum = sc.nextLine(); // 출차 차량 번호 입력

                    Car it = Utils.findCarInst(parkingLot.currentCars, outCarNum);
                    if (it == null) {break; }
                    // 1시간 추가
                    it.addAnHour();

                    NewPayment pay = new NewPayment(it, parkingLot);
                    if (memberList.mlist.findMember(outCarNum) != null) {
                        String test = memberList.mlist.findMember(outCarNum).carNum;
                        System.out.println("테스트용: " + test);
                    }
                    pay.setDiscount(memberList.mlist);

                    while (!pay.car.isPaid) {
                        pay.car.setTimeout();
                        System.out.printf(" %d원을 계산해 주세요.\n", pay.getAmount());
                        System.out.print("> ");
                        int amount = sc.nextInt();
                        pay.car = pay.execPay(amount);
                    }

                    Utils.showCarInfo(outCarNum, "출고");
                    Utils.showLotInfo(parkingLot.getCurrentSpace()); // 남은 주차 자리 출력
                    break; }

                else { System.out.println(parkingLot.errorType(2));
                    break;}
```

# 입출차관리 주요코드

```
10 interface InnerParkingLot
11 {
12     void carIn();
13     void carOut(Car car);
14     int getCurrentSpace();
15     String errorType(int errorCode);
16     void setPricePerTenMin(int price);
17     void showAllOutCarPayments();
18 }
19
20 public class ParkingLot implements InnerParkingLot{
21     CarArray currentCars = new CarArray(); // 데이터베이스 역할
22     CarArray outCars = new CarArray(); // 데이터베이스 역할
23     int spaceTotal = 70; // 전체 주차가능 수
24     // 현재 주차중인 수, private을 활용하여 외부 접근 제한하여 carIn과 carOut함수만으로 값을 수정 할 수 있도록 구현함.
25     private int space = this.currentCars.size();
26     int pricePerTenMin = 200; // 10분당 요금
27
28     // Payment 클래스로 구현하였으나 구조가 맞지 않아 새롭게 개발
29     public class NewPayment {
30         Car car;
31         ParkingLot parkingLot;
32         double discount = 0;
33
34         public NewPayment(Car car, ParkingLot parkingLot) {...}
35
36         public void setDiscount(double d) {...}
37
38         public void setDiscount(MemberArray ma) {...}
39
40         int getAmount() {...}
41
42         int getChanges(int amount, int receive) {...}
43
44         Car execPay(int receive) {...}
45     }
46 }
```



# 회원관리 주요코드

```
// 회원 정보 조회
case 2:
    runLMenu2_1 = true;

    if(memberList.mlist.size() != 0) {
        while(runLMenu2_1) {
            Utils.showUI("회원 정보 조회");
            List<String> subMenu2_1 = Arrays.asList("전체 회원 조회", "특정 회원 조회");
            selectLMenu2_1 = Utils.showMenu(subMenu2_1, sc);

            switch(selectLMenu2_1) {

                // 회원 전체 조회
                case 1:
                    Utils.showUI("전체 회원 조회");
                    for(Member m : memberList.mlist.sortMemberById()) {
                        m.memberPrint();
                    }
                    break;

                // 특정 회원 조회
                case 2:
                    Utils.showUI("특정 회원 조회");
                    System.out.print("조회할 회원 정보 입력(ID, 이름, 차량 번호 中 1) > ");
                    String whatFind = sc.nextLine(); // 정보 입력
                    memberList.mlist.showMember(whatFind);

                    break;
            } break;
        } break;
    } break;
}
else {
    System.out.println("조회할 정보가 없습니다.");
    break;
}
```

# 회원관리 주요코드

```
5 public class Member { // 회원 정보 모델
6     String memId;
7     String memName;
8     String carNum;
9     Instant issueDate = Instant.now();
10
11 > public Member(String memId, String memName, String carNum) { ...
16
17 > public String getId() { ...
20
21 > public String getCarNum() { ...
24
25 > public String getName() { ...
28
29 > public void setId(String memId) { ...
32
33 > public void setName(String memName) { ...
36
37 > public void setNum(String carNum) { ...
40
41 > public void memberPrint() { ...
44 }
```

```
5 public class MemberList {
6     MemberArray mlist = new MemberArray();
7     String memId;
8     String memName;
9     String carNum;
10
11 // MemberArray로 대체 할 예정이었으나
12 // 초기 메인 함수에 많이 사용되어 호환성을 위해 MemberArray를 필드로 받음.
13 > public void addMember() { ...
28
29 }

7 interface InnerMemberArray {
8     boolean isMember(String carNum);
9     Member findMember(String any);
10     void showAll();
11     void showMember(String numOrName);
12     MemberArray sortMemberById();
13     MemberArray sortMemberIssueDate();
14 }

15
16 // ArrayList로 extends 하여 ArrayList의 속성은 그대로 가져오면서 필요한 추가 기능을 구현
17 > public class MemberArray extends ArrayList<Member> implements InnerMemberArray { ...
```

# 주차장관리 주요코드

```
case 3:
    runLMenu3 = true;
    while(runLMenu3) {
        Utils.showUI("주차장 관리 메뉴 선택");
        List<String> subMenu3 = Arrays.asList("현재 주차된 차량 현황", "입출차 차량 통계", "결제된 금액 통계", "이전 메뉴");
        selectLMenu3 = Utils.showMenu(subMenu3, sc);

        switch(selectLMenu3) {
            // 현재 주차된 차량 현황
            case 1:
                Utils.showUI("현재 주차된 차량 현황");
                System.out.println("주차된 차량 목록");

                for(Car c : parkingLot.currentCars) {
                    c.carPrint();
                }
                System.out.println("주차된 차량 수 : " + parkingLot.getSpace());
                break;
            // 입출차 차량 통계
            case 2:
                Utils.showUI("입출차 차량 통계");
                parkingLot.outCars.showAllCar();
                break;
            // 결제된 금액 통계
            case 3:
                Utils.showUI("결제된 금액 통계");
                parkingLot.outCars.showAllPay();
                break;
            // 이전 메뉴
            case 4:
                runLMenu3 = false;
                break;
        }
    }
    break;

case 4:
    Utils.showUI("시스템 종료");
    runUMenu = false;
    break; }
} } }
```



# 실행결과



# 실행결과

◎ 메뉴 선택 ◎

1. 입출차 관리 > 1      2. 회원 관리      3. 주차장 관리      4. 종료

◎ 입출차 관리 메뉴 선택 ◎

1. 차량 입차 > 1      2. 출차 정산      3. 입차 차량 조회      4. 이전 메뉴

◎ 차량 입차 ◎

입차 차량 번호 입력(0 입력시 랜덤 생성) > 123가1234

1. 일반      2. 경차  
> 1

123가1234 차량이 입고 되었습니다.  
주차 자리 69개 남았습니다.

◎ 입출차 관리 메뉴 선택 ◎

1. 차량 입차 > 1      2. 출차 정산      3. 입차 차량 조회      4. 이전 메뉴

◎ 차량 입차 ◎

입차 차량 번호 입력(0 입력시 랜덤 생성) > 123나1234

1. 일반      2. 경차  
> 2

123나1234 차량이 입고 되었습니다.  
주차 자리 68개 남았습니다.

◎ 입출차 관리 메뉴 선택 ◎

1. 차량 입차 > 1      2. 출차 정산      3. 입차 차량 조회      4. 이전 메뉴

◎ 차량 입차 ◎

입차 차량 번호 입력(0 입력시 랜덤 생성) > 123다1234

1. 일반      2. 경차

◎ 입차 차량 조회 ◎

조회할 차량 번호 입력 > 123가1234

차량 번호: 123가1234 | 차량 종류: 일반 | 입차 시간: 2023-12-22 15:01:41

◎ 입출차 관리 메뉴 선택 ◎

1. 차량 입차 > 3      2. 출차 정산      3. 입차 차량 조회      4. 이전 메뉴

◎ 입차 차량 조회 ◎

조회할 차량 번호 입력 > 123나1234

차량 번호: 123나1234 | 차량 종류: 경차 | 입차 시간: 2023-12-22 15:01:45

◎ 입출차 관리 메뉴 선택 ◎

1. 차량 입차 > 4      2. 출차 정산      3. 입차 차량 조회      4. 이전 메뉴

◎ 메뉴 선택 ◎

1. 입출차 관리 > 3      2. 회원 관리      3. 주차장 관리      4. 종료

◎ 주차장 관리 메뉴 선택 ◎

1. 현재 주차된 차량 현황 > 1      2. 입출차 차량 통계      3. 결제된 금액 통계      4. 이전 메뉴

◎ 현재 주차된 차량 현황 ◎

주차된 차량 목록

차량 번호: 123가1234 | 차량 종류: 일반 | 입차 시간: 2023-12-22 15:01:41

차량 번호: 123나1234 | 차량 종류: 경차 | 입차 시간: 2023-12-22 15:01:45

차량 번호: 123다1234 | 차량 종류: 일반 | 입차 시간: 2023-12-22 15:01:51

차량 번호: 123라1234 | 차량 종류: 경차 | 입차 시간: 2023-12-22 15:01:57

차량 번호: 123마1234 | 차량 종류: 일반 | 입차 시간: 2023-12-22 15:02:04

주차된 차량 수 : 5

남은 주차 공간 : 65

# 실행결과

## ◎ 입출차 차량 통계 ◎

차량번호	차량타입	결제금액	입차시간
123가1234	일반	0원	2023-12-22 15:01:41
123나1234	경차	0원	2023-12-22 15:01:45
123다1234	일반	0원	2023-12-22 15:01:51
123라1234	경차	0원	2023-12-22 15:01:57
123마1234	일반	0원	2023-12-22 15:02:04

총 입차 차량: 5대

총 출차 차량: 0대

전체 입출차 차량: 5대

## ◎ 주차장 관리 메뉴 선택 ◎

1. 현재 주차된 차량 현황  
> 3
2. 입출차 차량 통계
3. 결제된 금액 통계
4. 이전 메뉴

## ◎ 결제된 금액 통계 ◎

출차된 차량이 없습니다.

## ◎ 주차장 관리 메뉴 선택 ◎

1. 현재 주차된 차량 현황  
> 4
2. 입출차 차량 통계
3. 결제된 금액 통계
4. 이전 메뉴

## ◎ 메뉴 선택 ◎

1. 입출차 관리  
> 1
2. 회원 관리
3. 주차장 관리
4. 종료

## ◎ 입출차 관리 메뉴 선택 ◎

1. 차량 입차  
> 2
2. 출차 정산
3. 입차 차량 조회
4. 이전 메뉴

## ◎ 출차 정산 ◎

출차 차량 번호 입력 > 123가1234

1200원을 계산해 주세요.

> 1000

다시 정산을 해주시기 바랍니다. 감사합니다.

200원을 계산해 주세요.

> 500

-----2023-12-22 15:02:57

거스름돈은 300원 입니다. 감사합니다.

123가1234 차량이 출고 되었습니다.

주차 자리 66개 남았습니다.

## ◎ 입출차 관리 메뉴 선택 ◎

1. 차량 입차  
> 2
2. 출차 정산
3. 입차 차량 조회
4. 이전 메뉴

## ◎ 출차 정산 ◎

출차 차량 번호 입력 > 123나1234

600원을 계산해 주세요.

> 600

-----2023-12-22 15:03:08

안녕히 가세요. 감사합니다.

123나1234 차량이 출고 되었습니다.

주차 자리 67개 남았습니다.

## ◎ 입출차 관리 메뉴 선택 ◎

1. 차량 입차  
> 3
2. 출차 정산
3. 입차 차량 조회
4. 이전 메뉴

## ◎ 입차 차량 조회 ◎

조회할 차량 번호 입력 > 123가1234

검색된 차량이 없습니다.

## ◎ 입출차 관리 메뉴 선택 ◎

1. 차량 입차  
> 4
2. 출차 정산
3. 입차 차량 조회
4. 이전 메뉴

## ◎ 메뉴 선택 ◎

1. 입출차 관리  
> 3
2. 회원 관리
3. 주차장 관리
4. 종료

## ◎ 주차장 관리 메뉴 선택 ◎

1. 현재 주차된 차량 현황  
> 1
2. 입출차 차량 통계
3. 결제된 금액 통계
4. 이전 메뉴

## ◎ 현재 주차된 차량 현황 ◎

주차된 차량 목록

차량 번호: 123다1234	차량 종류: 일반	입차 시간: 2023-12-22 15:01:51
-----------------	-----------	----------------------------

차량 번호: 123라1234	차량 종류: 경차	입차 시간: 2023-12-22 15:01:57
-----------------	-----------	----------------------------

차량 번호: 123마1234	차량 종류: 일반	입차 시간: 2023-12-22 15:02:04
-----------------	-----------	----------------------------

주차된 차량 수 : 3

남은 주차 공간 : 67



# 실행결과

## 회원 관리 메뉴 선택

1. 회원 자량 등록 > 2  
조회할 정보가 없습니다.

## 회원 관리 메뉴 선택

1. 회원 자량 등록 > 3  
조회할 정보가 없습니다.

## 회원 관리 메뉴 선택

1. 회원 자량 등록 > 1  
2. 회원 정보 조회  
3. 회원 정보 수정 및 삭제  
4. 이전 메뉴

## 회원 자량 등록

회원 정보를 입력하세요.  
회원 ID > 1011  
회원 이름 > 홍길동  
자량 번호 > 123다1234  
회원 등록 완료

## 회원 관리 메뉴 선택

1. 회원 자량 등록 > 1  
2. 회원 정보 조회  
3. 회원 정보 수정 및 삭제  
4. 이전 메뉴

## 회원 자량 등록

회원 정보를 입력하세요.  
회원 ID > 1010  
회원 이름 > 김길동  
자량 번호 > 123라1234  
회원 등록 완료

## 회원 관리 메뉴 선택

1. 회원 자량 등록 > 1  
2. 회원 정보 조회  
3. 회원 정보 수정 및 삭제  
4. 이전 메뉴

## 회원 자량 등록

회원 정보를 입력하세요.  
회원 ID > 1009  
회원 이름 > 최길동  
자량 번호 > 123마1234  
회원 등록 완료

## 회원 관리 메뉴 선택

1. 회원 자량 등록 > 2  
2. 회원 정보 조회  
3. 회원 정보 수정 및 삭제  
4. 이전 메뉴

## 회원 정보 조회

1. 전체 회원 조회 > 2  
2. 특정 회원 조회

## 특정 회원 조회

조회할 회원 정보 입력(ID, 이름, 자량 번호 中 1) > 1010  
아이디 자량번호 이름  
1010 123라1234 김길동

## 회원 관리 메뉴 선택

1. 회원 자량 등록 > 2  
2. 회원 정보 조회  
3. 회원 정보 수정 및 삭제  
4. 이전 메뉴

## 회원 정보 조회

1. 전체 회원 조회 > 2  
2. 특정 회원 조회

## 특정 회원 조회

조회할 회원 정보 입력(ID, 이름, 자량 번호 中 1) > 최길동  
아이디 자량번호 이름  
1009 123마1234 최길동

## 회원 관리 메뉴 선택

1. 회원 자량 등록 > 2  
2. 회원 정보 조회  
3. 회원 정보 수정 및 삭제  
4. 이전 메뉴

## 회원 정보 조회

1. 전체 회원 조회 > 2  
2. 특정 회원 조회

## 특정 회원 조회

조회할 회원 정보 입력(ID, 이름, 자량 번호 中 1) > 123다1234  
아이디 자량번호 이름  
1011 123다1234 홍길동

# 실행결과

출자 차량 번호 입력 > 123라1234  
420원을 계산해 주세요.  
> 420

-----2023-12-22 15:07:35

안녕히 가세요. 감사합니다.  
123라1234 차량이 출고 되었습니다.  
주차 자리 69개 남았습니다.

⊙ 입출차 관리 메뉴 선택 ⊙

1. 차량 입차                      2. 출차 정산                      3. 입차 차량 조회                      4. 이전 메뉴  
> 4

⊙ 메뉴 선택 ⊙

1. 입출차 관리                      2. 회원 관리                      3. 주차장 관리                      4. 종료  
> 3

⊙ 주차장 관리 메뉴 선택 ⊙

1. 현재 주차된 차량 현황                      2. 입출차 차량 통계                      3. 결제된 금액 통계                      4. 이전 메뉴  
> 1

⊙ 현재 주차된 차량 현황 ⊙

주차된 차량 목록  
차량 번호: 123마1234 | 차량 종류: 일반                      | 입차 시간: 2023-12-22 15:02:04  
주차된 차량 수 : 1  
남은 주차 공간 : 69

⊙ 주차장 관리 메뉴 선택 ⊙

1. 현재 주차된 차량 현황                      2. 입출차 차량 통계                      3. 결제된 금액 통계                      4. 이전 메뉴  
> 2

⊙ 입출차 차량 통계 ⊙

차량번호	차량타입	결제금액	입차시간	출차시간
123마1234	일반	0원	2023-12-22 15:02:04	출차 전
123가1234	일반	1200원	2023-12-22 14:01:41	2023-12-22 15:02:57
123나1234	경차	600원	2023-12-22 14:01:45	2023-12-22 15:03:08
123다1234	일반	840원	2023-12-22 14:01:51	2023-12-22 15:07:10
123라1234	경차	420원	2023-12-22 14:01:57	2023-12-22 15:07:35
총 입차 차량: 1대				
총 출차 차량: 4대				
전체 입출차 차량: 5대				

⊙ 주차장 관리 메뉴 선택 ⊙

1. 현재 주차된 차량 현황                      2. 입출차 차량 통계                      3. 결제된 금액 통계                      4. 이전 메뉴  
> 3

⊙ 결제된 금액 통계 ⊙

차량번호	결제금액	출차시간
123가1234	1200원	2023-12-22 14:01:41
123나1234	600원	2023-12-22 14:01:45
123다1234	840원	2023-12-22 14:01:51
123라1234	420원	2023-12-22 14:01:57
총 차량 수: 4		
합계: 3060원		

⊙ 주차장 관리 메뉴 선택 ⊙

1. 현재 주차된 차량 현황                      2. 입출차 차량 통계                      3. 결제된 금액 통계                      4. 이전 메뉴  
> 4

⊙ 메뉴 선택 ⊙

1. 입출차 관리                      2. 회원 관리                      3. 주차장 관리                      4. 종료  
> 4

⊙ 시스템 종료 ⊙

# 실행결과

## ◎ 메뉴 선택 ◎

1. 입출차 관리                      2. 회원 관리                      3. 주차장 관리                      4. 종료

> 김

잘못된 입력입니다.

1. 입출차 관리                      2. 회원 관리                      3. 주차장 관리                      4. 종료

> 5

범위 내 숫자 값을 입력해 주세요.

1. 입출차 관리                      2. 회원 관리                      3. 주차장 관리                      4. 종료

> 0

범위 내 숫자 값을 입력해 주세요.

1. 입출차 관리                      2. 회원 관리                      3. 주차장 관리                      4. 종료

> -10

범위 내 숫자 값을 입력해 주세요.

1. 입출차 관리                      2. 회원 관리                      3. 주차장 관리                      4. 종료

> 1

## ◎ 입출차 관리 메뉴 선택 ◎

1. 차량 입차                      2. 출차 정산                      3. 입차 차량 조회                      4. 이전 메뉴

> 2



# 자체평가



# 자체평가

## 신선호

- 다른 팀원들에 비해 배운 내용에 대한 이해도가 부족하여 프로젝트에 기여를 하지 못한 부분이 아쉽습니다.
- 팀원들의 도움으로 맡은 역할에 대해 완수할 수 있었고, 코드를 보고 이해하는 것만으로도 배운 내용에 대한 정리가 되어 많은 공부가 되었습니다.

## 안위재

- 프로그램 실행에 필요한 클래스와 메소드의 작성 후 메인 코드에 적용을 시킬 때 메소드나 객체에 대한 이해도가 부족해서 원활하게 구현하는 것에 어려움이 있어 아쉬웠습니다.
- 같이 구현하시는 분들의 설명이 공부에 많은 도움이 됐고 실행 시 오류를 찾고 같이 고민하며 수정하여 오류를 없애는 과정이 재미있었습니다.

## 이상목

프로그램의 구조 설계와 클래스 메소드 개발 및 오류 수정을 위주로 프로젝트에 참여 했고, 초기 설계 과정에서 부족한 부분과 실수들 때문에 열심히 하려는 팀원들에게 코드개발 분배가 적절히 이루어지지 않은 점이 아쉬웠습니다. 하지만 모두들 각자 맡은 분야에 열심히 하셔서 프로젝트를 잘 마무리 할 수 있었던 것 같아 유익한 시간이었습니다. 프로젝트를 진행 하면서 겪은 시행착오와 팀워크 덕분에 개인적으로도 많은 공부가 되었습니다.

## 이종운

- 다양한 클래스를 이용하여 구현한 메소드를 메인 클래스에 합치는 역할을 수행했는데, 다른 클래스에서 작성된 여러 메소드들의 호환을 적절히 적용하는 방법을 많이 연습하게 되었습니다. 또한 예외처리를 통해 메인 프로그램이 멈추지 않고 계속 작동하도록 하는 방법에 대해서도 많이 연습할 수 있었습니다.
- 메인 클래스에 직접 코드를 넣는 것이 아니라 다른 클래스에서 메소드를 구현하여 메인에서 실행하는 방식으로 메인 클래스를 가볍게 하는 기초 설계에 대한 연습을 더 해야할 것 같습니다.

## 최재희

Java언어 이해도가 뛰어난 팀장과 능력 있는 팀원간의 원활한 의사소통으로 큰 어려움 없이 프로젝트를 이행할 수 있었습니다. 아쉬운 점이 있다면 코딩에 적극적으로 참여하지 못한 것. 이번 프로젝트의 경험을 통해 앞으로 발전 가능성을 확인하고 나아가 다음 프로젝트의 코딩 참여를 위해 학습의지를 다잡을 계기가 되었습니다.



**Q&A**

**YOUR  
IDEAS  
MATTER**

**Write them down :)**

감사합니다