



ParkingLot Program System



**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

프로젝트 일정

12.19

프로젝트 주제 설정
역할 분담

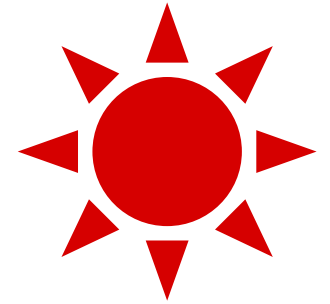
12.20

프로그램 코드 분담

12.21

12.22

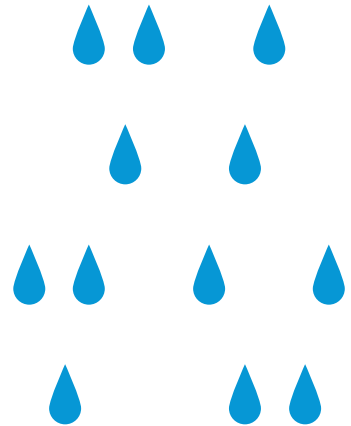
12.26



프로그램 구현

프로그램 오류 수정

PPT 제작

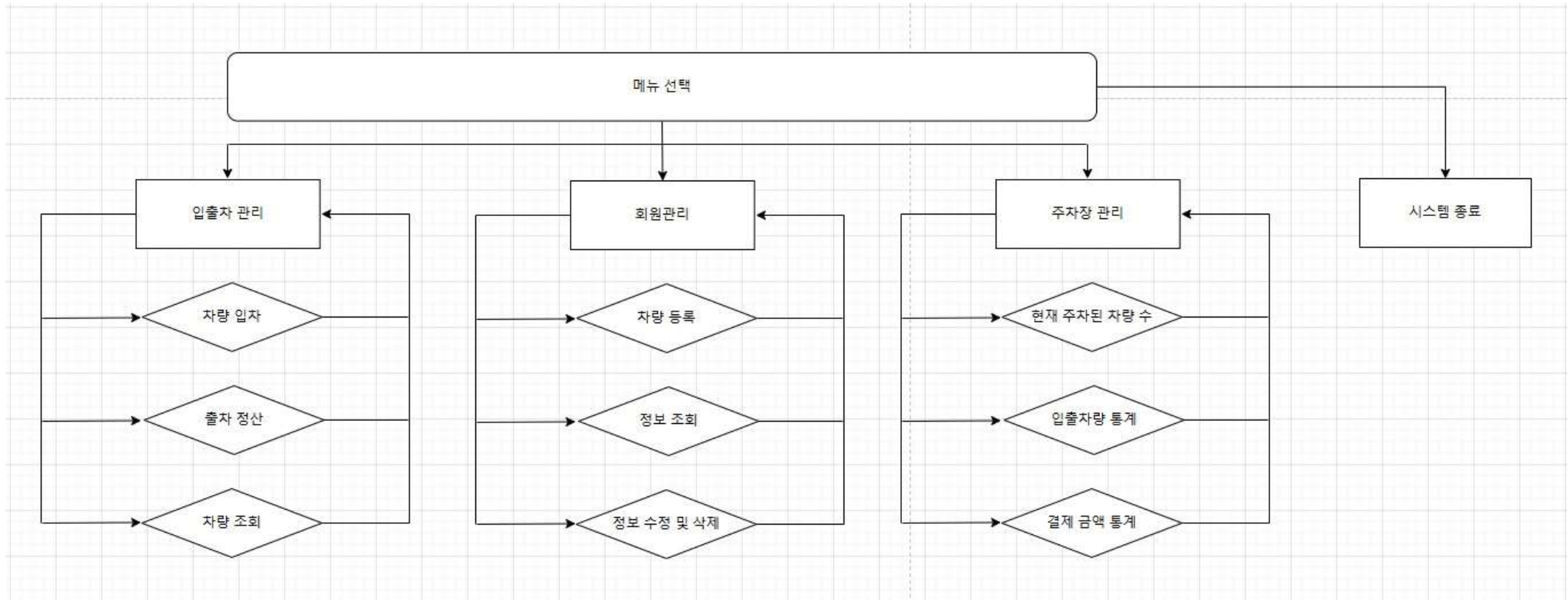


1 알고리즘

2 주요코드설명

3 실행결과

알고리즘



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", "종료");
            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_O it = Utils.findCarInst(parkingLot.currentCars, outCarNum);
                if (it == null) {
                    break;
                }

                NewPayment_O pay = new NewPayment_O(it, parkingLot);
                if (memberList.mlist.findMember(outCarNum) != null) {
                    // String test = memberList.mlist.findMember(outCarNum).carNum;
                }
                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

```
interface InnerParkingLot
{
    void carIn();
    void carOut(Car car);
    int getCurrentSpace();
    String errorType(int errorCode);
    void setPricePerTenMin(int price);
    // void showAllOutCarPayments();
}

public class ParkingLot implements InnerParkingLot{
    CarArray currentCars = new CarArray(); // 데이터베이스 역할
    CarArray outCars = new CarArray(); // 데이터베이스 역할
    int spaceTotal = 70; // 전체 주차가능 수
    private int space = this.currentCars.size(); // 현재 주차중인 수
    int pricePerTenMin = 200; //10분당 요금

    public class NewPayment {
        Car car;
        ParkingLot parkingLot;
        double discount = 0;
        int basicCharge = 1000;

        public NewPayment(Car car, ParkingLot parkingLot) {}

        public void setDiscount(double d) {}

        public void setDiscount(MemberArray ma) {}

        int getAmount() {}

        int getChanges(int amount, int receive) {}

        Car execPay(int receive) {}
    }
}
```

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

```
public class Member{ //회원 정보 모델
    String memId;
    String memName;
    String carNum;
    Instant issueDate = Instant.now();

    public Member(String memId, String memName, String carNum) {}

    public String getId() {}

    public String getCarNum() {}

    public String getName() {}

    public void setId(String memId) {}

    public void setName(String memName) {}

    public void setNum(String carNum) {}

    public void memberPrint() {}
}
```

```
public class MemberList {
    MemberArray mlist = new MemberArray();
    String memId;
    String memName;
    String carNum;

    public void addMember() {}
}

interface InnerMemberArray {
    boolean isMember(String carNum);
    Member findMember(String any);
    void showAll();
    void showMember(String numOrName);
    MemberArray sortMemberById();
    MemberArray sortMemberIssueDate();
}

public class MemberArray extends ArrayList<Member> implements InnerMemberArray {
    public boolean isMember(String carNum) {
        if (findMember(carNum) == null) {
            return false;
        }
        return true;
    }
}
```

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 (입출차 관리)

◎ 입출차 관리 메뉴 선택 ◎

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

◎ 차량 입차 ◎

입차 차량 번호 입력(0 입력시 랜덤 생성) > 123가1234

1. 일반 2. 경차
> 1

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

입차 차량 번호 입력(0 입력시 랜덤 생성) > 123나1234

1. 일반 2. 경차
> 2

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

입차 차량 번호 입력(0 입력시 랜덤 생성) > 123다1234

1. 일반 2. 경차
> 1

123다1234 차량이 입고 되었습니다.
주차 자리 67개 남았습니다.

입차 차량 번호 입력(0 입력시 랜덤 생성) > 123라1234

1. 일반 2. 경차
> 2

123라1234 차량이 입고 되었습니다.
주차 자리 66개 남았습니다.

◎ 입차 차량 조회 ◎

조회할 차량 번호 입력 > 123가1234

차량 번호: 123가1234 | 차량 종류: 일반 | 입차 시간: 2024-05-22 02:54:16

조회할 차량 번호 입력 > 123나1234

차량 번호: 123나1234 | 차량 종류: 경차 | 입차 시간: 2024-05-22 02:54:35

◎ 출차 정산 ◎

출차 차량 번호 입력 > 123가1234

1000원을 계산해 주세요.

> 800

다시 정산을 해주시기 바랍니다. 감사합니다.

200원을 계산해 주세요.

> 500

-----2024-05-22 02:56:26

거스름돈은 300원 입니다. 감사합니다.

123가1234 차량이 출고 되었습니다.

주차 자리 67개 남았습니다.

실행결과 (회원 관리)

◎ 회원 관리 메뉴 선택 ◎

1. 회원 자량 등록 2. 회원 정보 조회 3. 회원 정보 수정 및 삭제 4. 이전 메뉴 5. 테스트

> 2

조회할 정보가 없습니다.

◎ 회원 관리 메뉴 선택 ◎

1. 회원 자량 등록 2. 회원 정보 조회 3. 회원 정보 수정 및 삭제 4. 이전 메뉴 5. 테스트

> 1

◎ 회원 자량 등록 ◎

회원 정보를 입력하세요.

회원 ID > 1011

회원 이름 > 홍길동

자량 번호 > 123바1234

회원 등록 완료

◎ 회원 자량 등록 ◎

회원 정보를 입력하세요.

회원 ID > 1009

회원 이름 > 최길동

자량 번호 > 123바1234

회원 등록 완료

◎ 회원 정보 조회 ◎

1. 전체 회원 조회 2. 특정 회원 조회

> 2

◎ 특정 회원 조회 ◎

조회할 회원 정보 입력(ID, 이름, 자량 번호 中 1) > 1010

검색 결과 없음.

◎ 특정 회원 조회 ◎

조회할 회원 정보 입력(ID, 이름, 자량 번호 中 1) > 홍길동

아이디 자량번호 이름

1011 123바1234 홍길동

◎ 회원 정보 조회 ◎

1. 전체 회원 조회 2. 특정 회원 조회

> 1

◎ 전체 회원 조회 ◎

회원 번호: 1009		회원 이름: 최길동		자량 번호: 123바1234
회원 번호: 1011		회원 이름: 홍길동		자량 번호: 123바1234

실행결과 (주차장 관리)

◎ 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: 123나1234	차량 종류: 경차	입차 시간: 2024-05-22 02:54:35	
차량 번호: 123다1234	차량 종류: 일반	입차 시간: 2024-05-22 02:54:41	
차량 번호: 123라1234	차량 종류: 경차	입차 시간: 2024-05-22 02:54:48	
주차된 차량 수 : 3			
남은 주차 공간 : 67			

◎ 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
123나1234	경차	0원	2024-05-22 02:54:35	출차 전
123다1234	일반	0원	2024-05-22 02:54:41	출차 전
123라1234	경차	0원	2024-05-22 02:54:48	출차 전
123가1234	일반	1000원	2024-05-22 02:54:16	2024-05-22 02:56:26
총 입차 차량: 3대				
총 출차 차량: 1대				
전체 입출차 차량: 4대				

◎ 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	1000원	2024-05-22 02:54:16
총 차량 수: 1		
합계: 1000원		

Q&A

**YOUR
IDEAS
MATTER**

Write them down :)

감사합니다