

# [기말과제 3차 보고서 - 주차장 관리 시스템]

학과: 디지털미디어학과

학번: 2021111259

이름: 변혜빈

## 1. 프로그램 주제

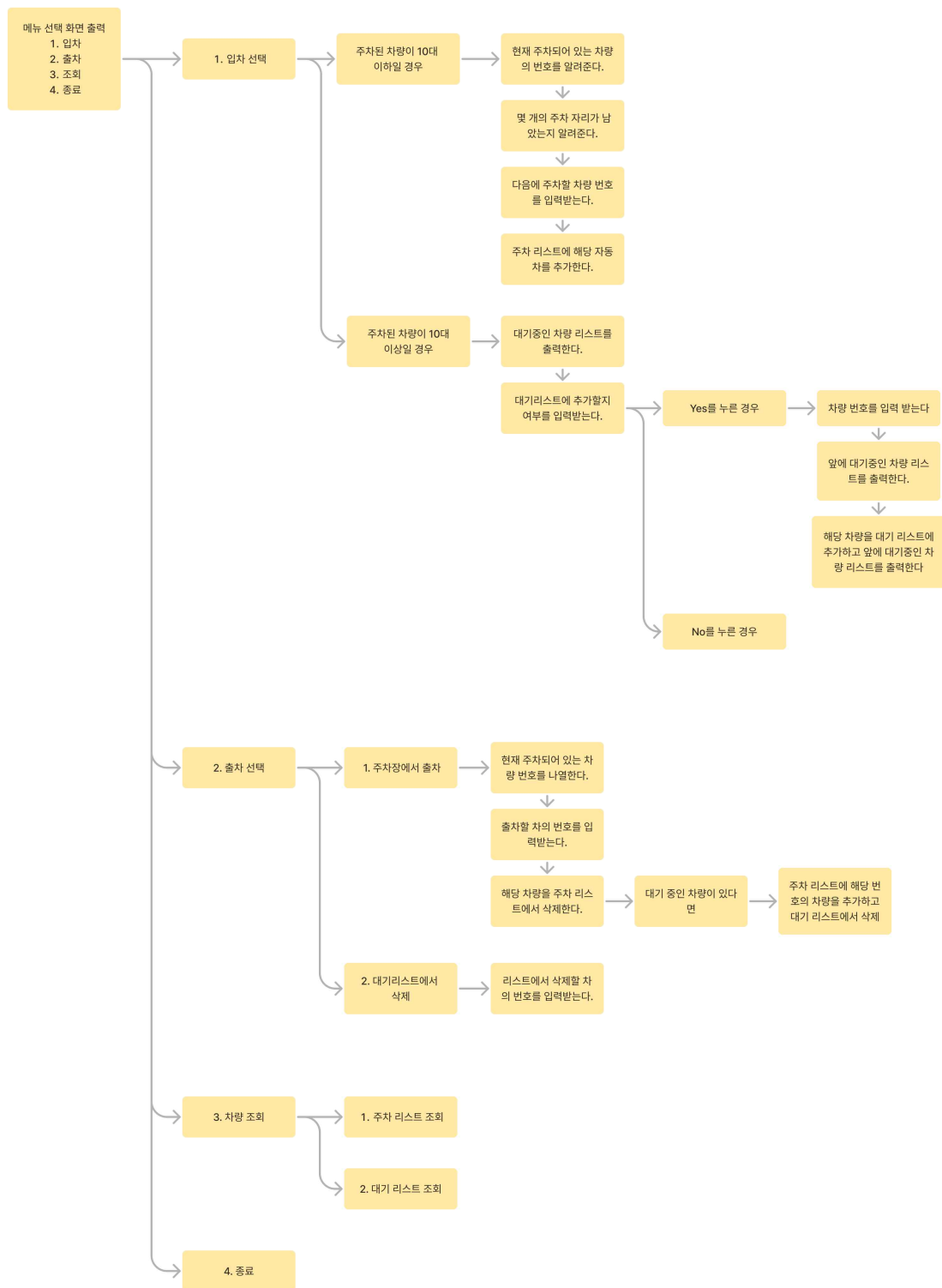
(1) 프로젝트 주제

- 이 프로젝트의 주제는 ‘주차장 관리 시스템’ 제작이다.
- 이 프로그램을 사용하는 대상자는 주차장을 관리하는 관리자이다.
- 주차장 관리자가 남은 주차공간 파악, 요금 정산, 주차 대기 차량 리스트 관리, 정기결제 리스트 관리 등의 주차장 업무를 효율적으로 수행하도록 돕는 것을 목적으로 한다.

## 2. 기존 프로그램 분석

## (1) 프로그램 분석

- 이 프로그램은 10개의 주차 공간이 마련된 임의의 주차장에서 차량 입·출차를 관리하는 프로그램이다.
- 출처 : [CarParkingSystem - Github](https://github.com/mohamedafzal03/java-car-parking-system)  
(<https://github.com/mohamedafzal03/java-car-parking-system>)
- 이 프로그램의 main 매소드는 car11 클래스 내부에 있다.
- 차량을 모델링한 Vehicle 클래스는 차량 번호를 나타내는 변수 no와 주차장 내부의 차량이 입출차하며 기존의 차량이 이동한 횟수를 나타내는 변수 mvs로 이루어져 있다. 차량 번호를 매개변수로 받는 명시적 생성자 1개와, mvs를 추가시키는 incmoves()를 매소드로 가지고 있다.
- 주차된 차량과 주차 대기 차량은 모두 vehicle 객체를 사용한 연결 리스트 구조에서 관리되고 있다.
- 이 프로그램에서는 java.util 패키지를 사용하고 있다. java.util 클래스는 Collection Framework, Set, List, Map 등 유용한 클래스들을 많이 포함하고 있다. 특히 이 프로그램에서는 util 패키지의 Linked List 클래스와 Scanner 클래스를 주로 사용하였다. Linked List 클래스는 이름 그대로 연결 리스트를 구현할 수 있도록 하는 클래스이다. Scanner 클래스는 사용자로부터 입력을 받기 위해 사용하는 클래스이다.
- 이 프로그램의 실행 과정은 다음과 같다.



→ 프로그램 실행 과정

- 프로그램이 실행되면 1. 입차 2. 출차 3. 조회 4. 종료의 메뉴를 출력한다. Scanner 클래스를 사용해 사용자의 입력을 받는다. 1~3의 메뉴를 선택 후 동작이 완료되면 다시 메인 메뉴를 출력하도록 while 반복문을 사용하였다. 단, 4번 메뉴를 선택하면 종료된다.
- Scanner 객체의 nextInt() 메소드를 이용해 사용자로부터 정수를 입력 받는다.
- 1번을 눌러 입차 메뉴를 선택했을 때, 주차된 차량이 10대 이하일 경우 for문을 이용해 현재 주차 리스트에 등록된 차량 번호를 화면에 출력한다. 리스트의 size() 메소드로 리스트 크기를 사용할 수 있다.
- 주차장의 주차 공간이 10개이므로, 10-numOfCars(현재 주차된 차량 수)로 몇 개의 주차 공간이 남았는지 출력한다.
- 주차할 차량 번호를 입력받아 주차 리스트에 추가한다. 입력받는 차량 번호는 주차 리스트의 현재 사이즈보다 1만큼 더 큰 수이며, 리스트의 인덱스와 같은 수이다. 차량 번호는 지정된 값만 입력받을 수 있다. 사실상 차량번호보다 주차 구역 번호에 가깝다.
- 주차된 차량이 10대 이상일 경우, 대기 리스트에 추가할지 여부를 다시 물어본다. 대기 리스트에 추가할 경우, 차량 번호를 입력 받아 대기 리스트에 추가한다. 이 때 차량 번호는 대기 리스트의 현재 사이즈보다 11만큼 더 큰 수이다. 차량번호는 정해진 값만 사용할 수 있다.
- 2번을 눌러 출차 메뉴를 선택한 경우, 주차 리스트에서 삭제할지 대기 리스트에서 삭제할지 다시 선택한다. 선택 후 리스트에 있는 차량 목록을 화면에 출력한 후, 어떤 차량을 삭제할지 정수로 입력받는다.
- 선택한 차량보다 이전에 입차한 차량의 mvs를 2씩 더하고, 이후에 입차한 차량의 mvs를 1씩 더한다. 아마 주차장의 구조 때문에 기존에 주차되어 있는 차량을 출차할 경우, 다른 주차된 차량들을 이동시켜야 하고 그 이동 횟수를 기록하는 내용으로 보인다.
- 입력받은 번호의 차량을 remove(인덱스) 메소드로 리스트에서 삭제한다.
- 3번을 눌러 조회 메뉴를 선택한 경우, 주차된 차량의 번호를 조회하거나 대기 중인 차량 번호를 조회 할 수 있다. for문을 이용해 리스트의 모든 요소를 화면에 출력한다.

## (2) 기획서

### ① GUI 구현

- 기존 프로그램에서는 GUI 방식으로 구성되지 않았고, 실행 결과를 콘솔창을 통해 출력하고 사용자로부터 입력받는 형식으로 구성되었다. 기존 프로그램의 기능을 GUI로 구현할 수 있도록 코드를 추가한다.
- 현재 남은 주차 공간은 그림으로 표현하여 한 눈에 볼 수 있도록 한다.

### ② Vehicle 클래스 재구성

- 현재의 Vehicle 클래스에 고유 차량 ID, 소유주, 소유주 전화번호 등의 변수와 필요한 기능들을 추가한다.
- Vehicle 클래스를 상속받아 경차, 중형차, 대형차, 장애인차량 등 차종을 구분하는 새로운 자식 클래스들을 작성한다.
- 차종마다 주차 요금이 다르다.

### ③ Parking Lot 클래스 작성

- 현재 프로그램에서는 주차장의 주차칸을 연결 리스트의 각 노드들로 표현 하고 있다. 때문에 각 차들의 고유 ID를 사용하지 않고 주차칸 번호를 사용하여 각 차량 객체를 구분하고 있다. 주차칸과 실제 자동차 객체를 완전히 분리하기 위하여, 주차칸 1칸을 모델링한 Parking Lot 클래스를 추가한다.
- Prtking Lot 클래스는 현재 해당 주차 구역을 사용중인지 구분하는 boolean 타입 변수 isFull을 가진다.

### ④ 메뉴 재구성

- 실제 유료 주차장에서 사용할 수 있도록 ‘정산’메뉴를 추가한다. 입차한 순간부터 출차한 순간까지 주차 구역을 사용한 시간만큼 요금을 부과한다. 사용자는 ‘정산’메뉴를 통해 요금을 정산할 수 있고, 정산하지 않고 출차할 경우 요금을 납부하라는 팝업을 띄운다.

### ⑤ 주차 정기권 (멤버십) 기능 추가

- 주차장을 이용하는 사용자를 일회성 사용자와 정기 등록 사용자로 나누어 관리한다.
- 정기 등록을 한 사용자는 전화번호나 차량 번호를 입력하는 것만으로 주차 차량을 조회할 수 있다.



### 3. 전체 소스코드

## (1) Vehicle.java

```
public class Vehicle {
    public String carIDString, nameString, phoneNumString, carModel; // 차 번호,
    소유주 이름, 전화번호, 기종
    public boolean isMember; // 정기결제 여부

    // 생성자
    public Vehicle () {
        this.carIDString = "차 번호";
        this.nameString = "소유주 이름";
        this.phoneNumString = "전화번호";
        this.carModel = "승용차";
        this.isMember = false;
    }
    public Vehicle(String carIDString) {
        this.carIDString = carIDString;
        this.nameString = "이름";
        this.phoneNumString = "전화번호";
        this.carModel = "승용차";
        this.isMember = false;
    }
    public Vehicle(String carIDString, String nameString, String phoneNumString,
String carModelString, boolean isMember) {
        this.carIDString = carIDString;
        this.nameString = nameString;
        this.phoneNumString = phoneNumString;
        this.carModel = carModelString;
        this.isMember = isMember;
    }
    public Vehicle(Vehicle vehicle) {
        this.carIDString = vehicle.carIDString;
        this.nameString = vehicle.nameString;
        this.phoneNumString = vehicle.phoneNumString;
        this.carModel = vehicle.carModel;
        this.isMember = vehicle.isMember;
    }
}
```

## (2) ParkingLot.java

```
public class ParkingLot {
    // 필드
    boolean isFull, isDisabled; // 주차되었는지 여부, 장애인 주차구역인지 여부
    int enterTime, exitTime; // 입차시각, 출차시각
    Vehicle aCar; // 현재 주차된 차량

    public ParkingLot() {
        aCar = new Vehicle();
        isFull = false;
        isDisabled = false;
        enterTime = 0;
        exitTime = 0;
    }
}
```

```

public ParkingLot(boolean isDisabled) {
    aCar = new Vehicle();
    isFull = false;
    this.isDisabled = isDisabled;
    enterTime = 0;
    exitTime = 0;
}

public ParkingLot(Vehicle aCar, boolean isFull, boolean isDisabled) {
    this.aCar = aCar;
    this.isFull = isFull;
    this.isDisabled = isDisabled;
    enterTime = 0;
    exitTime = 0;
}

public int calculate() {
    int sum = 0;
    int hour = 0;
    int min = 0;

    // 계산하는 코드
    sum = exitTime - enterTime;
    hour = (sum / 100) * 60; // 몫: 시간 -> 분으로 변환
    min = sum % 100; // 나머지 : 분
    sum = (hour + min * 100); // 10분에 1000원

    // 기종에 따라서 요금 가감.
    if (aCar.carModel.equals("승용차")) {
        sum = sum * 1;
    }
    else if (aCar.carModel.equals("승합차")) {
        sum = (int)(sum * 1.5f);
    }
    else if (aCar.carModel.equals("경차")) {
        sum = (int)(sum * 0.5f);
    }
    else if (aCar.carModel.equals("장애인 차량")) {
        sum = (int)(sum * 0.5f);
    }

    return sum;
}
}

```

### (3) SystemFrame.java

```

import java.awt.Button;
import java.awt.Color;
import java.awt.Font;
import java.awt.LayoutManager;
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;
import java.util.LinkedList;
import javax.swing.JButton;

```

```

import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;

// GUI가 실행될 프레임
public class SystemFrame extends JFrame {

    // 컴포넌트
    Vehicle currentVehicle = new Vehicle();
    ParkingLot[] parkingLot; // 주차장
    LinkedList<Vehicle> waitingList = new LinkedList<Vehicle>(); // 대기 리스트
    LinkedList<Vehicle> memberList = new LinkedList<Vehicle>(); // 정기결제 멤버
    리스트
    MenuPanel menuPanel;
    EnterPanel enterPanel;
    EnterPanel2 enterPanel2;
    EnterResultPanel enterResultPanel;
    ExitPanel exitPanel;
    ExitResultPanel exitResultPanel;
    FailPanel failPanel;
    FailPanelToEnter failPanelToEnter;
    AddWaitingListPanel addWaitingListPanel;
    DisplayPanel displayPanel;
    ParkingLotDisplayPanel parkingLotDisplayPanel;
    WaitingListDisplayPanel waitingListDisplayPanel;
    MemberPanel memberPanel;
    MemberDisplayPanel memberDisplayPanel;
    AddMemberPanel addMemberPanel;
    AddWaitingResultPanel addWaitingResultPanel;

    // 생성자
    public SystemFrame() {
        setTitle("자바시 공영주차장 관리 프로그램");
        setSize(1000, 600);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); // 프레임 닫으면 메모리에서
        제거
        setLocationRelativeTo(null); // 화면 가운데 배치
        setResizable(false); // 프레임 크기 고정

        // 주차장
        parkingLot = new ParkingLot[12];
        for(int i = 0; i < 12; i++) {
            if (i < 10) {
                parkingLot[i] = new ParkingLot();
            }
            else {
                parkingLot[i] = new ParkingLot(true);
            }
        }

        // 컴포넌트 생성
        menuPanel = new MenuPanel(this);
        enterPanel = new EnterPanel(this);
        enterPanel2 = new EnterPanel2(this);
        enterResultPanel = new EnterResultPanel(this);
        exitPanel = new ExitPanel(this);
        exitResultPanel = new ExitResultPanel(this);
        failPanel = new FailPanel(this);

```

```

        failPanelToEnter = new FailPanelToEnter(this);
        addWaitingListPanel = new AddWaitingListPanel(this);
        displayPanel = new DisplayPanel(this);
        ParkingLotDisplayPanel = new ParkingLotDisplayPanel(this);
        waitingListDisplayPanel = new WaitingListDisplayPanel(this);
        memberPanel = new MemberPanel(this);
        memberDisplayPanel = new MemberDisplayPanel(this);
        addMemberPanel = new AddMemberPanel(this);
        addWaitingResultPanel = new AddWaitingResultPanel(this);

        // 프레임에 메인 메뉴 패널 추가
        this.add(menuPanel);
        setVisible(true);
    }

    // 패널 변경 함수
    public void change(MyPanel nextPanel) {
        // 현재 패널을 지우고 다음 프레임에 패널을 추가
        getContentPane().removeAll();
        getContentPane().add(nextPanel);
        revalidate();
        repaint();
    }

    // Main Method
    public static void main (String[] args) {
        // 패널 생성
        SystemFrame f = new SystemFrame();
    }
}

```

#### (4) MyPanel.java

```

import java.awt.Button;
import java.awt.Color;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.FocusEvent;
import java.awt.event.FocusListener;
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;
import javax.swing.ButtonGroup;
import javax.swing.Icon;
import javax.swing.JButton;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JRadioButton;
import javax.swing.JTextField;

/** My Panel : 사용할 패널의 기본 */
class MyPanel extends JPanel{
    SystemFrame systemFrame;
    public static Color background = new Color (242, 230, 223, 255);
}

```

```

Font a = new Font("맑은 고딕", Font.PLAIN, 20);
Font b = new Font("맑은 고딕", Font.BOLD, 50);
Font c = new Font("맑은 고딕", Font.BOLD, 30);
Font d = new Font("맑은 고딕", Font.BOLD, 23);

public LocalTime time;
public DateFormatter formatter;

// 생성자
public MyPanel() {
    // 시계
    time = LocalTime.now();
    formatter = DateFormatter.ofPattern("HH:mm:ss");

    // 배치관리자 border vs null
    this.setBackground(background);
    this.setVisible(true);
}

}

/** Menu Panel */
class MenuPanel extends MyPanel {

    // 컴포넌트
    SystemFrame systemFrame; // 현재 패널을 포함한 프레임
    JLabel timeJLabel; // 현재 시각 시계
    JButton enterButton, exitButton, displayButton, memberButton; // 입차, 출차,
    조회, 정기결제 멤버십 관리

    // 생성자
    public MenuPanel(SystemFrame systemFrame) {

        this.systemFrame = systemFrame;

        // 배치관리자 : null (절대위치로 배치)
        this.setLayout(null);

        // 컴포넌트 생성
        timeJLabel = new JLabel(time.format(formatter));
        timeJLabel.setFont(b);
        timeJLabel.setHorizontalAlignment(JLabel.CENTER);
        // timer(); 추가
        enterButton = new JButton("주차하기");
        exitButton = new JButton("출차하기");
        memberButton = new JButton("정기결제 관리");
        displayButton = new JButton("조회하기");
        enterButton.setFont(a);
        exitButton.setFont(a);
        memberButton.setFont(a);
        displayButton.setFont(a);

        // 버튼 이벤트 처리 코드
        enterButton.addActionListener(e -> {
            systemFrame.enterPanel = new EnterPanel(systemFrame);
            systemFrame.change(systemFrame.enterPanel);
        });
        exitButton.addActionListener(e -> {

```

```

        systemFrame.exitPanel = new ExitPanel(systemFrame);
        systemFrame.change(systemFrame.exitPanel);
    });
    displayButton.addActionListener(e -> {
        systemFrame.change(systemFrame.displayPanel);
    });
    memberButton.addActionListener(e -> {
        systemFrame.change(systemFrame.memberPanel);
    });

    // 컴포넌트 추가
    this.add(timeJLabel);
    this.add(enterButton);
    this.add(exitButton);
    this.add(displayButton);
    this.add(memberButton);

    // 위치 설정
    timeJLabel.setBounds(0, 90, 1000, 50);
    enterButton.setBounds(350, 200, 300, 50);
    exitButton.setBounds(350, 280, 300, 50);
    displayButton.setBounds(350, 360, 300, 50);
    memberButton.setBounds(350, 440, 300, 50);
}

}

/** Enter Panel */
class EnterPanel extends MyPanel {

    // 컴포넌트
    SystemFrame systemFrame; // 현재 패널을 포함한 프레임
    JButton okButton; // 확인 버튼
    JTextField carIDField, phoneNumberField, nameField; // 자동차 번호, 전화번호,
이름
    JRadioButton sedan, van, light, disabled; // 차 기종 : 승용차, 승합차, 경차,
장애인
    ButtonGroup group; // 차 기종 라디오버튼 그룹
    JLabel carIDJLabel, phoneNumJLabel, nameJLabel, carModelLabel;
    boolean member;

    // 생성자
    public EnterPanel(SystemFrame systemFrame) {

        this.systemFrame = systemFrame;

        // 배치관리자 : null (절대위치로 배치)
        this.setLayout(null);

        // 컴포넌트 생성
        okButton = new JButton("입력하기");
        // JLabel
        carIDJLabel = new JLabel("차량 번호");
        phoneNumJLabel = new JLabel("전화번호");
        nameJLabel = new JLabel("이름");
        carModelLabel = new JLabel("기종");
        carIDJLabel.setFont(d);
        phoneNumJLabel.setFont(d);

```

```

nameJLabel.setFont(d);
carModelLabel.setFont(d);
// JField
carIDField = new JTextField();
phoneNumberField = new JTextField();
nameField = new JTextField();
carIDField.setFont(a);
phoneNumberField.setFont(a);
nameField.setFont(a);
// JRadioButton
sedan = new JRadioButton("승용차");
van = new JRadioButton("승합차");
light = new JRadioButton("경차");
disabled = new JRadioButton("장애인 차량");
sedan.setFont(a);
van.setFont(a);
light.setFont(a);
disabled.setFont(a);
sedan.setBackground(null);
van.setBackground(null);
light.setBackground(null);
disabled.setBackground(null);
// ButtonGroup (Radio Button)
group = new ButtonGroup();
group.add(sedan);
group.add(van);
group.add(light);
group.add(disabled);

// 이벤트 처리 코드
carIDField.addActionListener(e -> {
    // 정기결제 멤버라면 나머지 정보가 자동으로 채워짐
    for (int i=0; i< systemFrame.memberList.size(); i++) {
        if
(systemFrame.memberList.get(i).carIDString.equals(carIDField.getText())) {
            nameField.setText(systemFrame.memberList.get(i).nameString);
phoneNumberField.setText(systemFrame.memberList.get(i).phoneNumString);
            if (systemFrame.memberList.get(i).carModel == "승용차") {
                sedan.setSelected(true);
            }
            else if (systemFrame.memberList.get(i).carModel == "승합차") {
                van.setSelected(true);
            }
            else if (systemFrame.memberList.get(i).carModel == "경차") {
                light.setSelected(true);
            }
            else if (systemFrame.memberList.get(i).carModel == "장애인 차량")
{
                disabled.setSelected(true);
            }
            member = true;
        }
    }
});
okButton.addActionListener(e -> {
    // 차량 번호를 입력하지 않은 경우

```



```

        if (carIDField.getText().equals("")) {
            systemFrame.failPanel.failLabel.setText("차량 번호를 입력해주세요.");
            systemFrame.change(systemFrame.failPanel);
        }
        else {
            // current Vehicle에 차량 정보 입력
            String model = "승용차";
            if (sedan.isSelected() == true) {
                model = "승용차";
            }
            else if (van.isSelected() == true) {
                model = "승합차";
            }
            else if (light.isSelected() == true) {
                model = "경차";
            }
            else if (disabled.isSelected() == true) {
                model = "장애인 차량";
            }
            systemFrame.currentVehicle = new Vehicle(carIDField.getText(),
nameField.getText(), phoneNumberField.getText(), model, member);

            for (int i = 0; i < 12; i++) {
                if (systemFrame.parkingLot[i].isFull == false) {
                    if (i < 10) {
                        systemFrame.enterPanel2 = new EnterPanel2(systemFrame);
                        systemFrame.change(systemFrame.enterPanel2); // 다음 패널로
                        break;
                    }
                    else { // 장애인 주차구역
                        if (systemFrame.currentVehicle.carModel != "장애인 차량") {
                            systemFrame.change(systemFrame.addWaitingListPanel); //
                            break;
                        }
                        else {
                            systemFrame.enterPanel2 = new EnterPanel2(systemFrame);
                            systemFrame.change(systemFrame.enterPanel2); // 다음
                            break;
                        }
                    }
                }
            }
            else {
                if (i == 11) {
                    systemFrame.change(systemFrame.addWaitingListPanel); //
                }
            }
        }
    });

    // 컴포넌트 추가
    add(carIDJLabel);
    add(carIDField);

```

대기 패널로

패널로

대기 패널로

```

        add(nameJLabel);
        add(nameField);
        add(phoneNumJLabel);
        add(phoneNumberField);
        add(carModelLabel);
        add(sedan);
        add(van);
        add(light);
        add(disabled);
        add(okButton);

        // 위치 설정
        carIDJLabel.setBounds(50, 100, 200, 50);
        carIDField.setBounds(200, 100, 300, 50);
        nameJLabel.setBounds(50, 200, 200, 50);
        nameField.setBounds(200, 200, 300, 50);
        phoneNumJLabel.setBounds(50, 300, 200, 50);
        phoneNumberField.setBounds(200, 300, 300, 50);
        carModelLabel.setBounds(600, 100, 200, 50);
        sedan.setBounds(600, 150, 200, 50);
        van.setBounds(600, 200, 200, 50);
        light.setBounds(600, 250, 200, 50);
        disabled.setBounds(600, 300, 200, 50);
        okButton.setBounds(800, 450, 100, 50);
    }
}

/** Enter Panel 2 */
class EnterPanel2 extends MyPanel implements FocusListener {

    // 컴포넌트
    SystemFrame systemFrame; // 현재 패널을 포함한 프레임
    JButton okButton; // 확인 버튼
    JButton[] parkNumButtons; // 주차 자리 버튼
    JTextField carIDField, phoneNumberField, nameField, carModelField; // 자동차
    번호, 전화번호, 이름 -> 수정 불가능
    JLabel carIDJLabel, phoneNumJLabel, nameJLabel, carModelLabel;
    static int selectedNum = 0; // 현재 선택된 주차 자리

    // 생성자
    public EnterPanel2(SystemFrame systemFrame) {

        this.systemFrame = systemFrame;

        // 배치관리자 : null (절대위치로 배치)
        this.setLayout(null);

        // 컴포넌트 생성
        // JButton
        okButton = new JButton("주차하기");
        parkNumButtons = new JButton[12]; // 11, 12 : 장애인 주차 구역
        for (int i = 0; i < 12; i++) {
            if (i < 10) {
                parkNumButtons[i] = new JButton((i+1) + " 번");
            }
            else {
                parkNumButtons[i] = new JButton("장애인 " + (i - 9) + " 번");
            }
        }
    }
}

```

```

        add(parkNumButtons[i]); // 컴포넌트 추가
        parkNumButtons[i].setBackground(Color.white);
    }
    // 이미 주차된 자리는 버튼 비활성화
    for (int i = 0; i < 12; i++)
    {
        if (systemFrame.parkingLot[i].isFull == true) {
            parkNumButtons[i].setEnabled(false);
        }
    }

    // JLabel
    carIDJLabel = new JLabel("차량 번호");
    phoneNumJLabel = new JLabel("전화번호");
    nameJLabel = new JLabel("이름");
    carModelLabel = new JLabel("기종");
    carIDJLabel.setFont(new Font("맑은 고딕", Font.PLAIN, 15));
    phoneNumJLabel.setFont(new Font("맑은 고딕", Font.PLAIN, 15));
    nameJLabel.setFont(new Font("맑은 고딕", Font.PLAIN, 15));
    carModelLabel.setFont(new Font("맑은 고딕", Font.PLAIN, 15));
    // JField
    carIDField = new JTextField(systemFrame.currentVehicle.carIDString);
    phoneNumberField = new JTextField(systemFrame.currentVehicle.phoneNumString);
    nameField = new JTextField(systemFrame.currentVehicle.nameString);
    carModelField = new JTextField(systemFrame.currentVehicle.carModel);

    carIDField.setEditable(false);
    phoneNumberField.setEditable(false);
    nameField.setEditable(false);
    carModelField.setEditable(false);

    // 이벤트 처리 코드
    okButton.addActionListener(e -> {
        if ((selectedNum > 9) && (selectedNum < 12)) {
            if (systemFrame.currentVehicle.carModel == "장애인 차량") {
                // parkingLot 배열에 현재 차량을 저장
                systemFrame.parkingLot[EnterPanel2.selectedNum] = new
ParkingLot(systemFrame.currentVehicle, true, true);
                systemFrame.parkingLot[EnterPanel2.selectedNum].enterTime =
(time.getHour() * 100) + (time.getMinute());
                systemFrame.change(systemFrame.enterResultPanel); // 다음
패널로
            }
            else {
                systemFrame.failPanelToEnter.failLabel.setText("장애인 주차구역
입니다.");
                systemFrame.change(systemFrame.failPanelToEnter);
            }
        }
        else {
            // parkingLot 배열에 현재 차량을 저장
            systemFrame.parkingLot[EnterPanel2.selectedNum] = new
ParkingLot(systemFrame.currentVehicle, true, false);
            systemFrame.parkingLot[EnterPanel2.selectedNum].enterTime =
(time.getHour() * 100) + (time.getMinute());
            systemFrame.change(systemFrame.enterResultPanel); // 다음 패널로
        }
    }
}

```

```

    );
    // 주차 자리 버튼
    for (int i = 0; i < 12; i++) {
        final int a = i;
        parkNumButtons[i].addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                EnterPanel2.selectedNum = a;
            }
        });
        parkNumButtons[i].addFocusListener(this);
    }

    // 컴포넌트 추가
    add(carIDJLabel);
    add(carIDField);
    add(nameJLabel);
    add(nameField);
    add(phoneNumJLabel);
    add(phoneNumberField);
    add(carModelJLabel);
    add(carModelField);
    add(okButton);

    // 위치 설정
    carIDJLabel.setBounds(50, 50, 100, 20);
    carIDField.setBounds(140, 50, 100, 20);
    nameJLabel.setBounds(300, 50, 50, 20);
    nameField.setBounds(360, 50, 100, 20);
    phoneNumJLabel.setBounds(490, 50, 100, 20);
    phoneNumberField.setBounds(580, 50, 150, 20);
    carModelJLabel.setBounds(770, 50, 50, 20);
    carModelField.setBounds(830, 50, 100, 20);
    okButton.setBounds(800, 450, 100, 50);
    int x1 = 0, x2 = 0;
    for (int i = 0; i < 12; i++) {
        if (i < 7) {
            parkNumButtons[i].setBounds(50 + x1, 120, 100, 150);
            x1 += 130;
        }
        else {
            parkNumButtons[i].setBounds(50 + x2, 320, 100, 150);
            x2 += 130;
        }
    }
}

public void focusGained(FocusEvent e) {
    JButton buttonn = (JButton)e.getSource();
    buttonn.setBackground(new Color(124,205,255,100));
}

public void focusLost(FocusEvent e) {
    JButton buttonn = (JButton)e.getSource();
    buttonn.setBackground(Color.white);
}

}

/** Enter Result Panel */
class EnterResultPanel extends MyPanel {

```

```

// 컴포넌트
SystemFrame systemFrame; // 현재 패널을 포함한 프레임
JButton okButton; // 확인 버튼
JLabel okLabel; // 주차 완료 문구

// 생성자
public EnterResultPanel(SystemFrame systemFrame) {

    this.systemFrame = systemFrame;

    // 배치관리자 : null (절대위치로 배치)
    this.setLayout(null);

    // 컴포넌트 생성
    // JButton
    okButton = new JButton("돌아가기");
    // JLabel
    okLabel = new JLabel("주차가 완료되었습니다.");
    okLabel.setFont(b);
    okLabel.setHorizontalAlignment(JLabel.CENTER);

    // 이벤트 처리 코드
    okButton.addActionListener(e -> {
        systemFrame.change(systemFrame.menuPanel); // 메인 메뉴로
    });

    // 컴포넌트 추가
    add(okButton);
    add(okLabel);

    // 위치 설정
    okButton.setBounds(800, 450, 100, 50);
    okLabel.setBounds(0, 0, 1000, 500);
}

}

/** Exit Panel */
class ExitPanel extends MyPanel {

    // 컴포넌트
    SystemFrame systemFrame; // 현재 패널을 포함한 프레임
    JButton okButton; // 확인 버튼
    JTextField carIDField, phoneNumField, nameField, parkNumField; // 자동차
    번호, 전화번호, 이름, 주차 자리
    JLabel carIDJLabel, phoneNumJLabel, nameJLabel, parkNumJLabel, calculateJLabel,
    amountJLabel;
    int number; // 주차 번호

    // 생성자
    public ExitPanel(SystemFrame systemFrame) {

        this.systemFrame = systemFrame;

        // 배치관리자 : null (절대위치로 배치)
        this.setLayout(null);

```

```

// 컴포넌트 생성
okButton = new JButton("출차하기");
// JLabel
carIDJLabel = new JLabel("차량 번호");
phoneNumJLabel = new JLabel("전화번호");
nameJLabel = new JLabel("이름");
parkNumJLabel = new JLabel("주차 번호");
calculateJLabel = new JLabel("정산하기");
amountJLabel = new JLabel(); // 얼마인지 계산해서 넣기
carIDJLabel.setFont(d);
phoneNumJLabel.setFont(new Font("맑은 고딕", Font.BOLD, 18));
nameJLabel.setFont(new Font("맑은 고딕", Font.BOLD, 18));
parkNumJLabel.setFont(new Font("맑은 고딕", Font.BOLD, 18));
calculateJLabel.setFont(d);
amountJLabel.setFont(c);
// JField
carIDField = new JTextField();
phoneNumberField = new JTextField();
nameField = new JTextField();
parkNumField = new JTextField();
carIDField.setFont(a);
phoneNumberField.setFont(a);
nameField.setFont(a);
phoneNumberField.setEditable(false);
nameField.setEditable(false);
parkNumField.setEditable(false);

// 이벤트 처리 코드
// parkingLot에서 현재 아이디를 찾은 다음 나머지 필드를 채우고 현재 시간
저장
carIDField.addActionListener(e -> {
    for (int i = 0; i < 12; i++) {
        if
(systemFrame.parkingLot[i].aCar.carIDString.equals(carIDField.getText())) {
            // 현재 시간 저장 및 계산
            systemFrame.parkingLot[i].exitTime = (time.getHour() * 100) +
(time.getMinute());
            // 정기결제 여부
            if (systemFrame.parkingLot[i].aCar.isMember == true) {
                amountJLabel.setText("정기결제 회원");
            } else {
                amountJLabel.setText(systemFrame.parkingLot[i].calculate() + "
원");
            }
            // 화면에 해당 정보 출력
phoneNumberField.setText(systemFrame.parkingLot[i].aCar.phoneNumString);
nameField.setText(systemFrame.parkingLot[i].aCar.nameString);
parkNumField.setText((i + 1) + " 번");
number = i;
break;
        }
    }
    else if (i == 11) {
        // 해당 차량이 없다고 안내 후 main으로
        systemFrame.failPanel.failLabel.setText("해당 차량이 없습니다.");
        systemFrame.change(systemFrame.failPanel);
    }
}
}

```

```

});

oKButton.addActionListener(e -> {
    // parkingLot 배열에서 현재 차량 정보 삭제
    systemFrame.parkingLot[number] = new ParkingLot();
    // 대기 차량이 있다면 첫번째 차량을 해당 자리에 주차하기
    if (systemFrame.waitingList.size() > 0) {
        String model = systemFrame.waitingList.getFirst().carModel;
        if (model == "장애인 차량") {
            systemFrame.parkingLot[number] = new
ParkingLot(systemFrame.waitingList.getFirst(), true, true);
        }
        else {
            systemFrame.parkingLot[number] = new
ParkingLot(systemFrame.waitingList.getFirst(), true, false);
        }
        systemFrame.parkingLot[number].enterTime = (time.getHour() * 100) +
(time.getMinute());
        systemFrame.waitingList.removeFirst(); // 삭제
        systemFrame.change(systemFrame.addWaitingResultPanel); // 다음 패널로
    }
    systemFrame.change(systemFrame.exitResultPanel); // 다음 패널로
});

// 컴포넌트 추가
add(carIDJLabel);
add(carIDField);
add(nameJLabel);
add(parkNumJLabel);
add(calculateJLabel);
add(amountJLabel);
add(nameField);
add(phoneNumJLabel);
add(phoneNumberField);
add(parkNumField);
add(oKButton);

// 위치 설정
carIDJLabel.setBounds(50, 100, 200, 50);
carIDField.setBounds(200, 100, 300, 50);
nameJLabel.setBounds(50, 200, 200, 30);
nameField.setBounds(200, 200, 300, 30);
phoneNumJLabel.setBounds(50, 300, 200, 30);
phoneNumberField.setBounds(200, 300, 300, 30);
parkNumJLabel.setBounds(50, 400, 200, 30);
parkNumField.setBounds(200, 400, 300, 30);
oKButton.setBounds(800, 450, 100, 50);
calculateJLabel.setBounds(600, 100, 200, 50);
amountJLabel.setBounds(600, 200, 350, 50);
}
}

/** Exit Result Panel */
class ExitResultPanel extends MyPanel {

    // 컴포넌트
    SystemFrame systemFrame; // 현재 패널을 포함한 프레임
    JButton oKButton; // 확인 버튼

```

```

JLabel okLabel; // 주차 완료 문구

// 생성자
public ExitResultPanel(SystemFrame systemFrame) {

    this.systemFrame = systemFrame;

    // 배치관리자 : null (절대위치로 배치)
    this.setLayout(null);

    // 컴포넌트 생성
    // JButton
    okButton = new JButton("돌아가기");
    // JLabel
    okLabel = new JLabel("출차가 완료되었습니다.");
    okLabel.setFont(b);
    okLabel.setHorizontalAlignment(JLabel.CENTER);

    // 이벤트 처리 코드
    okButton.addActionListener(e -> {
        systemFrame.change(systemFrame.menuPanel); // 메인 메뉴로
    });

    // 컴포넌트 추가
    add(okButton);
    add(okLabel);

    // 위치 설정
    okButton.setBounds(800, 450, 100, 50);
    okLabel.setBounds(0, 0, 1000, 500);
}

}

/** Add Waiting Result Panel */
class AddWaitingResultPanel extends MyPanel {

    // 컴포넌트
    SystemFrame systemFrame; // 현재 패널을 포함한 프레임
    JButton okButton; // 확인 버튼
    JLabel okLabel; // 주차 완료 문구

    // 생성자
    public AddWaitingResultPanel(SystemFrame systemFrame) {

        this.systemFrame = systemFrame;

        // 배치관리자 : null (절대위치로 배치)
        this.setLayout(null);

        // 컴포넌트 생성
        // JButton
        okButton = new JButton("돌아가기");
        // JLabel
        okLabel = new JLabel(systemFrame.parkingLot[systemFrame.exitPanel.number] + "
        번 대기 차량을 주차하였습니다.");
        okLabel.setFont(c);
        okLabel.setHorizontalAlignment(JLabel.CENTER);
    }
}

```



```

// 이벤트 처리 코드
okButton.addActionListener(e -> {
    systemFrame.change(systemFrame.menuPanel); // 메인 메뉴로
});

// 컴포넌트 추가
add(okButton);
add(okLabel);

// 위치 설정
okButton.setBounds(800, 450, 100, 50);
okLabel.setBounds(0, 0, 1000, 500);
}
}

/** Display Panel */
class DisplayPanel extends MyPanel {

    // 컴포넌트
    SystemFrame systemFrame; // 현재 패널을 포함한 프레임
    JButton okButton, noButton; // 확인 버튼

    // 생성자
    public DisplayPanel(SystemFrame systemFrame) {

        this.systemFrame = systemFrame;

        // 배치관리자 : null (절대위치로 배치)
        this.setLayout(null);

        // 컴포넌트 생성
        // JButton
        okButton = new JButton("주차 리스트");
        noButton = new JButton("대기 리스트");
        okButton.setFont(c);
        noButton.setFont(c);

        // 이벤트 처리 코드
        okButton.addActionListener(e -> {
            systemFrame.ParkingLotDisplayPanel = new
ParkingLotDisplayPanel(systemFrame);
            systemFrame.change(systemFrame.ParkingLotDisplayPanel);
        });
        noButton.addActionListener(e -> {
            systemFrame.waitingListDisplayPanel = new
WaitingListDisplayPanel(systemFrame);
            systemFrame.change(systemFrame.waitingListDisplayPanel);
        });

        // 컴포넌트 추가
        add(okButton);
        add(noButton);

        // 위치 설정
        okButton.setBounds(250, 220, 200, 70);
        noButton.setBounds(550, 220, 200, 70);
    }
}

```

```

}

/** Waiting List Display Panel */
class WaitingListDisplayPanel extends MyPanel {

    // 컴포넌트
    SystemFrame systemFrame; // 현재 패널을 포함한 프레임
    JButton okButton; // 확인 버튼
    JLabel[] labels; // 출력 내용

    // 생성자
    public WaitingListDisplayPanel(SystemFrame systemFrame) {

        this.systemFrame = systemFrame;

        // 배치관리자 : null (절대위치로 배치)
        this.setLayout(null);

        // 컴포넌트 생성
        okButton = new JButton("돌아가기");
        // JLabel
        labels = new JLabel[12]; // 12개까지 표시함
        int y = 0;
        // 딸기
        System.out.println(systemFrame.waitingList.size());
        for (int i = 0; i < systemFrame.waitingList.size(); i++) {
            // 생성
            labels[i] = new JLabel((i+1) + " 번)      "
                + systemFrame.waitingList.get(i).carIDString + "      "
                + systemFrame.waitingList.get(i).nameString + "      "
                + systemFrame.waitingList.get(i).phoneNumString + "      "
                + systemFrame.waitingList.get(i).carModel + "      ");

            // 설정
            labels[i].setFont(new Font("맑은 고딕", Font.PLAIN, 15));
            // 추가
            add(labels[i]);
            // 위치 설정
            if (i < 7) {
                labels[i].setBounds(50, 50 + y, 400, 50);
            }
            else {
                labels[i].setBounds(500, y - 440, 400, 50);
            }
            y += 70;
        }

        // 이벤트 처리 코드
        okButton.addActionListener(e -> {
            systemFrame.change(systemFrame.menuPanel);
        });

        // 컴포넌트 추가
        add(okButton);

        // 위치 설정
        okButton.setBounds(800, 450, 100, 50);
    }
}

```

```

    }
}

/** Parking Lot Display Panel */
class ParkingLotDisplayPanel extends MyPanel {

    // 컴포넌트
    SystemFrame systemFrame; // 현재 패널을 포함한 프레임
    JButton okButton; // 확인 버튼
    JLabel[] labels; // 출력 내용

    // 생성자
    public ParkingLotDisplayPanel(SystemFrame systemFrame) {

this.systemFrame = systemFrame;

        // 배치관리자 : null (절대위치로 배치)
        this.setLayout(null);

        // 컴포넌트 생성
        okButton = new JButton("돌아가기");
        // JLabel
        labels = new JLabel[12];
        int y = 0;
        for (int i = 0; i < 12; i++) {
            // 생성
            labels[i] = new JLabel((i+1) + " 번)      "
                + systemFrame.parkingLot[i].aCar.carIDString + "      "
                + systemFrame.parkingLot[i].aCar.nameString + "      "
                + systemFrame.parkingLot[i].aCar.phoneNumString + "      "
                + systemFrame.parkingLot[i].aCar.carModel + "      ");
            // 설정
            labels[i].setFont(new Font("맑은 고딕", Font.PLAIN, 15));
            if (systemFrame.parkingLot[i].aCar.carIDString.equals("차 번호") ==
false) {
                add(labels[i]);
                // 위치 설정
                if (i < 7) {
                    labels[i].setBounds(50, 50 + y, 400, 50);
                }
                else {
                    labels[i].setBounds(500, y - 440, 400, 50);
                }
                y += 70;
            }
        }

        // 이벤트 처리 코드
        okButton.addActionListener(e -> {
            systemFrame.change(systemFrame.menuPanel);
        });

        // 컴포넌트 추가
        add(okButton);

        // 위치 설정
        okButton.setBounds(800, 450, 100, 50);
    }
}

```

```

    }
}

/** Member Panel */
class MemberPanel extends MyPanel {

    // 컴포넌트
    SystemFrame systemFrame; // 현재 패널을 포함한 프레임
    JButton okButton, noButton; // 확인 버튼

    // 생성자
    public MemberPanel(SystemFrame systemFrame) {

        this.systemFrame = systemFrame;

        // 배치관리자 : null (절대위치로 배치)
        this.setLayout(null);

        // 컴포넌트 생성
        // JButton
        okButton = new JButton("추가");
        noButton = new JButton("조회");
        okButton.setFont(c);
        noButton.setFont(c);

        // 이벤트 처리 코드
        okButton.addActionListener(e -> {
            systemFrame.addMemberPanel = new AddMemberPanel(systemFrame);
            systemFrame.change(systemFrame.addMemberPanel);
        });
        noButton.addActionListener(e -> {
            systemFrame.memberDisplayPanel = new MemberDisplayPanel(systemFrame);
            systemFrame.change(systemFrame.memberDisplayPanel);
        });

        // 컴포넌트 추가
        add(okButton);
        add(noButton);

        // 위치 설정
        okButton.setBounds(250, 220, 200, 70);
        noButton.setBounds(550, 220, 200, 70);
    }
}

/** Member Display Panel */
class MemberDisplayPanel extends MyPanel {

    // 컴포넌트
    SystemFrame systemFrame; // 현재 패널을 포함한 프레임
    JButton okButton; // 확인 버튼
    JLabel[] labels; // 출력 내용

    // 생성자
    public MemberDisplayPanel(SystemFrame systemFrame) {

        this.systemFrame = systemFrame;
    }
}

```

```

        // 배치관리자 : null (절대위치로 배치)
        this.setLayout(null);

        // 컴포넌트 생성
        okButton = new JButton("돌아가기");
        // JLabel
        labels = new JLabel[12]; // 12개까지 표시함
        int y = 0;
        for (int i = 0; i < systemFrame.memberList.size(); i++) {
            // 생성
            labels[i] = new JLabel((i+1) + " 번)      "
                + systemFrame.memberList.get(i).carIDString + "      "
                + systemFrame.memberList.get(i).nameString + "      "
                + systemFrame.memberList.get(i).phoneNumString + "      "
                + systemFrame.memberList.get(i).carModel + "      ");

            // 설정
            labels[i].setFont(new Font("맑은 고딕", Font.PLAIN, 15));
            // 추가
            add(labels[i]);
            // 위치 설정
            if (i < 7) {
                labels[i].setBounds(50, 50 + y, 400, 50);
            }
            else {
                labels[i].setBounds(500, y - 440, 400, 50);
            }
            y += 70;
        }

        // 이벤트 처리 코드
        okButton.addActionListener(e -> {
            systemFrame.change(systemFrame.menuPanel);
        });

        // 컴포넌트 추가
        add(okButton);

        // 위치 설정
        okButton.setBounds(800, 450, 100, 50);
    }
}

/** Add Member Panel */
class AddMemberPanel extends MyPanel {
    // 컴포넌트
    SystemFrame systemFrame; // 현재 패널을 포함한 프레임
    JButton okButton; // 확인 버튼
    JTextField carIDField, phoneNumField, nameField; // 자동차 번호, 전화번호,
이름
    JRadioButton sedan, van, light, disabled; // 차 기종 : 승용차, 승합차, 경차,
장애인
    ButtonGroup group; // 차 기종 라디오버튼 그룹
    JLabel carIDJLabel, phoneNumJLabel, nameJLabel, carModelLabel;

```

```

// 생성자
public AddMemberPanel(SystemFrame systemFrame) {

    this.systemFrame = systemFrame;

    // 배치관리자 : null (절대위치로 배치)
    this.setLayout(null);

    // 컴포넌트 생성
    okButton = new JButton("입력하기");
    // JLabel
    carIDJLabel = new JLabel("차량 번호");
    phoneNumJLabel = new JLabel("전화번호");
    nameJLabel = new JLabel("이름");
    carModelLabel = new JLabel("기종");
    carIDJLabel.setFont(d);
    phoneNumJLabel.setFont(d);
    nameJLabel.setFont(d);
    carModelLabel.setFont(d);
    // JField
    carIDField = new JTextField();
    phoneNumberField = new JTextField();
    nameField = new JTextField();
    carIDField.setFont(a);
    phoneNumberField.setFont(a);
    nameField.setFont(a);
    // JRadioButton
    sedan = new JRadioButton("승용차");
    van = new JRadioButton("승합차");
    light = new JRadioButton("경차");
    disabled = new JRadioButton("장애인 차량");
    sedan.setBackground(null);
    van.setBackground(null);
    light.setBackground(null);
    disabled.setBackground(null);
    sedan.setFont(a);
    van.setFont(a);
    light.setFont(a);
    disabled.setFont(a);
    // ButtonGroup (Radio Button)
    group = new ButtonGroup();
    group.add(sedan);
    group.add(van);
    group.add(light);
    group.add(disabled);

    // 이벤트 처리 코드
    okButton.addActionListener(e -> {
        // 정보를 모두 입력하지 않은 경우
        if ((carIDField.getText().equals("")) ||
(phoneNumberField.getText().equals(""))
        || (nameField.getText().equals("")) || (group.getSelection() ==
null)) {
            systemFrame.failPanel.failLabel.setText("정보를 정확하게
입력해주세요.");
            systemFrame.change(systemFrame.failPanel);
        } else {

```

```

        // current Vehicle에 차량 정보 입력
        String model = "승용차";
        if (sedan.isSelected() == true) {
            model = "승용차";
        } else if (van.isSelected() == true) {
            model = "승합차";
        } else if (light.isSelected() == true) {
            model = "경차";
        } else if (disabled.isSelected() == true) {
            model = "장애인 차량";
        }
        systemFrame.currentVehicle = new Vehicle(carIDField.getText(),
nameField.getText(),
            phoneNumberField.getText(), model, true);
        systemFrame.memberList.add(systemFrame.currentVehicle);
        systemFrame.change(systemFrame.menuPanel);
    });
}

// 컴포넌트 추가
add(carIDJLabel);
add(carIDField);
add(nameJLabel);
add(nameField);
add(phoneNumJLabel);
add(phoneNumberField);
add(carModelLabel);
add(sedan);
add(van);
add(light);
add(disabled);
add(okButton);

// 위치 설정
carIDJLabel.setBounds(50, 100, 200, 50);
carIDField.setBounds(200, 100, 300, 50);
nameJLabel.setBounds(50, 200, 200, 50);
nameField.setBounds(200, 200, 300, 50);
phoneNumJLabel.setBounds(50, 300, 200, 50);
phoneNumberField.setBounds(200, 300, 300, 50);
carModelLabel.setBounds(600, 100, 200, 50);
sedan.setBounds(600, 150, 200, 50);
van.setBounds(600, 200, 200, 50);
light.setBounds(600, 250, 200, 50);
disabled.setBounds(600, 300, 200, 50);
okButton.setBounds(800, 450, 100, 50);
    }
}

/** Fail Panel */
class FailPanel extends MyPanel {

    // 컴포넌트
    SystemFrame systemFrame; // 현재 패널을 포함한 프레임
    JButton okButton; // 확인 버튼
    JLabel failLabel; // 경고 문구

```

```

// 생성자
public FailPanel(SystemFrame systemFrame) {

    this.systemFrame = systemFrame;

    // 배치관리자 : null (절대위치로 배치)
    this.setLayout(null);

    // 컴포넌트 생성
    // JButton
    okButton = new JButton("돌아가기");
    // JLabel
    failLabel = new JLabel("안내문구");
    failLabel.setFont(b);
    failLabel.setHorizontalAlignment(JLabel.CENTER);

    // 이벤트 처리 코드
    okButton.addActionListener(e -> {
        systemFrame.change(systemFrame.menuPanel); // 메인 메뉴로
    });

    // 컴포넌트 추가
    add(okButton);
    add(failLabel);

    // 위치 설정
    okButton.setBounds(800, 450, 100, 50);
    failLabel.setBounds(0, 0, 1000, 500);
}

}

/** Fail Panel to Enter Panel */
class FailPanelToEnter extends MyPanel {

    // 컴포넌트
    SystemFrame systemFrame; // 현재 패널을 포함한 프레임
    JButton okButton; // 확인 버튼
    JLabel failLabel; // 경고 문구

    // 생성자
    public FailPanelToEnter(SystemFrame systemFrame) {

        this.systemFrame = systemFrame;

        // 배치관리자 : null (절대위치로 배치)
        this.setLayout(null);

        // 컴포넌트 생성
        // JButton
        okButton = new JButton("돌아가기");
        // JLabel
        failLabel = new JLabel("안내문구");
        failLabel.setFont(b);
        failLabel.setHorizontalAlignment(JLabel.CENTER);

        // 이벤트 처리 코드
        okButton.addActionListener(e -> {
            systemFrame.change(systemFrame.enterPanel2); // 메인 메뉴로
        });
    }
}

```



```

});

// 컴포넌트 추가
add(okButton);
add(faiLabel);

// 위치 설정
okButton.setBounds(800, 450, 100, 50);
faiLabel.setBounds(0, 0, 1000, 500);
}
}

/** Add Waiting List Panel */
class AddWaitingListPanel extends MyPanel {

    // 컴포넌트
    SystemFrame systemFrame; // 현재 패널을 포함한 프레임
    JButton okButton, noButton; // 확인 버튼
    JLabel label, label2; // 안내 문구

    // 생성자
    public AddWaitingListPanel(SystemFrame systemFrame) {

        this.systemFrame = systemFrame;

        // 배치관리자 : null (절대위치로 배치)
        this.setLayout(null);

        // 컴포넌트 생성
        // JButton
        okButton = new JButton("추가하기");
        noButton = new JButton("돌아가기");
        // JLabel
        label = new JLabel("주차장에 자리가 없습니다.");
        label2 = new JLabel("해당차량을 대기리스트에 추가하시겠습니까?");
        label.setFont(new Font("맑은 고딕", Font.BOLD, 35));
        label.setHorizontalAlignment(JLabel.CENTER);
        label2.setFont(new Font("맑은 고딕", Font.BOLD, 35));
        label2.setHorizontalAlignment(JLabel.CENTER);

        // 이벤트 처리 코드
        okButton.addActionListener(e -> {
            // 대기리스트에 추가
            systemFrame.waitingList.add(new Vehicle(systemFrame.currentVehicle));
            // 딸기 : 대기리스트에 추가되었습니다
            systemFrame.change(systemFrame.menuPanel);
        });
        noButton.addActionListener(e -> {
            systemFrame.failPanel.faiLabel.setText("대기리스트에 추가되었습니다.");
            systemFrame.change(systemFrame.failPanel); // 메인 메뉴로
        });

        // 컴포넌트 추가
        add(okButton);
        add(noButton);
        add(label);
        add(label2);
    }
}

```

```
// 위치 설정
oKButton.setBounds(800, 450, 100, 50);
noButton.setBounds(670, 450, 100, 50);
label.setBounds(0, 150, 1000, 100);
label2.setBounds(0, 250, 1000, 100);
}
```

## 4. 실행 결과 이미지

자바시 공영주차장 관리 프로그램

23:03:53

주차하기

출차하기

조회하기

정기결제 관리

시작 화면

자바시 공영주차장 관리 프로그램

차량 번호 12가3456

이름 변혜빈

전화번호

기종

☒ 승용차

☐ 승합차

☐ 경차

☐ 장애인 차량

입력하기

주차하기 메뉴 -> 정보 입력

자바시 공영주차장 관리 프로그램

차량 번호 12가3456 이름 변해빈 전화번호 기종 승용차

1 번	2 번	3 번	4 번	5 번	6 번	7 번
8 번	9 번	10 번	장애인 1 번	장애인 2 번		

주차 하기

입력하기 -> 주차 자리 선택

자바시 공영주차장 관리 프로그램

차량 번호 12가3456 정산하기

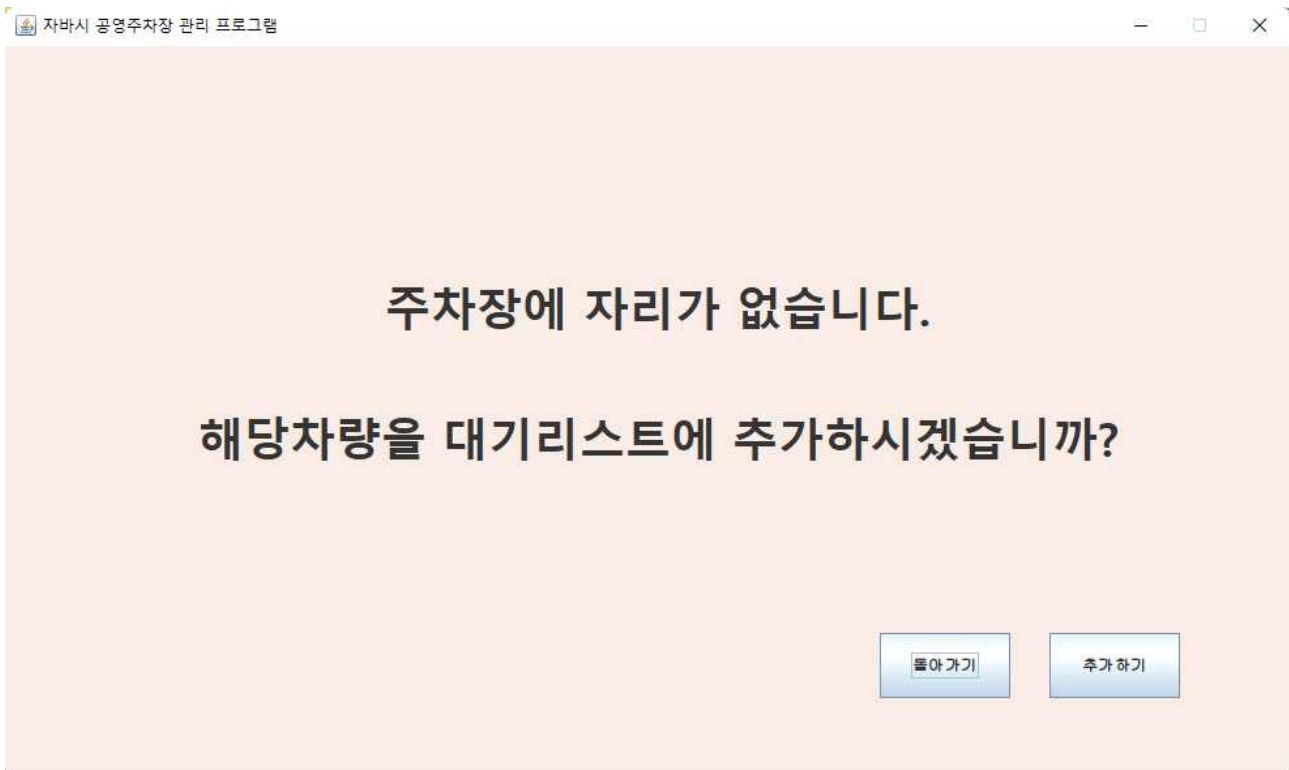
이름 변해빈 500 원

전화번호

주차 번호 3 번

출차 하기

정산하기 -> 차량번호 입력 후 엔터



주차하기 -> 입력하기 -> 주차자리가 없는 경우



조회하기 -> 대기리스트



정기결제 추가 -> 조회

## 5. 기획 대비 완성도



(1) 미완성 목록

- 실시간 시계 : 첫 메인메뉴 화면에서 실시간으로 현재 시각이 표시되는 시계를 표시하고 싶었지만 완성하지 못하였다.

(2) 완성률 : 95%

## 6. 후기

## (1) 느낀점

처음에는 주차장 이용 고객이 사용할 수 있는 자동 정산 키오스크를 만들고자 하였다. 하지만 레퍼런스로 삼았던 기존 프로그램의 코드가 관리자의 입장에 맞추어 설계되었다고 느꼈고, 주차장 관리자가 사용하는 프로그램으로 목적을 변경하였다. 관리자 입장에서 사용하는 메뉴의 구성은 사용자 키오스크와는 다른 점이 있어서 초기 구상과 최종 기획이 다른 점이 많았다.

레퍼런스로 삼은 기존 프로그램이 GUI가 아닌 CUI로 구성되어 있었다. 기존 프로그램 화면을 GUI 기반으로 바꾸면서, 기존 코드를 사용할 수 있을 거라고 생각했다. 하지만 GUI를 만들다 보니 코드 진행 방식이나 순서가 기존 코드와는 완전히 달라서 새로운 코드를 처음부터 다시 작성한 셈이 되었다. 기존 프로그램의 코드에서 필요 없다고 생각한 부분은 빼고, 필요한 부분을 추가하였더니, 기존 프로그램에서 프로그램 진행 순서를 제외하고는 완전히 새로운 코드가 되었다.

진행 절반 정도까지는 배치 관리자를 사용하여 GUI를 만들었는데, 생각보다 어려움이 있어서 절대배치방식으로 중간에 바꾸었다. 이 부분이 조금 아쉽다.

## (2) 기획서 변경 사항

- Vehicle 클래스를 상속받아 차종을 구분하는 자식클래스를 만들고자 기획하였으나, 새로운 클래스로 만들어야 할 필요성이 없다고 판단하여 구현하지 않았다. 차종은 Vehicle 클래스의 필드 carModel에 저장하였다.
- 메인 메뉴에 ‘정산’메뉴를 추가하고자 하였으나 ‘출차’메뉴와 비슷해 불필요하다고 느꼈다. 출차 메뉴에서 정산 금액을 보여주는 방식으로 바꾸었다.
- 정기등록을 한 사용자는 ‘차량번호’를 입력하는 것만으로 나머지 데이터를 자동으로 채워지도록 하였다. ‘전화번호’만을 입력받을 수 있도록 하는 기능은 불필요하다고 생각하여 구현하지 않았다.