

C#과 MSSQL을 이용한 차량관리 프로그램

박제성

경북산업직업전문학교

목차

01. 프로그램 개요

02. 프로그램 흐름

03. 프로그램 분석

01 프로그램 개요

주차 시 차량번호, 차주 이름,
전화번호를 등록하여 주차공간,
입차 시간 등을 표시하여
효율적으로 주차를 관리하는
프로그램을 제작



02 프로그램 흐름

The screenshot shows a Windows application window titled "주차 공간 현황 관리" (Parking Space Status Management). The interface includes several sections:

- Left Panel (Callout 1):** Contains input fields for "주차/출차" (Parking/Exit) with fields for "공간번호" (Space Number), "차량번호" (Vehicle Number), "차주이름" (Driver Name), and "전화번호" (Phone Number). There are "주차" (Park) and "출차" (Exit) buttons.
- Top Right Panel (Callout 2):** Contains a "주차 공간 현황 관리" section with a "공간번호" (Space Number) field and buttons for "해당 공간 조회" (Search for this space), "공간추가" (Add space), "공간삭제" (Delete space), and "전체 갱신" (Refresh all). A large blue "P" icon is also present.
- Center Panel (Callout 3):** A table titled "주차현황" (Parking Status) with columns: ParkingSpot, CarNumber, DriverName, PhoneNumber, and ParkingTime. The table has two rows: one with "1" in the first column and another with "2".
- Bottom Right Panel (Callout 4):** A small dialog box titled "주차공간 3이/가 추가 되었습니다." (Parking space 3 has been added.) with an "확인" (OK) button.
- Bottom Left Panel (Callout 5):** A status bar area showing a message: "[2022-02-DeviceDpi 01:58:12]주차공간 20이/가 추가 되었습니다." (Parking space 20 has been added at 2022-02-DeviceDpi 01:58:12).

The status bar at the bottom left shows the current time: "현재시간 : 2022-02-28 01:58:30".

1번(공간번호)에 번호를
입력후
2번(추가, 삭제, 조회)
클릭시
3번(dataGridView)에
공간생성 후 4번창이
뜨고 확인을 누르면
5번(listBox)창에
어떤날짜에 어떤행위를
했는지 표시가 됩니다

02 프로그램 흐름

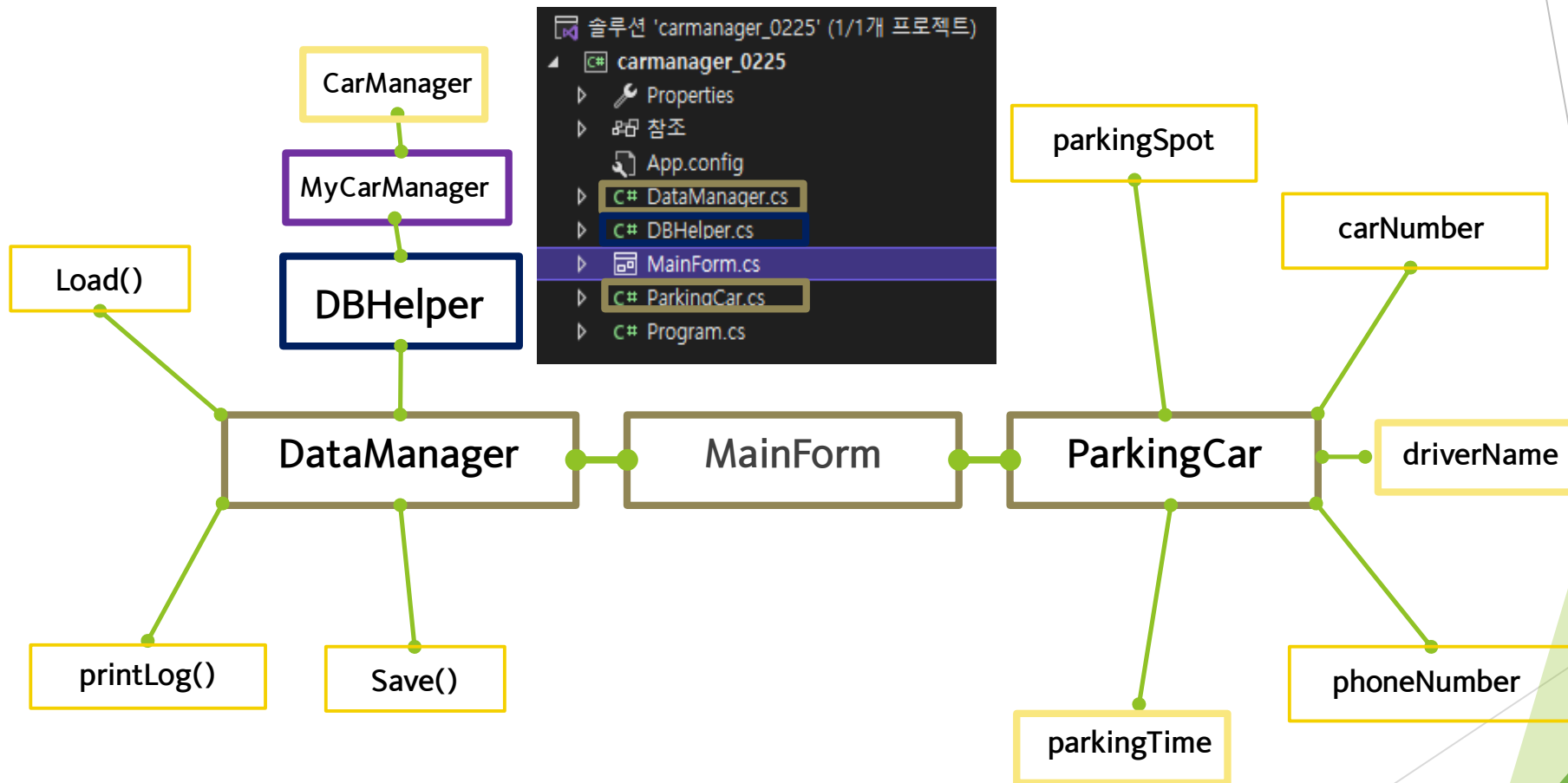
The screenshot shows a parking management application with the following components:

- 1** (Top Left): "주차/출차" (Parking/Exit) section with input fields for "공간번호" (Space Number: 1), "차량번호" (Car Number: 05오5555), "차주이름" (Driver Name: 박제성), and "전화번호" (Phone Number: 010-5555-5555). It includes "주차" (Park) and "출차" (Exit) buttons.
- 2** (Top Right): "주차 공간 현황 관리" (Parking Space Status Management) section with an input field for "공간번호" (Space Number) and buttons for "해당 공간 조회" (Search for this space), "공간추가" (Add space), "공간삭제" (Delete space), and "전체 갱신" (Refresh all). A large blue "P" icon is also present.
- 3** (Middle): "주차 현황" (Parking Status) table showing a list of parked cars. The first row is highlighted.
- 4** (Bottom Left): A log area showing the message "[2022-02-DeviceDpi 01:41:46] 주차공간1에 05오5555차를 주차했습니다." (Parking space 1 has parked car 05오5555).
- 3** (Bottom Center): A confirmation dialog box with the text "주차공간1에 05오5555차를 주차했습니다." (Parking space 1 has parked car 05오5555) and an "확인" (Confirm) button.

현재시간 : 2022-02-28 01:42:35

1번에 공간번호 지정후
정보를 입력한뒤
주차버튼을 클릭하면
3번창이 출력되면서
2번에 입력한 정보가
들어가게되고 4번에
어떤 날짜와 시간에
무슨일을 실행하였는지
저장이됩니다

02 프로그램 흐름



03 프로그램 분석

```
public class DBHelper
{
    private static SqlConnection conn = new SqlConnection();
    public static SqlDataAdapter da;
    public static DataSet ds;
    public static DataTable dt;

    참조 4개
    private static void ConnectDB()
    {
        conn.ConnectionString = string.Format("Data Source={0}; " +
            "initial Catalog = {1}; " +
            "integrated Security = {2}; " +
            "Timeout=3",
            "local", "MyCarManager", "SSPI");
        conn = new SqlConnection(conn.ConnectionString);
        conn.Open();
    }

    참조 3개
    public static void selectQuery(int parkingSpot = -1)
    {
        ConnectDB();

        SqlCommand cmd = new SqlCommand();
        cmd.Connection = conn;
        if(parkingSpot < 0)
        {
            cmd.CommandText = "select * from CarManager";
        }
        else
        {
            cmd.CommandText = "select * from CarManager where ParkingSpot = " + parkingSpot;
        }

        da = new SqlDataAdapter(cmd);
        ds = new DataSet();
        da.Fill(ds, "CarManager");
        dt = ds.Tables[0];

        conn.Close();
    }
}
```

SqlConnection을 통해 MSSQL을
연결시 사용할 객체를 선언하고
ConnectDB메소드를 통해 MSSQL과
프로그램을 연동합니다
MyCarManager라는 이름의
데이터베이스와
연결되며 쿼리문을 이용해
테이블을 불러오는
selectQuery메소드를 사용합니다

03 프로그램 분석

```
public static void insertQuery(int parkingSpot)
{
    try
    {
        ConnectDB();
        string sqlcommnd = "insert into CarManager (parkingSpot) values (@p1)";
        SqlCommand cmd = new SqlCommand();

        cmd.Connection=conn;
        cmd.CommandType = CommandType.Text;

        cmd.Parameters.AddWithValue("@p1", parkingSpot);
        cmd.CommandText = sqlcommnd;
        cmd.ExecuteNonQuery();

        conn.Close();
    }
    catch (Exception ex)
    {
        conn.Close();
        System.Windows.Forms.MessageBox.Show(ex.Message+ Environment.NewLine+ ex.StackTrace);
    }
}
```

insertQuery메서드를 통해
주차공간을 추가할수있는
버튼을 생성할수 있음

03 프로그램 분석

```
public static void Load()
{
    try
    {
        DBHelper.selectQuery();
        Cars.Clear();
        foreach(DataRow item in DBHelper.ds.Tables[0].Rows)
        {
            ParkingCar car = new ParkingCar();
            car.ParkingSpot = int.Parse(item["ParkingSpot"].ToString());
            car.CarNumber = item["CarNumber"].ToString();
            car.DriverName = item["DriverName"].ToString();
            car.PhoneNumber = item["PhoneNumber"].ToString();
            car.ParkingTime = item["ParkingTime"].ToString() == "" ? new DateTime() : DateTime.Parse(item["ParkingTime"].ToString());
            Cars.Add(car);
        }
    }
    catch (Exception ex)
    {
        System.Windows.Forms.MessageBox.Show(ex.Message+Environment.NewLine+ex.StackTrace);
    }
}
```

DBHelper에서 만든 테이블을
winForm의 dataGridView에
표로 만들어준다

03 프로그램 분석

```
public static void PrintLog(string contents)
{
    DirectoryInfo di = new DirectoryInfo("ParkingHistory");

    if (!di.Exists)
    {
        di.Create();
    }

    using (StreamWriter Writer = new StreamWriter("ParkingHistory" + "\\\" + "ParkingHistory" + ".txt", true))
    {
        Writer.WriteLine(contents);
    }
}
```

PrintLog메서드는
Debug폴더에
History.txt를생성
프로그램의 History를
text파일로 확인가능

03 프로그램 분석

```
private void button_parkingAdd_Click(object sender, EventArgs e)
{
    if(textBox_parkingSpot.Text.Trim() == "")
        MessageBox.Show("주차공간을 입력하세요.");
    else if (textBox_carNumber.Text.Trim() == "")
        MessageBox.Show("차량 번호를 입력해주세요.");
    else
    {
        try
        {
            ParkingCar car = DataManager.Cars.Single((x) => x.ParkingSpot.ToString() == textBox_parkingSpot.Text);

            if(car.CarNumber.Trim() != "")
            {
                MessageBox.Show("해당 공간에 이미 차가 있습니다.");
            }
            else
            {
                car.CarNumber = textBox_carNumber.Text;
                car.DriverName = textBox_driverName.Text;
                car.PhoneNumber = textBox_phoneNumber.Text;
                car.ParkingTime = DateTime.Now;

                dataGridView_parkingManager.DataSource = null;
                dataGridView_parkingManager.DataSource = DataManager.Cars;
            }
        }
        catch { }
    }
}
```

textBox 에 차량정보와
차주정보를 입력한뒤
parkingAdd 버튼을 클릭시
생성해둔 테이블에
차량정보, 차주정보를 저장

03 프로그램 분석

```
private void button_parkingRemove_Click(object sender, EventArgs e)
{
    if (textBox_parkingSpot.Text.Trim() == "")
        MessageBox.Show("주차공간을 입력하세요.");
    else
    {
        try
        {
            ParkingCar car = DataManager.Cars.Single((x) => x.ParkingSpot.ToString() == textBox_parkingSpot.Text);
            if (car.CarNumber.Trim() == "")
            {
                MessageBox.Show("해당 공간에 아직 차가 없습니다.");
            }
            else
            {
                string oldCar = car.CarNumber;
                car.CarNumber = "";
                car.DriverName = "";
                car.PhoneNumber = "";
                car.ParkingTime = new DateTime();

                dataGridView_parkingManager.DataSource = null;
                dataGridView_parkingManager.DataSource = DataManager.Cars;

                DataManager.Save(textBox_parkingSpot.Text, "", "", "", true);

                string contents = $"주차공간 {textBox_parkingSpot.Text}에서 {oldCar}차가 출차했습니다.";
                WriteLog(contents);
            }
        }
    }
}
```

parkingRemove 버튼은 출차
버튼으로 클릭시 테이블
공간에 차가 존재하지 않을시에
차가없다는
메세지가 출력되고 테이블에
저장되어있는 정보가 있을경우
삭제가 된다

03 프로그램 분석

```
private void timer_DisplayNow_Tick(object sender, EventArgs e)
{
    label_now_time.Text =
        "현재 시간 : " + DateTime.Now.ToString("yyyy/MM/dd HH:mm:ss");
}

참조 1개
private void button_add_Click(object sender, EventArgs e)
{
    int.TryParse(textBox_parkingSpot_lookUp.Text, out int parkingSpot);

    if(parkingSpot <=0 )
    {
        WriteLog("주차공간번호는 0 이상의 숫자여야 합니다.");
        return;
    }
    DBHelper.selectQuery(parkingSpot);

    if(DBHelper.dt.Rows.Count<1)
    {
        DBHelper.insertQuery(parkingSpot);
        string contents = $"주차공간 {parkingSpot}이/가 추가 되었습니다.";
        WriteLog(contents);
        button_refresh.PerformClick();
    }
    else
    {
        string contents = "해당 주차 공간 이미 존재";
        WriteLog(contents);
    }
}
```

Timer 메서드를 통해
label에 달력,시계를 생성

add_Click메서드에
DBHelper의
selectQuery메서드를
불러와 DB테이블에
주차공간을
생성 가능 deleteQuery를
사용시
DB테이블에서 주차공간
삭제도 가능

Thank you!