



물류배송 관리 프로그램

임수환

CONTENTS



01. 기본구조

02. 메인화면

03. 실행화면

04. 배송정보 관리



01. 기본구조

- 01.send.cs
- 02.Form(Designer , resx)
- 04.Customer.cs
- 03.DataManager.cs

기본 구조



Customer.cs

class



Form

디자인



send.cs

class

DataManager.cs

class



기본 구조(1)

send.cs

```
참조 7개  
class Send  
{  
    참조 7개  
    public string mName { get; set; } //물품 이름  
    참조 10개  
    public string mId { get; set; } //송장번호  
    참조 7개  
    public double mWeight { get; set; } //물품 중량  
    참조 7개  
    public int mCount { get; set; } //물품 무게  
    참조 3개  
    public DateTime RegDate { get; set; } //물품 입고 등록 날짜  
}
```

외부 클래스에서 변수 수정후
다른 클래스로 불러오는 class

참조 1개

```
public Form1()
{
```

```
    InitializeComponent();
```

```
    //https://sosobaba.tistory.com/265 //C# 탭
```

```
    DataManager.Load_Sends();
```

```
    dataGridView_Sends.DataSource = DataManager.Sends;
```

```
    DataManager.Load_Customers();
```

```
    dataGridView_customer.DataSource = DataManager.Customers;
```

```
    if(DataManager.Sends.Count > 0)
```

```
{
```

```
        textBox_name.Text = DataManager.Sends[0].mName;
```

```
        textBox_id.Text = DataManager.Sends[0].mId;
```

```
        textBox_weight.Text = DataManager.Sends[0].mWeight.ToString();
```

```
        textBox_count.Text = DataManager.Sends[0].mCount.ToString();
```

```
}
```

```
    if(DataManager.Customers.Count > 0 )
```

```
{
```

```
        textBox_customer_name.Text = DataManager.Customers[0].mCustomerName;
```

```
        textBox_customer_code.Text = DataManager.Customers[0].mCustomerCode;
```

기본 구조(2)

Form (1)고객 정보

XML파일에 있는 데이터들을
GridView에 기입을 시킵니다.

```
private void button_add_Click(object sender, EventArgs e)
{
    if(textBox_name.Text.Trim() != "" && textBox_id.Text.Trim() != ""
    && textBox_weight.Text.Trim() != "" && textBox_count.Text.Trim() != "")
    {
        bool isDuplicate = false;
        foreach (var item in DataManager.Sends)
        {
            if(item.mId == textBox_id.Text)
            {
                isDuplicate = true;
                break;
            }
        }
        if(isDuplicate)
        {
            MessageBox.Show("해당 송장번호의 고객정보는 이미 존재합니다.");
        }
        else
        {
            try
            {
                Send temp = new Send();
                temp.mName = textBox_name.Text;
                temp.mId = textBox_id.Text;
                double tempWeight = 0.0;
                double.TryParse(textBox_weight.Text, out tempWeight);
                temp.mWeight = tempWeight;
                temp.mCount = int.Parse(textBox_count.Text);
                temp.RegDate = DateTime.Now;
                DataManager.Sends.Add(temp);
                DataManager.PrintLog($"{textBox_id.Text} 학생 추가 완료");
                dataGridView_Sends.DataSource = null;
                dataGridView_Sends.DataSource = DataManager.Sends;
                DataManager.Save_Sends();
            }
            catch (Exception except)
            {
                MessageBox.Show(except.Message);
                DataManager.PrintLog(except.Message);
                DataManager.PrintLog(except.StackTrace);
            }
        }
    }
    else
    {
        MessageBox.Show("누락된 학생 정보가 있습니다.");
    }
}
```

Form

고객 정보 (2) 추가

입력한 고객 정보들을 추가시켜주는 코드입니다.
만약 정보가 하나라도 부족하거나 같은 정보이면 알려줄 메세지박스로 예외처리도 하였습니다.

```

private void button_dell_Click(object sender, EventArgs e)
{
    if (textBox_id.Text.Trim() != "")
    {
        bool isDuplicate = false;
        foreach (var item in DataManager.Sends)
        {
            if (item.mId == textBox_id.Text)
            {
                isDuplicate = true;
                DataManager.Sends.Remove(item);
                break;
            }
        }
        if (isDuplicate == false)
        {
            MessageBox.Show("해당 고객의 정보가 존재하지 않습니다.");
        }
        else
        {
            try
            {
                DataManager.PrintLog($"{textBox_id.Text} 고객 정보 제거 완료");
                dataGridView_Sends.DataSource = null;
                dataGridView_Sends.DataSource = DataManager.Sends;
                DataManager.Save_Sends();
            }
            catch (Exception except)
            {
                MessageBox.Show(except.Message);
                DataManager.PrintLog(except.Message);
                DataManager.PrintLog(except.StackTrace);
            }
        }
        else
        {
            MessageBox.Show("송장 번호 부분이 공백입니다.");
        }
    }
}

```

Form

고객 정보

(3) 삭제

추가한 고객 정보를 삭제 시켜주는 코드입니다.

만약 없는 고객정보이거나 송장 번호 부분이 공백일시에 제입력 하라는 메시지박스를 넣었습니다.

```

private void button_modify_Click(object sender, EventArgs e)
{
    if (textBox_name.Text.Trim() != "" && textBox_id.Text.Trim() != ""
        && textBox_weight.Text.Trim() != "" && textBox_count.Text.Trim() != "")
    {
        bool isDuplicate = false;
        foreach (var item in DataManager.Sends)
        {
            if (item.mId == textBox_id.Text)
            {
                item.mName = textBox_name.Text;
                double.TryParse(textBox_weight.Text, out double tempWeight);
                item.mWeight = tempWeight;
                int.TryParse(textBox_count.Text, out int tempCount);
                item.mCount = tempCount;
                isDuplicate = true;
                break;
            }
        }
        if (isDuplicate == false)
        {
            MessageBox.Show("(수정) 해당 고객의 정보가 존재하지 않습니다.");
        }
        else
        {
            try
            {
                DataManager.PrintLog($"{textBox_id.Text} 학생 수정 완료");
                dataGridView_Sends.DataSource = null;
                dataGridView_Sends.DataSource = DataManager.Sends;
                DataManager.Save_Sends();
            }
            catch (Exception except)
            {
                MessageBox.Show(except.Message);
                DataManager.PrintLog(except.Message);
                DataManager.PrintLog(except.StackTrace);
            }
        }
    }
    else
    {
        MessageBox.Show("(수정) 누락된 학생 정보가 있습니다.");
    }
}

```

Form

고객 정보

(4) 수정

추가한 고객정보를 수정하는 코드입니다.

```

private void button_select_Click(object sender, EventArgs e)
{
    if (textBox_id.Text.Trim() != "")
    {
        bool isDuplicate = false;
        foreach (var item in DataManager.Sends)
        {
            if (item.mId == textBox_id.Text)
            {
                string format = $"송장번호가 {item.mId}인 고객님의 정보입니다.{Environment.NewLine}" +
                    $"이름 : {item.mName}" + Environment.NewLine +
                    $"무게 : {item.mWeight}" + Environment.NewLine +
                    $"개수 : {item.mCount}";
                MessageBox.Show(format);
                DataManager.PrintLog(format);
                isDuplicate = true;
                break;
            }
        }
        if (isDuplicate == false)
        {
            MessageBox.Show("해당 송장번호의 정보가 존재하지 않습니다.");
        }
    }
    else
    {
        MessageBox.Show("(조회)송장번호 부분이 공백입니다.");
    }
}

```

Form

고객 정보 (5) 조회

추가한 고객의 정보를 조회할 때
쓰는 코드이며 송장번호 입력시
해당 고객의 정보를 불러옵니다

```
if (textBox_customer_name.Text.Trim() != "" && textBox_customer_code.Text.Trim() != "")  
{  
    bool isDuplicate = false;  
    foreach (var item in DataManager.Customers)  
    {  
        if (item.mCustomerCode == textBox_customer_code.Text)  
        {  
            isDuplicate = true;  
            break;  
        }  
    }  
    if (isDuplicate)  
    {  
        MessageBox.Show("해당 배송정보는 이미 존재합니다.");  
    }  
    else  
    {  
        try  
        {  
            Customer temp = new Customer();  
            temp.mCustomerName = textBox_customer_name.Text;  
            temp.mCustomerCode = textBox_customer_code.Text;  
            DataManager.Customers.Add(temp);  
            DataManager.PrintLog($"{textBox_customer_code.Text} 배송정보 추가 완료");  
            dataGridView_customer.DataSource = null;  
            dataGridView_customer.DataSource = DataManager.Customers;  
            DataManager.Save_Customers();  
        }  
        catch (Exception except)  
        {  
            MessageBox.Show(except.Message);  
            DataManager.PrintLog(except.Message);  
            DataManager.PrintLog(except.StackTrace);  
        }  
    }  
    else  
    {  
        MessageBox.Show("누락된 배송정보가 있습니다.");  
    }  
}
```

Form

(1-2) 배송 정보 (추가)

지역과 등기번호를 입력하여 추가하는 코드입니다.

```
if (textBox_customer_name.Text.Trim() != "" && textBox_customer_code.Text.Trim() != "")  
{  
    bool isDuplicate = false;  
    foreach (Customer item in DataManager.Customers)  
    {  
        if (item.mCustomerName == textBox_customer_name.Text &  
            item.mCustomerCode == textBox_customer_code.Text)  
        {  
            item.mCustomerName = textBox_customer_name.Text;  
            item.mCustomerCode = textBox_customer_code.Text;  
            isDuplicate = true;  
            break;  
        }  
    }  
    if (isDuplicate == false)  
    {  
        MessageBox.Show("해당 배송정보는 이미 존재합니다.");  
    }  
    else  
    {  
        try  
        {  
            DataManager.PrintLog($"{textBox_customer_code.Text} 배송정보 수정 완료");  
            dataGridView_customer.DataSource = null;  
            dataGridView_customer.DataSource = DataManager.Customers;  
            DataManager.Save_Customers();  
        }  
        catch (Exception except)  
        {  
            MessageBox.Show(except.Message);  
            DataManager.PrintLog(except.Message);  
            DataManager.PrintLog(except.StackTrace);  
        }  
    }  
}  
else  
{  
    MessageBox.Show("누락된 배송 정보가 있습니다.");  
}
```

Form

(1-2) 배송 정보 (수정)

추가한 배송 정보를 수정 하는
코드입니다.

```
private void button_del_major_Click(object sender, EventArgs e)
{
    if (textBox_customer_code.Text.Trim() != "")
    {
        bool isDuplicate = false;
        foreach (var item in DataManager.Customers)
        {
            if (item.mCustomerCode == textBox_customer_code.Text)
            {
                isDuplicate = true;
                DataManager.Customers.Remove(item);
                break;
            }
        }
        if (isDuplicate == false)
        {
            MessageBox.Show("해당 배송정보가 존재하지 않습니다.");
        }
        else
        {
            try
            {
                DataManager.PrintLog($"{textBox_customer_code.Text} 배송정보 제거 완료");
                dataGridView_customer.DataSource = null;
                dataGridView_customer.DataSource = DataManager.Customers;
                DataManager.Save_Customers();
            }
            catch (Exception except)
            {
                MessageBox.Show(except.Message);
                DataManager.PrintLog(except.Message);
                DataManager.PrintLog(except.StackTrace);
            }
        }
    }
    else
    {
        MessageBox.Show("등기번호 부분이 공백입니다.");
    }
}
```

Form

(1-2) 배송 정보 (삭제)

추가한 배송정보를 삭제하는 코드입니다.

```
private void button_select_major_Click(object sender, EventArgs e)
{
    if (textBox_customer_code.Text.Trim() != "")
    {
        bool isDuplicate = false;
        foreach (var item in DataManager.Customers)
        {
            if (item.mCustomerCode == textBox_customer_code.Text)
            {
                string format = $"등기번호가 {item.mCustomerCode}인 분의 정보입니다.{Environment.NewLine}" +
                    $"고객명 : {item.mCustomerName}";
                MessageBox.Show(format);
                DataManager.PrintLog(format);
                isDuplicate = true;
                break;
            }
        }
        if (isDuplicate == false)
        {
            MessageBox.Show("해당 배송정보가 존재하지 않습니다.");
        }
        else
        {
            MessageBox.Show("(조회)등기번호 부분이 공백입니다.");
        }
    }
}
```

Form

(1-2) 배송 정보 (조회)

추가한 배송정보를 조회하는 코
드입니다.

기본 구조(3)

Customer.cs

```
6  
7     namespace LogisticsManager  
8     {  
9         // 참조 7개  
10        class Customer  
11        {  
12            // 참조 7개  
13            public string mCustomerName { get; set; } //지역명  
14            // 참조 11개  
15            public string mCustomerCode { get; set; } //등기번호 int로 할 경우 문제 발생함.  
16            //y2k 예로 들기. 그리고 0330| 33으로 저장되는 것도 설명하기  
17        }  
18    }
```

(customer)XML파일을 저장하는 코드입니다.(※배송정보의 내용이 담김)

기본 구조(4)

DataManager.cs (1)

※고객 정보 부분을 담음.

(sends)XML파일을 불러오고 초

기화 하고 저장 하는코드입니다.

```
namespace LogisticsManager
{
    참조 56개
    class DataManager
    {
        public static List<Send> Sends = new List<Send>();
        public static List<Customer> Customers = new List<Customer>();

        const string SENDLIST = @"./SendsList.xml";
        const string CUSTOMERLIST = @"./customersList.xml";

        참조 2개
        public static void Load_Sends()
        {
            try
            {
                Sends.Clear();
                string sOutput = File.ReadAllText(SENDLIST);
                XElement sElement = XElement.Parse(sOutput);

                Foreach (var item in sElement.Descendants("Send"))
                {
                    string name = item.Element("name").Value;
                    string id = item.Element("id").Value;
                    double Weight = 0;
                    double.TryParse(item.Element("weight").Value, out Weight);
                    int Count = 0;
                    int.TryParse(item.Element("count").Value, out Count);
                    DateTime temp_RegDate = item.Element("regDate").Value == "" ? new DateTime() : DateTime.Parse(item.Element("regDate").Value); //DateTime.Now

                    Send tempSend = new Send()
                    {
                        mName = name,
                        mId = id,
                        mWeight = Weight,
                        mCount = Count,
                        RegDate = temp_RegDate
                    };
                    Sends.Add(tempSend);
                }
            }
            catch (Exception exception)
            {
                MessageBox.Show("파일이 누락되었습니다!" + Environment.NewLine + exception.Message);
                PrintLog(exception.StackTrace);
                CreateFile_Sends();
                Save_Sends();
                Load_Sends();
            }
        }
    }
}
```

```
public static void Save_Sends()
{
    string SendOutput = "";
    SendOutput += "<Sends>" + Environment.NewLine;
    if (Sends.Count > 0)
    {
        foreach (var item in Sends)
        {
            SendOutput += "<Send>" + Environment.NewLine;
            SendOutput += "    <name>" + item.mName + "</name>" + Environment.NewLine;
            SendOutput += "    <id>" + item.mId + "</id>" + Environment.NewLine;
            SendOutput += "    <weight>" + item.mWeight + "</weight>" + Environment.NewLine;
            SendOutput += "    <count>" + item.mCount + "</count>" + Environment.NewLine;
            SendOutput += "    <regDate>" + item.RegDate + "</regDate>" + Environment.NewLine;
            SendOutput += "</Send>" + Environment.NewLine;
        }
    }

    SendOutput += "</Sends>" + Environment.NewLine;

    File.WriteAllText(SNDLIST, SendOutput);
}
```

DataManager.cs (2)

(sends)Xml 저장 코드

DataManager.cs

(3)

```
접속 2/8
public static void Load_Customers()
{
    try
    {
        Customers.Clear();
        string sOutput = File.ReadAllText(CUSTOMERLIST);
        XElement xElement = XElement.Parse(sOutput);

        foreach (var item in xElement.Descendants("customer"))
        {
            string name = item.Element("name").Value;
            string code = item.Element("code").Value;

            Customer temp = new Customer()
            {
                mCustomerName = name,
                mCustomerCode = code
            };
            Customers.Add(temp);
        }
    }
    catch (Exception exception)
    {
        MessageBox.Show("파일이 누락되었습니까!" + Environment.NewLine + exception.Message);
        PrintLog(exception.StackTrace);
        CreateFile_majors();
        Save_Customers();
        Load_Customers();
    }
}
접속 4/8
public static void Save_Customers()
{
    string majorOutput = "";
    majorOutput += "<customers>" + Environment.NewLine;
    if (Customers.Count > 0)
    {
        foreach (var item in Customers)
        {
            majorOutput += "<customer>" + Environment.NewLine;
            majorOutput += "    <name>" + item.mCustomerName + "</name>" + Environment.NewLine;
            majorOutput += "    <code>" + item.mCustomerCode + "</code>" + Environment.NewLine;
            majorOutput += "</customer>" + Environment.NewLine;
        }
    }

    majorOutput += "</customers>" + Environment.NewLine;
    File.WriteAllText(CUSTOMERLIST, majorOutput);
}
접속 1/8
public static void CreateFile_majors()
{
    string fileName = CUSTOMERLIST;
    StreamWriter textWrite = File.CreateText(fileName); //파일이 없으면 자동으로 해당 파일을 생성한다.
    textWrite.Dispose(); //파이썬 예제. using 키워드로 할 수 있으나 파일생성한 후 바로 키워드로 별 차이 없다. 그 다음 때 using 키워드 이용함.
}
접속 22/8
public static void PrintLog(string contents)
{
    DirectoryInfo di = new DirectoryInfo("ProgramHistory");
    if (!di.Exists)
    {
        di.Create();
    }

    using (StreamWriter writer = new StreamWriter("ProgramHistory" + DateTime.Now.ToString("yyyy-MM-dd") + ".txt", true))
    {
        writer.WriteLine($"[{DateTime.Now.ToString("yyyy-MM-dd hh:mm:ss.fff")}]" + contents);
    }
}
```

customer.cs의 xml내용을 불러
오고 저장 삭제 해주는 코드



2. 메인 화면

Visual Studio main Form

Form1

고객정보 배송정보

	mName	mId	mWeight	mCount	RegDate
*					

★송장 번호만 치셔도 고객정보가 나옵니다.

추가 삭제 수정 조회

· 고객이름

· 송장번호

· 무게

· 개수

메인 화면

Visual Studio main Form

고객정보

메인 화면(2)

Visual Studio main Form

배송 정보

고객정보 배송정보

	mCustomerName	mCustomerCode
--	---------------	---------------

★고객 등기 번호만 치셔도 고객정보가 나옵니다.

추가

수정

삭제

조회

· 지역명

· 고객등기번호



3. 실행 결과

실행 결과

고객 정보 부분

Form1

고객정보 배송정보

mName	mId	mWeight	mCount	RegDate
▶ test1	1	123123	123	2021-06-02 오...

★송장 번호만 치셔도 고객정보가 나옵니다.

추가 삭제 수정 조회

· 고객이름 : test1
· 송장번호 : 1
· 무게 : 123123
· 개수 : 123

추가 기능

Form1

고객정보 배송정보

mName	mId	mWeight	mCount	RegDate
▶ test2	1	123123	123	2021-06-02 오...

★송장 번호만 치셔도 고객정보가 나옵니다.

추가 삭제 수정 조회

· 고객이름 : test2
· 송장번호 : 1
· 무게 : 123123
· 개수 : 123

수정 기능

Form1

고객정보 배송정보

mName	mId	mWeight	mCount	RegDate
▶ test1	1	123123	123	2021-06-02 오...

★송장 번호만 치셔도 고객정보가 나옵니다.

추가 삭제 수정 조회

· 고객이름 : test1
· 송장번호 : 1
· 무게 : 123123
· 개수 : 123

삭제 기능

Form1

고객정보 배송정보

mName	mId	mWeight	mCount	RegDate
▶ test2	1	123123	123	2021-06-02 오...

★송장 번호만 치셔도 고객정보가 나옵니다.

추가 삭제 수정 조회

· 고객이름 : test2
· 송장번호 : 1
· 무게 : 123123
· 개수 : 123

송장번호가 1인 고객님의 정보입니다.
이름 : test2
무게 : 123123
개수 : 123

확인

조회 기능

실행 결과

배송 정보 부분

Form1

고객정보 배송정보

mCustomerName	mCustomerCode
▶ test1	1

★고객 등기 번호만 치셔도 고객정보가 나옵니다.

추가 수정 삭제 조회

· 지역명 : test1

· 고객등기번호 : 1

추가 기능

Form1

고객정보 배송정보

mCustomerName	mCustomerCode
▶ test2	1

★고객 등기 번호만 치셔도 고객정보가 나옵니다.

추가 수정 삭제 조회

· 지역명 : test2

· 고객등기번호 : 1

수정 기능

Form1

고객정보 배송정보

mCustomerName	mCustomerCode
▶ test2	1

★고객 등기 번호만 치셔도 고객정보가 나옵니다.

추가 수정 삭제 조회

· 지역명 : test2

· 고객등기번호 : 1

삭제 기능

Form1

고객정보 배송정보

mCustomerName	mCustomerCode
▶ test2	1

★고객 등기 번호만 치셔도 고객정보가 나옵니다.

추가 수정 삭제 조회

· 지역명 : test2

· 고객등기번호 : 1

등기 번호가 1인 분의 정보입니다.
지역명 : test2

확인

조회 기능

Thank You

감사 합니다.

