

# 객체지향프로그래밍 기말과제

학번 : 2019775054

이름 : 전상훈

날짜 : 2023-06-12

과목 : 객체지향프로그래밍

# 목차

---

1. 원본소스
  2. 업그레이드 포인트
  3. 업그레이드 후 결과물
  4. 출처
-

# 1. 원본소스

Repository / IKioskRepository.cs

```
using kiosk.Model;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace kiosk.Repository
{
    interface IKioskRepository
    {
        // 주문
        void AddMenu(string name, int price);

        // 계산
        int Billing();

        // 주문한 메뉴 확인
        List<Menu> SelectOrders();

        // 주문한 갯수
        int CountingMenu();

        // 초기화
        void Reset();
    }
}
```

Model / Menu.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace kiosk.Model
{
    class Menu
    {
        // 메뉴 이름
        public string Name { get; set; }

        // 메뉴 가격
        public int Price { get; set; }

        // 주문 수량
        public int Each { get; set; }

        // 개체 출력 양식
        public override string ToString()
        {
            return $"{Name} : {Price} : {Each}개";
        }
    }
}
```

Repository / KioskRepository.cs

```
using kiosk.Model;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace kiosk.Repository
{
    class KioskRepository : IKioskRepository
    {
        // 주문한 메뉴 개체 인스턴스
        public static List<Menu> menus = new List<Menu>();
        // 주문
        public void AddMenu(string name, int price)
        {
            menus.Add(new Menu { Name = name, Price = price, Each = 1 });
            // 중복 제거, 특정 기준으로 그룹화를 시켜준 다음 그룹 중 첫번째 값만 리스트 형식으로 리턴
            // object가 아닌 literal value List 였다면 .Distinct().ToList() 사용
            menus = menus
                .GroupBy(item => item.Name)
                .Select(item => new Menu { Name = item.First().Name, Price =
                    item.First().Price, Each = (item.First().Each - 1) + item.Count() })
                .ToList();
        }

        // 계산
        public int Billing()
        {
            return menus.Sum((menu) => (menu.Price * menu.Each));
        }

        // 주문한 메뉴 확인
        public List<Menu> SelectOrders()
        {
            return menus;
        }

        // 주문한 갯수
        public int CountingMenu()
        {
            return menus.Sum(menu => menu.Each);
        }

        // 초기화
        public void Reset()
        {
            menus.Clear();
        }
    }
}
```

# 1. 원본소스

```
HambergerForm
배경 흰색과 기본 테두리
radio button
using kiosk.Repository;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace kiosk
{
    public partial class HambergerForm : UserControl
    {
        private KioskRepository repository = new KioskRepository();
        private TextBox textbox;
        private Label label1;
        private Label label2;

        public HambergerForm(TextBox textBox, Label label1, Label label2)
        {
            InitializeComponent();

            this.textbox = textBox;
            this.label1 = label1;
            this.label2 = label2;

            bulgogi.CheckedChanged += CheckItem;
            doublebulgogi.CheckedChanged += CheckItem;
            cheese.CheckedChanged += CheckItem;
            doublecheese.CheckedChanged += CheckItem;
        }

        private void CheckItem(object sender, EventArgs e)
        {
            RadioButton rd = (RadioButton)sender;
            bool flag = rd.Checked;
            if (flag)
            {
                string str = "";
                textbox.Text = "";
                repository.AddMenu(rd.Text, 5000);
                repository.SelectOrders().ForEach(menu => {
                    str += $"{menu.ToString()}\r\n";
                });
                textbox.Text = str;
                label1.Text = $"가격 : {repository.Billing()}";
                label2.Text = $"수량 : {repository.CountingMenu()}";
            }
        }
    }
}
```

```
SideForm
배경 흰색과 기본 테두리
check box
using kiosk.Repository;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace kiosk
{
    public partial class SideForm : UserControl
    {
        private KioskRepository repository = new KioskRepository();
        private TextBox textbox;
        private Label label1;
        private Label label2;

        public SideForm(TextBox textBox, Label label1, Label label2)
        {
            InitializeComponent();

            this.textbox = textBox;
            this.label1 = label1;
            this.label2 = label2;

            frenchfrice.CheckedChanged += CheckItem;
            marinaedfrise.CheckedChanged += CheckItem;
            applepie.CheckedChanged += CheckItem;
            egg Tart.CheckedChanged += CheckItem;
        }

        private void CheckItem(object sender, EventArgs e)
        {
            CheckBox ck = (CheckBox)sender;
            bool flag = ck.Checked;
            if (flag)
            {
                string str = "";
                textbox.Text = "";
                repository.AddMenu(ck.Text, 5000);
                repository.SelectOrders().ForEach(menu => {
                    str += $"{menu.ToString()}\r\n";
                });
                textbox.Text = str;
                label1.Text = $"가격 : {repository.Billing()}";
                label2.Text = $"수량 : {repository.CountingMenu()}";
            }
        }
    }
}
```

```
DrinksForm
배경 흰색, 기본 테두리
radio button
using kiosk.Repository;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace kiosk
{
    public partial class DrinksForm : UserControl
    {
        private KioskRepository repository = new KioskRepository();
        private TextBox textbox;
        private Label label1;
        private Label label2;
        public DrinksForm(TextBox textBox, Label label1, Label label2)
        {
            InitializeComponent();

            this.textbox = textBox;
            this.label1 = label1;
            this.label2 = label2;

            coke.CheckedChanged += CheckItem;
            zecoke.CheckedChanged += CheckItem;
            sider.CheckedChanged += CheckItem;
        }

        private void CheckItem(object sender, EventArgs e)
        {
            RadioButton rd = (RadioButton)sender;
            bool flag = rd.Checked;
            if (flag)
            {
                string str = "";
                textbox.Text = "";
                repository.AddMenu(rd.Text, 5000);
                repository.SelectOrders().ForEach(menu => {
                    str += $"{menu.ToString()}\r\n";
                });
                textbox.Text = str;
                label1.Text = $"가격 : {repository.Billing()}";
                label2.Text = $"수량 : {repository.CountingMenu()}";
            }
        }
    }
}
```

# 1. 원본소스

```
Form1
SplitContainer 사용
메뉴 선택은 button
주문 내역은 textbox(multiline)
주문은 button과 label
메뉴 보이는 부분은 Panel
using kiosk.Repository;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace kiosk
{
    public partial class Form1 : Form
    {
        private Control control = null;
        private KioskRepository repository = new KioskRepository();
        public Form1()
        {
            InitializeComponent();

            private void hanburger_Click(object sender, EventArgs e)
            {
                if (control != null) panel1.Controls.Remove(control);
                control = new HambergerForm(added, price, counter);
                panel1.Controls.Add(control);
            }

            private void side_Click(object sender, EventArgs e)
            {
                if (control != null) panel1.Controls.Remove(control);
                control = new SideForm(added, price, counter);
                panel1.Controls.Add(control);
            }

            private void drinks_Click(object sender, EventArgs e)
            {
                if (control != null) panel1.Controls.Remove(control);
                control = new DrinksForm(added, price, counter);
                panel1.Controls.Add(control);
            }
        }
    }
}
```

```
private void bill_Click(object sender, EventArgs e)
{
    MessageBox.Show($"총 결제 금액은 {price.Text.Replace("가격 : ", "")}원 입니다");

    String str1 = $"str1 : ko[{getString("str1")}]";
    String str2 = $"str2 : ko[{getString("str2")}]";
    String msg = str1 + "\n\n" + str2;
    MessageBox.Show(msg);
}

private void reset_Click(object sender, EventArgs e)
{
    repository.Reset();
    string str = "";
    added.Text = "";
    repository.SelectOrders().ForEach(menu => {
        str += $"{menu.ToString()}\r\n";
    });
    added.Text = str;
    price.Text = $"가격 : {repository.Billing()}";
    counter.Text = $"수량 : {repository.CountingMenu()}";
}

// 언어 지역화
// 기본값은 한국어로 하며, 설정파일(ini) 등에서 값을 가져와 저장하여 사용
// https://srcree.tistory.com/33
public static string sLanguage = "ko";
public static string getString(string name)
{
    return getString(sLanguage, name);
}

public static string getString(string code, string name)
{
    if (code.Equals("en"))
    {
        Thread.CurrentThread.CurrentUICulture = new System.Globalization.CultureInfo("en-US");
        return Properties.StringLib.ResourceManager.GetString(name);
    }
    else
    {
        Thread.CurrentThread.CurrentUICulture = new System.Globalization.CultureInfo("ko-KR");
        return Properties.StringLib.ResourceManager.GetString(name);
    }
}
}
```

# 1. 원본소스

---

Form1

햄버거

사이드

음료

☐ 볼고기   ☐ 더블볼고기   ☐ 치즈   ☐ 더블치즈

수량 :      가격 :        

Form1

햄버거

사이드

음료

☐ 감자튀김   ☐ 양념 감자튀김   ☐ 애플파이   ☐ 에그타르트

수량 :      가격 :

# 1. 원본소스

Form1

햄버거

사이드

음료

☐ 코카콜라 ☐ 제로콜라 ☐ 사이다

수량 :      가격 :      계산      리셋

Form1

햄버거

사이드

음료

☐ 불고기 ☐ 더블불고기 ☒ 치즈 ☐ 더블치즈

불고기 : 5000 : 1개  
더블불고기 : 5000 : 2개  
치즈 : 5000 : 2개  
감자튀김 : 5000 : 1개  
양념 감자튀김 : 5000 : 1개  
코카콜라 : 5000 : 1개

수량 : 8      가격 : 40000      계산      리셋

## 2. 업그레이드 포인트

---

- 1) 데이터베이스 연동
  - 2) 이미지 추가
  - 3) 다양한 창 추가
  - 4) 전체주문내역 추가
  - 5) 결제 시 문자발송 기능 추가
-



### 3. 업그레이드 후 결과물



### 3. 업그레이드 후 결과물

4



아메리카노  
3200



카페모카  
3900



카페라떼  
3700



카부치노  
3700



아인슈페너  
3700



바닐라라떼  
3900

메뉴명 : 카부치노 [Ice] [Tall]      가격 : 5100  
수량 : 1 / 샷 : 1 / 설탕 : 1 / 헤이즐 : 1 / 카라멜 : 1

메뉴명 : 아인슈페너 [Hot] [Grande]      가격 : 5000  
수량 : 1 / 샷 : 1 / 설탕 : 1 / 헤이즐 : 0 / 카라멜 : 0

메뉴명 : 아메리카노 [Hot] [Venti]      가격 : 5300  
수량 : 1 / 샷 : 1 / 설탕 : 0 / 헤이즐 : 1 / 카라멜 : 1

**결제하기**

가격 : 15400

5      주문확인

메뉴명 : 카부치노 [Ice] [Tall]      가격 : 5100  
수량 : 1 / 샷 : 1 / 설탕 : 1 / 헤이즐 : 1 / 카라멜 : 1

메뉴명 : 아인슈페너 [Hot] [Grande]      가격 : 5000  
수량 : 1 / 샷 : 1 / 설탕 : 1 / 헤이즐 : 0 / 카라멜 : 0

메뉴명 : 아메리카노 [Hot] [Venti]      가격 : 5300  
수량 : 1 / 샷 : 1 / 설탕 : 0 / 헤이즐 : 1 / 카라멜 : 1

주문 금액	₩	15400
할인 금액	₩	0
주문 금액	₩	15400

전체 취소

이전

다음

6      결제하기

삼성페이

애플페이

카카오페이

페이코

주문 금액	₩	15400
할인 금액	₩	0
주문 금액	₩	15400

전체 취소

이전

다음

### 3. 업그레이드 후 결과물

7

결제하기

IC카드 결제 시, 카드를 넣어 주세요.



삼성/애플페이 결제 시 휴대폰 뒷면을  
카드 리더기에 가까이 대주세요



주문 금액	₩	15400
할인 금액	₩	0
주문 금액	₩	15400

전체 취소

이전

다음

8

결제가 완료 되었습니다.

대기번호  
[ 96 ]

감사합니다.

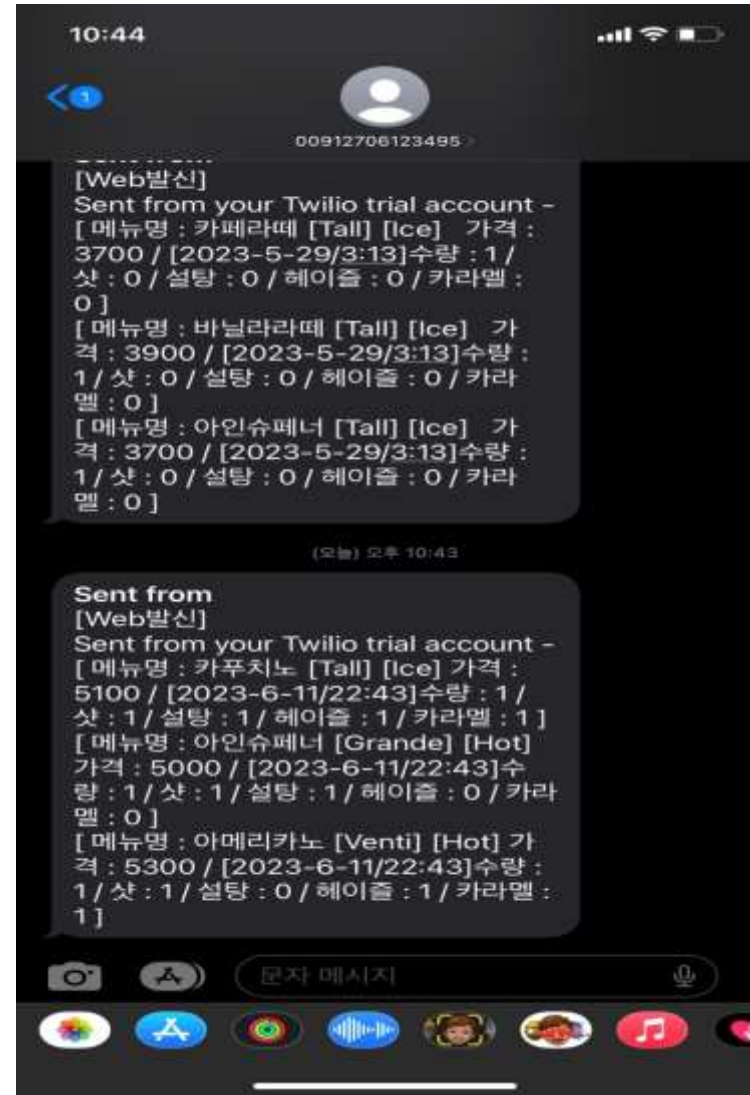
결제완료

### 3. 업그레이드 후 결과물

9



10



# 3. 데이터베이스 연동

---

## -데이터 베이스 연동

```
public static string connectionString = "server=-----
;user=newuser;database=oopcafe;port=3306;password=11111111!";
/*bool isConnected = TestConnection(connectionString);*/
public static DataTable dataTable = LoadTable(connectionString, "orderhistory");

public static DataTable LoadTable(string connectionString, string tableName)
{
    DataTable dataTable = new DataTable();

    using (MySqlConnection connection = new MySqlConnection(connectionString))
    {
        connection.Open();

        string query = $"SELECT * FROM {tableName}";

        using (MySqlCommand command = new MySqlCommand(query, connection))
        {
            using (MySqlDataReader reader = command.ExecuteReader())
            {
                dataTable.Load(reader);
            }
        }
    }

    return dataTable;
}
```

---

# 3. 데이터베이스 연동

---

-데이터 베이스 입력

```
public static void InsertOrderHistory(string coffeename, int coffeeprice, int coffeenum, int shotnum, int sugarnum, int hazlenum, int caranum, string coffeesize, string coffeetemperature, string date)
{
    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        connection.Open();

        string sqlCommand = "INSERT INTO orderhistory (coffeename, coffeeprice, coffeenum, shotnum, sugarnum, hazlenum, caranum, coffeesize, coffeetemperature, date) VALUES (@coffeename, @coffeeprice, @coffeenum, @shotnum, @sugarnum, @hazlenum, @caranum, @coffeesize, @coffeetemperature, @date)";

        using (SqlCommand command = new SqlCommand(sqlCommand, connection))
        {

            command.Parameters.AddWithValue("@coffeename", coffeename);
            command.Parameters.AddWithValue("@coffeeprice", coffeeprice);
            command.Parameters.AddWithValue("@coffeenum", coffeenum);
            command.Parameters.AddWithValue("@shotnum", shotnum);
            command.Parameters.AddWithValue("@sugarnum", sugarnum);
            command.Parameters.AddWithValue("@hazlenum", hazlenum);
            command.Parameters.AddWithValue("@caranum", caranum);
            command.Parameters.AddWithValue("@coffeesize", coffeesize);
            command.Parameters.AddWithValue("@coffeetemperature", coffeetemperature);
            command.Parameters.AddWithValue("@date", date);

            command.ExecuteNonQuery();
        }
    }
}
```

---

# 3. 데이터베이스 연동

-데이터 베이스 출력

```
public void setorderhistory()
{
    using (MySqlConnection connection = new MySqlConnection(MainForm.connectionString))
    {
        connection.Open();
        string sqlCommand = "SELECT coffeename, coffeeprice, coffeenum, shotnum, sugarnum, hazlenum, caranum, coffeesize, coffeetemperature, date FROM orderhistory";

        using (MySqlCommand command = new MySqlCommand(sqlCommand, connection))
        {
            using (MySqlDataReader reader = command.ExecuteReader())
            {
                while (reader.Read())
                {
                    string coffeename = reader["coffeename"].ToString();
                    int coffeeprice = Convert.ToInt32(reader["coffeeprice"]);
                    int coffeenum = Convert.ToInt32(reader["coffeenum"]);
                    int shotnum = Convert.ToInt32(reader["shotnum"]);
                    int sugarnum = Convert.ToInt32(reader["sugarnum"]);
                    int hazlenum = Convert.ToInt32(reader["hazlenum"]);
                    int caranum = Convert.ToInt32(reader["caranum"]);
                    string coffeesize = reader["coffeesize"].ToString();
                    string coffeetemperature = reader["coffeetemperature"].ToString();
                    string date = reader["date"].ToString();

                    string firstLine = $"메뉴명 : {coffeename} [{coffeesize}] [{coffeetemperature}] 가격 : {coffeeprice} / [{date}]";
                    string secondLine = $"수량 : {coffeenum} / 샷 : {shotnum} / 설탕 : {sugarnum} / 헤이즐 : {hazlenum} / 카라멜 : {caranum}";
                    orderhistorylist.Items.Add(firstLine);
                    orderhistorylist.Items.Add(secondLine);
                    orderhistorylist.Items.Add(""); // 이 줄을 추가하여 각 주문 사이에 빈줄을 추가합니다.
                }
            }
        }
    }
}
```



### 3. 데이터베이스 연동



The screenshot shows a window titled "데이터베이스 연동" (Database Connection) with a list of menu items. Each item is displayed with its name, options (Tail, Ice), price, and a timestamp. The items are:

- 메뉴명 : 바닐라라떼 [Tail] [Ice] 가격 : 3900 / [2023-5-29-1:39]
- 수량 : 1 / 샷 : 0 / 설탕 : 0 / 헤이즐 : 0 / 카라멜 : 0
- 메뉴명 : 바닐라라떼 [Tail] [Ice] 가격 : 3900 / [2023-5-29-1:44]
- 수량 : 1 / 샷 : 0 / 설탕 : 0 / 헤이즐 : 0 / 카라멜 : 0
- 메뉴명 : 바닐라라떼 [Tail] [Ice] 가격 : 3900 / [2023-5-29-2:28]
- 수량 : 1 / 샷 : 0 / 설탕 : 0 / 헤이즐 : 0 / 카라멜 : 0
- 메뉴명 : 바닐라라떼 [Tail] [Ice] 가격 : 3900 / [2023-5-29-2:37]
- 수량 : 1 / 샷 : 0 / 설탕 : 0 / 헤이즐 : 0 / 카라멜 : 0
- 메뉴명 : 아인슈페너 [Tail] [Ice] 가격 : 3700 / [2023-5-29-2:37]
- 수량 : 1 / 샷 : 0 / 설탕 : 0 / 헤이즐 : 0 / 카라멜 : 0
- 메뉴명 : 카무치노 [Tail] [Ice] 가격 : 3700 / [2023-5-29-2:37]
- 수량 : 1 / 샷 : 0 / 설탕 : 0 / 헤이즐 : 0 / 카라멜 : 0
- 메뉴명 : 카페라떼 [Venti] [Hot] 가격 : 9700 / [2023-5-29-2:51]
- 수량 : 1 / 샷 : 10 / 설탕 : 0 / 헤이즐 : 0 / 카라멜 : 0
- 메뉴명 : 카무치노 [Grande] [Ice] 가격 : 5400 / [2023-5-29-2:51]
- 수량 : 1 / 샷 : 0 / 설탕 : 4 / 헤이즐 : 0 / 카라멜 : 0
- 메뉴명 : 바닐라라떼 [Tail] [Ice] 가격 : 3900 / [2023-5-29-3:2]
- 수량 : 1 / 샷 : 0 / 설탕 : 0 / 헤이즐 : 0 / 카라멜 : 0
- 메뉴명 : 바닐라라떼 [Tail] [Ice] 가격 : 3900 / [2023-5-29-3:6]
- 수량 : 1 / 샷 : 0 / 설탕 : 0 / 헤이즐 : 0 / 카라멜 : 0
- 메뉴명 : 아인슈페너 [Tail] [Ice] 가격 : 3700 / [2023-5-29-3:6]
- 수량 : 1 / 샷 : 0 / 설탕 : 0 / 헤이즐 : 0 / 카라멜 : 0
- 메뉴명 : 아인슈페너 [Tail] [Ice] 가격 : 3700 / [2023-5-29-3:9]
- 수량 : 1 / 샷 : 0 / 설탕 : 0 / 헤이즐 : 0 / 카라멜 : 0
- 메뉴명 : 카페라떼 [Tail] [Ice] 가격 : 3700 / [2023-5-29-3:9]
- 수량 : 1 / 샷 : 0 / 설탕 : 0 / 헤이즐 : 0 / 카라멜 : 0
- 메뉴명 : 아메리카노 [Tail] [Ice] 가격 : 3200 / [2023-5-29-3:10]
- 수량 : 1 / 샷 : 0 / 설탕 : 0 / 헤이즐 : 0 / 카라멜 : 0

At the bottom of the window, there is a button labeled "닫기" (Close).

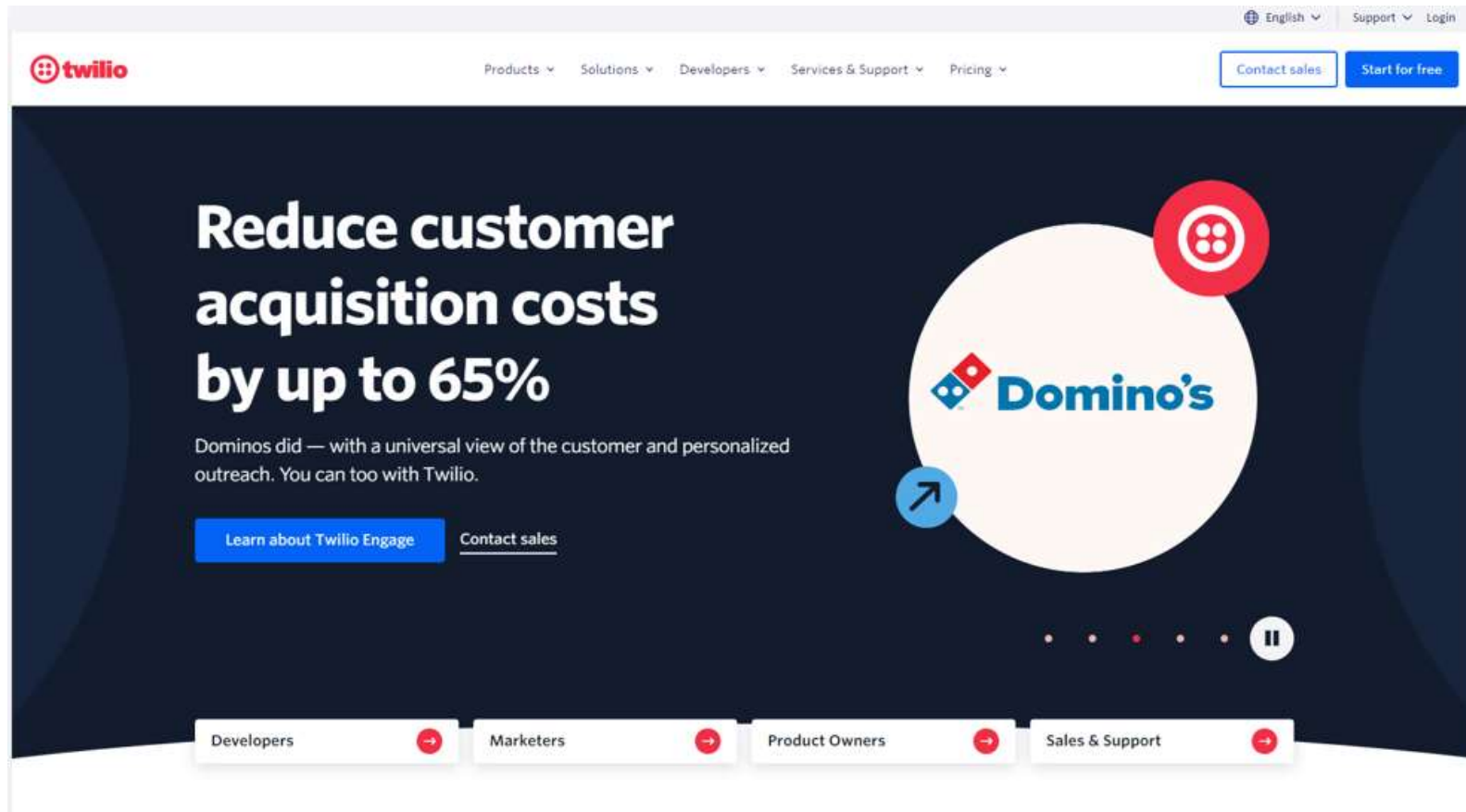
적용 예시



### 3. 결제 시 문자발송 기능

Twilio 의 API 를 이용

<https://www.twilio.com/>



- Twilio 홈페이지 사진

# 3. 결제 시 문자발송 기능

The screenshot displays the Twilio Console interface for a trial account. The top navigation bar includes the Twilio logo, account name 'My first Twilio account', a trial status indicator 'Trial: \$4.802 Upgrade', a search bar, and user account information 'Account Billing JS Jeon'. The left sidebar contains navigation links for 'View all accounts', 'Develop', 'Monitor', and various product categories like Phone Numbers, Messaging, Studio, Functions and Assets, and Voice. The main content area features a congratulatory message about receiving a Twilio phone number (+12706123495) with a 'Next' button. Below this is the 'Account Info' section, which is highlighted with a red box and contains three fields: 'Account SID', 'Auth Token' (masked with dots), and 'My Twilio phone number' (+12706123495). Each field has a copy icon. A warning message states: 'Always store your token securely to protect your account. Learn more'. To the right of the 'Account Info' section is a 'Helpful links' section with links to 'How does Twilio work?', 'SMS Quickstart guides', and 'Support help center'. At the bottom of the main content area, a note mentions the trial account's limitations: 'You are on a trial account. You can only send messages and make calls to verified phone numbers. Learn more about your trial account'. The browser's address bar at the bottom shows the URL 'https://www.twilio.com/console/projects/summary'.

Console  
My first Twilio account  
Trial: \$4.802 Upgrade  
Jump to...  
Account Billing JS Jeon

View all accounts  
Develop Monitor

› # Phone Numbers  
› Messaging  
› Studio  
› Functions and Assets  
› Voice

Explore Products +

Docs and Support

⏪  
https://www.twilio.com/console/projects/summary

🔔 Congratulations! You've got a Twilio phone number!  
+12706123495  
Next

▼ Account Info

Account SID  
Auth Token  
My Twilio phone number  
+12706123495

⚠️ Always store your token securely to protect your account. [Learn more](#)

You are on a trial account. You can only send messages and make calls to verified phone numbers. [Learn more about your trial account](#)

▼ Helpful links

[How does Twilio work?](#)  
Understand how to use Twilio in a 2-minute video.

[SMS Quickstart guides](#)  
Learn the basics of Twilio Messaging APIs.

[Support help center](#)  
Troubleshoot common issues.

### 3. 결제 시 문자발송 기능

---

```
DateTime nowDate = DateTime.Now;
    string date = nowDate.Year.ToString() + "-" + nowDate.Month.ToString() + "-" + nowDate.Day.ToString() + "/" + nowDate.Hour.ToString() +
":" + nowDate.Minute.ToString();
    string body = "";
    string firstr = "";
    string secstr = "";
    MainForm mainForm = new MainForm();
    mainForm.Show();
    foreach (Menuclass item in MenuForm.orderlist)
    {
        MainForm.InsertOrderHistory(item.coffeename,item.coffeeprice,item.coffeenum,item.shotnum,item.sugarnum,item.hazlenum,item.caranum,
item.coffeesize,item.coffeetemperature,date);
        firstr = $"메뉴명 : {item.coffeename} [{item.coffeesize}] [{item.coffeetemperature}] 가격 : {item.coffeeprice} / [{date}]";
        secstr = $"수량 : {item.coffeenum} / 샷 : {item.shotnum} / 설탕 : {item.sugarnum} / 헤이즐 : {item.hazlenum} / 카라멜 : {item.caranum}";
        body += "[ " + firstr+secstr + " ] \n";
    }

    const string accountSid = "11111111111111111111111111111111";
    const string authToken = "11111111111111111111111111111111";
    TwilioClient.Init(accountSid, authToken);

    var message = MessageResource.Create(
        body: body,
        from: new Twilio.Types.PhoneNumber("+12706123495"),
        to: new Twilio.Types.PhoneNumber("+8201036705235"));
    Console.WriteLine(message.Sid);
```

---

### 3. 결제 시 문자발송 기능

---

Sent from

[Web발신]

Sent from your Twilio trial account -

[메뉴명 : 카푸치노 [Tall] [Ice] 가격 :  
5100 / [2023-6-11/22:43]수량 : 1 /  
샷 : 1 / 설탕 : 1 / 헤이즐 : 1 / 카라멜 : 1]

[메뉴명 : 아인슈페너 [Grande] [Hot]  
가격 : 5000 / [2023-6-11/22:43]수  
량 : 1 / 샷 : 1 / 설탕 : 1 / 헤이즐 : 0 / 카라  
멜 : 0]

[메뉴명 : 아메리카노 [Venti] [Hot] 가  
격 : 5300 / [2023-6-11/22:43]수량 :  
1 / 샷 : 1 / 설탕 : 0 / 헤이즐 : 1 / 카라멜 :  
1]

문자발송 예시

## 4. 출처

---

<https://velog.io/@ansalstmd/C-%EC%95%8C%EC%95%84%EA%B0%80%EA%B8%B0-4-1.-C-%ED%82%A4%EC%98%A4%EC%8A%A4%ED%81%AC-%EB%8F%84%EC%A0%84>

<https://codenet.tistory.com/25>

ai.com

<https://wrt.n.ai/>

<https://github.com/JeonSangHoon0711/cafe-order-system>

twilio.com

[https://www.flaticon.com/kr/free-icon/coffee\\_4721028?term=%EC%BB%A4%ED%94%BC&page=1&position=90&origin=tag&related\\_id=4721028](https://www.flaticon.com/kr/free-icon/coffee_4721028?term=%EC%BB%A4%ED%94%BC&page=1&position=90&origin=tag&related_id=4721028)

이디야키오스크, 뽕다방키오스크