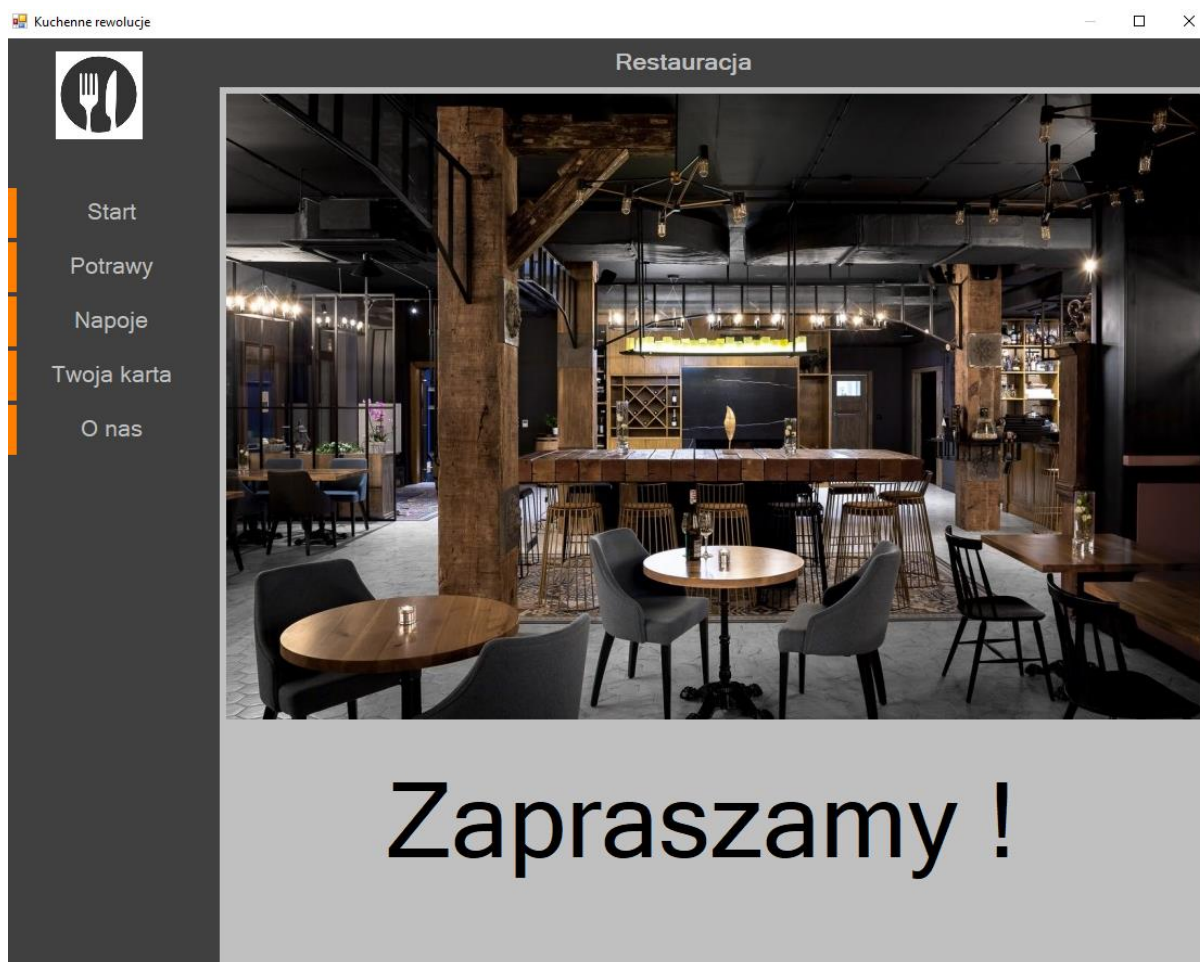


Aplikacja restauracyjna

Wykonał: Dawid Krasucki

Data:02.02.2021

Instrukcja obsługi



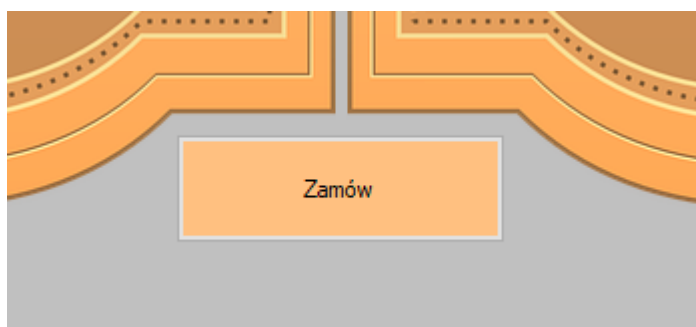
Po aplikacji przemieszczamy się za pomocą guzików po lewej stronie. Załączony obraz to zakładka **start**.



Potrawy – tutaj zamawiamy jedzenie zaznaczając interesującą nas pozycję.



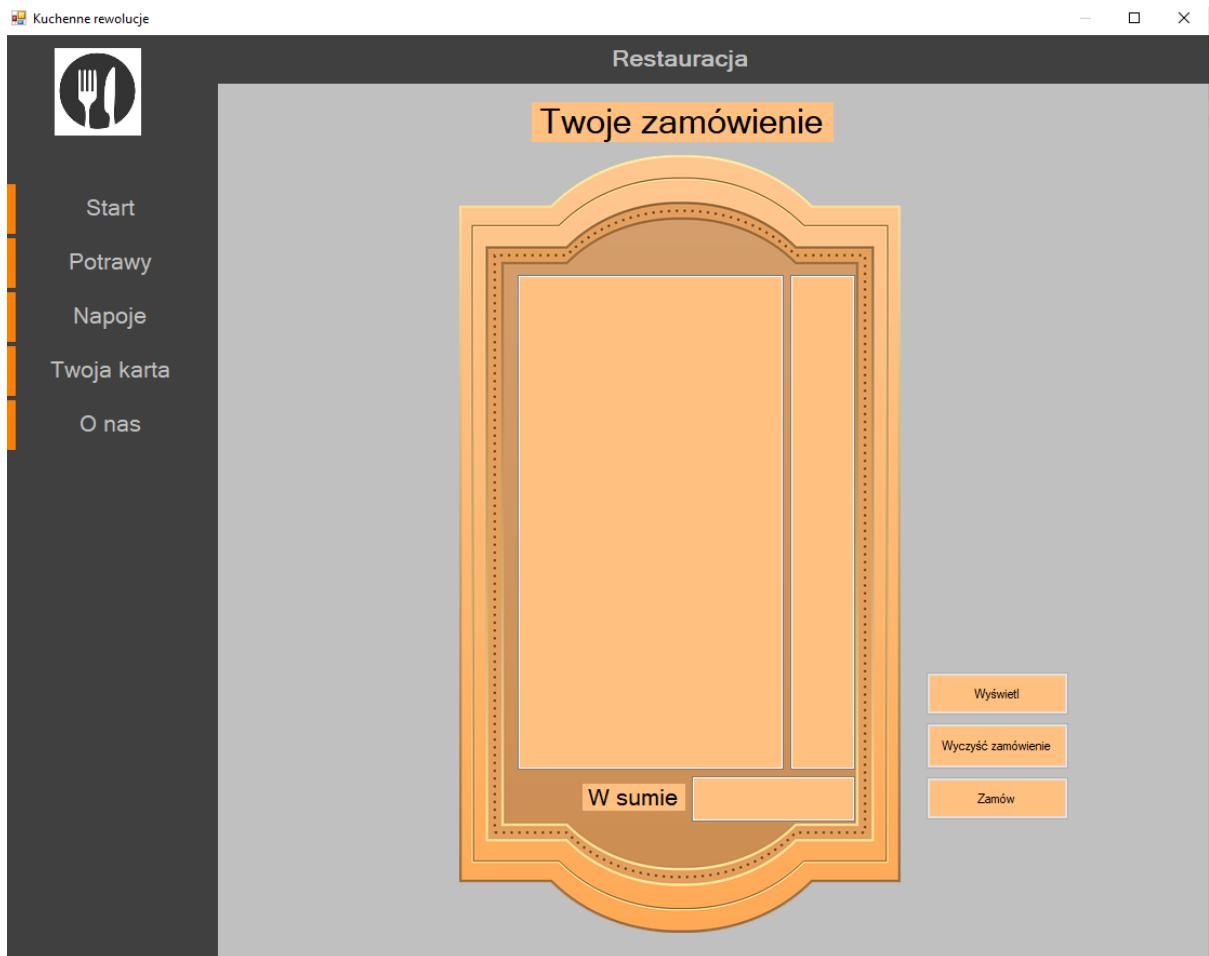
Następnie wybieramy ilość.



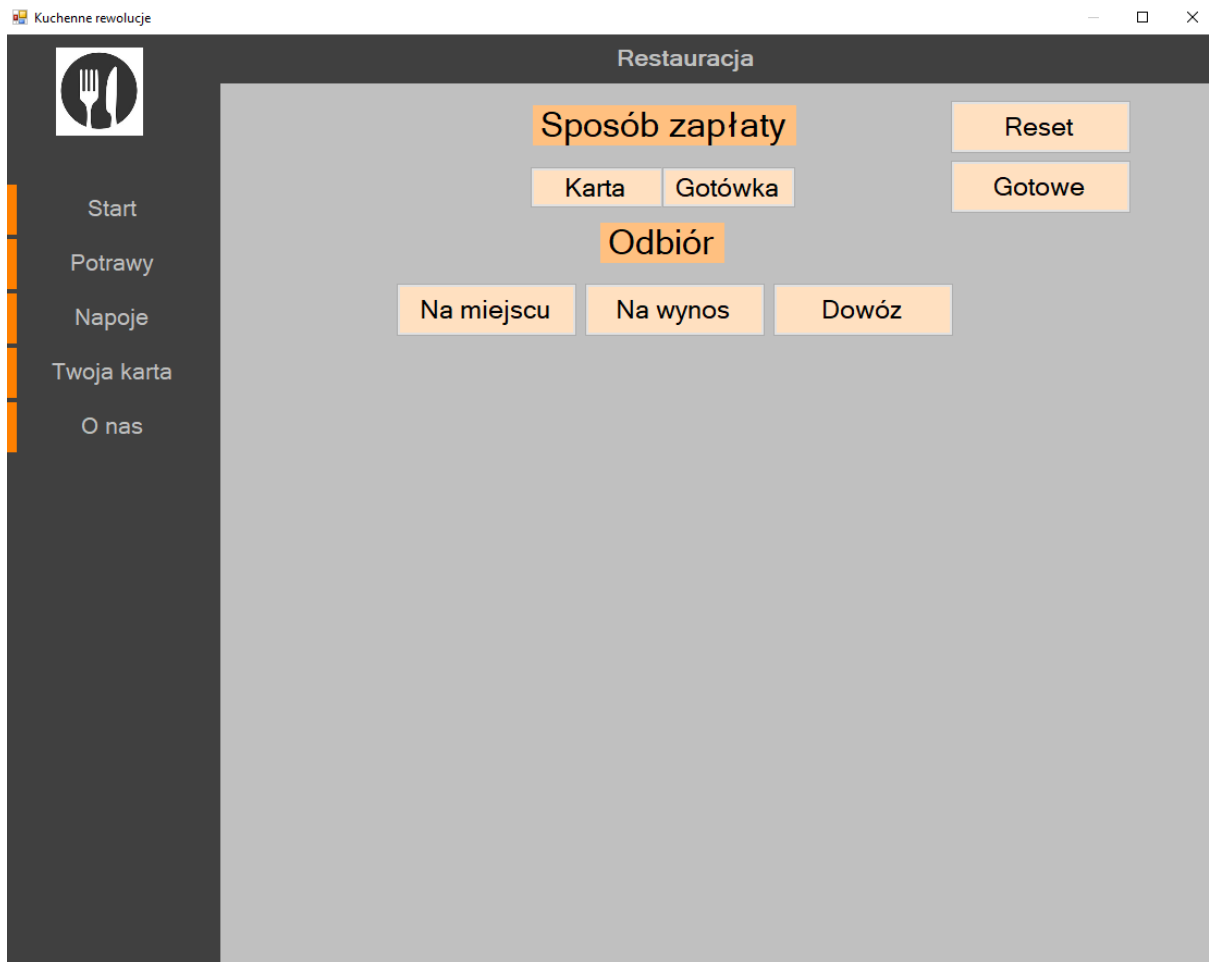
I zamawiamy !



Podobnie dzieję się w zakładce **"Napoje"**



"Twoja karta" to miejsce gdzie można sprawdzić swoje aktualne zamówienie (guzik wyświetl po prawej stronie) oraz w razie czego je skasować. Jeśli klikniemy **"Zamów"** zamówienia nie będzie można już cofnąć i zostaniemy przeniesieni do okna gdzie finalizujemy zamówienie.



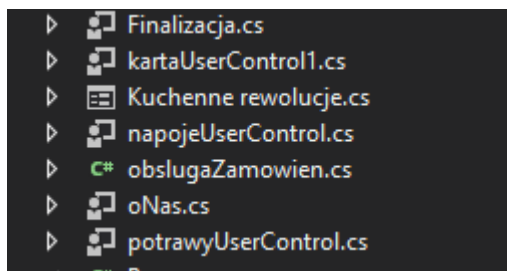
Po kliknięciu **“Dowóz”** wyświetlą się dodatkowe okna na wprowadzenie adresu. Jeśli będzie niepoprawny lub niepełny aplikacja nas o tym powiadomi. **“Reset”** jak nazwa wskazuje resetuje ale tylko te okno (w razie jeśli wprowadzimy złe dane). Na koniec klikamy **“Gotowe”**.



“O nas” to zakładka gdzie dowiemy się o aktualnych promocjach i ciekawych informacjach.

Kod źródłowy

Jest podzielony na:



Z czego najważniejsza jest klasa “**obsługaZamowien**” to w niej mamy metody i listy tworzące nasze zamówienie.

W ten sposób zbieramy zamówienie.

```
10 references
public void Wirtualny_Kelner(string potrawa, int cena, CheckBox cb, TextBox pokazIlosc, Button plus, Button minus)
{
    if (!zamowienie.ContainsKey(potrawa))
    {
        zamowienie.Add(potrawa, cena);
        pokazIlosc.Visible = true;
        plus.Visible = true;
        minus.Visible = true;
    }

    if (cb.Checked == false)
    {
        zamowienie.Remove(potrawa);
        pokazIlosc.Visible = false;
        plus.Visible = false;
        minus.Visible = false;
    }
}
```

A tak sprawdzamy ilość.


```

10 references
public void ZbierzIlosc(string potrawa, int ilosc)
{
    if (!iloscPotraw.ContainsKey(potrawa))
    {
        iloscPotraw.Add(potrawa, ilosc);
    }
    else if (zamowienie.ContainsKey(potrawa))
    {
        iloscPotraw.Remove(potrawa);
        iloscPotraw.Add(potrawa, ilosc);
    }
}

```

Podliczanie ceny i sprawdzanie czy zastosować promocję.
Cena jest zaokrąglana.

```

public void czySaZamowienia()
{
    if (zamowienie.Count == 0)
    {
        MessageBox.Show("Nic nie wybrałeś", "", MessageBoxButtons.OK);
    }
    calaIlosc = "";
    caleZamowienie = "";
    cena = 0;

    for(int i = 0; i < zamowienie.Count; i++)
    {
        temp = zamowienie.ElementAt(i).Key;
        caleZamowienie += zamowienie.ElementAt(i).Key + " " + zamowienie.ElementAt(i).Value + " zł \r\n";
        if (iloscPotraw.ContainsKey(temp))
        {
            calaIlosc += iloscPotraw[temp] + "\r\n";
            cena += iloscPotraw[temp] * zamowienie.ElementAt(i).Value;
        }

        if(cena > 50)
        {
            cena = (cena * 0.9) ;
        }
    }
    cena = Math.Ceiling(cena);
}

```

Po dokonaniu zamówienia trafia one do pliku tekstowego.

```

1 reference
public void Wydrukuj()
{
    FileInfo f = new FileInfo(sciezka_do_zamowien);
    sciezka_do_zamowien = f.FullName;
    using (StreamWriter sw = File.AppendText(sciezka_do_zamowien))
    {
        for (int i = 0; i < zamowienie.Count; i++)
        {
            temp = zamowienie.ElementAt(i).Key;
            caleZamowienie += zamowienie.ElementAt(i).Key + " " + zamowienie.ElementAt(i).Value + " zł \r\n";

            if (iloscPotraw.ContainsKey(temp))
            {
                sw.WriteLine(zamowienie.ElementAt(i).Key + " " + zamowienie.ElementAt(i).Value + " zł x" + iloscPotraw[temp]);
            }
        }
    }
}

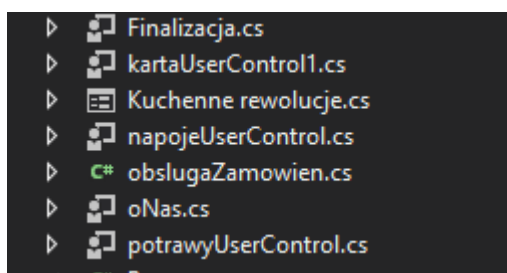
```

W tych zmiennych przechowywane są wszystkie informacje.

```

public Dictionary<string, int> zamowienie = new Dictionary<string, int>();
public Dictionary<string, int> iloscPotraw = new Dictionary<string, int>();
public bool gotowe = false;
public static obslugaZamowien podsumowanieZamowienia = new obslugaZamowien();
public string caleZamowienie;
public string calaIlosc;
string temp;
public double suma;
public string sciezka_do_zamowien = @"zamowienia\\zamowienia.txt";
public string sciezka_do_kwoty = @"ZAROBKI RESTAURACJI\\kwota.txt";
public double cena;

```



“Kuchenne rewolucje” odpowiedzialne jest za wyświetlanie prawidłowego okienka UserControl.

```

1 reference
private void b_potrawy_Click(object sender, EventArgs e)
{
    pb_start.Visible = false;
    potrawyUserControl.Show();
    napojeUserControl.Hide();
    kartaUserControl1.Hide();
    oNas1.Hide();
}

```

“Potrawy i napoje” to głównie wywoływanie metod z “obsługaZamowien”.

```

1 reference
private void cb_zapiekanka_CheckedChanged(object sender, EventArgs e)
{
    obslugaZamowien.podsumowanieZamowienia.Wirtualny_Kelner(l_zapiekanka_nazwa.Text, 15, cb_zapiekanka, tb_zapiekana_ilosc, b_plus_zapiekanka, b
}

```

Podobnie ma to miejsce w “kartaUserControl”.

```

public kartaUserControl()
{
    InitializeComponent();
}

// public Button b_Koniec { get { return b_zamow; } }
0 references
public Button b_Wyczysc { get { return b_wyczysc_zamowienie; } }

1 reference
private void b_podsumuj_Click(object sender, EventArgs e)
{
    tb_podsumowanie.Text = obslugaZamowien.podsumowanieZamowienia.caleZamowienie;
    tb_podsumowanie_ilosc.Text = obslugaZamowien.podsumowanieZamowienia.calaIlosc;
    tb_suma.Text = obslugaZamowien.podsumowanieZamowienia.cena.ToString() + " zł";
}

```

I “Finalizacja” to głównie obsługa guzików i wyłapywanie błędów.

```

obslugaZamowien.podsumowanieZamowienia.gotowe = false;
if (bool_karta == false && bool_gotowka == false)
{
    MessageBox.Show("Wybierz sposób zapłaty", "", MessageBoxButtons.OK);
}
else if (bool_dowoz == false && bool_na_miejscu == false && bool_na_wynos == false)
{
    MessageBox.Show("Wybierz odbiór", "", MessageBoxButtons.OK);
}

```

1 reference

```
private void b_dowoz_Click(object sender, EventArgs e)
{
    b_dowoz.BackColor = Color.Red;
    bool_dowoz = true;
    b_nawynos.Enabled = false;
    b_namiejscu.Enabled = false;
    tb_ulica.Visible = true;
    tb_numer.Visible = true;
    tb_miasto.Visible = true;
    l_ulica.Visible = true;
    l_numertelefonu.Visible = true;
    l_miasto.Visible = true;
}
```