



# **C# w aplikacjach desktopowych**

Zbigniew Kluczkowski

# C# w oknie



Aplikacje okienkowe to programy z interfejsem graficznym (GUI).



Umożliwiają interakcję użytkownika za pomocą przycisków, formularzy, okien dialogowych.



C# obsługuje tworzenie GUI za pomocą:



Windows Forms (WinForms)



WPF (Windows Presentation Foundation)



MAUI (następca Xamarin.Forms)

# Kontrolki i zdarzenia w języku C#

Kontrolka	Najczęstsze zdarzenia	Przykład kodu
<b>Button</b>	Click	<pre>private void button1_Click(object sender, EventArgs e) { MessageBox.Show("Kliknięto przycisk!"); }</pre>
<b>TextBox</b>	TextChanged, KeyPress	<pre>private void textBox1_TextChanged(object sender, EventArgs e) { label1.Text = textBox1.Text; } private void textBox1_KeyPress(object sender, KeyPressEventArgs e) { if (!char.IsDigit(e.KeyChar) &amp;&amp; !char.IsControl(e.KeyChar)) e.Handled = true; }</pre>
<b>CheckBox</b>	CheckedChanged	<pre>private void checkBox1_CheckedChanged(object sender, EventArgs e) { if (checkBox1.Checked) MessageBox.Show("Zaznaczono!"); else MessageBox.Show("Odznaczono!"); }</pre>
<b>RadioButton</b>	CheckedChanged	<pre>private void radioButton1_CheckedChanged(object sender, EventArgs e) { if (radioButton1.Checked) label1.Text = "Wybrano opcję 1"; }</pre>
<b>ComboBox</b>	SelectedIndexChanged	<pre>private void comboBox1_SelectedIndexChanged(object sender, EventArgs e) { label1.Text = "Wybrałeś: " + comboBox1.SelectedItem; }</pre>
<b>ListBox</b>	SelectedIndexChanged	<pre>private void listBox1_SelectedIndexChanged(object sender, EventArgs e) { label1.Text = "Wybrano: " + listBox1.SelectedItem; }</pre>
<b>PictureBox</b>	Click	<pre>private void pictureBox1_Click(object sender, EventArgs e) { MessageBox.Show("Kliknięto obrazek!"); }</pre>
<b>ProgressBar</b>	(sterowane kodem – Value)	<pre>progressBar1.Value = 50; // ustawia pasek na 50%</pre>

# Kontrolki i zdarzenia w języku C#

Kontrolka	Najczęstsze zdarzenia	Przykład kodu
<b>TrackBar</b>	Scroll, ValueChanged	<pre>private void trackBar1_Scroll(object sender, EventArgs e) { label1.Text = trackBar1.Value.ToString(); }</pre>
<b>DateTimePicker</b>	ValueChanged	<pre>private void dateTimePicker1_ValueChanged (object sender, EventArgs e) { label1.Text = dateTimePicker1.Value.ToShort DateString(); }</pre>
<b>Form</b>	Load, FormClosing	<pre>private void Form1_Load(object sender, EventArgs e) { MessageBox.Show("Formularz załadowany!"); } private void Form1_FormClosing(object sender, FormClosingEventArgs e) { MessageBox.Show("Zamykanie aplikacji!"); }</pre>
<b>Timer</b>	Tick	<pre>private void timer1_Tick(object sender, EventArgs e) { label1.Text = DateTime.Now.ToLongTimeStrin g(); }</pre>
<b>DataGridView</b>	CellClick, CellValueChanged	<pre>private void dataGridView1_CellClick(object sender, DataGridViewCellEventArgs e) { MessageBox.Show("Klik w komórkę: " + e.RowIndex + ", " + e.ColumnIndex); }</pre>

# Przykładowe kontrolki i zdarzenia

```
using System;
using System.Windows.Forms;

namespace TestKontrolki
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            // BUTTON
            Button btn = new Button();
            btn.Text = "Kliknij mnie";
            btn.Location = new
            System.Drawing.Point(20, 20);
            btn.Click += Btn_Click;
            this.Controls.Add(btn);

            // TEXTBOX
            TextBox txt = new TextBox();
            txt.Location = new
            System.Drawing.Point(20, 60);
            txt.TextChanged += Txt_TextChanged;
            this.Controls.Add(txt);

            // LABEL
            Label lbl = new Label();
            lbl.Text = "Tutaj pojawi się tekst";
            lbl.Location = new
            System.Drawing.Point(20, 100);
            lbl.AutoSize = true;
            lbl.Name = "lblInfo";
            this.Controls.Add(lbl);

            // CHECKBOX
            CheckBox chk = new CheckBox();
            chk.Text = "Zaznacz mnie";
            chk.Location = new
            System.Drawing.Point(20, 140);
            chk.CheckedChanged +=
            Chk_CheckedChanged;
            this.Controls.Add(chk);

            // RADIOBUTTON
            RadioButton rdo = new RadioButton();
            rdo.Text = "Opcja 1";
            rdo.Location = new
            System.Drawing.Point(20, 180);
            rdo.CheckedChanged +=
            Rdo_CheckedChanged;
            this.Controls.Add(rdo);

            // COMBOBOX
            ComboBox cmb = new ComboBox();
            cmb.Location = new
            System.Drawing.Point(20, 220);
            cmb.Items.AddRange(new string[] { "C#",
            "Java", "Python" });
            cmb.SelectedIndexChanged +=
            Cmb_SelectedIndexChanged;
            this.Controls.Add(cmb);

            private void Btn_Click(object sender,
            EventArgs e)
            {
                MessageBox.Show("Kliknięto przycisk!");
            }

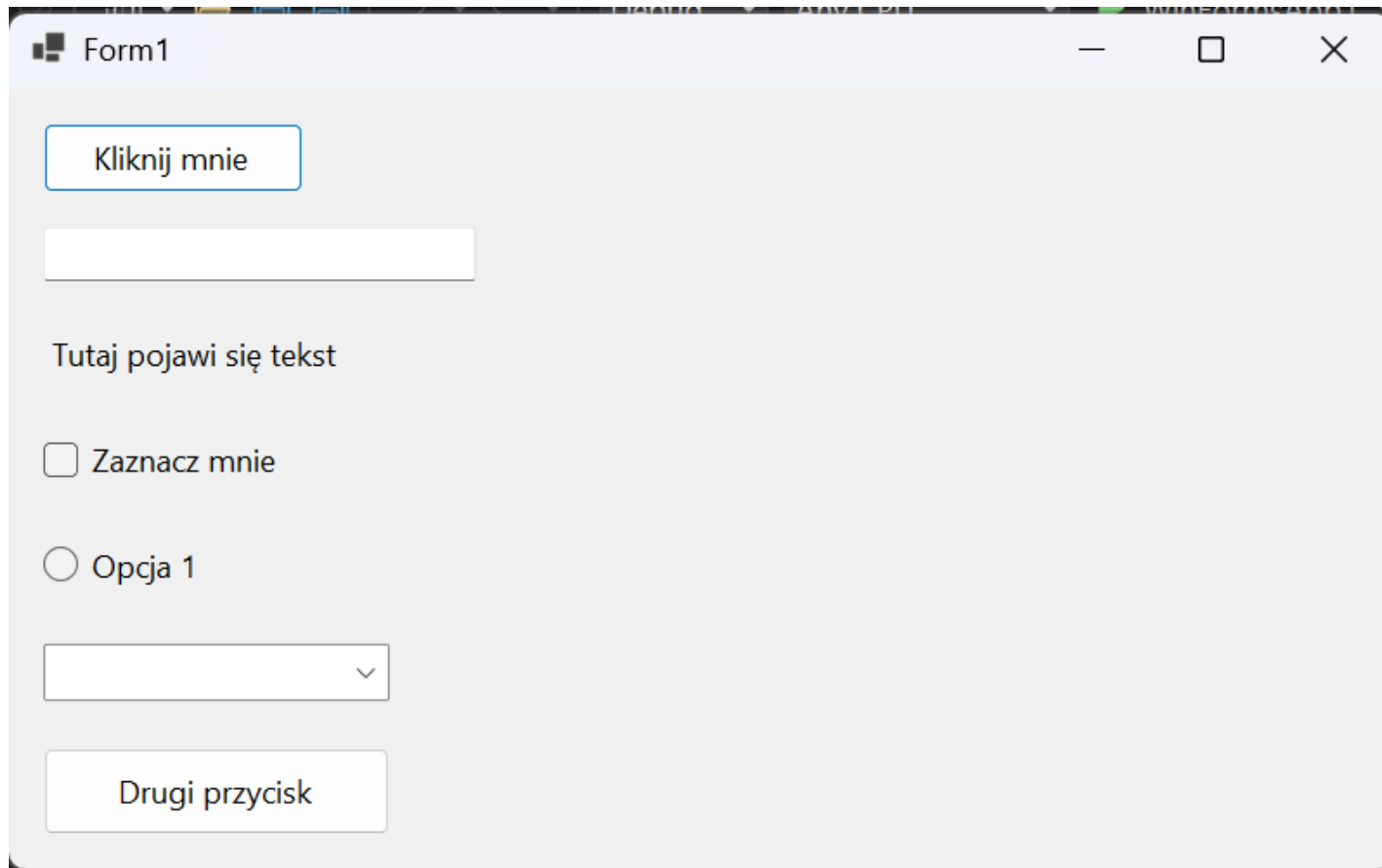
            private void Txt_TextChanged(object sender,
            EventArgs e)
            {
                Label lbl = this.Controls["lblInfo"] as Label;
                if (lbl != null)
                    lbl.Text = (sender as TextBox).Text;
            }

            private void Chk_CheckedChanged(object
            sender, EventArgs e)
            {
                CheckBox chk = sender as CheckBox;
                MessageBox.Show(chk.Checked ?
                "Zaznaczono!" : "Odznaczono!");
            }

            private void Rdo_CheckedChanged(object
            sender, EventArgs e)
            {
                RadioButton rdo = sender as RadioButton;
                if (rdo.Checked)
                    MessageBox.Show("Wybrano opcję 1!");
            }

            private void Cmb_SelectedIndexChanged(object
            sender, EventArgs e)
            {
                ComboBox cmb = sender as ComboBox;
                MessageBox.Show("Wybrano: " +
                cmb.SelectedItem);
            }
        }
    }
}
```

# Przykładowe kontrolki i zdarzenia



The image shows a screenshot of a Windows application window titled "Form1". The window contains several user interface controls:

- A button labeled "Kliknij mnie" (Click me) with a blue border.
- A text box.
- A label "Tutaj pojawi się tekst" (Text will appear here).
- A checkbox labeled "Zaznacz mnie" (Check me).
- A radio button labeled "Opcja 1" (Option 1).
- A dropdown menu.
- A button labeled "Drugi przycisk" (Second button).

# Kod prostego okna z kontrolkami

```
using System;
using System.Windows.Forms;

namespace TestKontrolkek
{
    public partial class Form1 : Form
    {
        // Pola dla kontrolkek (zapamiętujemy je)
        private Label lblInfo;
        private TextBox txtBox;

        public Form1()
        {
            InitializeComponent();

            // BUTTON
            Button btn = new Button();
            btn.Text = "Kliknij mnie";
            btn.Location = new
            System.Drawing.Point(20, 20);
            btn.Click += Btn_Click;
            this.Controls.Add(btn);

            // TEXTBOX
            txtBox = new TextBox();
            txtBox.Location = new
            System.Drawing.Point(20, 60);
            txtBox.TextChanged += Txt_TextChanged;
            this.Controls.Add(txtBox);

            // LABEL
            lblInfo = new Label();
            lblInfo.Text = "Tutaj pojawi się tekst";
            lblInfo.Location = new
            System.Drawing.Point(20, 100);
            lblInfo.AutoSize = true;

            this.Controls.Add(lblInfo);

            // CHECKBOX
            CheckBox chk = new CheckBox();
            chk.Text = "Zaznacz mnie";
            chk.Location = new
            System.Drawing.Point(20, 140);
            chk.CheckedChanged +=
            Chk_CheckedChanged;
            this.Controls.Add(chk);

            // RADIOBUTTON
            RadioButton rdo = new RadioButton();
            rdo.Text = "Opcja 1";
            rdo.Location = new
            System.Drawing.Point(20, 180);
            rdo.CheckedChanged +=
            Rdo_CheckedChanged;
            this.Controls.Add(rdo);

            // COMBOBOX
            ComboBox cmb = new ComboBox();
            cmb.Location = new
            System.Drawing.Point(20, 220);
            cmb.Items.AddRange(new string[] { "C#",
            "Java", "Python" });
            cmb.SelectedIndexChanged +=
            Cmb_SelectedIndexChanged;
            this.Controls.Add(cmb);
        }

        // Zdarzenia
        private void Btn_Click(object sender,
            EventArgs e)
        {
            MessageBox.Show("Kliknięto przycisk!");
        }

        private void Txt_TextChanged(object sender,
            EventArgs e)
        {
            lblInfo.Text = txtBox.Text;
        }

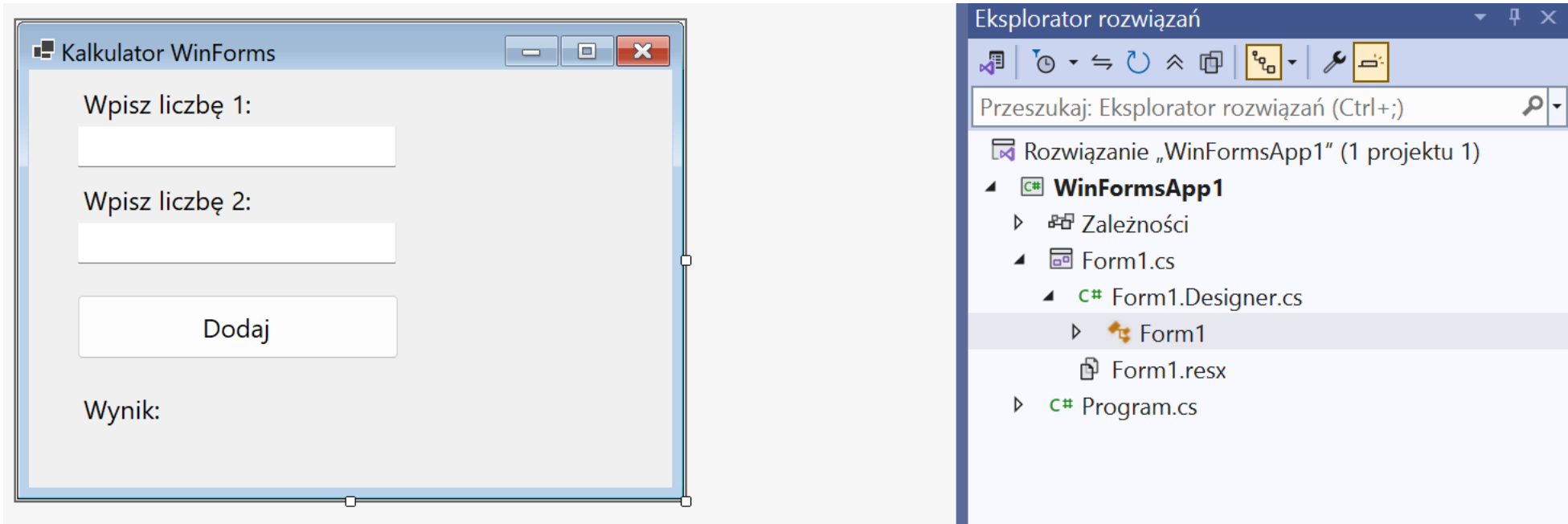
        private void Chk_CheckedChanged(object sender,
            EventArgs e)
        {
            CheckBox chk = sender as CheckBox;
            MessageBox.Show(chk.Checked ?
            "Zaznaczono!" : "Odznaczono!");
        }

        private void Rdo_CheckedChanged(object sender,
            EventArgs e)
        {
            RadioButton rdo = sender as RadioButton;
            if (rdo.Checked)
                MessageBox.Show("Wybrano opcję 1!");
        }

        private void Cmb_SelectedIndexChanged(object sender,
            EventArgs e)
        {
            ComboBox cmb = sender as ComboBox;
            MessageBox.Show("Wybierałeś: " +
            cmb.SelectedItem);
        }
    }
}
```

# Windows Forms

- Starsza, ale bardzo popularna technologia.
- Szybki rozwój aplikacji z prostym interfejsem.



Przybornik

Przeszukaj przybornik

Wszystkie kontrolki Windows Forms

Typowe kontrolki

- Wskaźnik
- Button
- CheckBox
- CheckedListBox
- ComboBox
- DateTimePicker
- Label
- LinkLabel
- ListBox
- ListView
- MaskedTextBox
- MonthCalendar
- NotifyIcon

- NumericUpDown
- PictureBox
- ProgressBar

RadioButton

RichTextBox


TextBox

skalowanie na ekranie głównym jest ustawione na 150%. Rozważ otwarcie Projektanta WinForms w trybie be

Form1

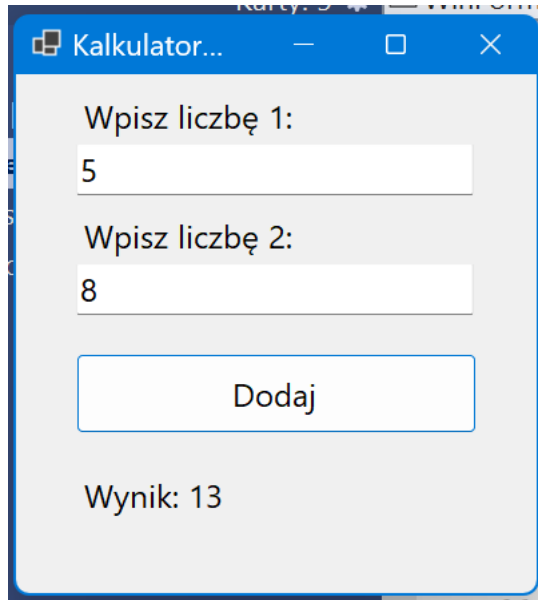
## Okno z przybornikiem

# Podpinanie zdarzeń

- Kliknij prawym przyciskiem kontrolkę → **Properties** → zakładka  **Events**
- Wybierz zdarzenie, np. Click dla przycisku
- Visual Studio automatycznie wygeneruje metodę np.:

```
private void button1_Click(object sender, EventArgs e)
{
    MessageBox.Show("Kliknięto przycisk!");
}
```

# Przykładowy Forms - kalkulator



Kalkulator...

Wpisz liczbę 1:  
5

Wpisz liczbę 2:  
8

Dodaj

Wynik: 13

Form.cs

```
using System.Windows.Forms;
namespace KalkulatorWinForms
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void btnDodaj_Click(object sender, EventArgs e)
        {
            try
            {
                double a = double.Parse(txtLiczba1.Text);
                double b = double.Parse(txtLiczba2.Text);
                double suma = a + b;
                lblWynik.Text = $"Wynik: {suma}";
            }
            catch
            {
                lblWynik.Text = "Błąd danych wejściowych.";
            }
        }
    }
}
```

# Przykładowy Forms - kalkulator

```
namespace KalkulatorWinForms
{
    partial class Form1
    {
        private System.ComponentModel.IContainer components = null;

        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }
    }
}
```

#region Windows Form Designer generated code

```
private void InitializeComponent()
{
    this.txtLiczba1 = new System.Windows.Forms.TextBox();
    this.txtLiczba2 = new System.Windows.Forms.TextBox();
    this.btnDodaj = new System.Windows.Forms.Button();
    this.lblWynik = new System.Windows.Forms.Label();
    this.lblLiczba1 = new System.Windows.Forms.Label();
    this.lblLiczba2 = new System.Windows.Forms.Label();
    this.SuspendLayout();
    //
    // lblLiczba1
    //
    this.lblLiczba1.Font = new System.Drawing.Font("Segoe UI", 12F);
    this.lblLiczba1.Location = new System.Drawing.Point(30, 10);
    this.lblLiczba1.Name = "lblLiczba1";
    this.lblLiczba1.Size = new System.Drawing.Size(200, 25);
    this.lblLiczba1.Text = "Wpisz liczbę 1:";
    //
    // txtLiczba1
```

```
//
this.txtLiczba1.Font = new System.Drawing.Font("Segoe UI", 12F);
this.txtLiczba1.Location = new System.Drawing.Point(30, 35);
this.txtLiczba1.Name = "txtLiczba1";
this.txtLiczba1.Size = new System.Drawing.Size(200, 29);
this.txtLiczba1.TabIndex = 0;
//
// lblLiczba2
//
this.lblLiczba2.Font = new System.Drawing.Font("Segoe UI", 12F);
this.lblLiczba2.Location = new System.Drawing.Point(30, 70);
this.lblLiczba2.Name = "lblLiczba2";
this.lblLiczba2.Size = new System.Drawing.Size(200, 25);
this.lblLiczba2.Text = "Wpisz liczbę 2:";
//
// txtLiczba2
//
this.txtLiczba2.Font = new System.Drawing.Font("Segoe UI", 12F);
this.txtLiczba2.Location = new System.Drawing.Point(30, 95);
this.txtLiczba2.Name = "txtLiczba2";
this.txtLiczba2.Size = new System.Drawing.Size(200, 29);
this.txtLiczba2.TabIndex = 1;
//
// btnDodaj
//
this.btnDodaj.Font = new System.Drawing.Font("Segoe UI", 12F);
this.btnDodaj.Location = new System.Drawing.Point(30, 140);
this.btnDodaj.Name = "btnDodaj";
this.btnDodaj.Size = new System.Drawing.Size(200, 40);
this.btnDodaj.TabIndex = 2;
this.btnDodaj.Text = "Dodaj";
this.btnDodaj.UseVisualStyleBackColor = true;
this.btnDodaj.Click += new System.EventHandler(this.btnDodaj_Click);
//
// lblWynik
//
```

```
this.lblWynik.Font = new System.Drawing.Font("Segoe UI", 12F);
this.lblWynik.Location = new System.Drawing.Point(30, 200);
this.lblWynik.Name = "lblWynik";
this.lblWynik.Size = new System.Drawing.Size(300, 30);
this.lblWynik.TabIndex = 3;
this.lblWynik.Text = "Wynik:";
//
// Form1
//
this.AutoScaleMode = new System.Drawing.SizeF(8F, 20F);
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
this.ClientSize = new System.Drawing.Size(400, 260);
this.Controls.Add(this.lblLiczba1);
this.Controls.Add(this.txtLiczba1);
this.Controls.Add(this.lblLiczba2);
this.Controls.Add(this.txtLiczba2);
this.Controls.Add(this.btnDodaj);
this.Controls.Add(this.lblWynik);
this.Name = "Form1";
this.Text = "Kalkulator WinForms";
this.ResumeLayout(false);
this.PerformLayout();
}

#endregion

private System.Windows.Forms.TextBox txtLiczba1;
private System.Windows.Forms.TextBox txtLiczba2;
private System.Windows.Forms.Button btnDodaj;
private System.Windows.Forms.Label lblWynik;
private System.Windows.Forms.Label lblLiczba1;
private System.Windows.Forms.Label lblLiczba2;
}
}
```



Designer.cs

# Przykładowy Forms – lista zadań

ToDo Lista

ugotować obiad

Dodaj

Usuń

zrobić pranie

```
using System;
using System.Windows.Forms;

namespace ToDoListWinForms
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            private void btnDodaj_Click(object sender, EventArgs e)
            {
                string zadanie =
                txtZadanie.Text.Trim();

                if (!string.IsNullOrEmpty(zadanie))
                {
                    listZadan.Items.Add(zadanie);

                    private void btnUsun_Click(object sender, EventArgs e)
                    {
                        while
                        (listZadan.SelectedItems.Count > 0)
                        {
                            listZadan.Items.Remove(listZadan.Selecte
                            dItems[0]);
                        }
                    }
                }
            }

            txtZadanie.Clear();
        }
    }
}
```

# Przykładowy Forms – lista zadań

```
namespace ToDoListWinForms
{
    partial class Form1
    {
        private System.ComponentModel.IContainer components = null;

        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        private void InitializeComponent()
        {
            this.txtZadanie = new System.Windows.Forms.TextBox();
            this.btnAdd = new System.Windows.Forms.Button();
            this.btnUsun = new System.Windows.Forms.Button();
            this.listZadan = new System.Windows.Forms.ListBox();
            this.SuspendLayout();

            //
            // txtZadanie
            //
            this.txtZadanie.Font = new System.Drawing.Font("Segoe UI", 14F);
            this.txtZadanie.Location = new System.Drawing.Point(30, 30); // wyżej
            this.txtZadanie.Name = "txtZadanie";
            this.txtZadanie.Size = new System.Drawing.Size(400, 32);
            this.txtZadanie.TabIndex = 0;

            //
            // btnDodaj
            //
            this.btnAdd.Font = new System.Drawing.Font("Segoe UI", 14F);
            this.btnAdd.Location = new System.Drawing.Point(450, 30);
            this.btnAdd.Name = "btnDodaj";
            this.btnAdd.Size = new System.Drawing.Size(120, 55);
            this.btnAdd.TabIndex = 1;
            this.btnAdd.Text = "Dodaj";
            this.btnAdd.UseVisualStyleBackColor = true;
            this.btnAdd.Click += new System.EventHandler(this.btnAdd_Click);

            //
            // btnUsun
            //
            this.btnUsun.Font = new System.Drawing.Font("Segoe UI", 14F);
            this.btnUsun.Location = new System.Drawing.Point(30, 90); // zwiększo odległość
            this.btnUsun.Name = "btnUsun";
            this.btnUsun.Size = new System.Drawing.Size(120, 55);
            this.btnUsun.TabIndex = 2;
            this.btnUsun.Text = "Usuń";
            this.btnUsun.UseVisualStyleBackColor = true;
            this.btnUsun.Click += new System.EventHandler(this.btnUsun_Click);

            //
            // listZadan
            //
            this.listZadan.Font = new System.Drawing.Font("Segoe UI", 14F);
            this.listZadan.FormatEnabled = true;
            this.listZadan.ItemHeight = 25;
            this.listZadan.Location = new System.Drawing.Point(30, 150); // zwiększo odstęp
            this.listZadan.Name = "listZadan";
            this.listZadan.SelectionMode = System.Windows.Forms.SelectionMode.MultiExtended;
            this.listZadan.Size = new System.Drawing.Size(540, 254);

            this.listZadan.TabIndex = 3;

            //
            // Form1
            //
            this.ClientSize = new System.Drawing.Size(600, 450); // powiększone okno
            this.Controls.Add(this.listZadan);
            this.Controls.Add(this.btnAdd);
            this.Controls.Add(this.txtZadanie);
            this.Name = "Form1";
            this.Text = "ToDo Lista";
            this.ResumeLayout(false);
            this.PerformLayout();
        }

        private System.Windows.Forms.TextBox txtZadanie;
        private System.Windows.Forms.Button btnDodaj;
        private System.Windows.Forms.Button btnUsun;
        private System.Windows.Forms.ListBox listZadan;
    }
}
```

# Obsługa plików w Windows Forms

## Plik Form1.cs

```
using System;
using System.IO;
using System.Windows.Forms;

namespace WinFormsFileExample
{
    public partial class Form1 : Form
    {
        string sciezka = "dane.txt";

        public Form1()
        {
            InitializeComponent();
        }

        private void btnZapisz_Click(object sender,
EventArgs e)
        {
            File.WriteAllText(sciezka, txtTresc.Text);
            MessageBox.Show("Zapisano do pliku.");
        }

        private void btnWczytaj_Click(object sender,
EventArgs e)
        {
            if (File.Exists(sciezka))
                txtTresc.Text = File.ReadAllText(sciezka);
            else
                MessageBox.Show("Plik nie istnieje.");
        }
    }
}
```

# Integracja z bazą danych

## System.Data.SQLite.Core

```
using System;
using System.Data.SQLite;
using System.IO;
using System.Windows.Forms;

namespace WinFormsSQLiteExample
{
    public partial class Form1 : Form
    {
        string dbPath = "baza.db";

        public Form1()
        {
            InitializeComponent();
            UtworzBaze();
            WczytajDane();
        }

        private void UtworzBaze()
        {
            if (!File.Exists(dbPath))
            {
                SQLiteConnection.CreateFile(dbPath);
```

```
                using (var conn = new SQLiteConnection($"Data
                Source={dbPath}"))
                {
                    conn.Open();

                    string sql = "CREATE TABLE Osoby (Id INTEGER
                    PRIMARY KEY AUTOINCREMENT, Imie TEXT)";

                    SQLiteCommand cmd = new SQLiteCommand(sql,
                    conn);

                    cmd.ExecuteNonQuery();
                }
            }

            private void btnDodaj_Click(object sender, EventArgs e)
            {
                using (var conn = new SQLiteConnection($"Data
                Source={dbPath}"))
                {
                    conn.Open();

                    string sql = "INSERT INTO Osoby(Imie)
                    VALUES(@imie)";

                    SQLiteCommand cmd = new SQLiteCommand(sql,
                    conn);

                    cmd.Parameters.AddWithValue("@imie",
                    txtImie.Text);
```

```
                    cmd.ExecuteNonQuery();
                }
            }

            WczytajDane();
        }

        private void WczytajDane()
        {
            listBoxOsoby.Items.Clear();

            using (var conn = new SQLiteConnection($"Data
            Source={dbPath}"))
            {
                conn.Open();

                string sql = "SELECT Imie FROM Osoby";

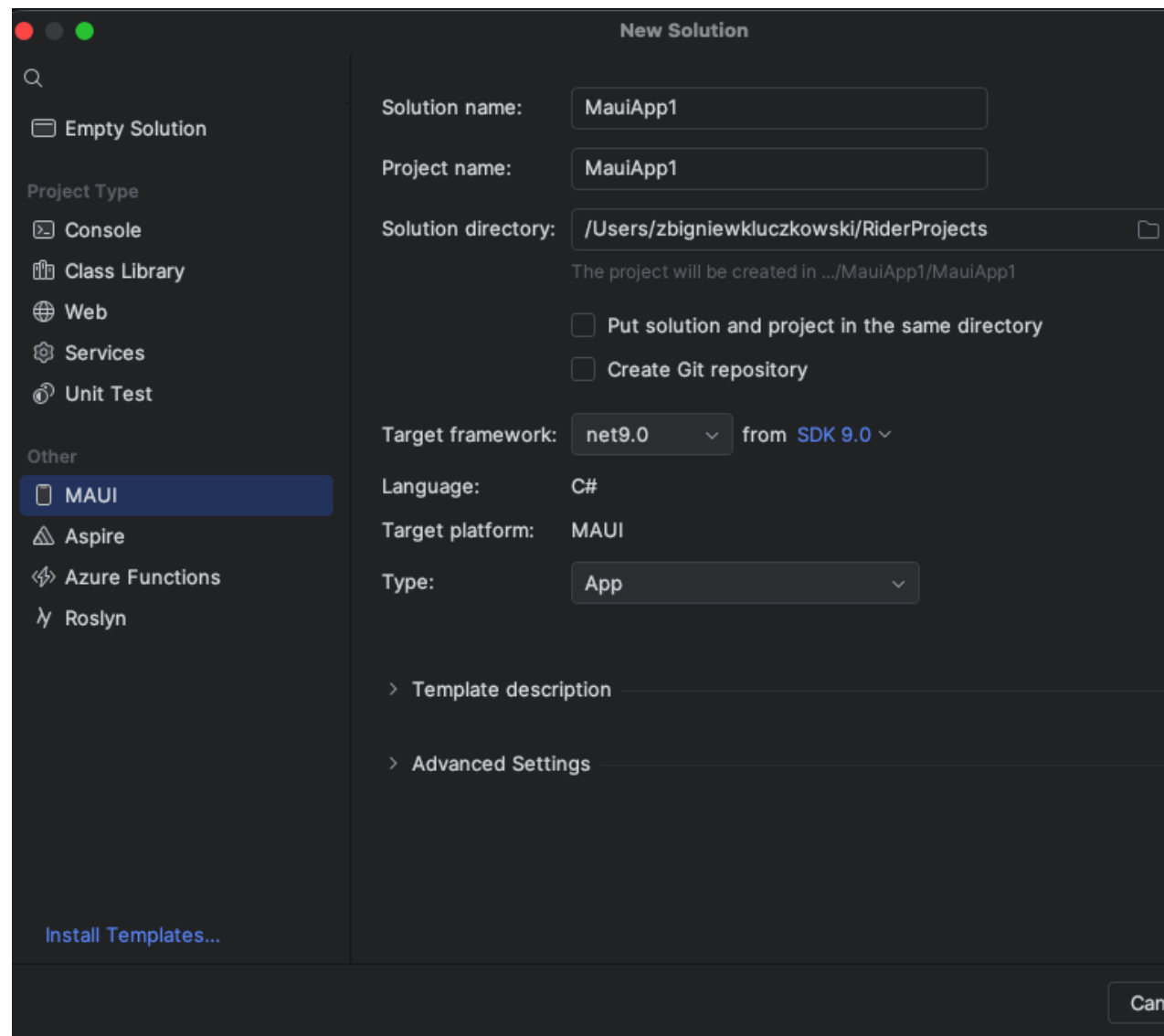
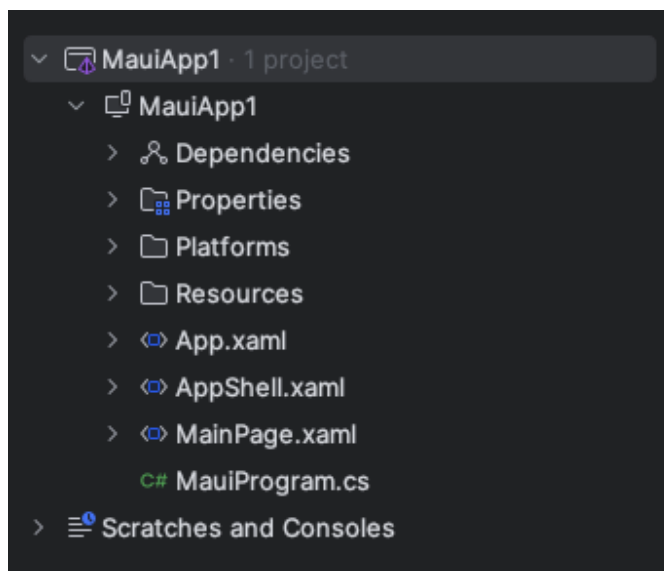
                SQLiteCommand cmd = new SQLiteCommand(sql,
                conn);

                SQLiteDataReader reader = cmd.ExecuteReader();

                while (reader.Read())
                {
                    listBoxOsoby.Items.Add(reader["Imie"].ToString());
                }
            }
        }
    }
}
```

# Rider – alternatywa od JetBrains

Problematiczny w konfiguracji, ale łatwiejszy w obsłudze niż Visual Studio. Doskonale sprawdza się w środowisku innym niż Windows.



# Rider i MacOS – C# na Maku

```
using Microsoft.Maui.Controls;

namespace MauiApp3
{
    public partial class MainPage :
    ContentPage
    {
        public MainPage()
        {
            InitializeComponent(); // musi być,
            żeby widżety z XAML działały
        }

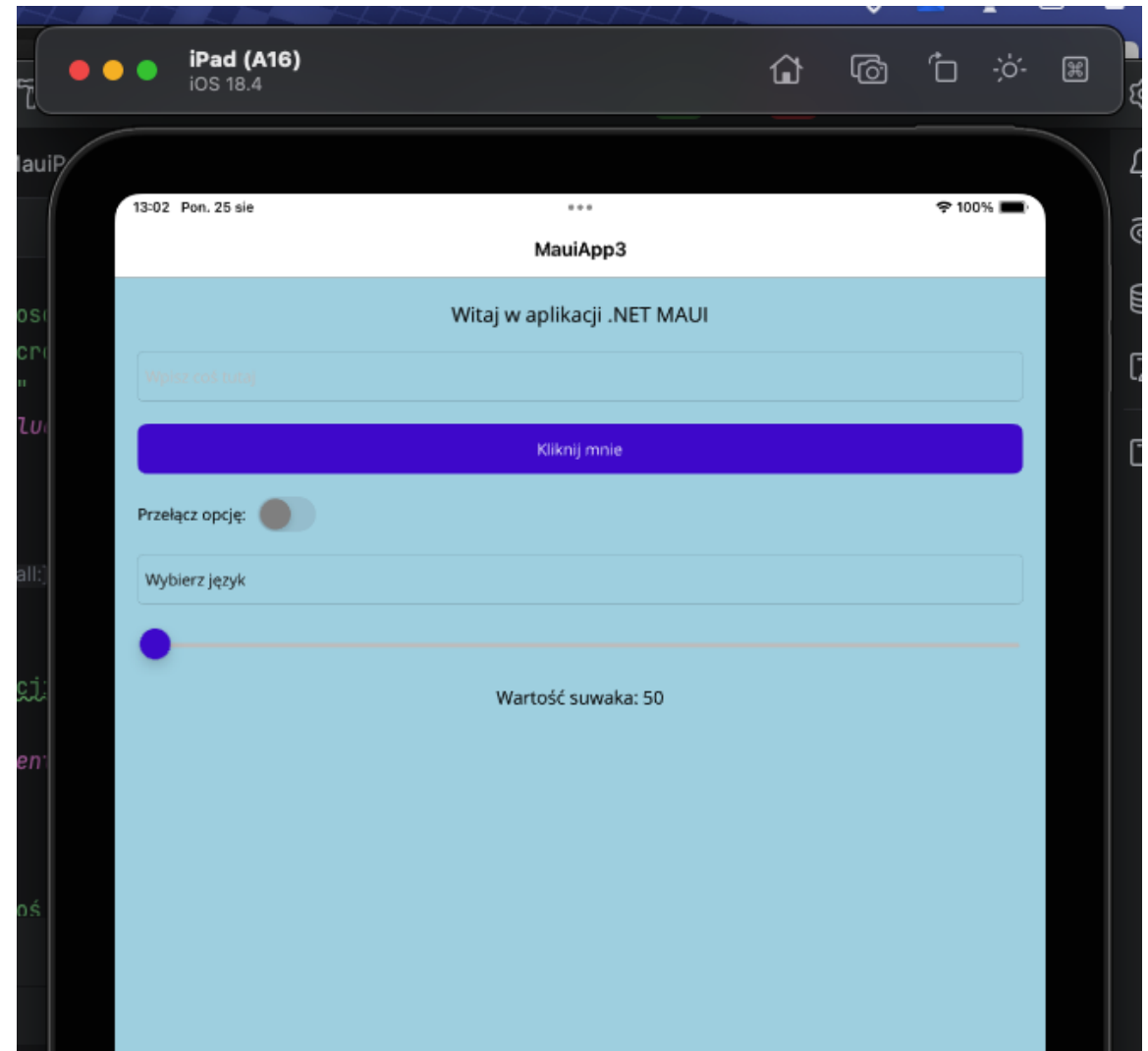
        // Obsługa przycisku
        private void Btn_Clicked(object sender,
        EventArgs e)
        {
            lblInfo.Text = "Kliknięto przycisk!
            Wpisano: " + txtEntry.Text;
        }

        // Obsługa zmiany tekstu w Entry
        private void
        TxtEntry_TextChanged(object sender,
        TextChangedEventArgs e)
        {
            lblInfo.Text = "Wpisujesz: " +
            e.NewTextValue;
        }

        // Obsługa przełącznika Switch
        private void
        SwitchOption_Toggled(object sender,
        ToggledEventArgs e)
        {
            lblInfo.Text = e.Value ? "Opcja
            włączona" : "Opcja wyłączona";
        }

        // Obsługa wyboru w Pickerze
        private void
        PickerLang_SelectedIndexChanged(object
        sender, EventArgs e)
        {
            if (pickerLang.SelectedIndex != -1)
            {
                lblInfo.Text = "Wybrano język: " +
                pickerLang.SelectedItem.ToString();
            }
        }

        // Obsługa suwaka Slider
        private void
        SliderValue_ValueChanged(object sender,
        ValueChangedEventArgs e)
        {
            lblSlider.Text = $"Wartość suwaka:
            {Math.Round(e.NewValue)}";
        }
    }
}
```



# Rider i MacOS – C# na Maku

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage
xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
xmlns:x="http://schemas.microsoft.com/winfx/2009/xaanged"
ml"
  x:Class="MauiApp3.MainPage"
  BackgroundColor="LightBlue"
  Title="MauiApp3">
  <ScrollView>
    <VerticalStackLayout Padding="20" Spacing="20">
      <!-- Label -->
      <Label x:Name="lblInfo"
        Text="Witaj w aplikacji .NET MAUI"
        FontSize="18"
        HorizontalOptions="Center"/>
      <!-- Entry (TextBox) -->
      <Entry x:Name="txtEntry"
        Placeholder="Wpisz coś tutaj"
        TextChanged="TxtEntry_TextChanged"/>
      <!-- Button -->
      <Button Text="Kliknij mnie"
        Clicked="Btn_Clicked"/>
      <!-- Switch -->
      <HorizontalStackLayout Spacing="10">
        <Label Text="Przełącz opcję:"
          VerticalOptions="Center"/>
        <Switch x:Name="switchOption"
          Toggled="SwitchOption_Toggled"/>
      </HorizontalStackLayout>
      <!-- Picker -->
      <Picker x:Name="pickerLang"
        Title="Wybierz język"
        SelectedIndexChanged="PickerLang_SelectedIndexCh
anged">
        <Picker.ItemsSource>
          <x:Array Type="{x:Type x:String}">
            <x:String>C#</x:String>
            <x:String>Java</x:String>
            <x:String>Python</x:String>
          </x:Array>
        </Picker.ItemsSource>
      </Picker>
      <!-- Slider -->
      <Slider x:Name="sliderValue"
        Minimum="0"
        Maximum="100"
        ValueChanged="SliderValue_ValueChanged"/>
      <Label x:Name="lblSlider"
        Text="Wartość suwaka: 50"
        FontSize="16"
        HorizontalOptions="Center"/>
    </VerticalStackLayout>
  </ScrollView>
</ContentPage>
```

# **NET MAUI w aplikacjach desktopowych**

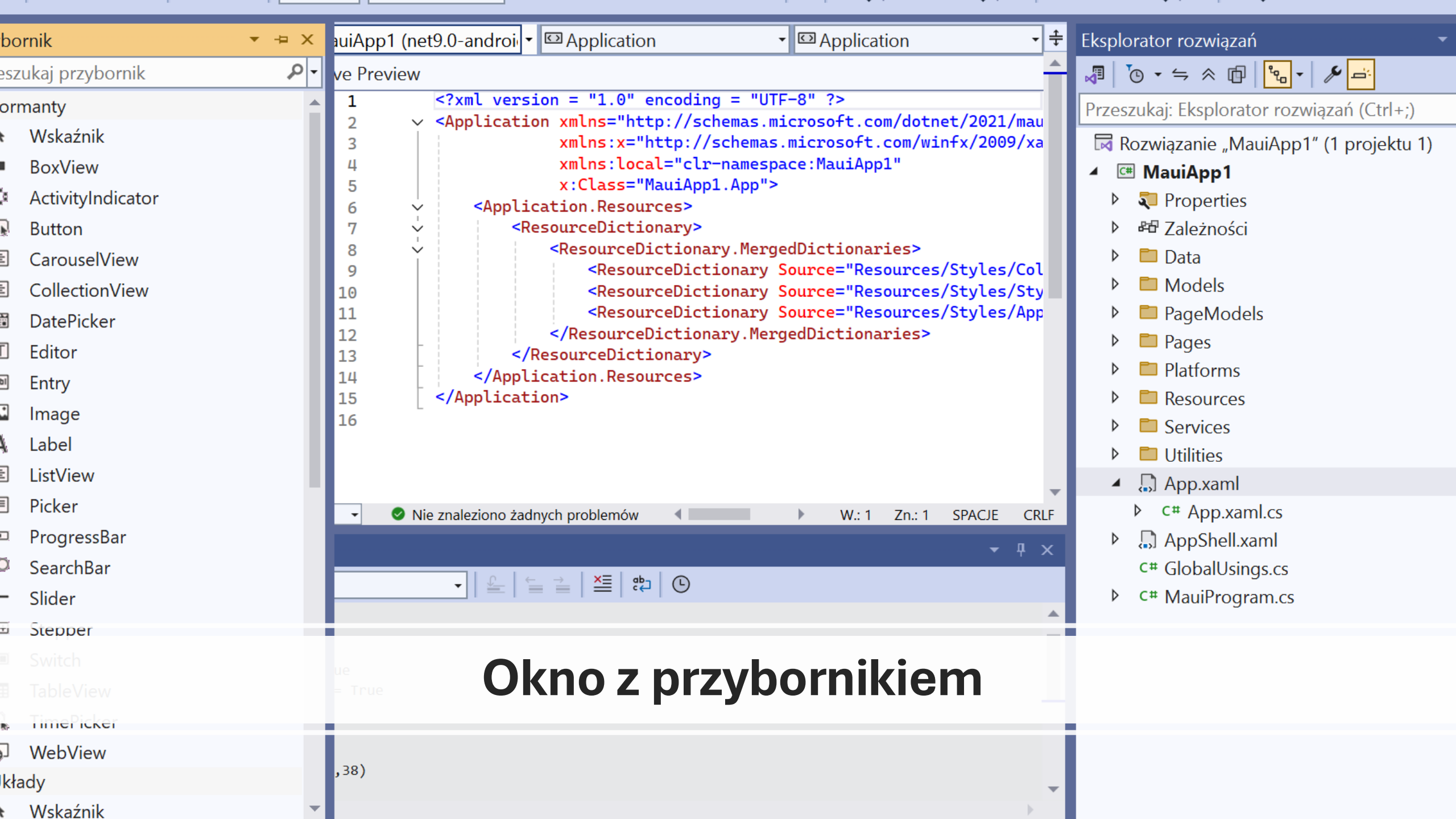
Jeden projekt – działa na Windows i macOS

Nowoczesny wygląd aplikacji

Integracja z kontrolkami natywnymi (np. FilePicker, Camera)

Obsługa stylów i motywów (jasny/ciemny tryb)

Łatwa rozbudowa o funkcje mobilne w przyszłości

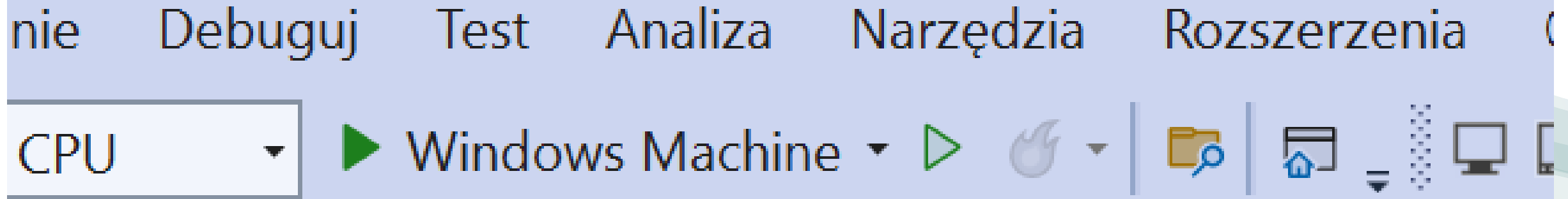


- Wyszukaj przybornik
- Wskaznik
  - BoxView
  - ActivityIndicator
  - Button
  - CarouselView
  - CollectionView
  - DatePicker
  - Editor
  - Entry
  - Image
  - Label
  - ListView
  - Picker
  - ProgressBar
  - SearchBar
  - Slider
  - Stepper
  - Switch
  - TableView
  - TimePicker
  - WebView
  - Wskaznik

```
<?xml version = "1.0" encoding = "UTF-8" ?>
<Application xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
xmlns:local="clr-namespace:MauiApp1"
x:Class="MauiApp1.App">
  <Application.Resources>
    <ResourceDictionary>
      <ResourceDictionary.MergedDictionaries>
        <ResourceDictionary Source="Resources/Styles/Colors.xaml" />
        <ResourceDictionary Source="Resources/Styles/Styles.xaml" />
        <ResourceDictionary Source="Resources/Styles/App.xaml" />
      </ResourceDictionary.MergedDictionaries>
    </ResourceDictionary>
  </Application.Resources>
</Application>
```

- Eksplorator rozwiązań
- Przeszukaj: Eksplorator rozwiązań (Ctrl+;)
- Rozwiązanie „MauiApp1” (1 projektu 1)
  - MauiApp1
    - Properties
    - Zależności
    - Data
    - Models
    - PageModels
    - Pages
    - Platforms
    - Resources
    - Services
    - Utilities
    - App.xaml
      - App.xaml.cs
      - AppShell.xaml
      - GlobalUsings.cs
      - MauiProgram.cs

Okno z przybornikiem



## Emulatory

Można wygenerować podgląd dla każdego Windows, MacOS i Linux. Możliwe jest wygenerowanie podglądu dla różnych urządzeń: telefony z Androidem, iOS czy inne, które trzeba pobrać.

Formularz z  
zapisem do  
pliku

zapisplik

## Dane użytkownika

Uzupełnij poniższy formularz:

Imię:

Email:

Zapisz do pliku

Otwórz folder z plikiem

# Plik MainPage.xaml

```
<ContentPage
xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
```

```
xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
```

```
    x:Class="zapisplik.Pages.MainPage">
```

```
    <VerticalStackLayout Padding="20"
Spacing="15">
```

```
        <Label Text="Uzupełnij poniższy
formularz:"
```

```
            FontSize="24"
```

```
            HorizontalOptions="Center" />
```

```
        <Label Text="Imię:" />
```

```
        <Entry x:Name="entryImie"
```

```
            Placeholder="Wpisz imię" />
```

```
        <Label Text="Email:" />
```

```
        <Entry x:Name="entryEmail"
```

```
            Placeholder="Wpisz adres e-mail" />
```

```
        <Button Text="Zapisz do pliku"
```

```
            Clicked="OnZapiszClicked" />
```

```
        <Button Text="Otwórz folder z plikiem"
```

```
            Clicked="OnOtworzFolderClicked"
```

```
            Margin="0,10,0,0" />
```

```
        <Label x:Name="lblStatus"
```

```
            TextColor="Green"
```

```
            FontAttributes="Italic"
```

```
            HorizontalOptions="Center" />
```

```
    </VerticalStackLayout>
```

```
</ContentPage>
```

Rozwiązanie „zapisplik” (1 projektu 1)

zapisplik

Properties

Zależności

Data

Models

PageModels

Pages

Controls

MainPage.xaml

MainPage.xaml.cs

ManageMetaPage.xaml

ProjectDetailPage.xaml

ProjectListPage.xaml

TaskDetailPage.xaml

Platforms

Resources

Services

Utilities

App.xaml

AppShell.xaml

GlobalUsings.cs

MauiProgram.cs

# Plik C#

```
using System;
using System.IO;
using System.Diagnostics;

namespace zapisplik.Pages;

public partial class MainPage : ContentPage
{
    private readonly string sciezkaPliku;

    public MainPage()
    {
        InitializeComponent();
        sciezkaPliku =
        Path.Combine(FileSystem.AppDataDirectory,
        "formularz_dane.txt");
    }

    private async void OnZapiszClicked(object sender,
    EventArgs e)
    {
        string imie = entryImie.Text?.Trim();
        string email = entryEmail.Text?.Trim();

        if (string.IsNullOrEmpty(imie) ||
        string.IsNullOrEmpty(email))
        {
            lblStatus.Text = "Uzupelnij wszystkie pola.";
            return;
        }

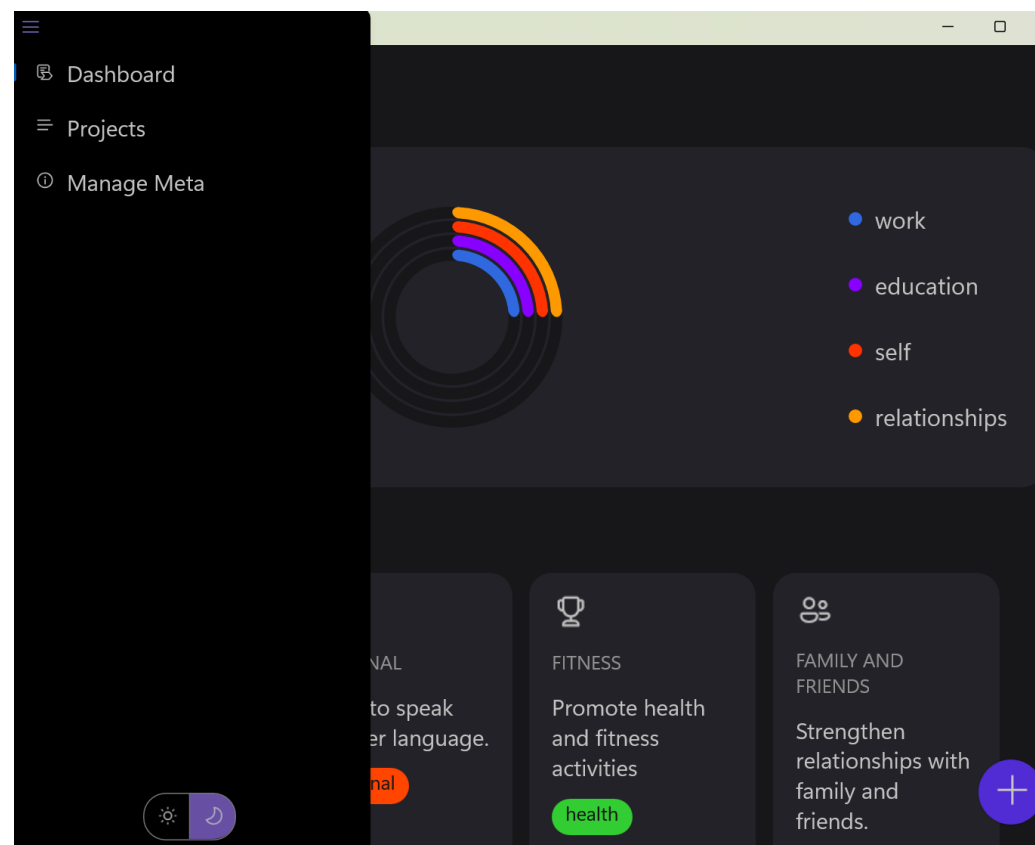
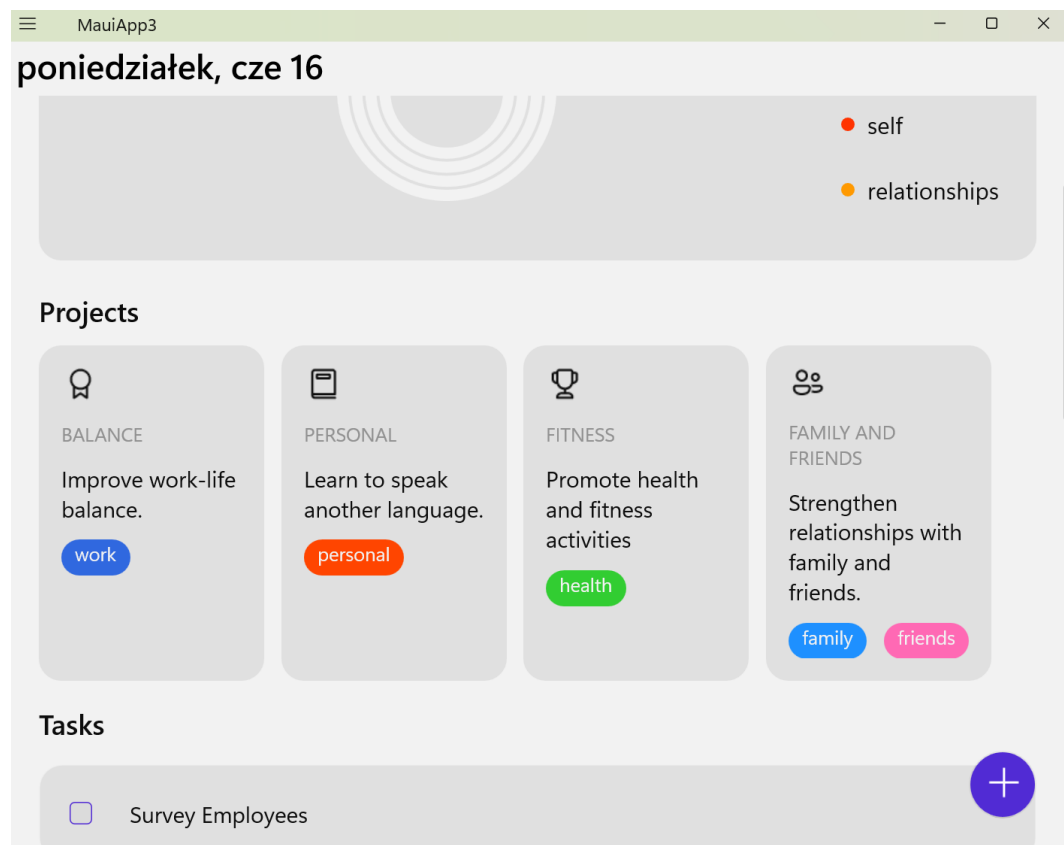
        string linia = $"Imię: {imie}, Email:
        {email}{Environment.NewLine}";

        try
        {
            File.AppendAllText(sciezkaPliku, linia);
            lblStatus.Text = "Dane zapisane.";
            entryImie.Text = "";
            entryEmail.Text = "";

            // Informacja o lokalizacji pliku
            await DisplayAlert("Zapisano dane", $"Plik
            zapisano w:\n{sciezkaPliku}", "OK");
        }
        catch (Exception ex)
        {
            lblStatus.Text = $"Błąd: {ex.Message}";
        }
    }

    private void OnOtworzFolderClicked(object
    sender, EventArgs e)
    {
        #if WINDOWS
        string folderPath =
        Path.GetDirectoryName(sciezkaPliku);
        if (folderPath != null)
        {
            try
            {
                Process.Start("explorer.exe", folderPath);
            }
            catch (Exception ex)
            {
                lblStatus.Text = $"Nie można otworzyć
                folderu: {ex.Message}";
            }
        }
        #else
        lblStatus.Text = "Funkcja dostępna tylko na
        Windows.";
        #endif
    }
}
```

# Aplikacja do zarządzania - kalendarz



Aplikacja do  
zarządzania -  
kalendarz

Task 🗑️ Delete

Task

Completed

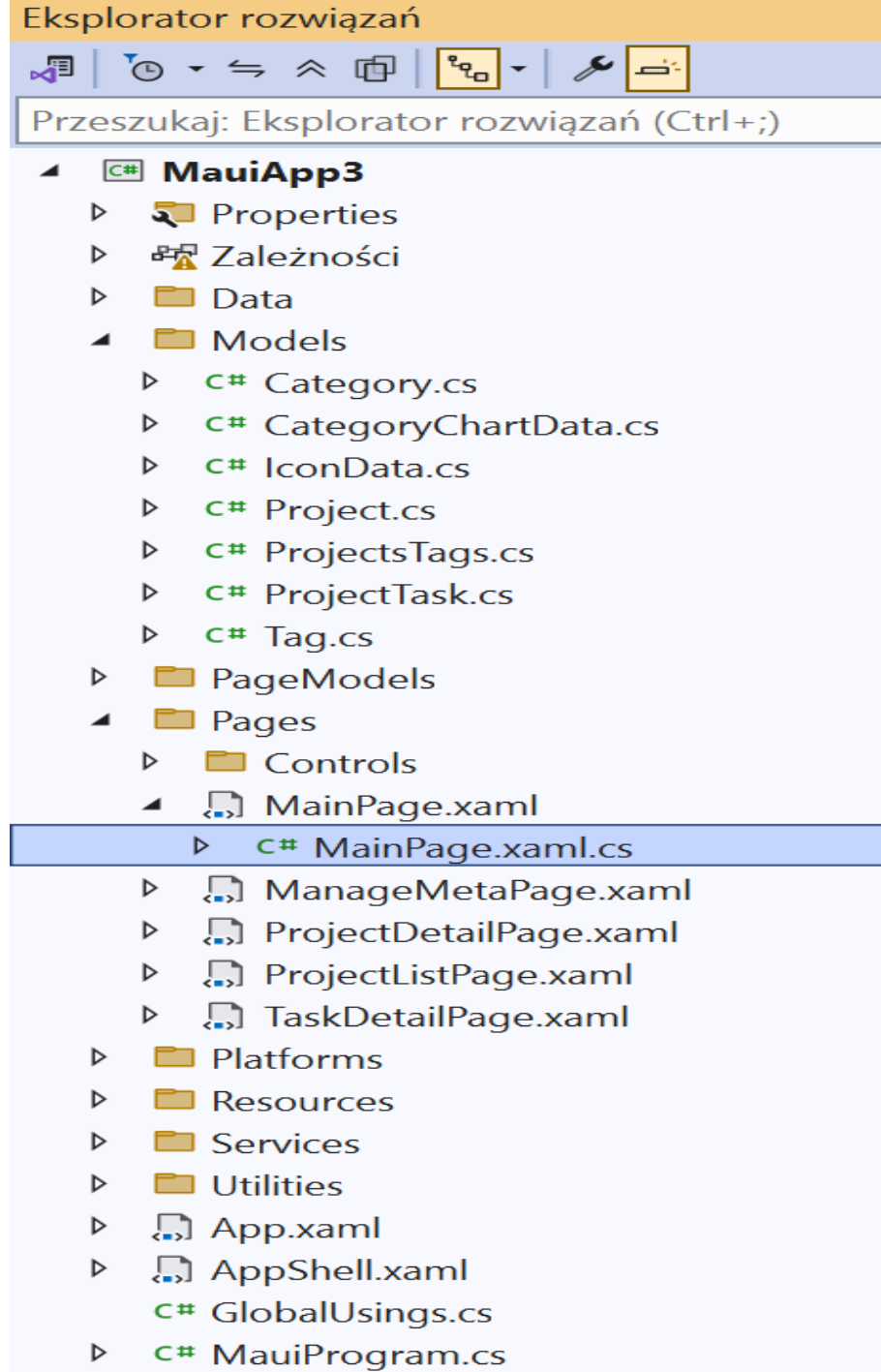
Project

Save

# Struktura projektu

Model danych znajduje się w folderze Models. W plikach Xaml znajduje się opis interfejsu użytkownika.

MAUI ma zestaw przykładowych aplikacji, które losowo ładują się jako przykładowe dane i można je dostosować do własnych potrzeb za pomocą plików Xaml.



# Wykorzystanie XCode w aplikacjach desktopowych

## Co można tworzyć w Xcode dla komputerów Mac?

- Aplikacje graficzne z interfejsem użytkownika (GUI) dla **macOS**.
- Narzędzia wiersza poleceń.
- Usługi systemowe (np. rozszerzenia).
- Widżety i aplikacje typu „menu bar”.

## Zalety Xcode przy tworzeniu aplikacji desktopowych:

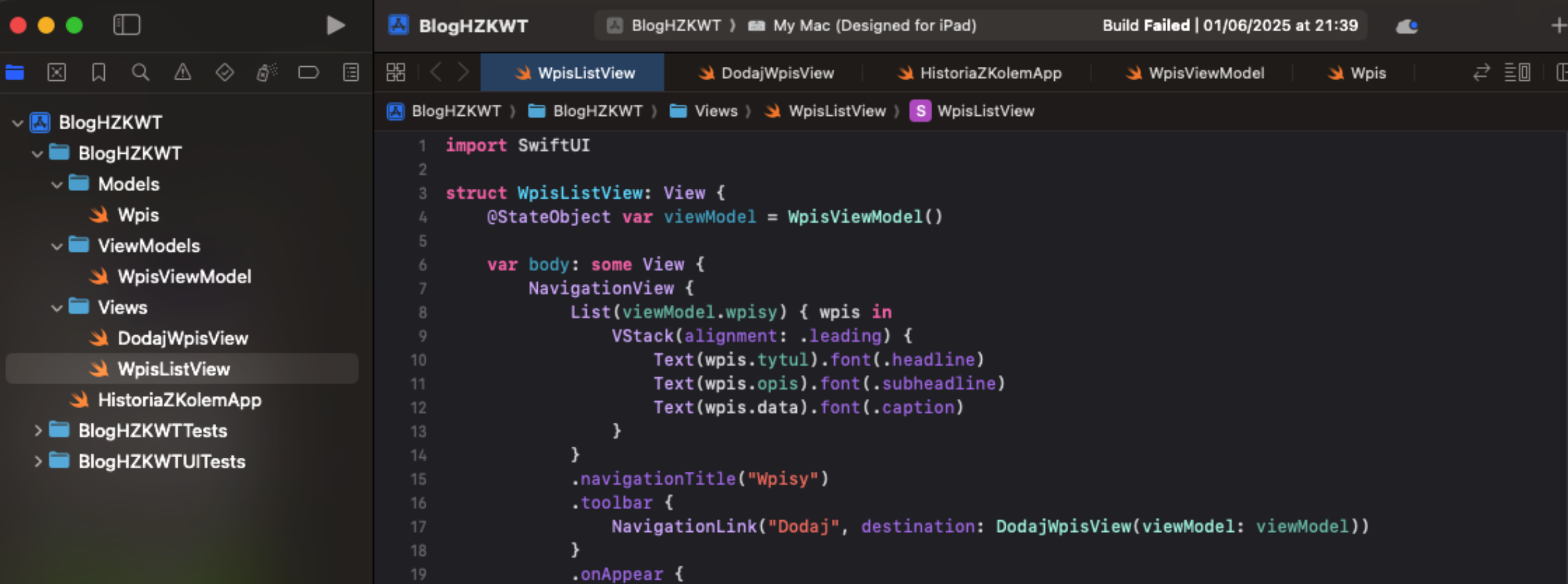
- Zaawansowany edytor kodu z podpowiedziami.
- Graficzny edytor UI (Interface Builder, SwiftUI Preview).
- Narzędzia debugowania i testowania.
- Symulatory i profilowanie wydajności.
- Automatyczne podpisywanie kodu i archiwizacja aplikacji.

```
import SwiftUI

@main
struct MyApp: App {
    var body: some Scene {
        WindowGroup {
            ContentView()
        }
    }
}

struct ContentView: View {
    @State private var wynik = 0

    var body: some View {
        VStack {
            Text("Wynik: \(wynik)")
            HStack {
                Button("Dodaj") { wynik += 1 }
                Button("Odejmij") { wynik -= 1 }
            }
        }.padding()
    }
}
```



# Wady Xcode

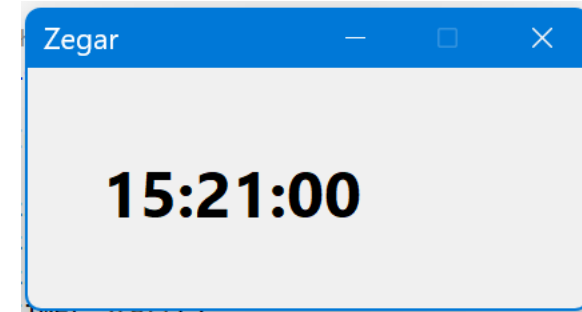
## Wady i ograniczenia:

- Dostępne tylko na macOS.
- Wymaga sprzętu Apple (Mac, MacBook).
- Brak natywnego wsparcia dla Windows/Linux.
- SwiftUI jeszcze nie obsługuje wszystkich zaawansowanych elementów GUI.

# Zadania do samodzielnego wykonania

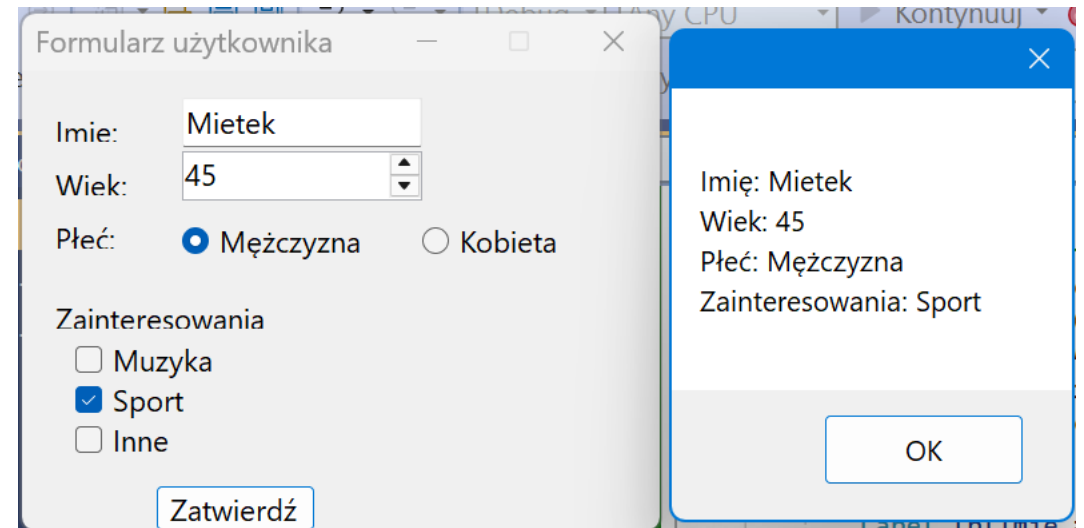
## Zadanie 1

Zaprojektuj i wykonaj zadanie w postaci cyfrowego zegarka.  
Możesz użyć dodatkowego formatowania okna.



## Zadanie 2

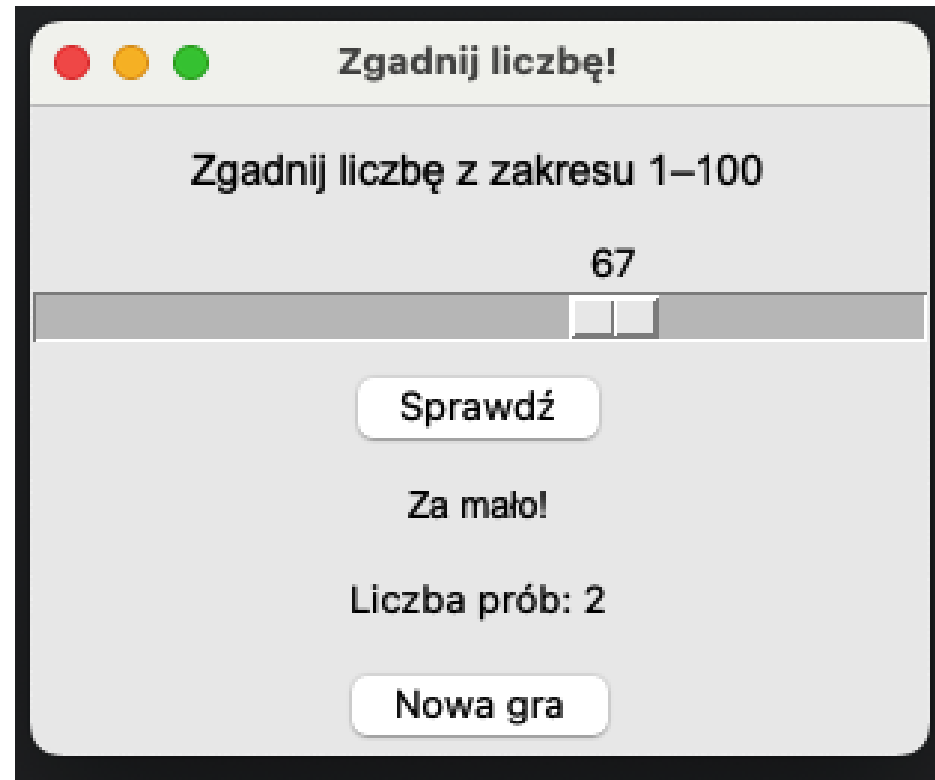
Na podstawie wzoru przedstawionego obok zaprojektuj i wykonaj formularz, który pobiera dane od użytkownika a następnie wyświetla je w wyskakującym okienku.



# Zadania do samodzielnego wykonania

## Zadanie 3

Wykonaj zadanie, w którym użytkownik za pomocą suwaka wybiera liczbę pomiędzy 1 a 100. Po naciśnięciu przycisku program sprawdza czy użytkownik podał liczbę większą od tej, którą wylosował program. Każda próba zgadywania liczby jest zliczana.



# Zadania do samodzielnego wykonania

## Zadanie 4 (zadanie na 5)

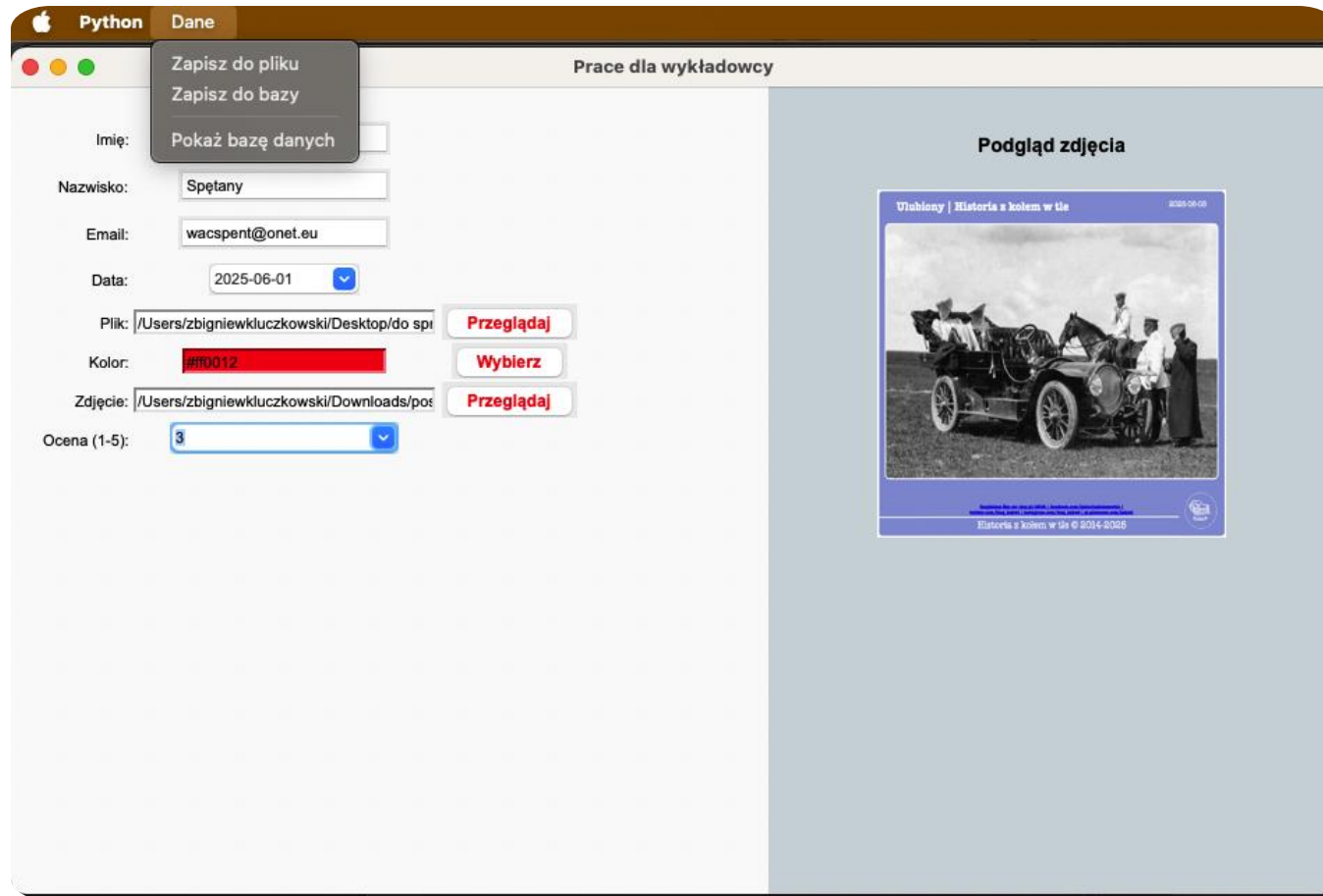
Zadanie w formie przeglądarki zdjęć. W dolnej części okna znajdują się miniatury.

Po naciśnięciu miniatury obraz w pełnej skali wyświetla się w górnej części okna.

Rozmiar miniatury wynosi 20% rozmiaru całego obrazka. Wykorzystano Tk Image, OS oraz Canvas.



# Zadania do samodzielnego wykonania



## Zadanie 5 (na 6)

Za pomocą języka C# stwórz aplikację desktopową według wzoru zamieszczonego obok. Aplikacja pozwala na zapisanie danych do pliku oraz do bazy SQL.

Wybrane zdjęcie pojawia się w drugiej części okna.

# Zadania do samodzielnego wykonania

## Zadanie 6 (na 5)

Przed Tobą prosta gra w „kółko i krzyżyk”.  
Gra korzysta z layout opartego o siatkę  
oraz z przycisków.

