Politechnika Częstochowska

Wydział Inżynierii Mechanicznej i Informatyki



Dokumentacja aplikacji z przedmiotu

Programowanie komponentowe

Aplikacja wykorzystująca

kontrolkę ActiveX

Artur Śnioszek

nr. 113055

Piotr Zyszczak

nr. 113066

Damian Łukasik

nr. 112993

II stopień, 2 semestr , 1 rok

Częstochowa 11 stycznia 2016 r.

1. Przedstawienie aplikacji

Niniejsza dokumentacja ma na celu opisanie aplikacji okienkowej wykorzystująca kontrolkę ActiveX w ramach zaliczenia z przedmiotu *Programowanie komponentowe*. Omawiana aplikacja jest przeznaczona na komputery z systemem operacyjnym z rodziny *Windows* firmy *Microsoft*.

Projekt zakładał stworzenie aplikacji okienkowej, która korzysta z kontrolki ActiveX w postaci biblioteki *dll*. Kontrolka ActiveX jest rozszerzeniem kontrolki *TextBox*,z zestawu *System.Windows.Forms*. Do kontrolki wprowadza się wartości liczbowe oraz znaki „-” i „.”, aby móc wprowadzać liczb ujemne lub liczby rzeczywiste. Dodatkową funkcjonalnością jest podświetlanie tła kontrolki na czerwono w przypadku wprowadzania liczby ujemnych oraz na zielono w przypadku liczb dodatnich. W kontrolce można wprowadzić tylko jeden znak „.”, a znak minus tylko na początku. Stworzona aplikacja ma na celu przetestowanie nowej kontrolki wykorzystując technologie ActiveX.

W dokumentacji załączono zrzuty ekranu w postaci *printscreen* oraz fragmenty kodów.

1. Wymagania sprzętowe

Projekt *ProjektActivexXLukasik* został napisany w języku C#, w środowisku programistycznym w środowisku programistycznym *Microsoft Visual Studio Community 2015* w wersji *14.0.24720.00* firmy *Microsoft*.

Aby kontrolka mogła poprawnie działać, należy spełnić jej wymaga sprzętowe:

* Komputer z systemem operacyjnym z rodziny Windows.
* Pamięć RAM : 3 MB.
* Nie wymagane połączenie z Internetem.
* 13,5 KB pamięci fizycznej.
* Monitor wyświetlający obraz.

1. Wykorzystana technologia

ActiveX jest technologią opartą na COM. Pozwala na tworzenie kontrolek *.ocx* lub *.dll*. W rzeczywistości ActiveX to obiekt COM, tyle że posiadający własny interfejs użytkownika. Tak więc mogliśmy tworzyć kontrolki ActiveX, wykorzystując np. *Delphi* oraz jego zalety projektowania wizualnego. Można było korzystać ze wszystkich komponentów i, ogólnie rzecz biorąc, projektowanie było łatwiejsze, niż w przypadku zwykłych obiektów COM. Dodatkowo ActiveX pozwala na wygenerowanie kodu umożliwiającego umieszczenie aplikacji na stronie WWW. Platforma .NET jest następczynią COM, która zakłada integralność pomiędzy programami. Do tej pory programiści mogli budować osobne kontrolki, które później dawało się wykorzystywać w innych aplikacjach. Wiązało się to z rejestracją tej kontrolki i dodawaniem odpowiednich wpisów w rejestrze Windows. W .NET komunikacja, między aplikacjami będzie ułatwiona, a dany program będzie mógł dziedziczyć po klasie z innego. [[[1]](#footnote-2)]

1. Kod źródłowy

Zawartość pliku *AssemblyInfo.cs*:

using System.Reflection;

using System.Runtime.CompilerServices;

using System.Runtime.InteropServices;

// General Information about an assembly is controlled through the following

// set of attributes. Change these attribute values to modify the information

// associated with an assembly.

[assembly: AssemblyTitle("ProjektActiveXLukasik")]

[assembly: AssemblyDescription("")]

[assembly: AssemblyConfiguration("")]

[assembly: AssemblyCompany("")]

[assembly: AssemblyProduct("ProjektActiveXLukasik")]

[assembly: AssemblyCopyright("Copyright © 2016")]

[assembly: AssemblyTrademark("")]

[assembly: AssemblyCulture("")]

// Setting ComVisible to false makes the types in this assembly not visible

// to COM components. If you need to access a type in this assembly from

// COM, set the ComVisible attribute to true on that type.

[assembly: ComVisible(true)]

// The following GUID is for the ID of the typelib if this project is exposed to COM

[assembly: Guid("f901ce39-bdce-47e0-b318-2b04174dfa40")]

// Version information for an assembly consists of the following four values:

//

// Major Version

// Minor Version

// Build Number

// Revision

//

// You can specify all the values or you can default the Build and Revision Numbers

// by using the '\*' as shown below:

// [assembly: AssemblyVersion("1.0.\*")]

[assembly: AssemblyVersion("1.0.0.0")]

[assembly: AssemblyFileVersion("1.0.0.0")]

Zawartość pliku *MyTextBox.cs*:

using System;

using System.Collections.Generic;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace ProjektActiveXLukasik

{

class MyTextBox : TextBox

{

public MyTextBox() : base()

{

}

protected override void OnKeyPress(KeyPressEventArgs e)

{

if (e.KeyChar == '.')

e.KeyChar = ',';

if (e.KeyChar == ',' && this.Text.Contains(','))

e.Handled = true;

if ((e.KeyChar == '0' || e.KeyChar == ',') && this.SelectionStart == 0)

e.Handled = true;

if ((e.KeyChar == '0' || e.KeyChar == ',') && this.SelectionStart == 1 && this.Text.Contains('-'))

e.Handled = true;

if (e.KeyChar == '-' && this.SelectionStart != 0)

e.Handled = true;

if (char.IsDigit(e.KeyChar) || e.KeyChar == ',' || e.KeyChar == (char)Keys.Back || e.KeyChar == '-')

base.OnKeyPress(e);

else

e.Handled = true;

}

protected override void OnTextChanged(EventArgs e)

{

if (this.Text.Length == 0)

{

BackColor = Color.White;

}

else

{

if (this.Text.Contains('-'))

{

BackColor = Color.Red;

}

else

{

BackColor = Color.Green;

}

}

}

}

}

Zawartość pliku *ProjektActiveXLukasikCtrl.Designer.cs*:

namespace ProjektActiveXLukasik

{

partial class ProjektActiveXLukasikCtrl

{

/// <summary>

/// Required designer variable.

/// </summary>

private System.ComponentModel.IContainer components = null;

/// <summary>

/// Clean up any resources being used.

/// </summary>

/// <param name="disposing">true if managed resources should be disposed; otherwise, false.</param>

protected override void Dispose(bool disposing)

{

if (disposing && (components != null))

{

components.Dispose();

}

base.Dispose(disposing);

}

#region Component Designer generated code

/// <summary>

/// Required method for Designer support - do not modify

/// the contents of this method with the code editor.

/// </summary>

private void InitializeComponent()

{

this.textbox1 = new ProjektActiveXLukasik.MyTextBox();

this.label1 = new System.Windows.Forms.Label();

this.SuspendLayout();

//

// textbox1

//

this.textbox1.Location = new System.Drawing.Point(3, 26);

this.textbox1.Name = "textbox1";

this.textbox1.Size = new System.Drawing.Size(109, 20);

this.textbox1.TabIndex = 0;

//

// label1

//

this.label1.AutoSize = true;

this.label1.Font = new System.Drawing.Font("Microsoft Sans Serif", 10F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(238)));

this.label1.Location = new System.Drawing.Point(3, 6);

this.label1.Name = "label1";

this.label1.Size = new System.Drawing.Size(86, 17);

this.label1.TabIndex = 1;

this.label1.Text = "Wpisz liczbe";

//

// ProjektActiveXLukasikCtrl

//

this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);

this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

this.Controls.Add(this.label1);

Controls.Add(textbox1);

this.Name = "ProjektActiveXLukasikCtrl";

this.Size = new System.Drawing.Size(160, 46);

this.ResumeLayout(false);

this.PerformLayout();

}

#endregion

private MyTextBox textbox1;

private System.Windows.Forms.Label label1;

}

}

Zawartość pliku *ProjektActiveXLukasikCtrl.cs*:

#region Using directives

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Text;

using System.Windows.Forms;

using System.Runtime.InteropServices;

using Microsoft.Win32;

using System.Reflection;

using System.Security.Permissions;

#endregion

namespace ProjektActiveXLukasik

{

[Guid("9051DE03-5FF2-45DA-9E9E-E52A92D1ABCC")]

public interface axProjektActiveXLukasikCtrl

{

#region Właściwości

bool Visible { get; set; } // Typical control property

bool Enabled { get; set; } // Typical control property

int ForeColor { get; set; } // Typical control property

int BackColor { get; set; } // Typical control property

float FloatProperty { get; set; } // Custom property

#endregion

#region Metody

void Refresh(); // Typical control method

string HelloWorld(); // Custom method

#endregion

}

[Guid("E029E53F-CFA5-436E-8050-94D894AEF83A")]

[InterfaceType(ComInterfaceType.InterfaceIsIDispatch)]

// punbliczny interface do zdarzeń

public interface axProjektActiveXLukasikCtrlEvents

{

#region Zdarzenia

// Must explicitly define DISPID for each event, otherwise, the

// callback address cannot be found when the event is fired.

[DispId(1)]

void Click();

[DispId(2)]

void FloatPropertyChanging(float NewValue, ref bool Cancel);

#endregion

}

[ProgId("ProjektActiveX.MojaKontrolka")]

[ClassInterface(ClassInterfaceType.None)]

[ComSourceInterfaces(typeof(axProjektActiveXLukasikCtrlEvents))]

[Guid("6F468F05-5C15-4679-825D-67BA02FD977F")]

public partial class ProjektActiveXLukasikCtrl : UserControl, axProjektActiveXLukasikCtrl

{

#region Rejestracja kontrolki ActiveX

// These routines perform the additional COM registration needed by

// ActiveX controls

[EditorBrowsable(EditorBrowsableState.Never)]

[ComRegisterFunction()]

public static void Register(Type t)

{

try

{

ActiveXCtrlHelper.RegasmRegisterControl(t);

}

catch (Exception ex)

{

Console.WriteLine(ex.Message); // Log the error

throw; // Re-throw the exception

}

}

[EditorBrowsable(EditorBrowsableState.Never)]

[ComUnregisterFunction()]

public static void Unregister(Type t)

{

try

{

ActiveXCtrlHelper.RegasmUnregisterControl(t);

}

catch (Exception ex)

{

Console.WriteLine(ex.Message); // Log the error

throw; // Re-throw the exception

}

}

#endregion

#region Inicjalizacja

public ProjektActiveXLukasikCtrl()

{

InitializeComponent();

// These functions are used to handle Tab-stops for the ActiveX

// control (including its child controls) when the control is

// hosted in a container.

this.LostFocus += new EventHandler(ProjektActiveXLukasikCtrl\_LostFocus);

this.ControlAdded += new ControlEventHandler(

ProjektActiveXLukasikCtrl\_ControlAdded);

// Raise custom Load event

this.OnCreateControl();

}

// This event will hook up the necessary handlers

void ProjektActiveXLukasikCtrl\_ControlAdded(object sender, ControlEventArgs e)

{

// Register tab handler and focus-related event handlers for

// the control and its child controls.

ActiveXCtrlHelper.WireUpHandlers(e.Control, ValidationHandler);

}

// Ensures that the Validating and Validated events fire properly

internal void ValidationHandler(object sender, System.EventArgs e)

{

if (this.ContainsFocus) return;

this.OnLeave(e); // Raise Leave event

if (this.CausesValidation)

{

CancelEventArgs validationArgs = new CancelEventArgs();

this.OnValidating(validationArgs);

if (validationArgs.Cancel && this.ActiveControl != null)

this.ActiveControl.Focus();

else

this.OnValidated(e); // Raise Validated event

}

}

[SecurityPermission(SecurityAction.LinkDemand,

Flags = SecurityPermissionFlag.UnmanagedCode)]

protected override void WndProc(ref System.Windows.Forms.Message m)

{

const int WM\_SETFOCUS = 0x7;

const int WM\_PARENTNOTIFY = 0x210;

const int WM\_DESTROY = 0x2;

const int WM\_LBUTTONDOWN = 0x201;

const int WM\_RBUTTONDOWN = 0x204;

if (m.Msg == WM\_SETFOCUS)

{

// Raise Enter event

this.OnEnter(System.EventArgs.Empty);

}

else if (m.Msg == WM\_PARENTNOTIFY && (

m.WParam.ToInt32() == WM\_LBUTTONDOWN ||

m.WParam.ToInt32() == WM\_RBUTTONDOWN))

{

if (!this.ContainsFocus)

{

// Raise Enter event

this.OnEnter(System.EventArgs.Empty);

}

}

else if (m.Msg == WM\_DESTROY &&

!this.IsDisposed && !this.Disposing)

{

// Used to ensure the cleanup of the control

this.Dispose();

}

base.WndProc(ref m);

}

// Ensures that tabbing across the container and the .NET controls

// works as expected

void ProjektActiveXLukasikCtrl\_LostFocus(object sender, EventArgs e)

{

ActiveXCtrlHelper.HandleFocus(this);

}

#endregion

#region Właściwości

public new int ForeColor

{

get { return ActiveXCtrlHelper.GetOleColorFromColor(base.ForeColor); }

set { base.ForeColor = ActiveXCtrlHelper.GetColorFromOleColor(value); }

}

public new int BackColor

{

get { return ActiveXCtrlHelper.GetOleColorFromColor(base.BackColor); }

set { base.BackColor = ActiveXCtrlHelper.GetColorFromOleColor(value); }

}

private float fField = 0;

/// <summary>

/// A custom property with both get and set accessor methods.

/// </summary>

public float FloatProperty

{

get { return this.fField; }

set

{

bool cancel = false;

// Raise the event FloatPropertyChanging

if (null != FloatPropertyChanging)

FloatPropertyChanging(value, ref cancel);

if (!cancel)

{

this.fField = value;

}

}

}

#endregion

#region Metody

public string HelloWorld()

{

return "HelloWorld";

}

#endregion

#region Zdarzenia

// This section shows the examples of exposing a control's events.

// Typically, you just need to

// 1) Declare the event as you want it.

// 2) Raise the event in the appropriate control event.

[ComVisible(false)]

public delegate void ClickEventHandler();

public new event ClickEventHandler Click = null;

void ProjektActiveXLukasikCtrl\_Click(object sender, EventArgs e)

{

if (null != Click) Click(); // Raise the new Click event.

}

[ComVisible(false)]

public delegate void FloatPropertyChangingEventHandler(float NewValue, ref bool Cancel);

public event FloatPropertyChangingEventHandler FloatPropertyChanging = null;

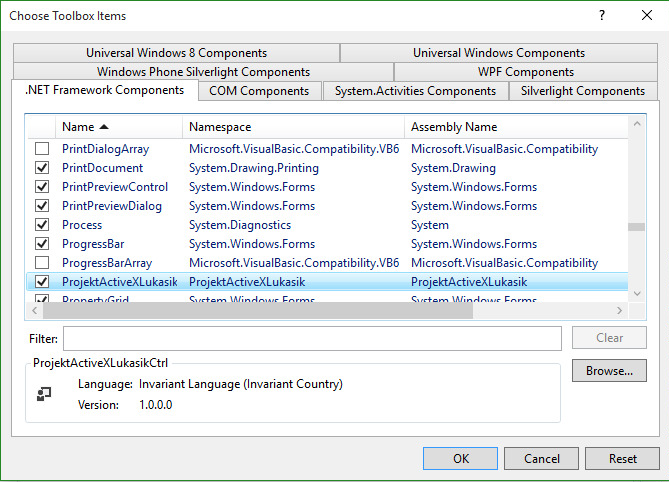
#endregion

}

}

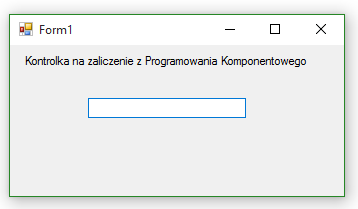
1. Wykorzystanie kontrolki

Wygenerowano bibliotekę *dll* projektu *ProjektActivexXLukasik* w folderze *Release.*

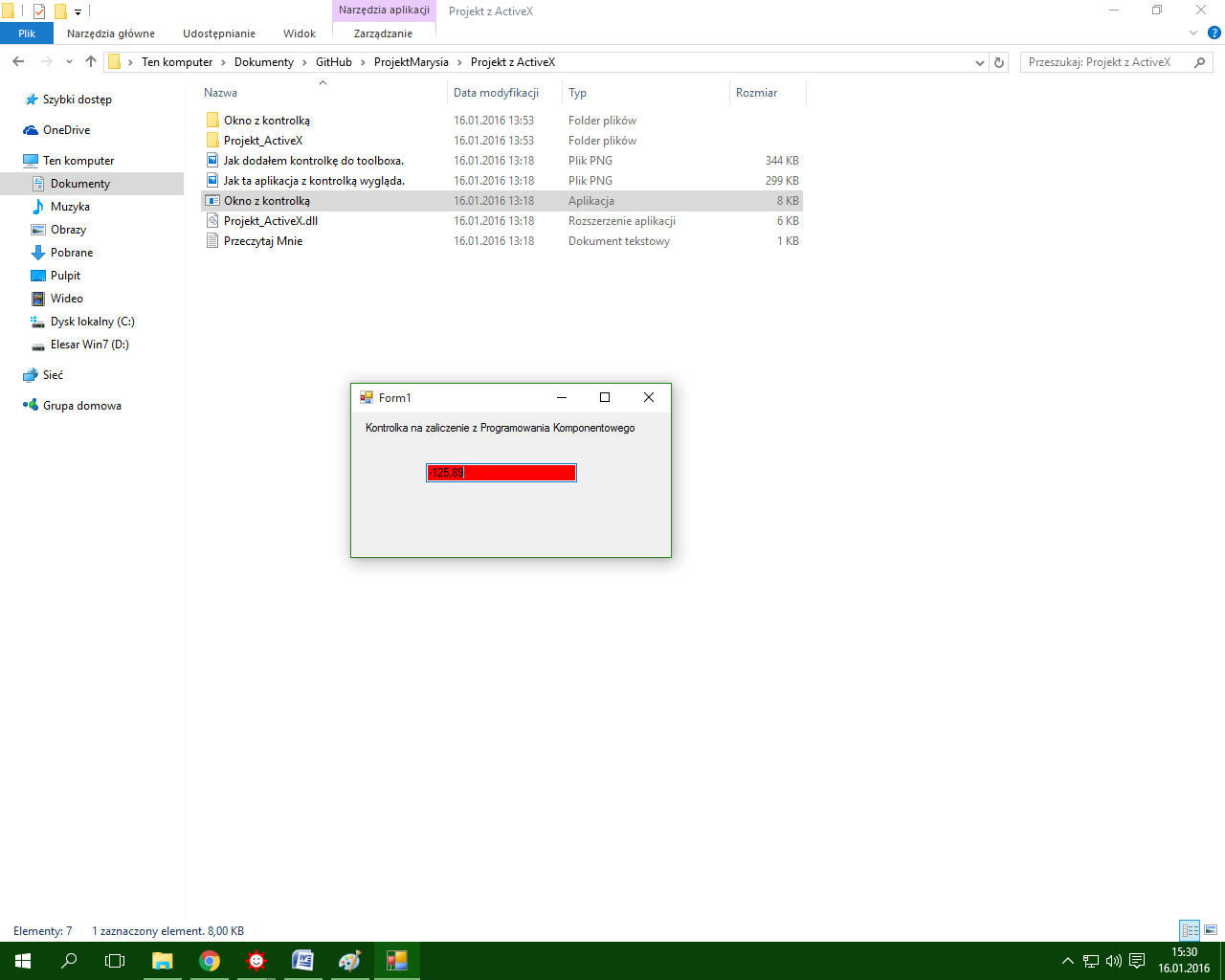


*Rys1. Widok okna Choose Toolbox Items.*

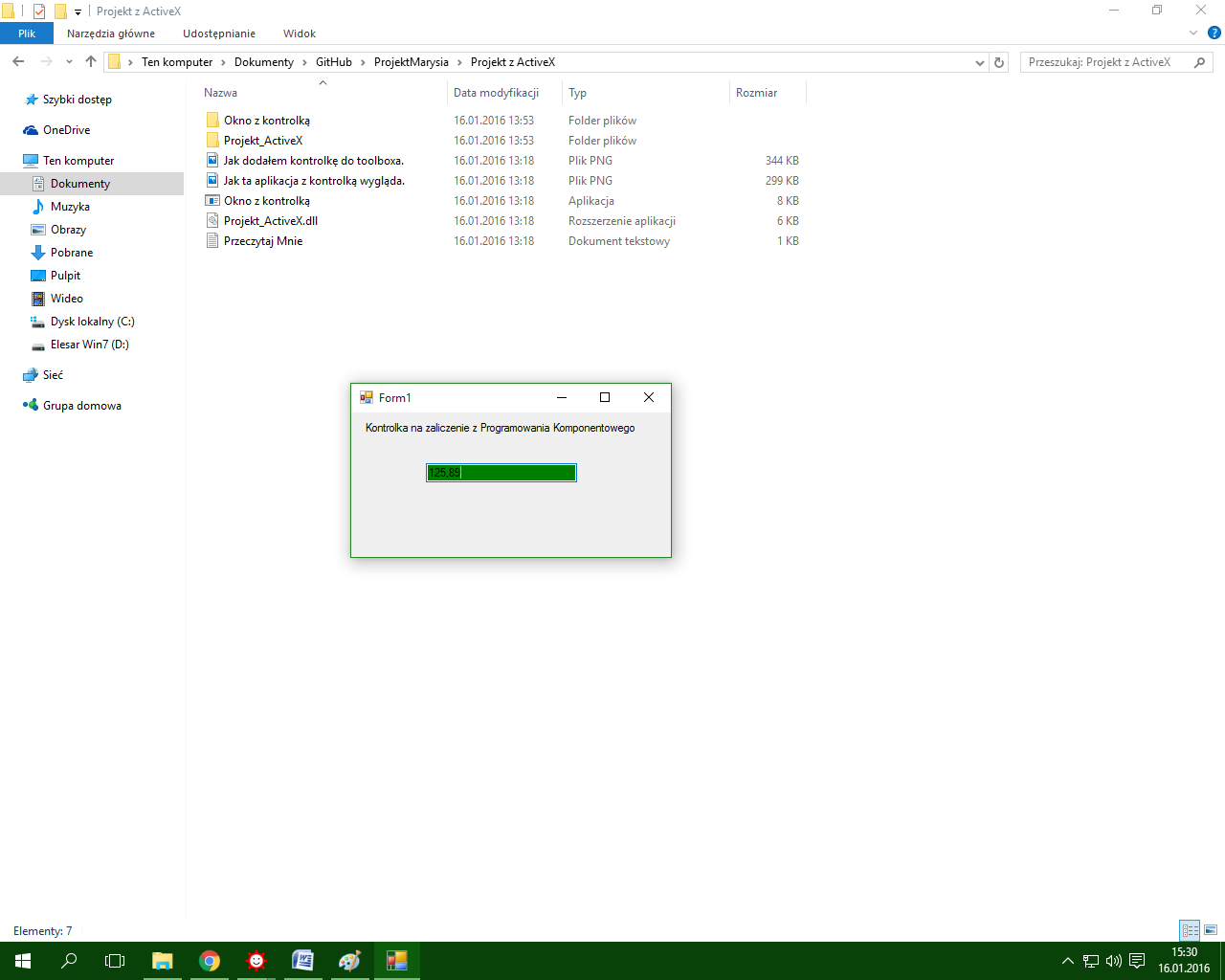
Działanie kontrolki zostały przedstawione na poniższych zrzutach ekranu.



*Rys2. Widok okna z kontrolką.*



*Rys3. Widok okna z kontrolką po wpisaniu wartości ujemnych.*



*Rys4. Widok okna z kontrolką po wpisaniu wartości dodatnich.*

1. Podsumowanie

Projekt *ProjektActivexXLukasik* jest napisany w języku C#, w środowisku programistycznym *Microsoft Visual Studio Community 2015* w wersji *14.0.24720.00* firmy *Microsoft*.

1. [C# Wprowadzenie Rozdział 11. Podzespoły .Net](http://4programmers.net/C_sharp/Wprowadzenie/Rozdział_11" \l "id-ActiveX) [↑](#footnote-ref-2)