Politechnika Częstochowska

Wydział Inżynierii Mechanicznej i Informatyki



Dokumentacja aplikacji z przedmiotu

Programowanie komponentowe

Aplikacja wykorzystująca

kontrolkę ActiveX

Damian Łukasik

nr. 112993

II stopień, 2 semestr , 1 rok

Częstochowa 11 stycznia 2016 r.

# 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.

# Wymagania sprzętowe

Aplikacja „Okno z kontrolką” została napisana w języku C#, w środowisku programistycznym w środowisku programistycznym *Microsoft Visual Studio Community 2015* w wersji *14.0.24720.00* firmy *Microsoft*.

Aby aplikacja 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.

# 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-1)]

# Kod źródłowy

Zawartość pliku *AssemblyInfo.cs* z folderu *Properties* projektu *Projekt\_ActiveX*:

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("Projekt\_ActiveX")]

[assembly: AssemblyDescription("")]

[assembly: AssemblyConfiguration("")]

[assembly: AssemblyCompany("")]

[assembly: AssemblyProduct("Projekt\_ActiveX")]

[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)]

//ustawienie widoczności klasy dla COMa

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

[assembly: Guid("ca790c33-169e-4358-9162-10ad0bc7c29a")]

// 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 IMojaKontrolka.cs z folderu głównego projektu *Projekt\_ActiveX*:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Runtime.InteropServices;

using System.Windows.Forms;

namespace Projekt\_ActiveX

{

// identyfikator globalnie unikatowy – guid

//można wygenerować z Tools > Create GUID

//InterfaceType - typ interfejsu

[InterfaceType(ComInterfaceType.InterfaceIsDual)]

//guidy muszą być różne

[Guid("713F2503-8ADB-48C6-87C6-6219029D6599")]

public interface IMojaKontrolka

{

void show();

}

}

Zawartość pliku MojaKontrolka.cs z folderu głównego projektu *Projekt\_ActiveX*:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Runtime.InteropServices;

using System.Windows.Forms;

using System.Drawing;

namespace Projekt\_ActiveX

{

[ClassInterface(ClassInterfaceType.None)]

//guidy muszą być różne

[Guid("58ED5856-C818-4923-A7EA-0DF6F4998AFC")]

[ProgId("Kontrolka.MojaKontrolka")]

public partial class MojaKontrolka : TextBox, IMojaKontrolka

{

public void show()

{

Console.WriteLine("Test");

}

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 Form1.cs z folderu głównego projektu *Okno z Kontrolką*:

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 Okno\_z\_kontrolką

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

}

}

Zawartość pliku Form1.Designer.cs z folderu głównego projektu *Okno z Kontrolką*:

namespace Okno\_z\_kontrolką

{

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

this.mojaKontrolka1 = new Projekt\_ActiveX.MojaKontrolka();

this.SuspendLayout();

//

// label1

//

this.label1.AutoSize = true;

this.label1.Location = new System.Drawing.Point(12, 9);

this.label1.Name = "label1";

this.label1.Size = new System.Drawing.Size(289, 13);

this.label1.TabIndex = 0;

this.label1.Text = "Kontrolka na zaliczenie z Programowania Komponentowego";

//

// mojaKontrolka1

//

this.mojaKontrolka1.BackColor = System.Drawing.Color.White;

this.mojaKontrolka1.Location = new System.Drawing.Point(78, 53);

this.mojaKontrolka1.Name = "mojaKontrolka1";

this.mojaKontrolka1.Size = new System.Drawing.Size(158, 20);

this.mojaKontrolka1.TabIndex = 1;

//

// Form1

//

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

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

this.ClientSize = new System.Drawing.Size(334, 151);

this.Controls.Add(this.mojaKontrolka1);

this.Controls.Add(this.label1);

this.Name = "Form1";

this.Text = "Form1";

this.ResumeLayout(false);

this.PerformLayout();

}

#endregion

private System.Windows.Forms.Label label1;

private Projekt\_ActiveX.MojaKontrolka mojaKontrolka1;

}

}

Zawartość pliku Program.cs z folderu głównego projektu *Okno z Kontrolką*:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Okno\_z\_kontrolką

{

static class Program

{

/// <summary>

/// The main entry point for the application.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Form1());

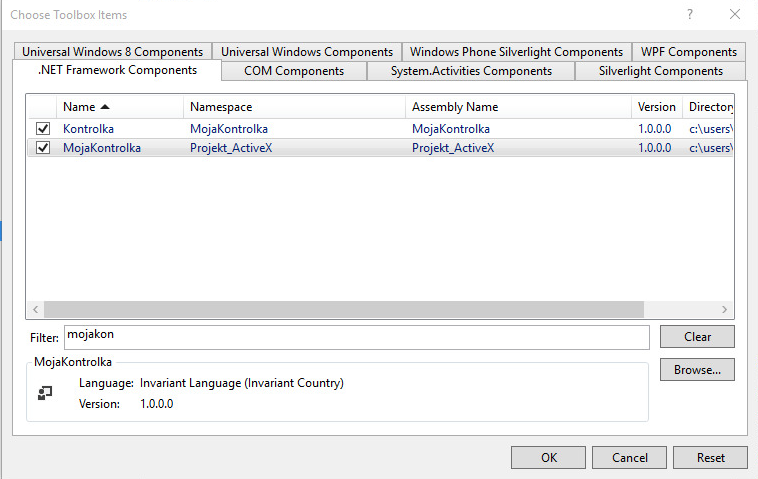
}

}

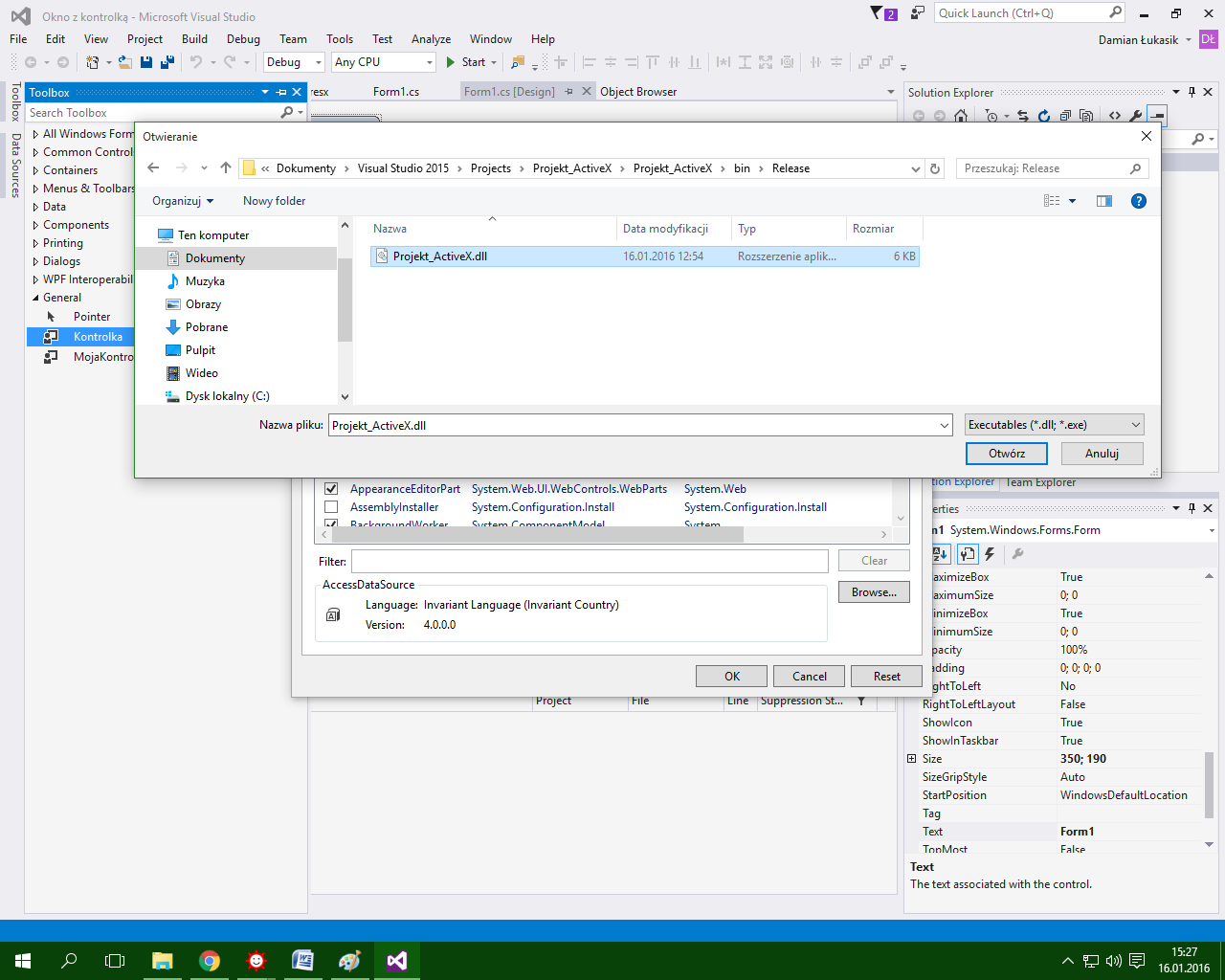
}

# Wykorzystanie kontrolki

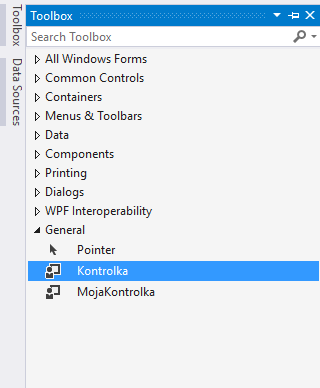
Wygenerowano bibliotekę *dll* projektu *Projekt\_ActiveX* w folderze *Release*, następnie stworzono drugi projekt *Okno z Kontrolką*, w którym dodano do *Toolboxa* nową kontrolkę poprzez kliknięciem prawym przyciskiem myszy i wybraniu z menu pozycje *Choose Items*, gdzie w zakładce *.Net Framework Components* wybrano z folderu *bin/Release* projektu *Projekt\_ActiveX* wygenerowaną bibliotekę *dll*, za pośrednictwem przycisku *Browse*. Pojawia nam się nasza kontrolka w zakładce *General* w *Toolboxie*, którą można wstawić do widoku okna tak samo jak zwykłą kontrolkę.



*Rys1. Widok okna Choose Toolbox Items.*

**

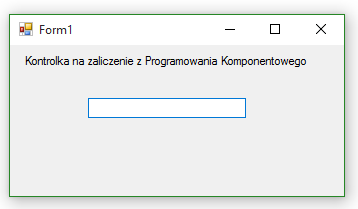
*Rys2. Widok eksploratora Windows wyszukujący bibliotekę dll.*

**

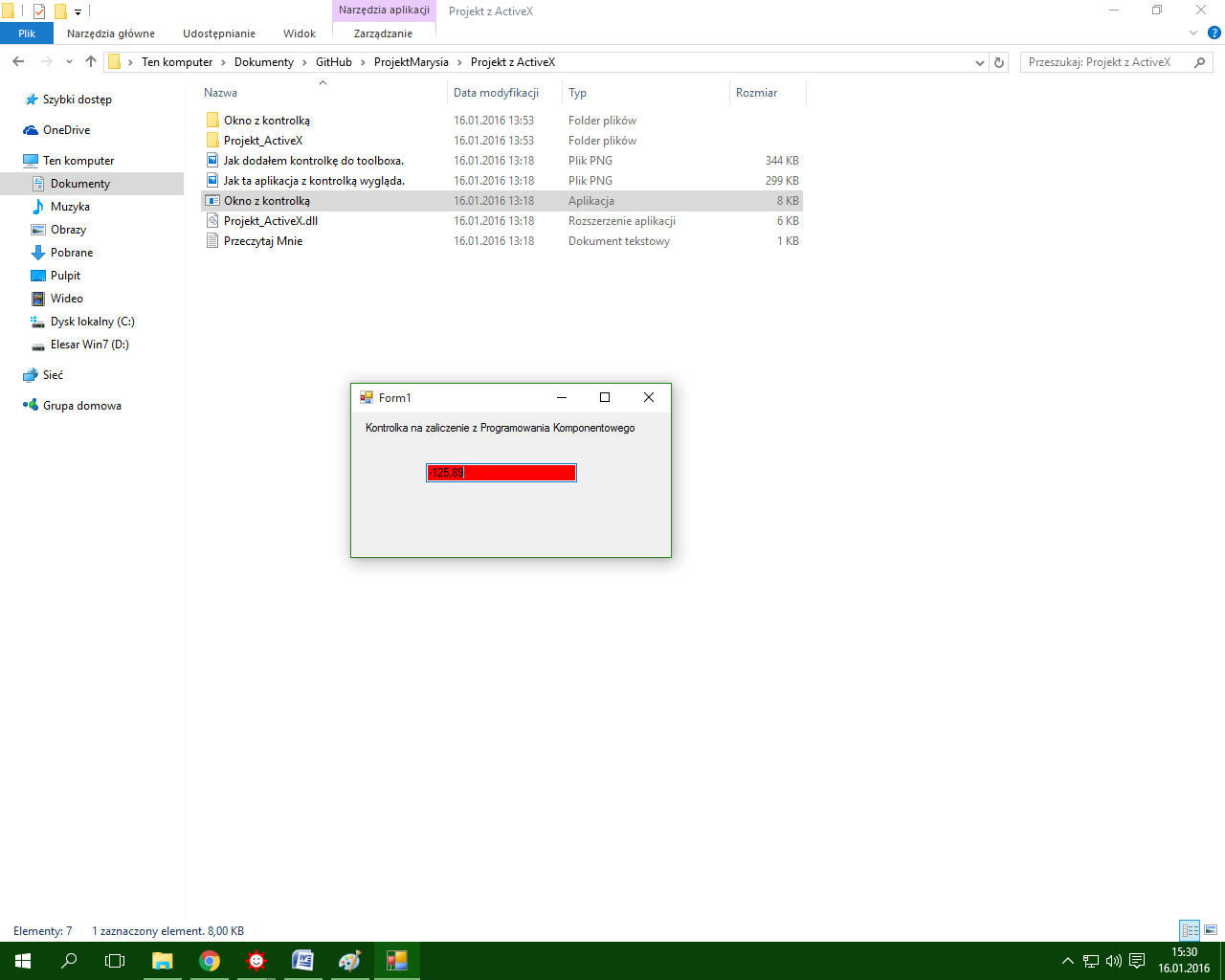
*Rys3. Widok zakładki General w Toolboxie.*

# Działanie kontrolki

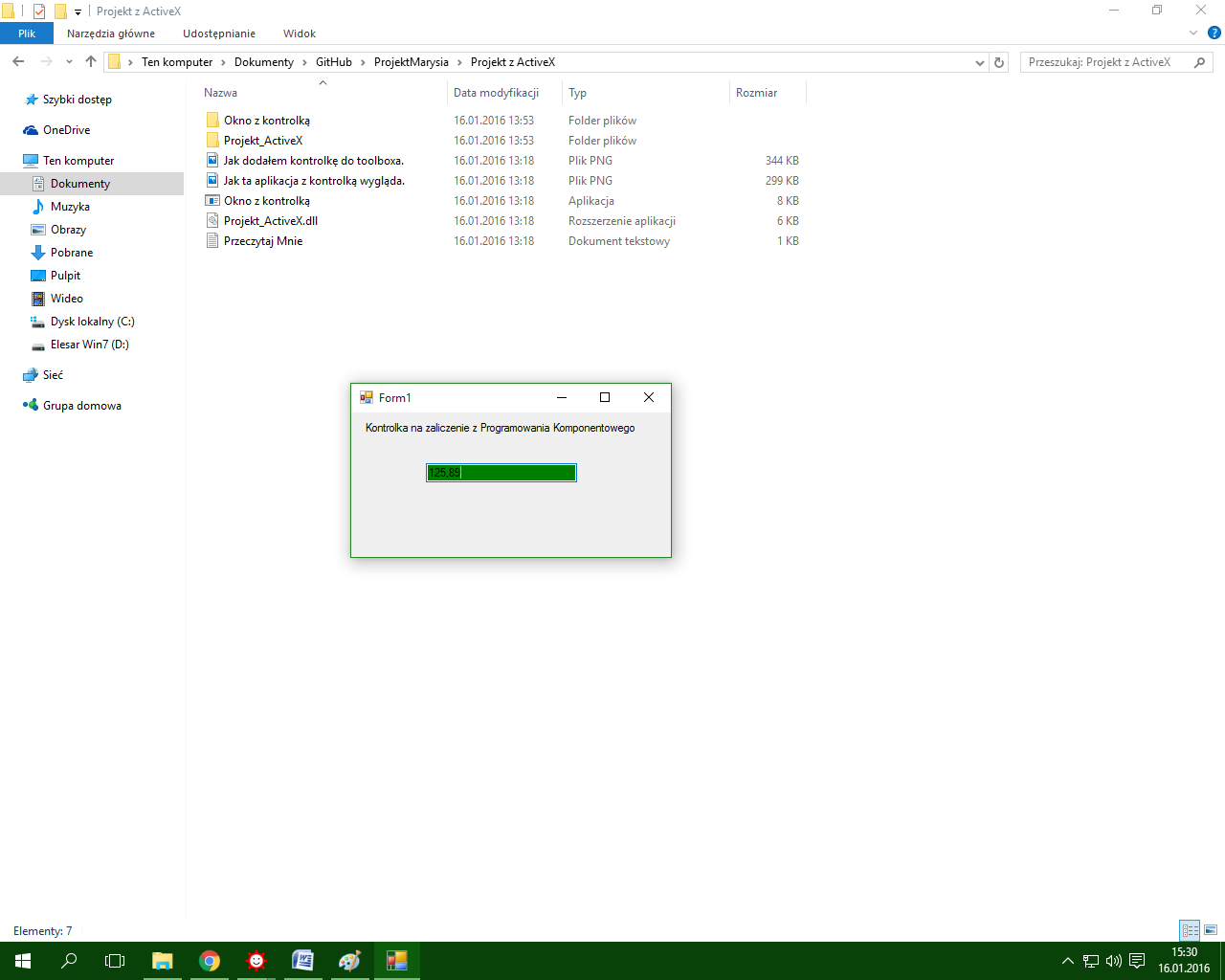
Działanie kontrolki zostały przedstawione na poniższych zrzutach ekranu.

**

*Rys4. Widok okna z kontrolką.*

**

*Rys5. Widok okna z kontrolką po wpisaniu wartości ujemnych.*

**

*Rys6. Widok okna z kontrolką po wpisaniu wartości dodatnich.*

# Podsumowanie

Aplikacja "Okno z Kontrolką" jest napisana w języku C#, w środowisku programistycznym *Microsoft Visual Studio Community 2015* w wersji *14.0.24720.00* firmy *Microsoft*. Wykorzystuje bibliotekę dll wygenerowaną z projektu *Projekt\_ActiveX*.

Aby aplikacja mogła poprawnie działać należy spełnić wymagania sprzętowe zawarte w rozdziale 2, a także pliki *Projekt\_ActiveX.dll* oraz *Okno z kontrolką.exe* muszą znajdować się w tym samym folderze oraz dodać te pliki do wyjątków w programie antywirusowym, ponieważ są traktowane jako wirusy.

1. [C# Wprowadzenie Rozdział 11. Podzespoły .Net](http://4programmers.net/C_sharp/Wprowadzenie/Rozdzia%C5%82_11#id-ActiveX) [↑](#footnote-ref-1)