Notebook

Software Documentation

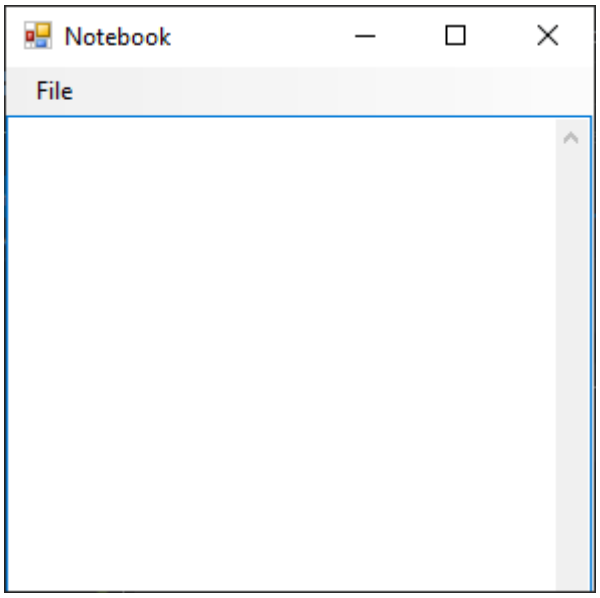
Author: matiwa

Table of contents

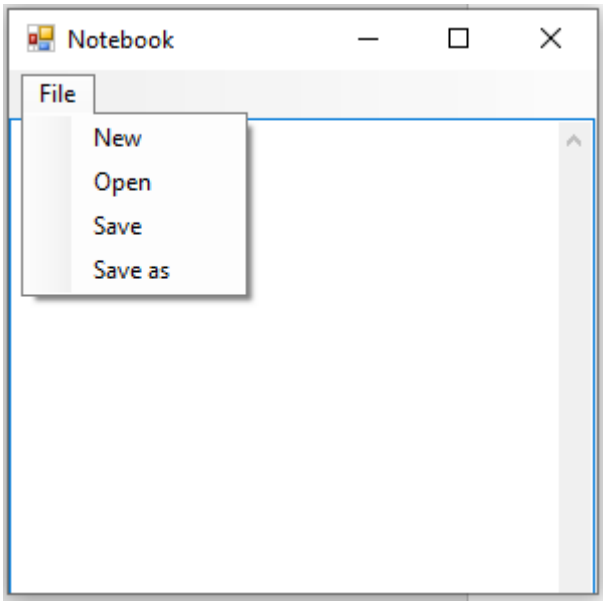
Introduction

This software documentation includes: description of the application's operation, what is needed for use, algorithms used, interface description and source code description. This application serves as a simple notepad.

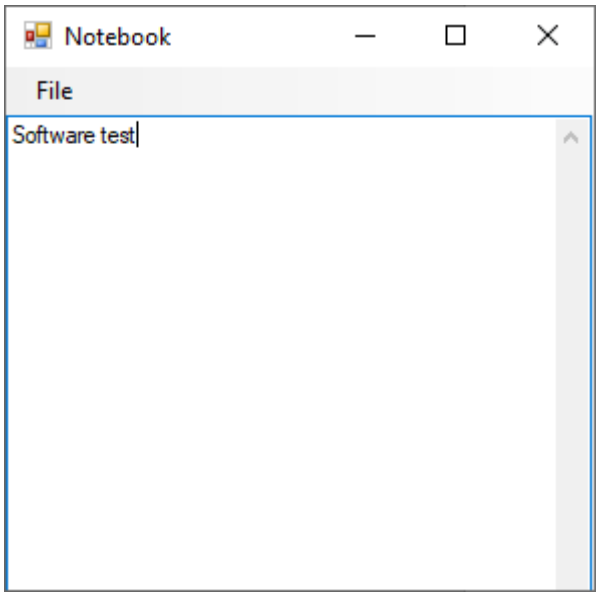
Describing of the application’s operation



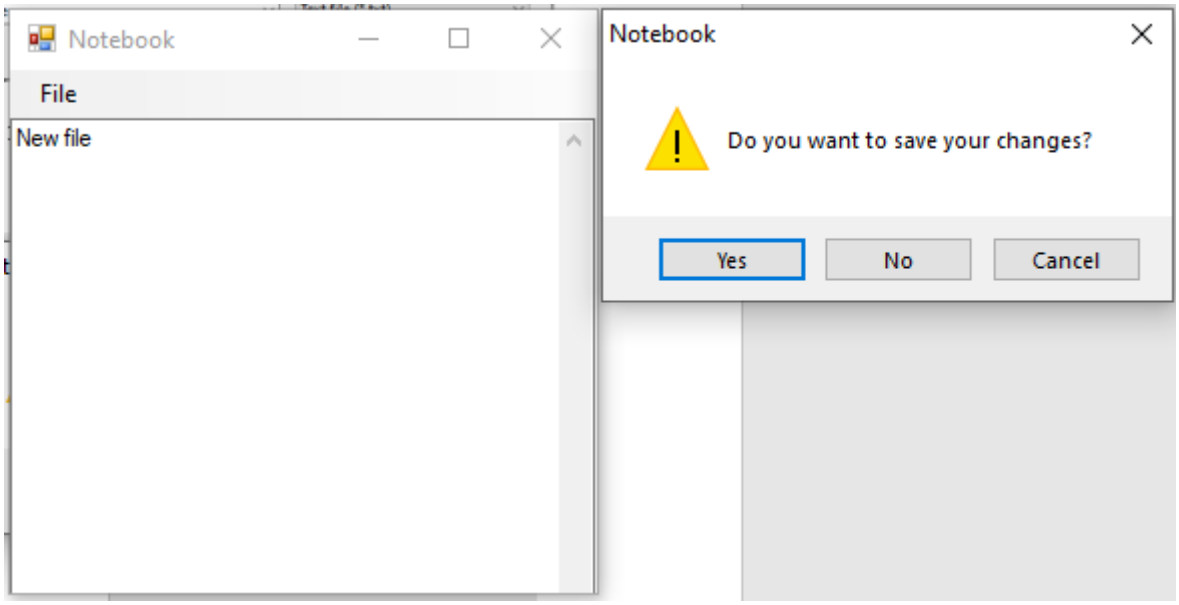
Drawing 1: The beginning of the application's operation [own study]



Drawing 2: Menu view [own study]

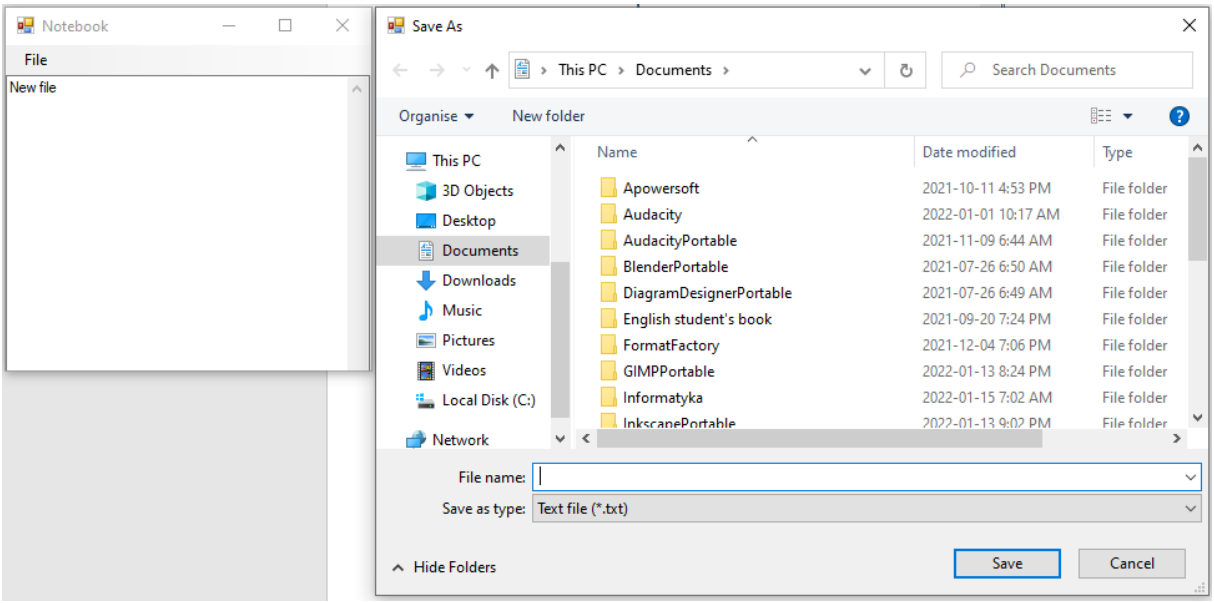


Drawing 3: Sample text [own study]

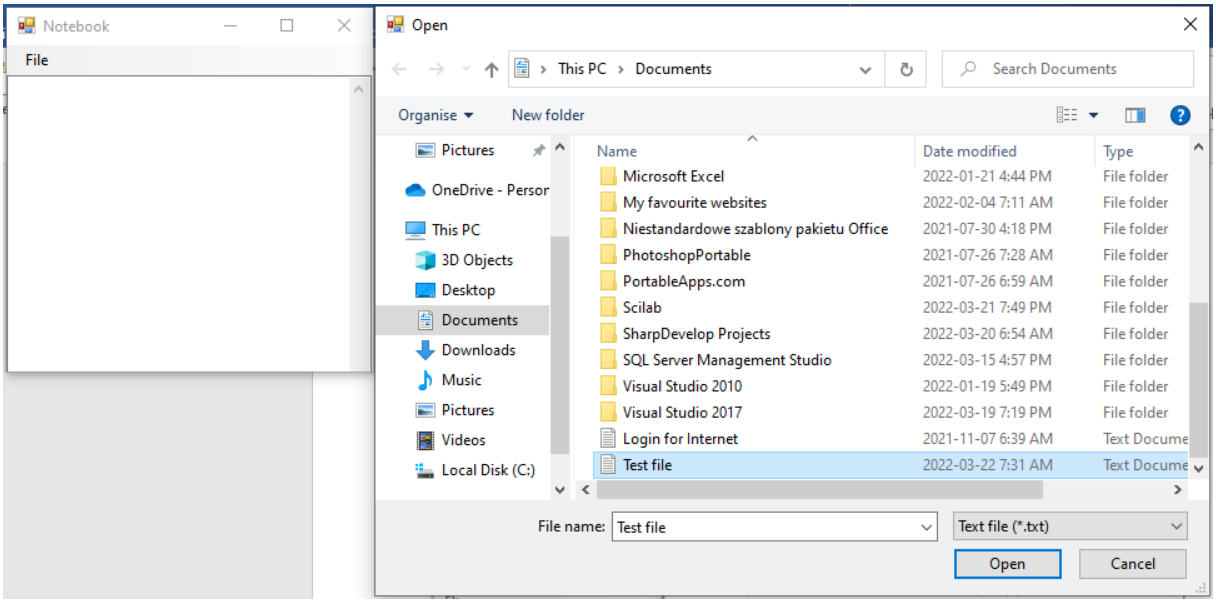


Drawing 4: Save changes dialog (after pressing New) [own study]

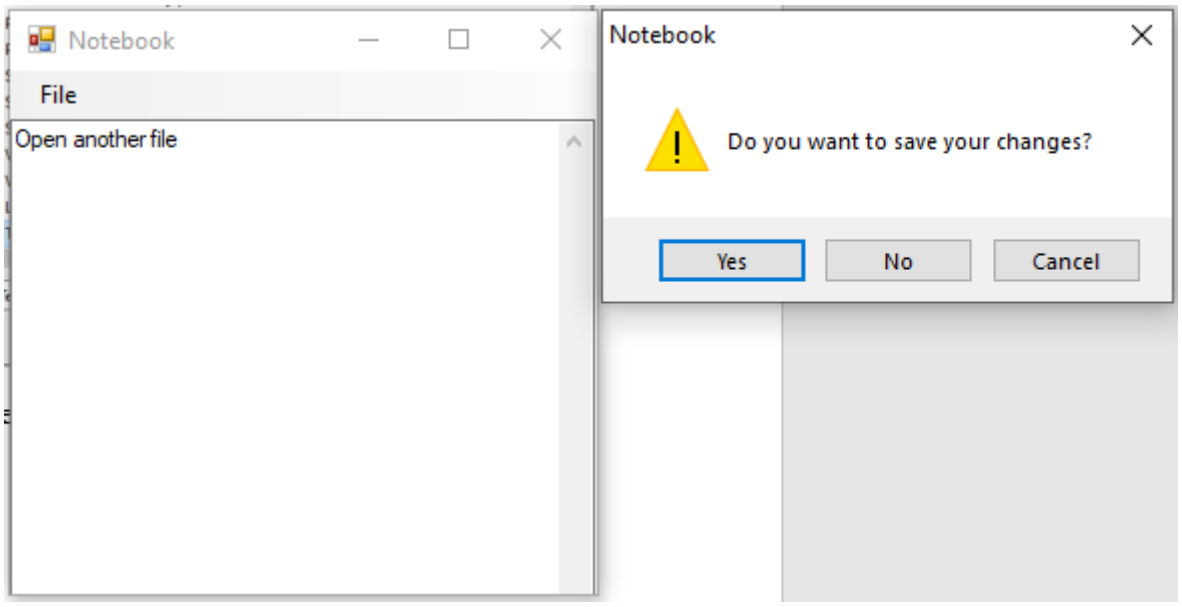
Notepad starts with an empty text box. In addition to entering text into the text field, the user can open another file, save changes to an existing file and save a new one (all with the * .txt extension). If he wants to choose an option, he clicks on File and selects the option, moreover, it is also possible to create a new file.



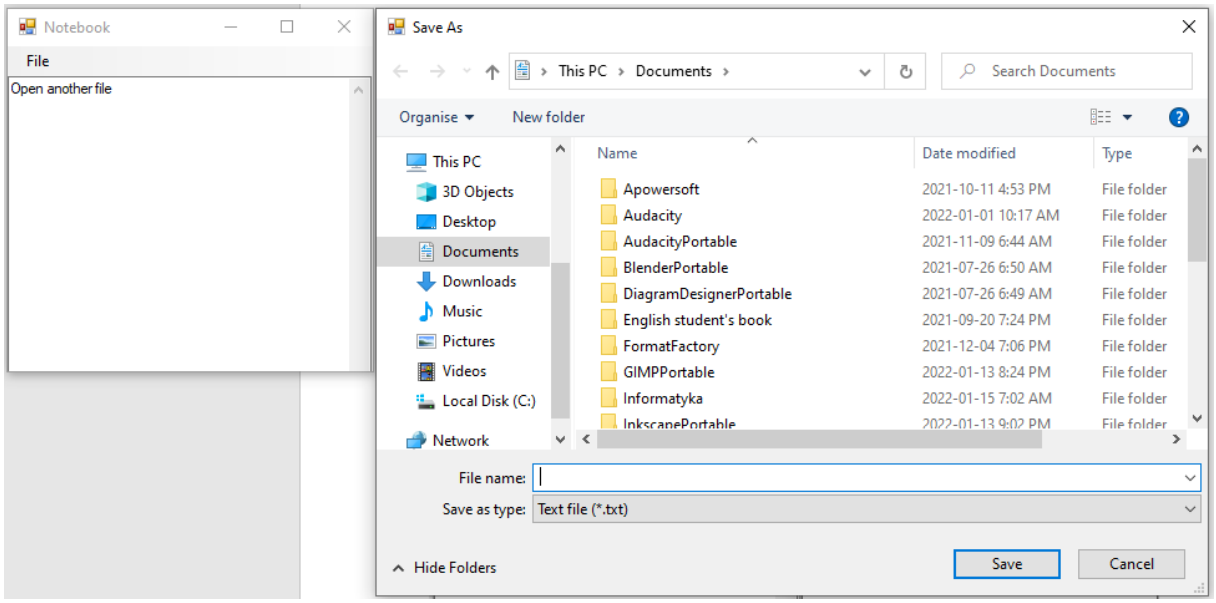
Drawing 5: File save dialog (New) after selecting Yes from File save dialog [own study]



Drawing 6: Open file dialog (text field empty) [own study]



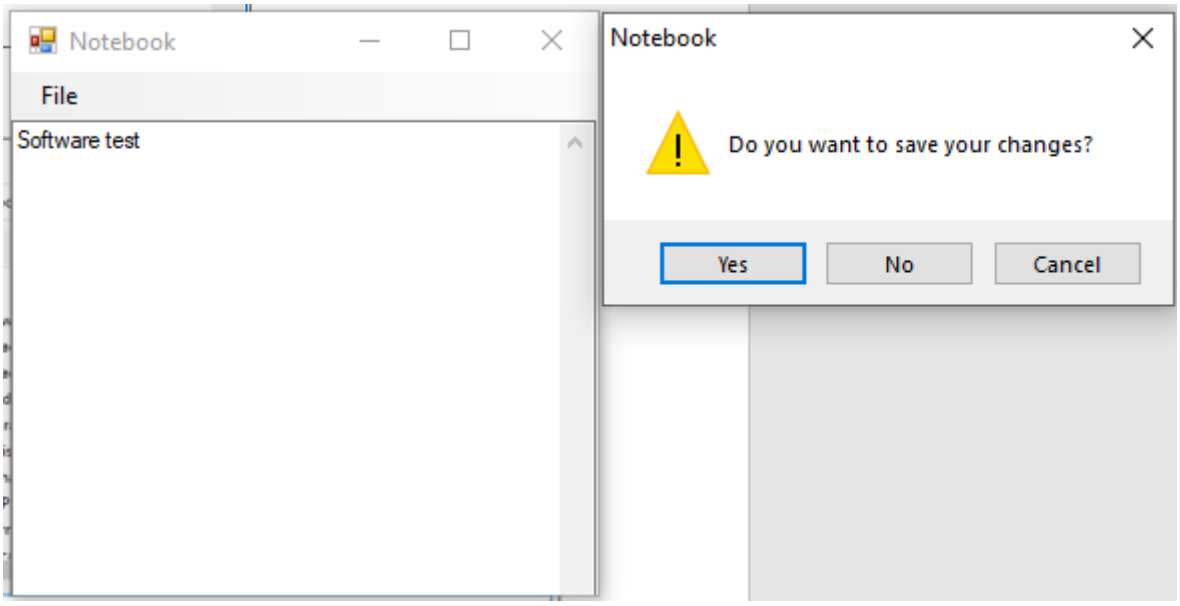
Drawing 7: Save changes dialog (after pressing Open) [own study]



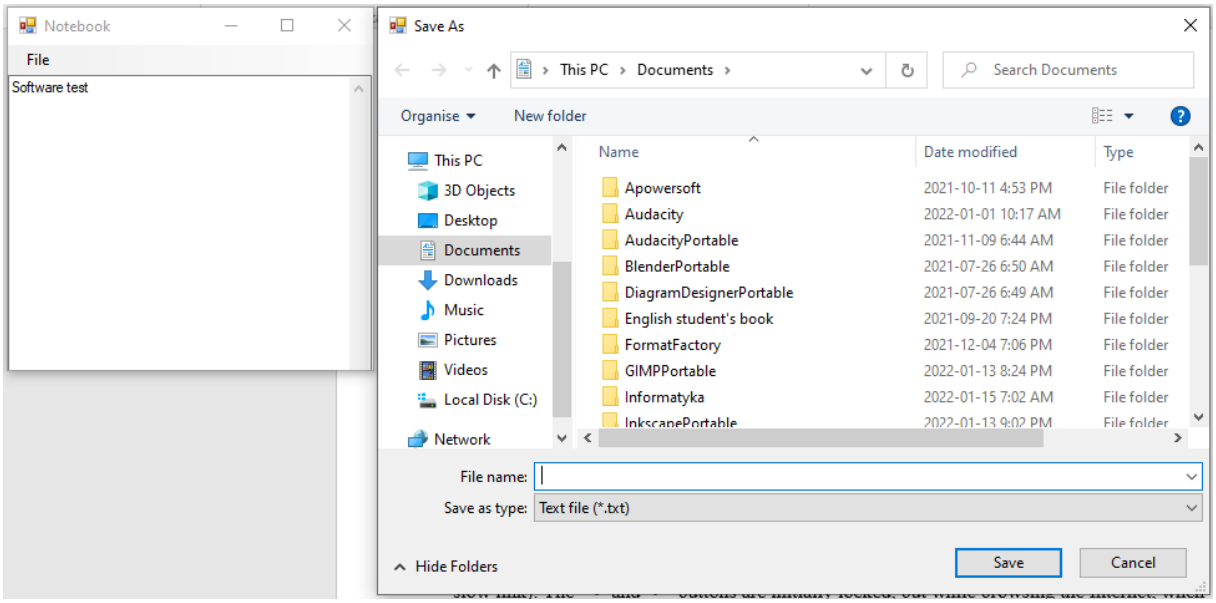
Drawing 8: File save dialog (Open) after selecting Yes from File save dialog [own study]

The new file function works for a blank field and for content in a text field. If there is any text, it is possible to save it. When wishing to open a different file, the user can also do so in these two situations. When there is text in the field, it is possible to save the content. He just needs to confirm it. The save option is closely related to the save as function. It works on the same principle. The previous functions, new and open, are linked to the recording when the user makes the appropriate decision. The user has the option to save the text when he presses the X in the upper right corner of the application pane.

There are certainly shortcomings of the application that its creator did not detect.



Drawing 9: Save changes dialog (after pressing X) [own study]



Drawing 10: Save changes dialog (options Save i Save As) [own study]

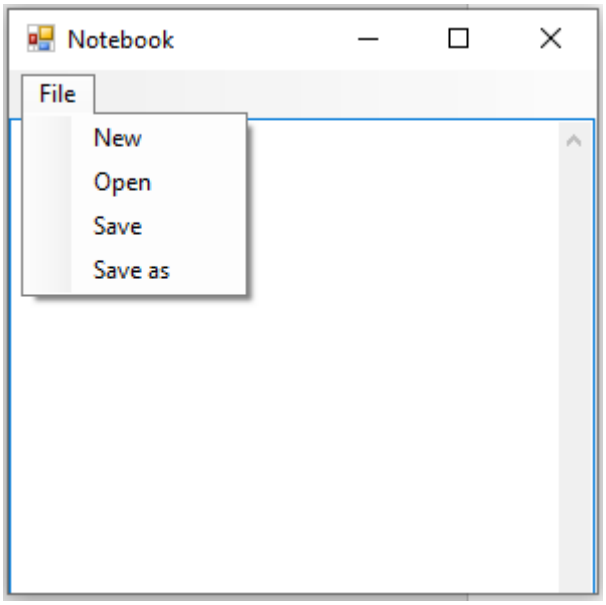
What is needed for use?

The application does not require installation. It only needs the Windows operating system as it runs in the same session as the Windows system notebook. However, this is a truncated version.

Algorithm used

The basic form of the algorithm can be deduced from the previous section. To sum up, the application is a typical notebook compared to the Windows system notebook.

Interface description



Drawing 11: Graphical interface [own study]

The interface is typical for a Windows Forms Application. There are essential components: menustrip and text box.

Source code description

The project was made in the C# programming language, in the Visual Studio Community 2017 programming environment. All work was done on the Windows 10 operating system. The application's source code looks like this.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO;

namespace Notebook
{
    public partial class frmNotatnik : Form
    {
        private string plik = "";

        public frmNotatnik()
        {
            InitializeComponent();
        }
```

```
        private DialogResult Czyzapisac()
        {
            DialogResult odp = MessageBox.Show("Do you want to save your changes?",
"Notebook",
            MessageBoxButtons.YesNoCancel, MessageBoxIcon.Exclamation);
            if (odp == DialogResult.Yes)
                MSave_Click(null, null);
            return odp;
        }

        private void MNowy_Click(object sender, EventArgs e)
        {
            if (TxtTresc.Text != "")
            {
                DialogResult odp = Czyzapisac();
                if (odp == DialogResult.Cancel)
                    return;
                plik = "";
                TxtTresc.Clear();
            }
        }

        private void MOpen_Click(object sender, EventArgs e)
        {
            if (TxtTresc.Text != "")
            {
                DialogResult odp = Czyzapisac();
                if (odp == DialogResult.Cancel)
                    return;
                plik = "";
                TxtTresc.Clear();
            }
            OpenFileDialog dialog = new OpenFileDialog();
            dialog.Filter = "Text file (*.txt)|*.txt";
            dialog.Multiselect = false;
            dialog.ShowDialog();
            if (dialog.FileName != "")
            {
                plik = dialog.FileName;
                StreamReader f = new StreamReader(plik);
                TxtTresc.Text = f.ReadToEnd();
                f.Close();
            }
        }

        private void MSave_Click(object sender, EventArgs e)
        {
            if (plik != "")
            {
                StreamWriter f = new StreamWriter(plik);
                f.Write(TxtTresc.Text);
                f.Close();
            }
            else MSaveAs_Click(sender, e);
        }

        private void MSaveAs_Click(object sender, EventArgs e)
        {
            SaveFileDialog dialog = new SaveFileDialog();
            dialog.Filter = "Text file (*.txt)|*.txt";
            dialog.ShowDialog();
            if (dialog.FileName != "")
```

```
            {
                plik = dialog.FileName;
                StreamWriter f = new StreamWriter(plik);
                f.Write(TxtTresc.Text);
                f.Close();
            }
        }

        private void frmNotatnik_FormClosing(object sender, FormClosingEventArgs e)
        {
            if (TxtTresc.Text != "")
            {
                DialogResult odp = Czyzapisac();
                if (odp == DialogResult.Cancel)
                    e.Cancel = true;
            }
        }
    }
}
```

Listing 1: Source code [own study]

List of drawings

List of listings