|  |
| --- |
| College LaSalle |
| Project - Oriented Object Programming User and Technical Manual |
|  |
| Presented to: Mihai Maftei. |

|  |
| --- |
| Karina Piekarski Siviero Contini  4/18/2023  Version: 1.5 |

1. **Short description of your project, and the languages (technologies) used:**
2. Language: C#
3. Tools (IDE): Visual Studio 2022

Multiform (.NET Framework) project, that covers 6 applications:

* Lotto Max
* Lotto 649
* Money Exchange
* Temperature Conversion
* Simple Calculator
* IPv4 Validator

1. **Forms and a detailed description of the functionalities:**

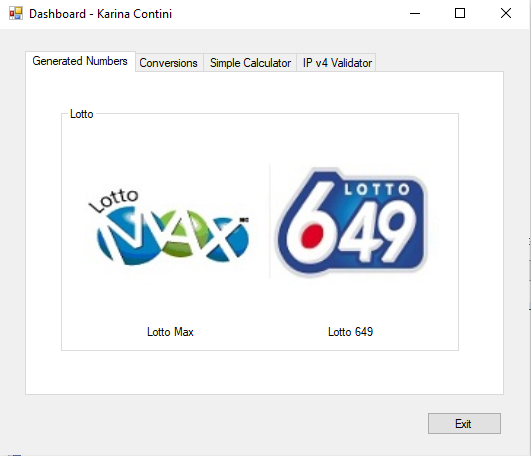


Figure – Dashboard (Generated Numbers as default tab)

The main form is the Dashboard. It has 4 tabs. As presented on Figure 1, the tab “Generated Numbers” is shown by default. This tab has two images (Lotto Max and Lotto 649).

1. If you click on the image Lotto Max, the Lotto Max application will be open, as presented on Figure 2.
   1. If you click on the button Generate, 8 unique numbers will be randomly generated and presented to the user in a text box. They will also be saved in a LottoNbrs.txt text file, identified by the name of the lottery and the current date and time.
   2. If you click on the button Read File, the file LottoNbrs.txt will be displayed in a text message with all records.
   3. If you click on the button Exit, the user can close the application by pressing Yes, or return to the application by pressing No.

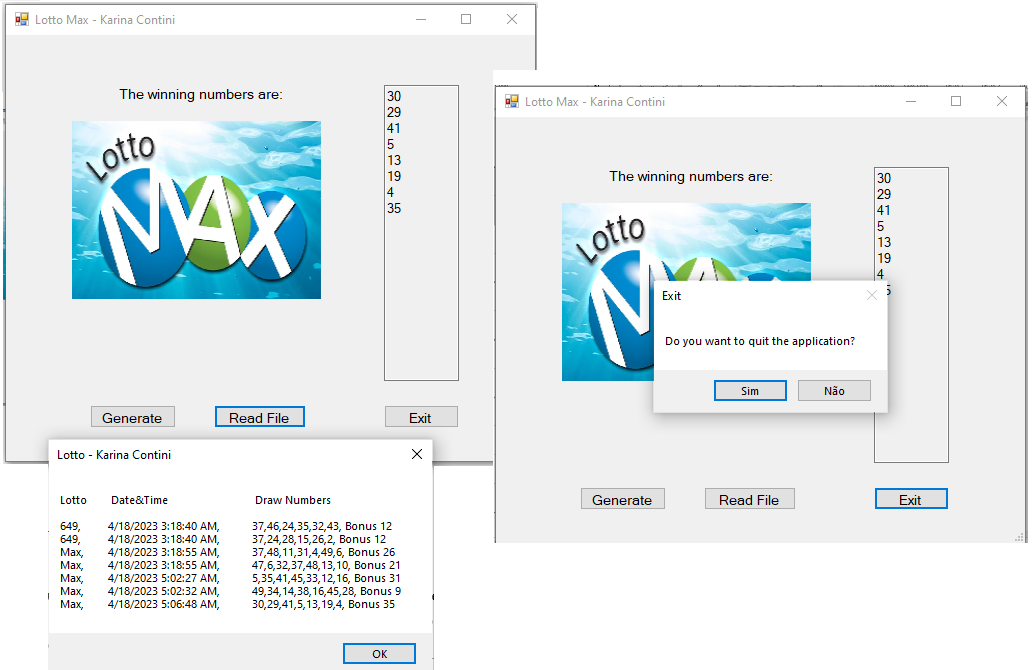


Figure - Lotto Max application

1. If you click on the image Lotto 649, the Lotto 649 application will be open, as presented on Figure 3.
   1. If you click on the button Generate, 7 unique numbers will be randomly generated and presented to the user in a text box. They will also be saved in a LottoNbrs.txt text file, identified by the name of the lottery and the current date and time.
   2. If you click on the button Read File, the file LottoNbrs.txt will be displayed in a text message with all records.
   3. If you click on the button Exit, the user can close the application by pressing Yes, or return to the application by pressing No.

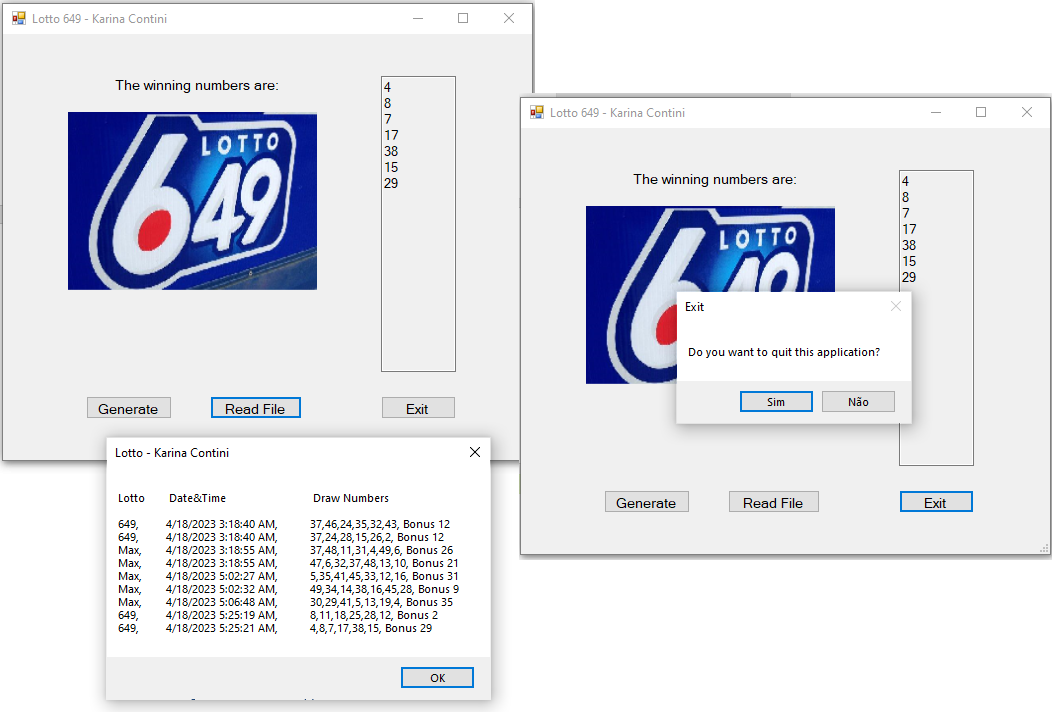


Figure - Lotto 649 application

1. If you click on the Conversions tb, the Dashboard will show two images (Money Exchange and Temperature Convert) as presented on Figure 4;

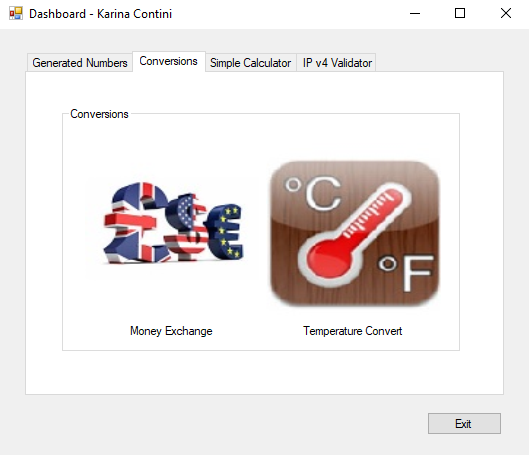


Figure - Conversions tab

1. If you click on the image Money Exchange, the Money Exchange application will be open as presented on Figure 5. The application allows the choice between 5 currencies (CAD, USD, EUR, GBP, AUD).
   1. If you click on the button Convert, the application will convert the amount entered on the text box from/to the selected currencies. Each conversion will also be saved in a MoneyConversions.txt text file, identified with the current date and time.
   2. If you click on the button Read File, the file MoneyConversions.txt will be displayed in a text message with all records.
   3. If you click on the button Exit, the user can close the application by pressing Yes, or return to the application by pressing No. After closing the application, the usage time of the form will be displayed in a message, as presented on Figure 6.

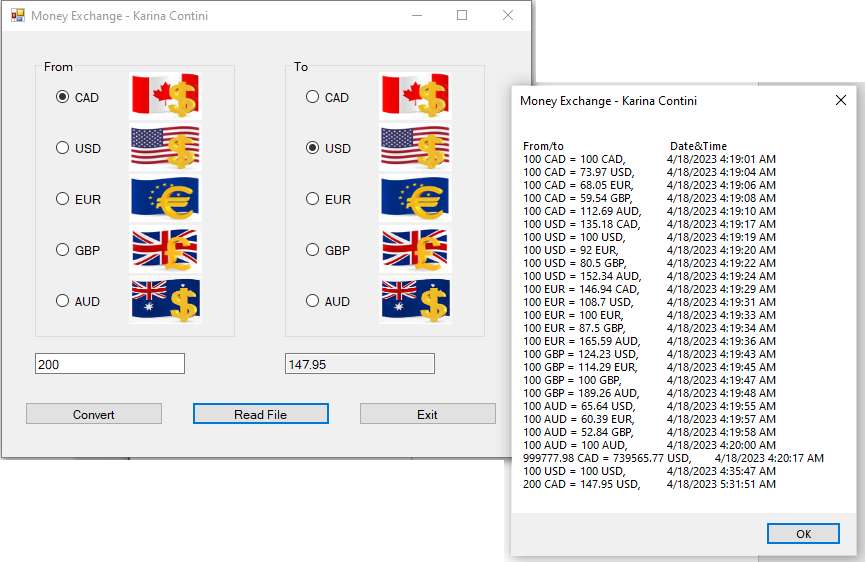


Figure - Money Exchange

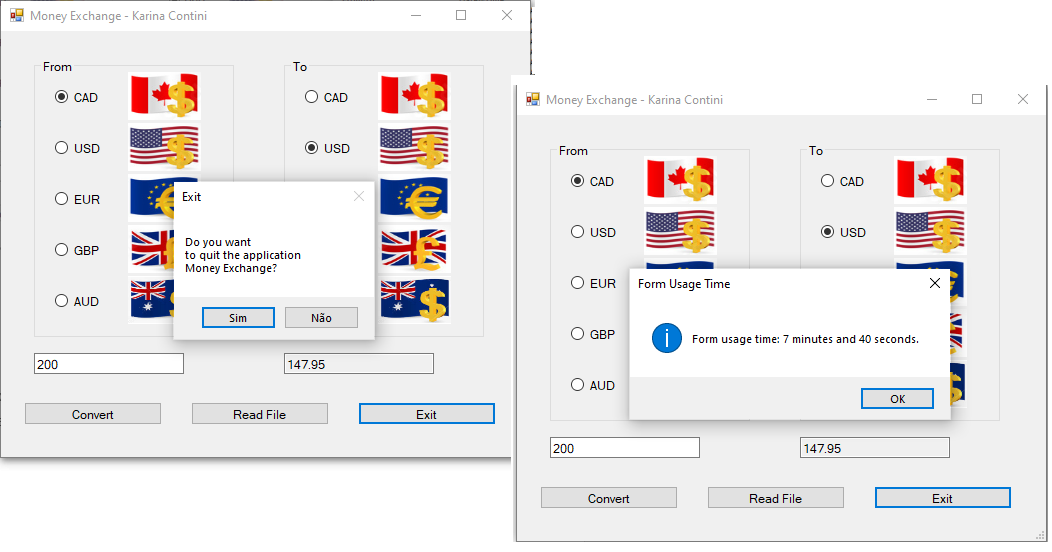


Figure - Money Exchange - Exit and Form Usage Time

1. If you click on the image Temperature Convert, the Temperature Conversions application will be open as presented on Figure 7. The application allows the choice for conversion from °C to °F or from °F to °C.
   1. If you click on the button Convert, the application will convert the temperature entered on the text box from/to the selected scales. Each conversion will also be registered in a TempConversions.txt text file, identified with the current date and time and a respective message (if applicable).
   2. If you click on the button Read File, the file TempConversions.txt will be displayed in a text message with all records.
   3. If you click on the button Exit, the user can close the application by pressing Yes, or return to the application by pressing No.

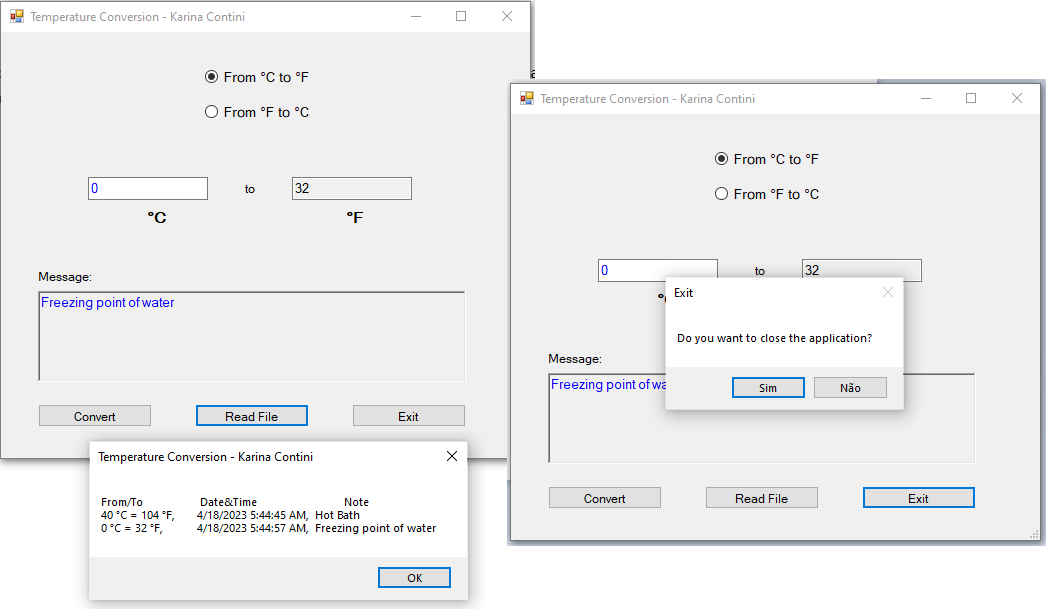


Figure - Temperature Conversion

1. If you click on the Simple Calculator tab, the Dashboard will show an image of a calculator as presented on Figure 8.

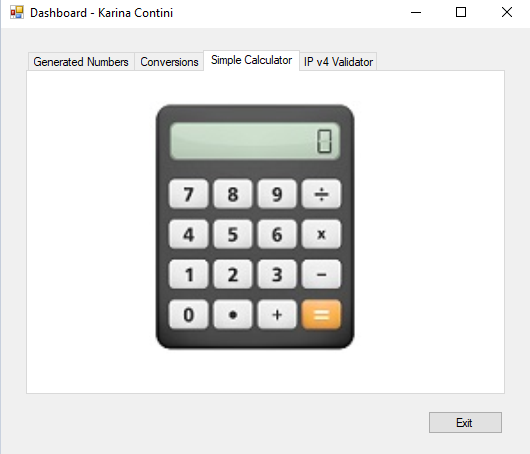


Figure - Simple Calculator tab

1. If you click on the image of the calculator, the Simple Calculator application will be open as presented on Figure 9. The application allows operations “+”, “-”, “\*”, “/” between the numbers entered.
   1. Each operation will also be registered in a Calculator.txt text file.
   2. If you click on the button Clear, all the values ​​in operation will be cleared.
   3. If you click on the button Exit, the user can close the application by pressing Yes, or return to the application by pressing No.

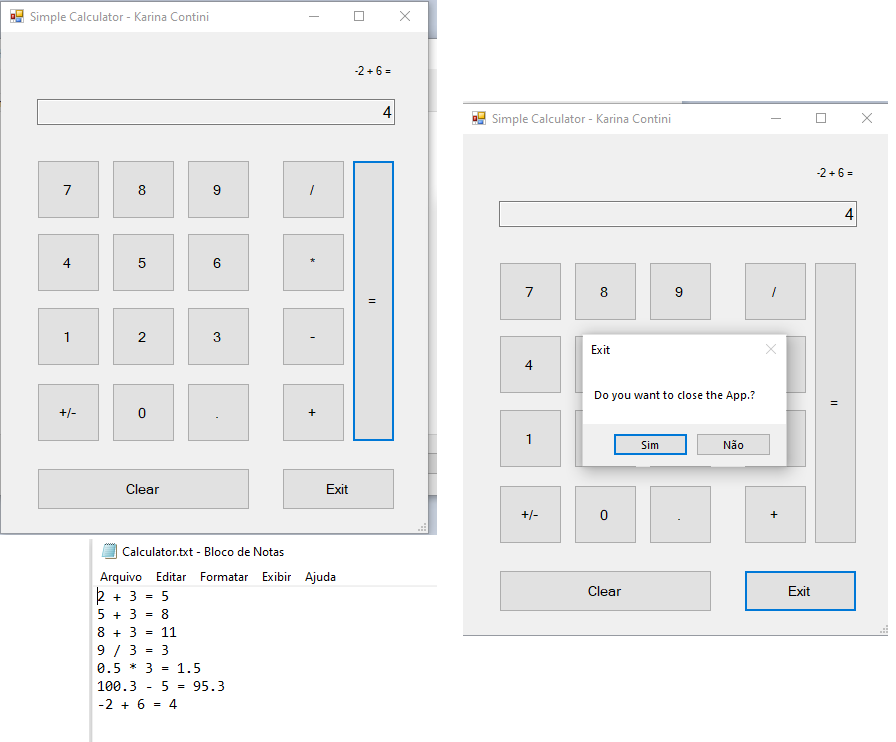


Figure - Calculator

1. If you click on IP v4 Validator tab, the Dashboard will show an image for the IP Validator as presented on Figure 10.

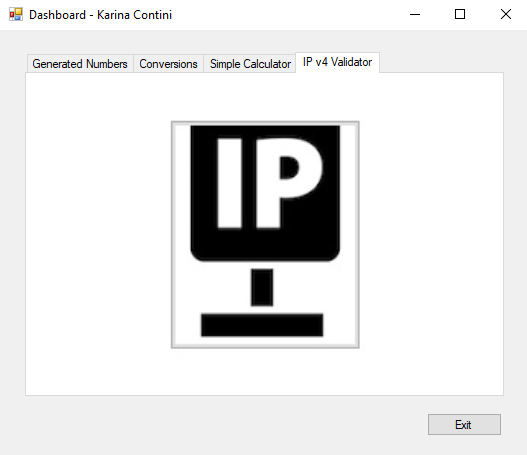


Figure - IPv4 Validator tab

1. If you click on the image of IP Validator, the IP v4 Validator application will be open as presented on Figure 11.
   1. If you click on the button Validate IP, the application will check the IP address inserted on the text box and show a message if it is valid or not. Each valid IP address will also be saved in a IP\_binary.dat file, identified with the current date and time.
   2. If you click on the button Reset, it will clear the text box for a new input.
   3. If you click on the button Exit, the user can close the application by pressing Yes, or return to the application by pressing No. After closing the application, the usage time of the form will be displayed in a message.

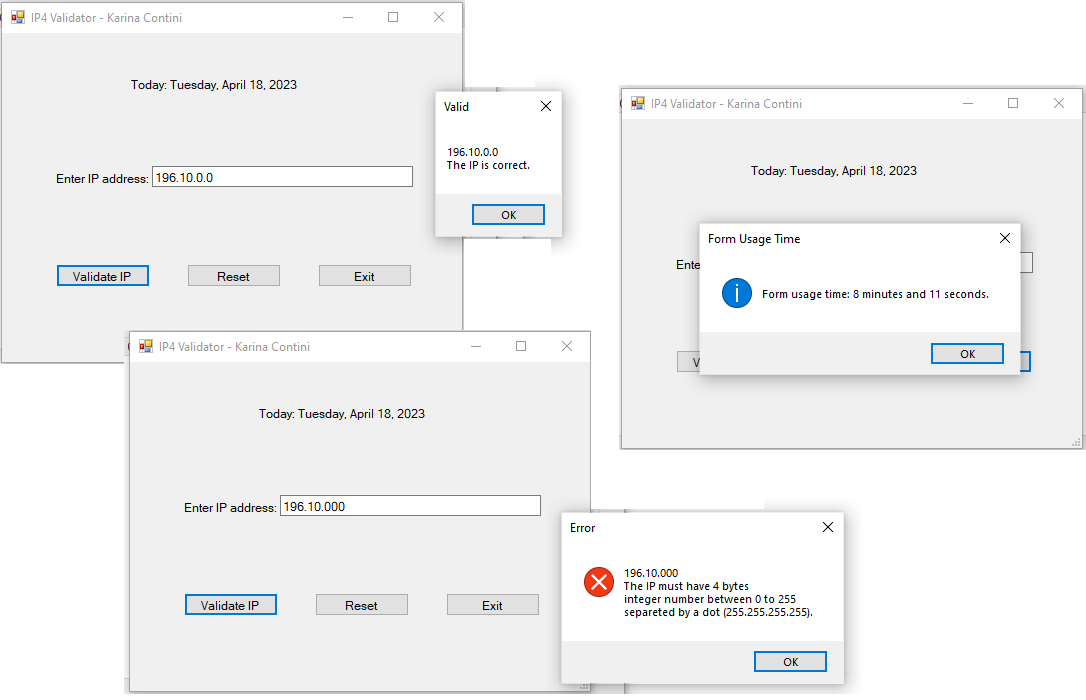


Figure - IP4 Validator

* 1. If you click on the Exit button on Dashboard, the main application will be closed.

1. **Code of application (forms).**

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;

/\*

Karina Contini - 2230591

4/18/2023

Final project

Form Dashboard with 4 tabs calling 6 applications:

- Lotto Max

- Lotto 649

- Money Exchange

- Temperature Conversion

- Simple Calculator

- IP Validator.

\*/

namespace FinalProject\_KarinaContini

{

public partial class FrmDashboard : Form

{

public FrmDashboard()

{

InitializeComponent();

}

private void btnExit\_Click(object sender, EventArgs e)

{

if (MessageBox.Show("Do you want to close the application?", "Exit", MessageBoxButtons.YesNo).ToString() == "Yes")

{

Application.Exit();

}

}

private void pbxLottoMax\_Click(object sender, EventArgs e)

{

FrmLottoMax obj = new FrmLottoMax();

obj.ShowDialog();

}

private void pbxLotto649\_Click(object sender, EventArgs e)

{

FrmLotto649 obj = new FrmLotto649();

obj.ShowDialog();

}

private void pbxTempConversion\_Click(object sender, EventArgs e)

{

FrmTempConversions obj = new FrmTempConversions();

obj.ShowDialog();

}

private void pbxIP\_Validator\_Click(object sender, EventArgs e)

{

FrmIP4\_Validator obj = new FrmIP4\_Validator();

obj.ShowDialog();

}

private void pbxMoneyConversion\_Click(object sender, EventArgs e)

{

FrmMoneyConversions obj = new FrmMoneyConversions();

obj.ShowDialog();

}

private void pbxCalculator\_Click(object sender, EventArgs e)

{

FrmCalculator obj = new FrmCalculator();

obj.ShowDialog();

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace FinalProject\_KarinaContini

{

public partial class FrmLotto649 : Form

{

public FrmLotto649()

{

InitializeComponent();

}

private void btnExit\_Click(object sender, EventArgs e)

{

if (MessageBox.Show("Do you want to quit this application?", "Exit", MessageBoxButtons.YesNo).ToString() == "Yes")

{

this.Close();

}

else

{

btnGenerate.Focus();

}

}

string filePath = @".\LottoNbrs.txt";

private void btnGenerate\_Click(object sender, EventArgs e)

{

Random random = new Random();

int[] randomNumber = new int[7]; // to generate from 1 to 49 range should be random.Next(1, 50);

txtLotto649.Clear();

DateTime currentDateTime = DateTime.Now;

string current = currentDateTime.ToShortDateString()+" "+currentDateTime.ToLongTimeString();

FileStream fs1 = new FileStream(filePath, FileMode.Append, FileAccess.Write);

StreamWriter objW = new StreamWriter(fs1);

objW.Write("649,\t"+ current+",\t");

for (int i = 0; i < randomNumber.Length; i++)

{

randomNumber[i] = random.Next(1,50);

for (int k = 0; k < i; k++)

{

while (randomNumber[k] == randomNumber[i])

{

randomNumber[i] = random.Next(1, 50);

k = 0;

}

}

txtLotto649.Text += randomNumber[i].ToString() + "\t";

if(i == randomNumber.Length - 1)

objW.WriteLine(" Bonus "+randomNumber[i].ToString());

else

objW.Write(randomNumber[i].ToString() + ",");

}

objW.Close();

fs1.Close();

}

private void btnReadFile\_Click(object sender, EventArgs e)

{

FileStream fs2 = null;

try

{

fs2 = new FileStream(filePath, FileMode.OpenOrCreate,

FileAccess.Read);

// code that uses the file stream to read and write data file

// create the object for the input stream for a text file

StreamReader textIn = new StreamReader(fs2);

string textToPrint = "Lotto\t Date&Time\t\t Draw Numbers\n\n";

// read the data from the file and store it in the list

string content = textIn.ReadToEnd();

textToPrint += content;

if (content != null || content.Trim() != null)

{

MessageBox.Show(textToPrint,"Lotto - Karina Contini", MessageBoxButtons.OK);

}

else

MessageBox.Show("There are no registers.", "Lotto - Karina Contini", MessageBoxButtons.OK);

// close the input stream for the text file

textIn.Close();

}

catch (IOException ex)

{ MessageBox.Show(ex.Message, "IOException"); }

finally { if (fs2 != null) fs2.Close(); }

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace FinalProject\_KarinaContini

{

public partial class FrmLottoMax : Form

{

public FrmLottoMax()

{

InitializeComponent();

}

private void btnExit\_Click(object sender, EventArgs e)

{

if (MessageBox.Show("Do you want to quit the application?", "Exit", MessageBoxButtons.YesNo).ToString() == "Yes")

{

this.Close();

}

else

{

btnGenerate.Focus();

}

}

string filePath = @".\LottoNbrs.txt";

private void btnGenerate\_Click(object sender, EventArgs e)

{

Random random = new Random();

int[] randomNumber = new int[8];

txtLottoMax.Clear();

DateTime currentDateTime = DateTime.Now;

string current = currentDateTime.ToShortDateString() + " " + currentDateTime.ToLongTimeString();

FileStream fs1 = new FileStream(filePath, FileMode.Append, FileAccess.Write);

StreamWriter objW = new StreamWriter(fs1);

objW.Write("Max,\t" + current + ",\t");

for (int i = 0; i < randomNumber.Length; i++)

{

randomNumber[i] = random.Next(1, 51);

for (int k = 0; k < i; k++)

{

while (randomNumber[k] == randomNumber[i])

{

randomNumber[i] = random.Next(1, 51);

k = 0;

}

}

txtLottoMax.Text += randomNumber[i].ToString() + "\t";

if (i == randomNumber.Length - 1)

objW.WriteLine(" Bonus " + randomNumber[i].ToString());

else

objW.Write(randomNumber[i].ToString() + ",");

}

objW.Close();

fs1.Close();

}

private void btnReadFile\_Click(object sender, EventArgs e)

{

FileStream fs2 = null;

try

{

fs2 = new FileStream(filePath, FileMode.OpenOrCreate,

FileAccess.Read);

// code that uses the file stream to read and write data file

// create the object for the input stream for a text file

StreamReader textIn = new StreamReader(fs2);

string textToPrint = "Lotto\t Date&Time\t\t Draw Numbers\n\n";

// read the data from the file and store it in the list

string content = textIn.ReadToEnd();

textToPrint += content;

if (content != null && content.Trim() != null)

{

MessageBox.Show(textToPrint, "Lotto - Karina Contini", MessageBoxButtons.OK);

}

else

MessageBox.Show("There are no registers.", "Lotto - Karina Contini", MessageBoxButtons.OK);

// close the input stream for the text file

textIn.Close();

}

catch (IOException ex)

{ MessageBox.Show(ex.Message, "IOException"); }

finally { if (fs2 != null) fs2.Close(); }

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading.Tasks;

using System.Windows.Forms;

using static System.Net.WebRequestMethods;

using static System.Windows.Forms.VisualStyles.VisualStyleElement.Button;

using static System.Windows.Forms.VisualStyles.VisualStyleElement;

using System.Xml.Linq;

using System.Data.SqlTypes;

/\*

\* Exchange factor - Apr16,2023

\*

\* 1 CAD = 0.73973 USD

\* 1 CAD = 0.68055 EUR

\* 1 CAD = 0.59545 GBP

\* 1 CAD = 1.12692 AUD;

\*/

namespace FinalProject\_KarinaContini

{

public partial class FrmMoneyConversions : Form

{

public FrmMoneyConversions()

{

InitializeComponent();

}

string filePath = @".\MoneyConversions.txt";

DateTime initialDateTime;

private void btnConvert\_Click(object sender, EventArgs e)

{

double inputMoney;

double convertedMoney;

string inputCode="", convertedCode="";

const double CAD\_FACTOR = 1;

const double USD\_FACTOR = 0.73973;

const double EUR\_FACTOR = 0.68055;

const double GBP\_FACTOR = 0.59545;

const double AUD\_FACTOR = 1.12692;

double iFactor = 0;

double oFactor = 0;

//check valid input

try

{

txtInputMoney.Text = txtInputMoney.Text.Trim();

inputMoney = Convert.ToDouble(txtInputMoney.Text);

Regex myRegex = new Regex(@"^(\d{0,6})(\.\d\*)?$");

if (myRegex.IsMatch(txtInputMoney.Text) == true)

{

//check input

if (rdBtnFromCAD.Checked)

{

iFactor = CAD\_FACTOR;

inputCode = "CAD";

}

else if(rdBtnFromUSD.Checked)

{

iFactor = USD\_FACTOR;

inputCode = "USD";

}

else if (rdBtnFromEUR.Checked)

{

iFactor = EUR\_FACTOR;

inputCode = "EUR";

}

else if (rdBtnFromGBP.Checked)

{

iFactor = GBP\_FACTOR;

inputCode = "GBP";

}

else if (rdBtnFromAUD.Checked)

{

iFactor = AUD\_FACTOR;

inputCode = "AUD";

}

//check input

if (rdBtnToCAD.Checked)

{

oFactor = CAD\_FACTOR;

convertedCode = "CAD";

}

else if (rdBtnToUSD.Checked)

{

oFactor = USD\_FACTOR;

convertedCode = "USD";

}

else if (rdBtnToEUR.Checked)

{

oFactor = EUR\_FACTOR;

convertedCode = "EUR";

}

else if (rdBtnToGBP.Checked)

{

oFactor = GBP\_FACTOR;

convertedCode = "GBP";

}

else if (rdBtnToAUD.Checked)

{

oFactor = AUD\_FACTOR;

convertedCode = "AUD";

}

// conversion

convertedMoney = (inputMoney \* oFactor) / iFactor;

//round 2 decimals

convertedMoney = Math.Round(convertedMoney, 2);

txtConvertedMoney.Text = convertedMoney.ToString();

//get date and time

DateTime currentDateTime = DateTime.Now;

string current = currentDateTime.ToShortDateString() + " " + currentDateTime.ToLongTimeString();

//write text file

FileStream fs1 = new FileStream(filePath, FileMode.Append, FileAccess.Write);

StreamWriter objW = new StreamWriter(fs1);

objW.Write(txtInputMoney.Text + " "+inputCode+" = " + txtConvertedMoney.Text +" " + convertedCode + ",\t");

objW.WriteLine(current);

objW.Close();

fs1.Close();

}

else

{

MessageBox.Show("Please, enter a positive number < 1,000,000.", "Attention", MessageBoxButtons.OK, MessageBoxIcon.Warning);

txtInputMoney.Focus();

}

}

catch (Exception)

{

MessageBox.Show("Please, enter a number to be converted.", "Attention", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

private void btnReadFile\_Click(object sender, EventArgs e)

{

FileStream fs2 = null;

try

{

fs2 = new FileStream(filePath, FileMode.OpenOrCreate,

FileAccess.Read);

// code that uses the file stream to read and write data file

// create the object for the input stream for a text file

StreamReader textIn = new StreamReader(fs2);

string textToPrint = "From/to\t\t\t Date&Time\n";

// read the data from the file and store it in the list

string content = textIn.ReadToEnd();

textToPrint += content;

if (content != null && content.Trim() != null)

{

MessageBox.Show(textToPrint, "Money Exchange - Karina Contini", MessageBoxButtons.OK);

}

else

MessageBox.Show("There are no registers.", "Money Exchange - Karina Contini", MessageBoxButtons.OK);

// close the input stream for the text file

textIn.Close();

}

catch (IOException ex)

{ MessageBox.Show(ex.Message, "IOException"); }

finally { if (fs2 != null) fs2.Close(); }

}

private void btnExit\_Click(object sender, EventArgs e)

{

if (MessageBox.Show("Do you want\nto quit the application\nMoney Exchange?", "Exit", MessageBoxButtons.YesNo).ToString() == "Yes")

{

DateTime finalTime = DateTime.Now;

TimeSpan difference = finalTime.Subtract(initialDateTime);

MessageBox.Show("Form usage time: " + difference.Minutes + " minutes and " + difference.Seconds + " seconds.", "Form Usage Time", MessageBoxButtons.OK, MessageBoxIcon.Information);

//+ difference.Hours + " hours, "

this.Close();

}

else

{

txtInputMoney.Focus();

}

}

private void FrmMoneyConversions\_Load(object sender, EventArgs e)

{

rdBtnFromCAD.Checked = true;

rdBtnToUSD.Checked = true;

initialDateTime = DateTime.Now;

}

private void grpFrom\_Enter(object sender, EventArgs e)

{

txtInputMoney.Clear();

txtConvertedMoney.Clear();

txtInputMoney.Focus();

}

private void grpTo\_Enter(object sender, EventArgs e)

{

txtConvertedMoney.Clear();

txtInputMoney.Focus();

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Text.RegularExpressions;

using System.IO;

namespace FinalProject\_KarinaContini

{

public partial class FrmTempConversions : Form

{

public FrmTempConversions()

{

InitializeComponent();

}

private void rdBtnCtoF\_CheckedChanged(object sender, EventArgs e)

{

lblInputTemp.Text = "°C";

lblConvertedTemp.Text = "°F";

txtConvertedTemp.Clear();

txtInputTemp.Clear();

txtMessage.Clear();

}

private void rdBtnFtoC\_CheckedChanged(object sender, EventArgs e)

{

lblInputTemp.Text = "°F";

lblConvertedTemp.Text = "°C";

txtConvertedTemp.Clear();

txtInputTemp.Clear();

txtMessage.Clear();

}

string filePath = @".\TempConversions.txt";

private void btnConvert\_Click(object sender, EventArgs e)

{

double temp;

double convertedTemp = 0;

double noteNumber = 0;

//check valid input

try

{

temp = Convert.ToDouble(txtInputTemp.Text.Trim());

Regex myRegex = new Regex(@"^-?(\d{0,3})(\.\d\*)?$"); //^(|-?\d+)$

if (myRegex.IsMatch(txtInputTemp.Text.Trim()) == true)

{

//make conversion

if (rdBtnCtoF.Checked)

{

convertedTemp = (temp \* 9) / 5 + 32;

noteNumber = temp;

}

else if (rdBtnFtoC.Checked)

{

convertedTemp = ((temp - 32) \* 5) / 9;

noteNumber = convertedTemp;

}

txtConvertedTemp.Text = convertedTemp.ToString();

//set the message and color

if (noteNumber < 103 && noteNumber >= 97)

{

txtMessage.ForeColor = Color.DarkRed;

txtInputTemp.ForeColor = Color.DarkRed;

txtMessage.Text = "Water boils";

}

else if (noteNumber < 43 && noteNumber >= 38)

{

txtMessage.ForeColor = Color.Red;

txtInputTemp.ForeColor = Color.Red;

txtMessage.Text = "Hot Bath";

}

else if (noteNumber < 37.6 && noteNumber >= 36.5)

{

txtMessage.ForeColor = Color.Orange;

txtInputTemp.ForeColor = Color.Orange;

txtMessage.Text = "Body temperature";

}

else if (noteNumber < 36.5 && noteNumber >= 30)

{

txtMessage.ForeColor = Color.Orange;

txtInputTemp.ForeColor = Color.Orange;

txtMessage.Text = "Beach weather";

}

else if (noteNumber < 25 && noteNumber >= 19)

{

txtMessage.ForeColor = Color.Green;

txtInputTemp.ForeColor = Color.Green;

txtMessage.Text = "Room temperature";

}

else if (noteNumber <= 10 && noteNumber > 0)

{

txtMessage.ForeColor = Color.DeepSkyBlue;

txtInputTemp.ForeColor = Color.DeepSkyBlue;

txtMessage.Text = "Cold Day";

}

else if (noteNumber == 0)

{

txtMessage.ForeColor = Color.Blue;

txtInputTemp.ForeColor = Color.Blue;

txtMessage.Text = "Freezing point of water";

}

else if (noteNumber < 0 && noteNumber >= -27)

{

txtMessage.ForeColor = Color.Blue;

txtInputTemp.ForeColor = Color.Blue;

txtMessage.Text = "Very cold day";

}

else if (noteNumber < -27 && noteNumber > -40)

{

txtMessage.ForeColor = Color.MediumPurple;

txtInputTemp.ForeColor = Color.MediumPurple;

txtMessage.Text = "Extremely Cold Day";

}

else if (noteNumber == -40)

{

txtMessage.ForeColor = Color.MediumPurple;

txtInputTemp.ForeColor = Color.MediumPurple;

txtMessage.Text = "Extremely Cold Day (and the same number!)";

}

//get date and time

DateTime currentDateTime = DateTime.Now;

string current = currentDateTime.ToShortDateString() + " " + currentDateTime.ToLongTimeString();

//write text file

FileStream fs1 = new FileStream(filePath, FileMode.Append, FileAccess.Write);

StreamWriter objW = new StreamWriter(fs1);

objW.Write(txtInputTemp.Text + " " + lblInputTemp.Text + " = " + txtConvertedTemp.Text + " " + lblConvertedTemp.Text + ",\t");

objW.Write(current + ", ");

objW.WriteLine(txtMessage.Text);

objW.Close();

fs1.Close();

}

else

{

MessageBox.Show("Please, enter a number < 1000 and > -1000.", "Attention", MessageBoxButtons.OK, MessageBoxIcon.Warning);

txtInputTemp.Focus();

}

}

catch (Exception)

{

lblInputTemp.ForeColor = Color.Red;

MessageBox.Show("Please, enter a number to be converted.", "Attention", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

private void btnExit\_Click(object sender, EventArgs e)

{

if (MessageBox.Show("Do you want to close the application?", "Exit", MessageBoxButtons.YesNo).ToString() == "Yes")

{

this.Close();

}

else

{

txtInputTemp.Focus();

}

}

private void btnReadFile\_Click(object sender, EventArgs e)

{

FileStream fs2 = null;

try

{

fs2 = new FileStream(filePath, FileMode.OpenOrCreate,

FileAccess.Read);

// code that uses the file stream to read and write data file

// create the object for the input stream for a text file

StreamReader textIn = new StreamReader(fs2);

string textToPrint = "From/To\t\t Date&Time\t\t Note\n";

// read the data from the file and store it in the list

string content = textIn.ReadToEnd();

textToPrint += content;

if (content != null && content.Trim() != null)

{

MessageBox.Show(textToPrint, "Temperature Conversion - Karina Contini", MessageBoxButtons.OK);

}

else

MessageBox.Show("There are no registers.", "Temperature Conversion - Karina Contini", MessageBoxButtons.OK);

// close the input stream for the text file

textIn.Close();

}

catch (IOException ex)

{ MessageBox.Show(ex.Message, "IOException"); }

finally { if (fs2 != null) fs2.Close(); }

}

private void txtInputTemp\_TextChanged(object sender, EventArgs e)

{

lblInputTemp.ForeColor = Color.Black;

txtInputTemp.ForeColor = Color.Black;

txtConvertedTemp.Clear();

txtMessage.Clear();

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace FinalProject\_KarinaContini

{

public partial class FrmCalculator : Form

{

public FrmCalculator()

{

InitializeComponent();

}

Calculator calc = new Calculator();

Boolean operatorOn = false;

Boolean equalOn = false;

string filePath = @".\Calculator.txt";

private void btnExit\_Click(object sender, EventArgs e)

{

if (MessageBox.Show("Do you want to close the App.?", "Exit", MessageBoxButtons.YesNo).ToString() == "Yes")

{

this.Close();

}

else

{

btnClear.Focus();

}

}

private void btnClear\_Click(object sender, EventArgs e)

{

calc.Clear();

txtDisplay.Text = "0";

lblRecord.Text = "";

}

private void btn1\_Click(object sender, EventArgs e)

{

if (txtDisplay.Text=="0" || operatorOn == true)

{

txtDisplay.Text = "1";

operatorOn = false;

equalOn = false;

}

else

{

txtDisplay.Text += "1";

}

}

private void btn2\_Click(object sender, EventArgs e)

{

if (txtDisplay.Text == "0" || operatorOn == true)

{

txtDisplay.Text = "2";

operatorOn = false;

equalOn = false;

}

else

{

txtDisplay.Text += "2";

}

}

private void btn3\_Click(object sender, EventArgs e)

{

if (txtDisplay.Text == "0" || operatorOn == true)

{

txtDisplay.Text = "3";

operatorOn = false;

equalOn = false;

}

else

{

txtDisplay.Text += "3";

}

}

private void btn4\_Click(object sender, EventArgs e)

{

if (txtDisplay.Text == "0" || operatorOn == true)

{

txtDisplay.Text = "4";

operatorOn = false;

equalOn = false;

}

else

{

txtDisplay.Text += "4";

}

}

private void btn5\_Click(object sender, EventArgs e)

{

if (txtDisplay.Text == "0" || operatorOn == true)

{

txtDisplay.Text = "5";

operatorOn = false;

equalOn = false;

}

else

{

txtDisplay.Text += "5";

}

}

private void btn6\_Click(object sender, EventArgs e)

{

if (txtDisplay.Text == "0" || operatorOn == true)

{

txtDisplay.Text = "6";

operatorOn = false;

equalOn = false;

}

else

{

txtDisplay.Text += "6";

}

}

private void btn7\_Click(object sender, EventArgs e)

{

if (txtDisplay.Text == "0" || operatorOn == true)

{

txtDisplay.Text = "7";

operatorOn = false;

equalOn = false;

}

else

{

txtDisplay.Text += "7";

}

}

private void btn8\_Click(object sender, EventArgs e)

{

if (txtDisplay.Text == "0" || operatorOn == true)

{

txtDisplay.Text = "8";

operatorOn = false;

equalOn = false;

}

else

{

txtDisplay.Text += "8";

}

}

private void btn9\_Click(object sender, EventArgs e)

{

if (txtDisplay.Text == "0" || operatorOn == true)

{

txtDisplay.Text = "9";

operatorOn = false;

equalOn = false;

}

else

{

txtDisplay.Text += "9";

}

}

private void btn0\_Click(object sender, EventArgs e)

{

if (txtDisplay.Text == "0" || operatorOn == true)

{

txtDisplay.Text = "0";

operatorOn = false;

equalOn = false;

}

else

{

txtDisplay.Text += "0";

}

}

private void btnDot\_Click(object sender, EventArgs e)

{

if (txtDisplay.Text == "" || operatorOn == true)

{

txtDisplay.Text = "0.";

operatorOn = false;

equalOn = false;

}

else

{

Regex myRegex = new Regex(@"^\d\*\.\d\*$");

if (myRegex.IsMatch(txtDisplay.Text) == true)

{

txtDisplay.Text = txtDisplay.Text;

}

else

{

txtDisplay.Text += ".";

}

}

}

private void btnAdd\_Click(object sender, EventArgs e)

{

if (txtDisplay.Text == "")

{

MessageBox.Show("Invalid format used.");

}

else

{

lblRecord.Text = txtDisplay.Text + " + ";

operatorOn = true;

equalOn = false;

calc.Add(Convert.ToDecimal(txtDisplay.Text));

}

}

private void btnSubt\_Click(object sender, EventArgs e)

{

if (txtDisplay.Text == "")

{

MessageBox.Show("Invalid format used.");

}

else

{

lblRecord.Text = txtDisplay.Text + " - ";

operatorOn = true;

equalOn = false;

calc.Subtract(Convert.ToDecimal(txtDisplay.Text));

}

}

private void btnMult\_Click(object sender, EventArgs e)

{

if (txtDisplay.Text == "")

{

MessageBox.Show("Invalid format used.");

}

else

{

lblRecord.Text = txtDisplay.Text + " \* ";

operatorOn = true;

equalOn = false;

calc.Multiply(Convert.ToDecimal(txtDisplay.Text));

}

}

private void btnDiv\_Click(object sender, EventArgs e)

{

if (txtDisplay.Text == "")

{

MessageBox.Show("Invalid format used.");

}

else

{

lblRecord.Text = txtDisplay.Text + " / ";

operatorOn = true;

equalOn = false;

calc.Divide(Convert.ToDecimal(txtDisplay.Text));

}

}

private void btnEqual\_Click(object sender, EventArgs e)

{

if (equalOn == true)

{

int index = lblRecord.Text.IndexOf(" ");

lblRecord.Text = calc.CurrentValue.ToString() + lblRecord.Text.Substring(index);

calc.Equals();

operatorOn = true;

txtDisplay.Text = calc.CurrentValue.ToString();

}

else

{

calc.Equals(Convert.ToDecimal(txtDisplay.Text));

operatorOn = true;

lblRecord.Text += txtDisplay.Text + " =";

txtDisplay.Text = calc.CurrentValue.ToString();

equalOn = true;

}

//write text file

FileStream fs1 = new FileStream(filePath, FileMode.Append, FileAccess.Write);

StreamWriter objW = new StreamWriter(fs1);

objW.Write(lblRecord.Text+" ");

objW.WriteLine(calc.CurrentValue.ToString());

objW.Close();

fs1.Close();

}

private void btnSignal\_Click(object sender, EventArgs e)

{

calc.Signal(Convert.ToDecimal(txtDisplay.Text));

txtDisplay.Text = calc.CurrentValue.ToString();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace FinalProject\_KarinaContini

{

internal class Calculator

{

//private fields

private decimal currentValue; //A decimal that stores the result currently displayed by the calculator

private decimal operand1; //A decimal that stores the value of the first operand.

private decimal operand2; //A decimal that stores the value of the second operand.

private string op; //A string type that stores the value of the operator

//Properties

public decimal CurrentValue

{

get { return currentValue; }

set { currentValue = value; }

}

public Calculator()

{ } //Creates a Calculator object with default values. The default value for the op field is Null.

////Sets the operand1 and currentValue fields to the value that’s passed to it and sets the op field to "+".

public void Add(decimal displayValue)

{

operand1 = displayValue;

CurrentValue = displayValue;

op = "+";

}

//Sets the operand1 and currentValue fields to the value that’s passed to it and sets the op field to "-".

public void Subtract(decimal displayValue)

{

operand1 = displayValue;

CurrentValue = displayValue;

op = "-";

}

//Sets the operand1 and currentValue fields to the value that’s passed to it and sets the op field to "\*".

public void Multiply(decimal displayValue)

{

operand1 = displayValue;

CurrentValue = displayValue;

op = "\*";

}

//Sets the operand1 and currentValue fields to the value that’s passed to it and sets the op field to "/".

public void Divide(decimal displayValue)

{

operand1 = displayValue;

CurrentValue = displayValue;

op = "/";

}

public void Signal(decimal displayValue)

{

//operand1 = displayValue \* (-1);

CurrentValue = displayValue \* (-1);

}

//Performs the operation specified by the op field on the operand1 and operand2 fields, and stores the result in the operand1 field.

public void Equals()

{

operand1 = CurrentValue;

if (op == "+")

{

CurrentValue = operand1 + operand2;

}

else if (op == "-")

{

CurrentValue = operand1 - operand2;

}

else if (op == "\*")

{

CurrentValue = operand1 \* operand2;

}

else if (op == "/" && operand2 != 0)

{

CurrentValue = operand1 / operand2;

}

else if (op == "/" && operand2 == 0)

{

MessageBox.Show("Cannot divide by zero.", "Attention", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

//Sets the operand2 field to the value that’s passed to it.Then, performs the operation specified by the op field on the operand1 and operand2 fields, and stores the result in the operand1 field.

public void Equals(decimal displayValue)

{

operand2 = displayValue;

if (op == "+")

{

CurrentValue = operand1 + operand2;

}

else if (op == "-")

{

CurrentValue = operand1 - operand2;

}

else if (op == "\*")

{

CurrentValue = operand1 \* operand2;

}

else if (op == "/" && operand2 != 0)

{

CurrentValue = operand1 / operand2;

}

else if (op == "/" && operand2 == 0)

{

MessageBox.Show("Cannot divide by zero.", "Attention", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

//Sets the private fields to their default values.

public void Clear()

{

operand1 = 0;

operand2 = 0;

CurrentValue = 0;

op = "";

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading.Tasks;

using System.Windows.Forms;

using static System.Windows.Forms.VisualStyles.VisualStyleElement;

namespace FinalProject\_KarinaContini

{

public partial class FrmIP4\_Validator : Form

{

public FrmIP4\_Validator()

{

InitializeComponent();

}

string pathBin = @".\IP\_binary";

DateTime initialDateTime;

private void FrmIP4\_Validator\_Load(object sender, EventArgs e)

{

initialDateTime = DateTime.Now;

string initial = initialDateTime.ToLongDateString();

lblDay.Text = "Today: "+initial;

}

private void btnExit\_Click(object sender, EventArgs e)

{

if (MessageBox.Show("Do you want to quit the application?", "Exit", MessageBoxButtons.YesNo).ToString() == "Yes")

{

DateTime finalTime = DateTime.Now;

TimeSpan difference = finalTime.Subtract(initialDateTime);

MessageBox.Show("Form usage time: "+difference.Minutes+" minutes and "+difference.Seconds+" seconds.", "Form Usage Time", MessageBoxButtons.OK, MessageBoxIcon.Information);

//+difference.Hours+" hours, "

this.Close();

}

else

{

txtIP\_Address.Focus();

}

}

private void btnValidateIP\_Click(object sender, EventArgs e)

{

txtIP\_Address.Text = txtIP\_Address.Text.Trim();

try

{

Regex myRegex = new Regex(@"^(25[0-5]|2[0-4]\d|[0-1]?\d?\d)(\.(25[0-5]|2[0-4]\d|[0-1]?\d?\d)){3}$");

if (myRegex.IsMatch(txtIP\_Address.Text) == true)

{

DateTime currentDateTime = DateTime.Now;

string current = currentDateTime.ToShortDateString() + " " + currentDateTime.ToLongTimeString();

FileStream fs3 = new FileStream(pathBin, FileMode.Append, FileAccess.Write);

BinaryWriter binaryOut = new BinaryWriter(fs3);

// write the fields into text file

binaryOut.Write(txtIP\_Address.Text);

binaryOut.Write(current);

// close the output stream for the text file

binaryOut.Close();

fs3.Close();

MessageBox.Show(txtIP\_Address.Text + "\nThe IP is correct.", "Valid", MessageBoxButtons.OK);

}

else

{

MessageBox.Show(txtIP\_Address.Text + "\nThe IP must have 4 bytes\ninteger number between 0 to 255\nsepareted by a dot (255.255.255.255).", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

catch (Exception ex)

{

Console.WriteLine("Error: " + ex.Message);

}

}

private void btnReset\_Click(object sender, EventArgs e)

{

txtIP\_Address.Clear();

txtIP\_Address.Focus();

}

}

}

1. **Classes and/or methods created in the project.**

|  |  |
| --- | --- |
| **Class/Method Name** | **Description** |
| 1. Class FrmDashboard | Create the form Dashboard that can call the other applications. |
| 1. Class FrmLotto649 | Create the form for the Lotto 649 application. |
| 1. Class FrmLottoMax | Create the form for the Lotto Max application. |
| 1. Class FrmMoneyConversions | Create the form for the Money Exchange application. |
| 1. Class FrmTempConversions | Create the form for the Temperature Convert application. |
| 1. Class FrmCalculator | Create the form for the Simple Calculator application. |
| 1. Class FrmIP4-Validator | Create the form for the IP v4 Validator application. |
| 1. Class Calculator | Performs the required basic operation on the Calculator application. |

1. **Difficulties that I had, what was the hardest and the easiest part of the project.**

The biggest difficulty was managing time. The project is full of details and extensive.

On the other hand, throughout the course we acquire the necessary knowledge to develop it.