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

|  |
| --- |
| Your name: Rakotoarimanana Tsiory  10/7/2023  Version 3.0 |

1. **Start by adding a short description of your project, and the languages (technologies) used:**
2. Language C#
3. Used tool(s): Visual Studio version 2022

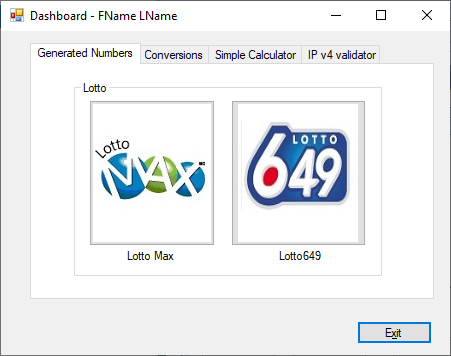
The first section is about a widows form which gives the user the possibility to generate numbers:

By selecting either lotto 649 or lotto max 7 to 8 random numbers will be displayed and the numbers are unique from 1 to 50 for lotto max and from 1 to 49 for lotto 649.

The second section is about a windows form which allows the user to convert currency or to convert temperature from C to F and vice versa by selecting a radio button.

The third section is a windows form with calculator and Ip validator. The calculator allows the user to perform simple calculations and the Ip validator validates if Ip address is valid.

1. **Present the print screens of yours forms, and have a detailed description of the functionalities (step by step).**





The lotto Max and the lotto 649 has the same function but the lotto max just have one more number.

There are 3 buttons :

1. Generate 6+1 numbers out of 50
2. Read and display text file
3. Exit

And 1 textbox

Let’s look at the buttons :

1. **Generate 6+1 numbers out of 50**



It generates random number all different form each other and also write the number generated into a text file

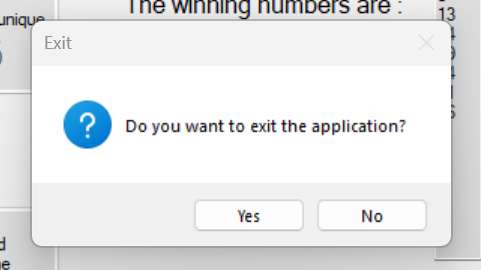
**2.Read and display text file:**



It will display the numbers on the text file on a new window.

3- Exit

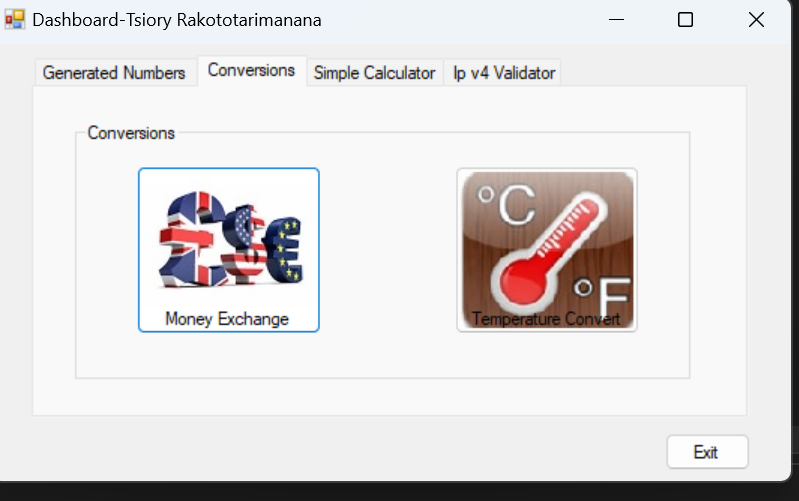
It helps the user to exit the application and giving them the choice to exit or to stay



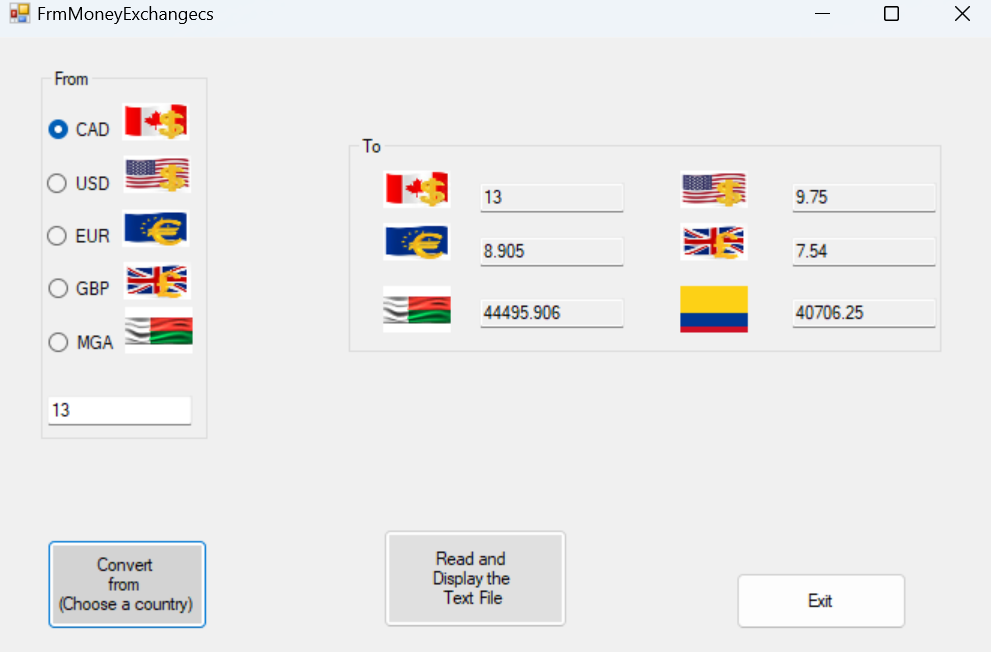
Conversion:

The dashboard displays 2 choices to the user :

The left one is money exchange and the right one is temperature conversion

****

**Money Conversion**

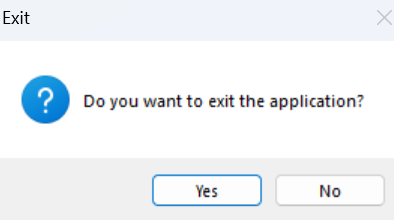
****

There are radio button which select the initial currency we want to convert from .

After selecting the radio button : by selecting Convert from button , the currency will be converted to the currency of the countries on the right.

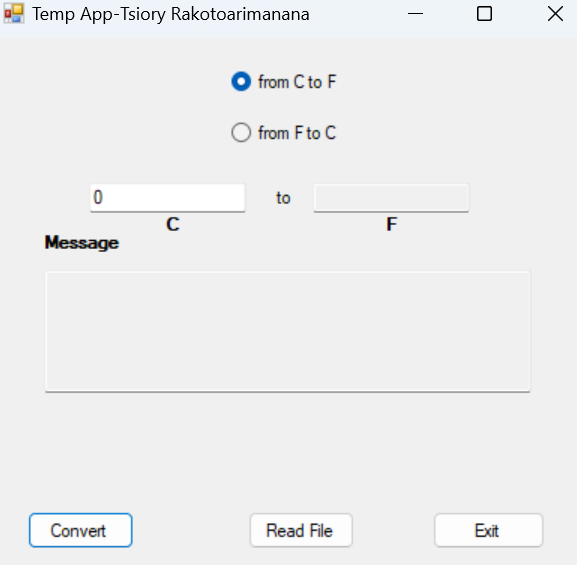


The button Read and Display the Text File will allow the user to print on the screen the data converted previously , following the same format displayed on the picture above.

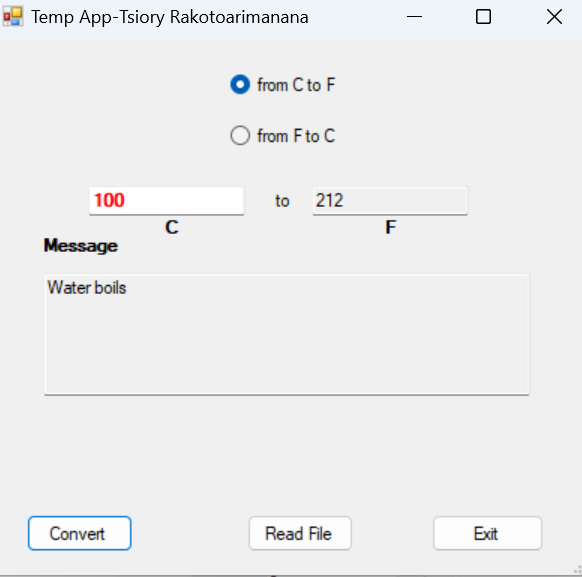


Finally the button **Exit** allows the user to quit the money conversion dashboard

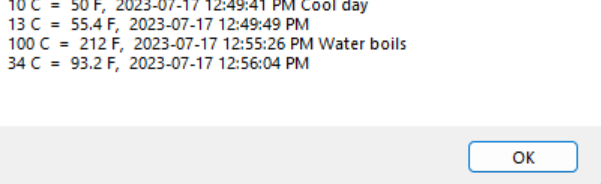
Temperature Convert:



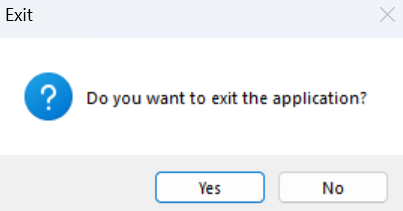
There are radio button which allow the user to select the unit of temperature he/she wants to convert from , either from F to C or from C to F.



By selecting the button Convert , the temperature will be converted to the desired temperature unit. Depending on the temperature, different color will be shown from red, yellow, green blue black referring to the intensity of the temperature , also in specific instances some message will be displayed such as “Water boils".



The button Read File will allow the user to print on the screen the data converted previously , following the same format displayed on the picture above.

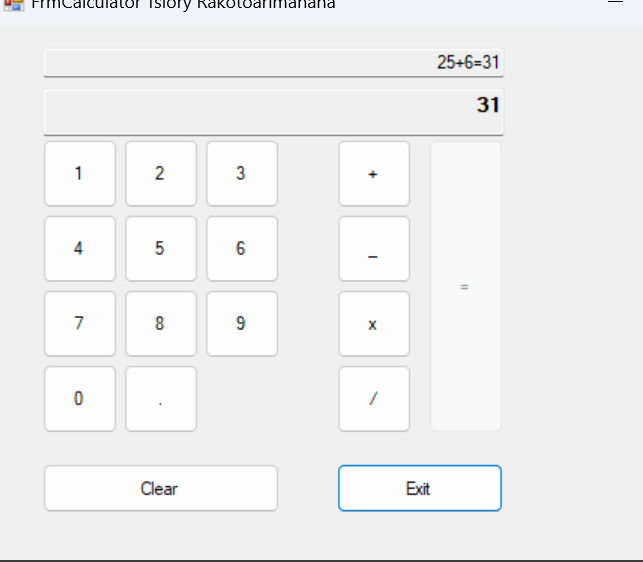


Finally the button **Exit** allows the user to quit the money conversion dashboard

Calculator:

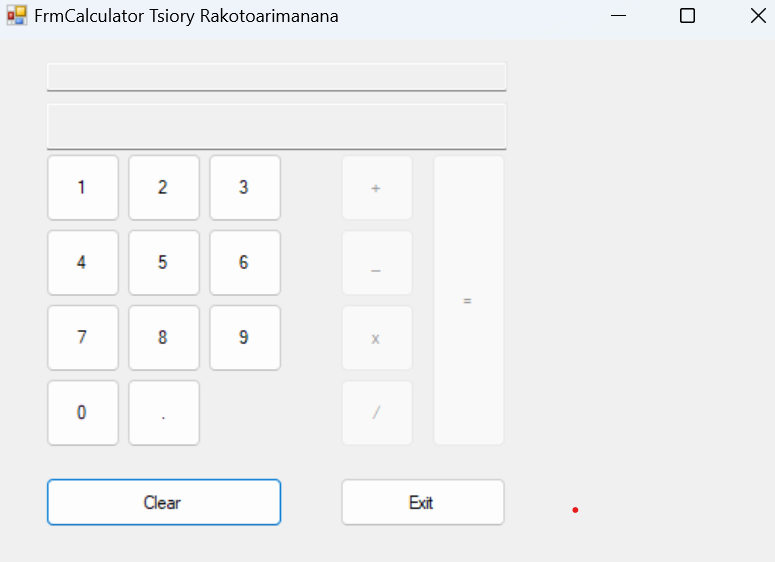


The dashboard show a button with a calculator image , when the user press it opens a windows form.

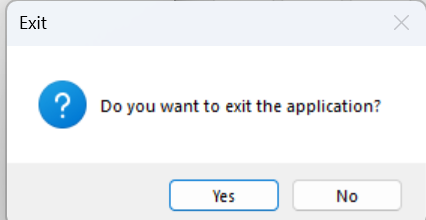


The calculator is performing simple calculation such as addition , substraction,multiplication and division . there is 2 textbox the first one is to display the operation, the textbox on the bottom displays the result of the operation .

By choosing the operand , operations are done and by pressing the button = , it performs the calculation



The button clear, clears up the textboxes so that the user can perform new operation

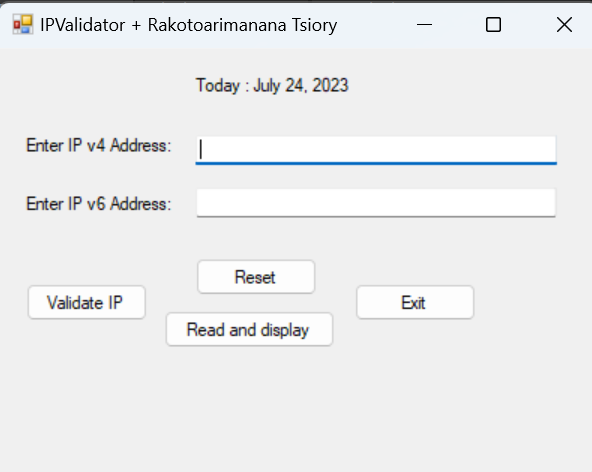


The button exit allows the user to exit the application.

IP Validator

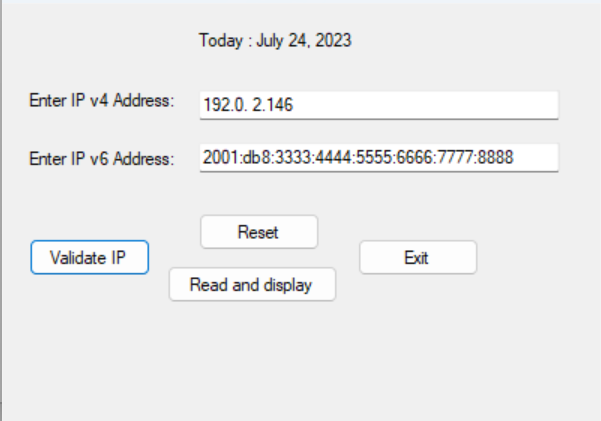


By pressing the button ip with the black logo, the user is taken to a windows form presented as below



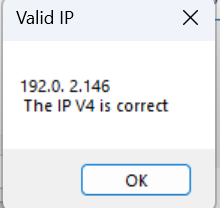
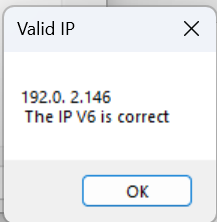
It is composed of 3 textboxes and 4 buttons:

The top textbox is for validating the format of Ipv4 whereas the bottom textbox is for validating the format of the ipv6.

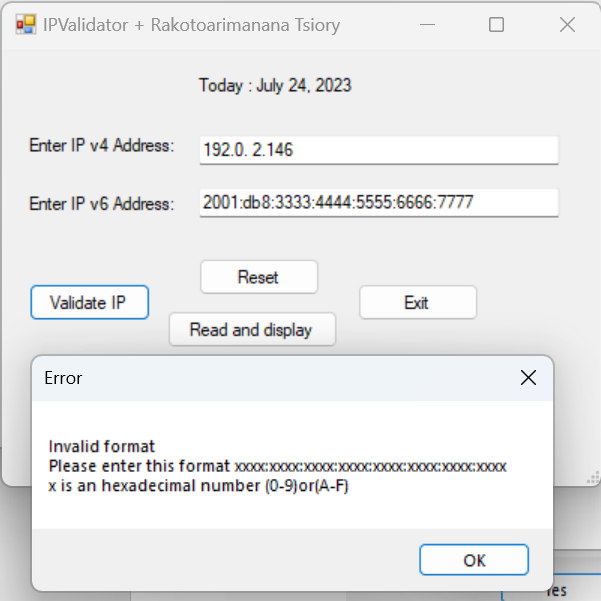


When pressing the button validate Ip : the format are being checked and the result are being written in a textfile.

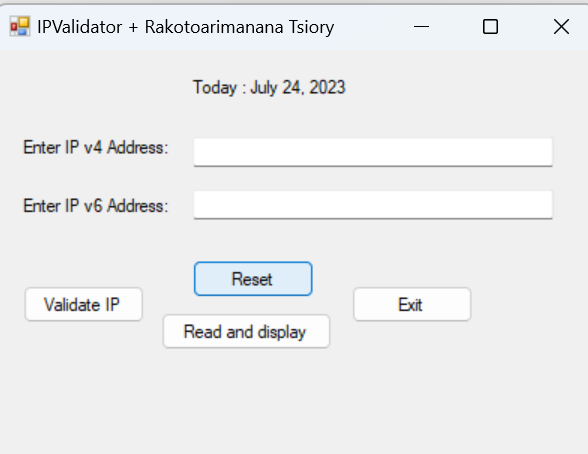
If the format are write it is going to display some message box as below



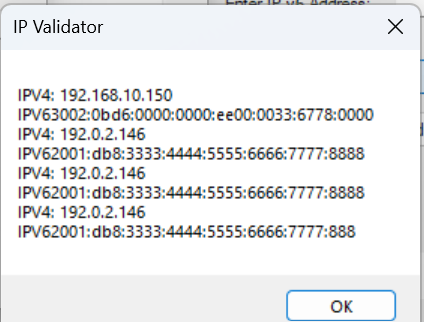
If the format are not acceptable an error message is being displayed to the user :



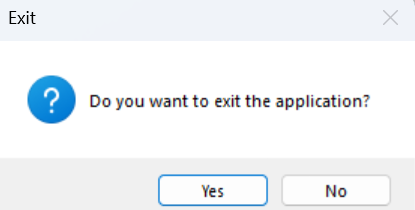
The button reset , resets the textboxes to empty



The button read and display displays the textfile with the valid Ip copied inside



And last one the button exit allows the user to exit the application



1. **Present the code of your application (forms).**

**Form1**

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;

//Tsiory Rakotoarimanana

namespace Final\_Project

{

public partial class frm23Dashboard : Form

{

public frm23Dashboard()

{

InitializeComponent();

}

private void button3\_Click(object sender, EventArgs e)

{

Application.Exit();

}

private void button1\_Click(object sender, EventArgs e)

{

frmMax obj = new frmMax();

//obj.Show();

obj.ShowDialog();

}

private void btn649\_Click(object sender, EventArgs e)

{

frm649 obj = new frm649();

//obj.Show();

obj.ShowDialog();

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

}

private void button4\_Click(object sender, EventArgs e)

{

this.Close();

}

private void button1\_Click\_1(object sender, EventArgs e)

{

FrmMoneyExchanges obj= new FrmMoneyExchanges();

obj.ShowDialog();

}

}

}

**Class Lotto**

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

//Tsiory Rakotoarimanana

namespace Final\_Project

{

internal class Lotto

{

private int Randomnumbers;

public int RandomNumbers { get { return Randomnumbers; } }

public Lotto()

{

}

public Lotto(int randomNumbers)

{

this.Randomnumbers = randomNumbers;

}

string dir = @".\Final Project\";

string path = @".\Final Project\LottoNbrs.txt";

FileStream fs = null;

public string WinningNumbers(int numberWithoutExtra,int minLimitOfRandom,int maxLimitOfRandom)

{

Random random = new Random();

int[] randomNumbers = new int[numberWithoutExtra];

string result = "";

int lottoNumber;

for (int numberIndex = 0; numberIndex < numberWithoutExtra; numberIndex++)

{

do

{

lottoNumber = random.Next(minLimitOfRandom, maxLimitOfRandom);

}

while (randomNumbers.Contains(lottoNumber));

randomNumbers[numberIndex] = lottoNumber;

result = result + randomNumbers[numberIndex].ToString()+"\t";

}

int extra;

do

{

extra = random.Next(minLimitOfRandom, maxLimitOfRandom);

} while (randomNumbers.Contains(extra));

string finalResult = result + extra.ToString();

return finalResult;

}

public void Display (string toDisplay, TextBox winningNumbers)

{

winningNumbers.Text = toDisplay;

}

public void WriteTextFile()

{

}

}

}

**LotoMAX(frmMax)**

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;

using System.IO;

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

//Tsiory Rakotoarimanana

namespace Final\_Project

{

public partial class frmMax : Form

{

public frmMax()

{

InitializeComponent();

}

private void button3\_Click(object sender, EventArgs e)

{

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

{

this.Close();

}

}

string dir = @".\Final Project\";

string path = @".\Final Project\LottoNbrsMax.txt";

FileStream fs = null;

private void button1\_Click(object sender, EventArgs e)

{

Lotto obj1 = new Lotto();

string result = obj1.WinningNumbers(7, 1, 50);

obj1.Display(result, txtDisplay);

result.Split();

string[] results = result.Split('\t');

try

{

fs = new FileStream(path, FileMode.Append, FileAccess.Write);

StreamWriter textOut = new StreamWriter(fs);

textOut.Write("Max, " + DateTime.Now.ToString("yyyy/MM/dd hh:mm:ss tt") + ", ");

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

{

if (i < 6)

{

textOut.Write(results[i] + ",");

}

if (i == 6)

{

textOut.Write(results[i] + " ");

}

if (i == 7)

{

textOut.Write("\tExtra: " + results[i] + "\n");

}

}

textOut.Close();

}

catch (FileNotFoundException)

{

MessageBox.Show(path + " not found.", "File Not Found");

}

catch (DirectoryNotFoundException)

{

MessageBox.Show(path + " not found.", "Directory Not Found");

}

catch (IOException ex)

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

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

}

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

{

if (!Directory.Exists(dir))

Directory.CreateDirectory(dir);

}

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

{

try

{

fs = new FileStream(path, FileMode.OpenOrCreate, FileAccess.Read);

StreamReader textIn = new StreamReader(fs);

string textToPrint = "";

while (textIn.Peek() != -1)

{

string row = textIn.ReadLine().Trim();

textToPrint += row + "\n";

}

MessageBox.Show(textToPrint, "Lotto:");

textIn.Close();

}

catch (FileNotFoundException)

{

MessageBox.Show(path + " not found.", "File Not Found");

}

catch (DirectoryNotFoundException)

{

MessageBox.Show(path + " not found.", "Directory Not Found");

}

catch (IOException ex)

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

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

}

}

}

**Loto649(frm649)**

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;

//Tsiory Rakotoarimanana

namespace Final\_Project

{

public partial class frm649 : Form

{

public frm649()

{

InitializeComponent();

}

string dir = @".\Final Project\";

string path = @".\Final Project\LottoNbrs.txt";

FileStream fs = null;

private void btnGenerate649\_Click(object sender, EventArgs e)

{

Lotto obj1 = new Lotto();

string result = obj1.WinningNumbers(6, 1, 49);

obj1.Display(result, txtDisplay);

result.Split();

string[] results = result.Split('\t');

try

{

fs = new FileStream(path, FileMode.Append, FileAccess.Write);

StreamWriter textOut = new StreamWriter(fs);

textOut.Write("649, " + DateTime.Now.ToString("yyyy/MM/dd hh:mm:ss tt") + ", ");

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

{

if (i < 5)

{

textOut.Write(results[i]+ ",");

}

if (i == 5)

{

textOut.Write(results[i] + " ");

}

if (i == 6)

{

textOut.Write("\tExtra: " + results[i] + "\n");

}

}

textOut.Close();

}

catch (FileNotFoundException)

{

MessageBox.Show(path + " not found.", "File Not Found");

}

catch (DirectoryNotFoundException)

{

MessageBox.Show(path + " not found.", "Directory Not Found");

}

catch (IOException ex)

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

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

}

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

{

try

{

fs = new FileStream(path, FileMode.OpenOrCreate, FileAccess.Read);

StreamReader textIn = new StreamReader(fs);

string textToPrint = "";

while (textIn.Peek() != -1)

{

string row = textIn.ReadLine().Trim();

textToPrint += row + "\n";

}

MessageBox.Show(textToPrint, "Lotto:");

textIn.Close();

}

catch (FileNotFoundException)

{

MessageBox.Show(path + " not found.", "File Not Found");

}

catch (DirectoryNotFoundException)

{

MessageBox.Show(path + " not found.", "Directory Not Found");

}

catch (IOException ex)

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

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

}

private void frm649\_Load(object sender, EventArgs e)

{

if (!Directory.Exists(dir))

Directory.CreateDirectory(dir);

}

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

{

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

{

this.Close();

}

}

}

}

**Money Convert**

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;

using System.IO;

//Tsiory Rakotoarimanana

//10/07/2023

namespace Final\_Project

{

public partial class FrmMoneyExchanges : Form

{

string dir = @".\Final Project\";

string path = @".\Final Project\MoneyConverter.txt";

FileStream fs = null;

DateTime initial;

DateTime final;

public FrmMoneyExchanges()

{

InitializeComponent();

initial = DateTime.Now;

}

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

{

if (radioCad.Checked)

{

MoneyConvert(1,txtFrom,txtCad,txtEur,txtMga,txtUsd,txtGbp,txtCop);

}

if (radioUsd.Checked)

{

MoneyConvert(2, txtFrom, txtCad, txtEur, txtMga, txtUsd, txtGbp, txtCop);

}

if (radioEur.Checked)

{

MoneyConvert(3, txtFrom, txtCad, txtEur, txtMga, txtUsd, txtGbp, txtCop);

}

if (radioGbp.Checked)

{

MoneyConvert(4, txtFrom, txtCad, txtEur, txtMga, txtUsd, txtGbp, txtCop);

}

if (radioMga.Checked)

{

MoneyConvert(5, txtFrom, txtCad, txtEur, txtMga, txtUsd, txtGbp, txtCop);

}

}

static string currency;

int valueToConvert;

private void MoneyConvert(int num,TextBox from,TextBox t1, TextBox t2, TextBox t3, TextBox t4, TextBox t5, TextBox t6)

{

try

{

valueToConvert= Convert.ToInt32(from.Text);

}

catch(Exception ex)

{

MessageBox.Show("Please enter a numeric value");

num = 0;

from.Focus();

}

if (valueToConvert<0)

{

MessageBox.Show("Please enter a positive value");

t1.Text = "XXXXXXX";

t2.Text = "XXXXXXX";

t3.Text = "XXXXXXX";

t4.Text = "XXXXXXX";

t5.Text = "XXXXXXX";

t6.Text = "XXXXXXX";

num = 0;

from.Focus();

}

if (num == 1)

{

t1.Text= valueToConvert.ToString();

t2.Text= (valueToConvert\*0.685).ToString();

t3.Text = (valueToConvert \* 3422.762).ToString();

t4.Text = (valueToConvert \* 0.75).ToString();

t5.Text = (valueToConvert \* 0.58).ToString();

t6.Text = (valueToConvert \* 3131.25).ToString();

currency = "CAD";

}

if (num == 2)

{

t1.Text = (valueToConvert\*1.32).ToString();

t2.Text = (valueToConvert \* 0.90).ToString();

t3.Text = (valueToConvert \* 4545.70).ToString();

t4.Text = (valueToConvert).ToString();

t5.Text = (valueToConvert \* 0.77).ToString();

t6.Text = (valueToConvert \* 4154.17).ToString();

currency = "USD";

}

if (num == 3)

{

t1.Text = (valueToConvert \* 1.46).ToString();

t2.Text = (valueToConvert ).ToString();

t3.Text = (valueToConvert \* 5000).ToString();

t4.Text = (valueToConvert\*1.1).ToString();

t5.Text = (valueToConvert \* 0.85).ToString();

t6.Text = (valueToConvert \* 4572.93).ToString();

currency = "EUR";

}

if (num == 4)

{

t1.Text = (valueToConvert \* 1.70).ToString();

t2.Text = (valueToConvert\*1.16).ToString();

t3.Text = (valueToConvert \* 5847.98).ToString();

t4.Text = (valueToConvert \* 1.28).ToString();

t5.Text = (valueToConvert ).ToString();

t6.Text = (valueToConvert \* 5346.41).ToString();

currency = "GBP";

}

if (num == 5)

{

t1.Text = (valueToConvert \* 0.00029).ToString();

t2.Text = (valueToConvert \* 0.00019).ToString();

t3.Text = (valueToConvert ).ToString();

t4.Text = (valueToConvert \* 0.00022).ToString();

t5.Text = (valueToConvert\*0.00017).ToString();

t6.Text = (valueToConvert \* 0.91).ToString();

currency = "MGA";

}

if (num>=1 && num<=5)

{

try

{

fs = new FileStream(path, FileMode.Append, FileAccess.Write);

StreamWriter textOut = new StreamWriter(fs);

textOut.Write(DateTime.Now.ToString("yyyy/MM/dd hh:mm:ss tt\n"));

textOut.Write(valueToConvert + " " + currency + "= " + t1.Text + " CAD; " + t2.Text + " EUR; " + t3.Text + " MGA; " + t4.Text + " USD; " + t5.Text + " GBP; " + t6.Text + " COP; \n");

textOut.Close();

}

catch (FileNotFoundException)

{

MessageBox.Show(path + " not found.", "File Not Found");

}

catch (DirectoryNotFoundException)

{

MessageBox.Show(path + " not found.", "Directory Not Found");

}

catch (IOException ex)

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

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

}

}

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

{

if (!Directory.Exists(dir))

Directory.CreateDirectory(dir);

}

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

{

try

{

fs = new FileStream(path, FileMode.OpenOrCreate, FileAccess.Read);

StreamReader textIn = new StreamReader(fs);

string textToPrint = "";

while (textIn.Peek() != -1)

{

string row = textIn.ReadLine().Trim();

textToPrint += row + "\n\n";

}

MessageBox.Show(textToPrint, "Money Exchange");

textIn.Close();

}

catch (FileNotFoundException)

{

MessageBox.Show(path + " not found.", "File Not Found");

}

catch (DirectoryNotFoundException)

{

MessageBox.Show(path + " not found.", "Directory Not Found");

}

catch (IOException ex)

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

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

}

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

{

final=DateTime.Now;

TimeSpan timeSpent = final.Subtract(initial);

if (MessageBox.Show("Do you want to quit the app? You have been here "+timeSpent.Minutes+"min "+timeSpent.Seconds+"s", "Exit", MessageBoxButtons.YesNo, MessageBoxIcon.Question).ToString() == "Yes")

{

this.Close();

}

}

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

{

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

{

this.Close();

}

}

}

}

**Temperature Convert**

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 Final\_Project

{

public partial class frmTemp : Form

{

double valueEntered;

double result;

string message;

char symbolTO;

char symbolFrom;

bool validate = false;

string dir = @".\Final Project\";

string path = @".\Final Project\TemperatureConversions.txt";

FileStream fs = null;

bool color = false;

public frmTemp()

{

InitializeComponent();

}

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

{

try

{

valueEntered = Convert.ToDouble(txtFrom.Text);

}

catch (Exception)

{

MessageBox.Show("Please enter a valid number ");

validate= true;

txtTo.Focus();

}

message = " ";

txtFrom.ForeColor= Color.Black;

if (radioCtoF.Checked)

{

result = (valueEntered \* 9 / 5) + 32;

txtTo.Text = result.ToString();

//if (result > 212 && result <= 212 && result > 104)

//if ( result == 212)

//{

// message = "Water boils";

// txtFrom.ForeColor= Color.Red;

// txtFrom.Font = new Font(txtFrom.Font, FontStyle.Bold) ;

// color= true;

//}

//if (result==104 )

//{

// message = "Hot Bath";

// txtFrom.ForeColor = Color.Red;

// txtFrom.Font = new Font(txtFrom.Font, FontStyle.Bold);

// color = true;

//}

//if (result == 98.6)

//{

// message = "Body Temperature";

// txtFrom.ForeColor = Color.Yellow;

// txtFrom.Font = new Font(txtFrom.Font, FontStyle.Bold);

// color = true;

//}

//if (result == 86)

//{

// message = "Beach Weather";

// txtFrom.ForeColor = Color.Yellow;

// txtFrom.Font = new Font(txtFrom.Font, FontStyle.Bold);

// color = true;

//}

//if (result == 70)

//{

// message = "Room Temperature";

// txtFrom.ForeColor = Color.Green;

// color= true;

// txtDisplay.Text = message;

//}

//if (result == 50)

//{

// message = "Cool day";

// txtFrom.ForeColor = Color.DarkBlue;

// color= true;

// txtFrom.Font = new Font(txtFrom.Font, FontStyle.Bold);

// txtDisplay.Text = message;

//}

//if (result == 32)

//{

// message = "Freezing point of water";

// txtFrom.ForeColor = Color.DarkBlue;

// txtFrom.Font = new Font(txtFrom.Font, FontStyle.Bold);

// color = true;

// txtDisplay.Text = message;

//}

//if (result == 0 )

//{

// message = "Very Cold Day";

// color= true;

// txtDisplay.Text = message;

//}

//if (result == -40)

//{

// message = "Extremely Cold Day\n(and the same number!)";

// color= true;

// txtDisplay.Text = message;

// txtFrom.Font = new Font(txtFrom.Font, FontStyle.Bold);

//}

if (result >= 104)

{

txtFrom.ForeColor = Color.Red;

txtFrom.Font = new Font(txtFrom.Font, FontStyle.Bold);

if (result == 212)

{

message = "Water boils";

}

if (result == 104)

{

message = "Hot Bath";

txtFrom.ForeColor = Color.Red;

txtFrom.Font = new Font(txtFrom.Font, FontStyle.Bold);

}

}

if (result >= 86 && result <= 98.6)

{

txtFrom.ForeColor = Color.Yellow;

txtFrom.Font = new Font(txtFrom.Font, FontStyle.Bold);

if (result == 98.6)

{

message = "Body Temperature";

}

if (result == 86)

{

message = "Beach Weather";

txtFrom.ForeColor = Color.Yellow;

txtFrom.Font = new Font(txtFrom.Font, FontStyle.Bold);

}

}

if (result >= 70 && result < 86)

{

txtFrom.ForeColor = Color.Green;

if (result == 70)

{

message = "Room Temperature";

}

}

if (result < 70 && result >= 32)

{

txtFrom.ForeColor = Color.DarkBlue;

txtFrom.Font = new Font(txtFrom.Font, FontStyle.Bold);

if (result == 50)

{

message = "Cool day";

}

if (result == 32)

{

message = "Freezing point of water";

txtFrom.ForeColor = Color.DarkBlue;

txtFrom.Font = new Font(txtFrom.Font, FontStyle.Bold);

}

}

if (result >= 0 && result < 32)

{

if (result == 0)

{

message = "Very Cold day";

}

}

if (result < 0)

{

txtFrom.Font = new Font(txtFrom.Font, FontStyle.Bold);

if (result == -40)

{

message = "Extremely Cold Day\n(and the same number!)";

}

}

txtDisplay.Text = message;

symbolFrom = 'C';

symbolTO = 'F';

}

if (radioFtoC.Checked)

{

//lblC. = "F";

//lblF.Text = "C";

result = (valueEntered - 32) \* 5 / 9;

txtTo.Text = result.ToString();

if (valueEntered>=104)

{

txtFrom.ForeColor = Color.Red;

txtFrom.Font = new Font(txtFrom.Font, FontStyle.Bold);

if (valueEntered == 212)

{

message = "Water boils";

}

if (valueEntered == 104 )

{

message = "Hot Bath";

txtFrom.ForeColor = Color.Red;

txtFrom.Font = new Font(txtFrom.Font, FontStyle.Bold);

}

}

if (valueEntered>=86 && valueEntered<=98.6)

{

txtFrom.ForeColor = Color.Yellow;

txtFrom.Font = new Font(txtFrom.Font, FontStyle.Bold);

if (valueEntered == 98.6 )

{

message = "Body Temperature";

}

if (valueEntered == 86 )

{

message = "Beach Weather";

txtFrom.ForeColor = Color.Yellow;

txtFrom.Font = new Font(txtFrom.Font, FontStyle.Bold);

}

}

if (valueEntered>=70 && valueEntered<86)

{

txtFrom.ForeColor = Color.Green;

if (valueEntered == 70)

{

message = "Room Temperature";

}

}

if (valueEntered<70 && valueEntered>=32)

{

txtFrom.ForeColor = Color.DarkBlue;

txtFrom.Font = new Font(txtFrom.Font, FontStyle.Bold);

if (valueEntered == 50)

{

message = "Cool day";

}

if (valueEntered == 32)

{

message = "Freezing point of water";

txtFrom.ForeColor = Color.DarkBlue;

txtFrom.Font = new Font(txtFrom.Font, FontStyle.Bold);

}

}

if (valueEntered>=0 && valueEntered<32)

{

if (valueEntered == 0 )

{

message = "Very Cold day";

}

}

if (valueEntered<0)

{

txtFrom.Font = new Font(txtFrom.Font, FontStyle.Bold);

if (valueEntered == -40)

{

message = "Extremely Cold Day\n(and the same number!)";

}

}

txtDisplay.Text = message;

symbolFrom = 'F';

symbolTO = 'C';

}

string check;

if (validate == false)

{

try

{

fs = new FileStream(path, FileMode.Append, FileAccess.Write);

StreamWriter textOut = new StreamWriter(fs);

textOut.WriteLine(valueEntered + " " + symbolFrom + " = " + result + " " + symbolTO+ DateTime.Now.ToString(", yyyy/MM/dd hh:mm:ss tt ") + message);

textOut.Write(" ");

textOut.Close();

}

catch (FileNotFoundException)

{

MessageBox.Show(path + " not found.", "File Not Found");

}

catch (DirectoryNotFoundException)

{

MessageBox.Show(path + " not found.", "Directory Not Found");

}

catch (IOException ex)

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

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

}

}

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

{

if (!Directory.Exists(dir))

Directory.CreateDirectory(dir);

}

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

{

try

{

fs = new FileStream(path, FileMode.OpenOrCreate, FileAccess.Read);

StreamReader textIn = new StreamReader(fs);

string textToPrint = "";

while (textIn.Peek() != -1)

{

string row = textIn.ReadLine().Trim();

textToPrint += row + "\n";

}

MessageBox.Show(textToPrint, "Temperature conversion");

textIn.Close();

}

catch (FileNotFoundException)

{

MessageBox.Show(path + " not found.", "File Not Found");

}

catch (DirectoryNotFoundException)

{

MessageBox.Show(path + " not found.", "Directory Not Found");

}

catch (IOException ex)

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

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

}

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

{

if (MessageBox.Show("Do you want to quit the app? You have been here ", "Exit", MessageBoxButtons.YesNo, MessageBoxIcon.Question).ToString() == "Yes")

{

this.Close();

}

}

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

{

lblC.Text = "F";

lblF.Text= "C";

}

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

{

lblC.Text = "C";

lblF.Text = "F";

}

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

{

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

{

this.Close();

}

}

}

//private void frmTemp\_Load(object sender, EventArgs e)

// {

// }

//private void button1\_Click(object sender, EventArgs e)

//{

//}

}

**frMCalculator**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Reflection.Emit;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

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

// Tsiory Rakotoarimanana

namespace Final\_Project

{

public partial class FrmCalculator : Form

{

public FrmCalculator()

{

InitializeComponent();

}

static string messageOperation = "";

static bool dot;

static bool oper;

static char sign;

static decimal result;

bool retour;

bool resultExist;

bool start;

bool deuxiemeOperation;//second operation after you click result

bool allow; //boolean from the operator button

bool exist;

bool clearStart;

string dir = @".\Final Project\";

string path = @".\Final Project\Calculator.txt";

FileStream fs = null;

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

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

{

clearStart = true;

if (clearStart == true)

{

btnAdd.Enabled = true;

btnDivide.Enabled = true;

btnSubtract.Enabled = true;

btnMultiply.Enabled = true;

btnResult.Enabled = false;

// allow = false;

start = false;

}

if (allow == true)

{

start = true;

btnAdd.Enabled = false;

btnDivide.Enabled = false;

btnSubtract.Enabled = false;

btnMultiply.Enabled = false;

}

if (start == true)

{

btnResult.Enabled = true;

}

else

{

btnResult.Enabled = false;

}

resultExist = false;

messageOperation = messageOperation + "1";

textBox1.Text = messageOperation;

}

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

{

clearStart = true;

if (clearStart == true)

{

btnAdd.Enabled = true;

btnDivide.Enabled = true;

btnSubtract.Enabled = true;

btnMultiply.Enabled = true;

// allow = false;

start = false;

}

if (allow == true)

{

start = true;

btnAdd.Enabled = false;

btnDivide.Enabled = false;

btnSubtract.Enabled = false;

btnMultiply.Enabled = false;

}

if (start == true)

{

btnResult.Enabled = true;

}

else

{

btnResult.Enabled = false;

}

if (retour == true)

resultExist = false;

messageOperation = messageOperation + "2";

textBox1.Text = messageOperation;

}

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

{

clearStart = true;

if (clearStart == true)

{

btnAdd.Enabled = true;

btnDivide.Enabled = true;

btnSubtract.Enabled = true;

btnMultiply.Enabled = true;

start = false;

}

if (allow == true)

{

start = true;

btnAdd.Enabled = false;

btnDivide.Enabled = false;

btnSubtract.Enabled = false;

btnMultiply.Enabled = false;

}

if (start == true)

{

btnResult.Enabled = true;

}

else

{

btnResult.Enabled = false;

}

resultExist = false;

messageOperation = messageOperation + "3";

textBox1.Text = messageOperation;

}

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

{

clearStart = true;

if (clearStart == true)

{

btnAdd.Enabled = true;

btnDivide.Enabled = true;

btnSubtract.Enabled = true;

btnMultiply.Enabled = true;

start = false;

}

if (allow == true)

{

start = true;

btnAdd.Enabled = false;

btnDivide.Enabled = false;

btnSubtract.Enabled = false;

btnMultiply.Enabled = false;

}

if (start == true)

{

btnResult.Enabled = true;

}

else

{

btnResult.Enabled = false;

}

resultExist = false;

messageOperation = messageOperation + "4";

textBox1.Text = messageOperation;

}

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

{

if (clearStart == true)

{

btnAdd.Enabled = true;

btnDivide.Enabled = true;

btnSubtract.Enabled = true;

btnMultiply.Enabled = true;

start = false;

}

if (allow == true)

{

start = true;

btnAdd.Enabled = false;

btnDivide.Enabled = false;

btnSubtract.Enabled = false;

btnMultiply.Enabled = false;

}

if (start == true)

{

btnResult.Enabled = true;

}

else

{

btnResult.Enabled = false;

}

resultExist = false;

messageOperation = messageOperation + "5";

textBox1.Text = messageOperation;

}

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

{

clearStart = true;

if (clearStart == true)

{

btnAdd.Enabled = true;

btnDivide.Enabled = true;

btnSubtract.Enabled = true;

btnMultiply.Enabled = true;

// allow = false;

start = false;

}

if (allow == true)

{

start = true;

btnAdd.Enabled = false;

btnDivide.Enabled = false;

btnSubtract.Enabled = false;

btnMultiply.Enabled = false;

}

if (start == true)

{

btnResult.Enabled = true;

}

else

{

btnResult.Enabled = false;

}

resultExist = false;

messageOperation = messageOperation + "6";

textBox1.Text = messageOperation;

}

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

{

clearStart = true;

if (clearStart == true)

{

btnAdd.Enabled = true;

btnDivide.Enabled = true;

btnSubtract.Enabled = true;

btnMultiply.Enabled = true;

// allow = false;

start = false;

}

if (allow == true)

{

start = true;

btnAdd.Enabled = false;

btnDivide.Enabled = false;

btnSubtract.Enabled = false;

btnMultiply.Enabled = false;

}

if (start == true)

{

btnResult.Enabled = true;

}

else

{

btnResult.Enabled = false;

}

resultExist = false;

messageOperation = messageOperation + "7";

textBox1.Text = messageOperation;

}

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

{

clearStart = true;

if (clearStart == true)

{

btnAdd.Enabled = true;

btnDivide.Enabled = true;

btnSubtract.Enabled = true;

btnMultiply.Enabled = true;

// allow = false;

start = false;

}

if (allow == true)

{

start = true;

btnAdd.Enabled = false;

btnDivide.Enabled = false;

btnSubtract.Enabled = false;

btnMultiply.Enabled = false;

}

if (start == true)

{

btnResult.Enabled = true;

}

else

{

btnResult.Enabled = false;

}

resultExist = false;

messageOperation = messageOperation + "8";

textBox1.Text = messageOperation;

}

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

{

clearStart = true;

if (clearStart == true)

{

btnAdd.Enabled = true;

btnDivide.Enabled = true;

btnSubtract.Enabled = true;

btnMultiply.Enabled = true;

// allow = false;

start = false;

}

if (allow == true)

{

start = true;

btnAdd.Enabled = false;

btnDivide.Enabled = false;

btnSubtract.Enabled = false;

btnMultiply.Enabled = false;

}

if (start == true)

{

btnResult.Enabled = true;

}

else

{

btnResult.Enabled = false;

}

resultExist = false;

messageOperation = messageOperation + "9";

textBox1.Text = messageOperation;

}

private void button10\_Click(object sender, EventArgs e)

{

clearStart = true;

if (clearStart == true)

{

btnAdd.Enabled = true;

btnDivide.Enabled = true;

btnSubtract.Enabled = true;

btnMultiply.Enabled = true;

// allow = false;

start = false;

}

if (allow == true)

{

start = true;

btnAdd.Enabled = false;

btnDivide.Enabled = false;

btnSubtract.Enabled = false;

btnMultiply.Enabled = false;

}

if (start == true)

{

btnResult.Enabled = true;

}

else

{

btnResult.Enabled = false;

}

resultExist = false;

messageOperation = messageOperation + "0";

textBox1.Text = messageOperation;

}

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

{

resultExist = false;

messageOperation = messageOperation + ".";

dot = true;

if (dot == true)

{

btndecimal.Enabled = false;

}

else

{

btndecimal.Enabled = true;

}

textBox1.Text = messageOperation;

}

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

{

Calculator obj = new Calculator();

if (resultExist == true)

{

messageOperation = Convert.ToString(result);

}

dot = false;

messageOperation = messageOperation + "+";

oper = true;

if (oper == true)

{

btnAdd.Enabled = false;

btnDivide.Enabled = false;

btnSubtract.Enabled = false;

btnMultiply.Enabled = false;

btndecimal.Enabled = true;

// btnResult.Enabled = true ;

}

textBox1.Text = messageOperation;

sign = '+';

allow = true;

start = true;

// result = obj.Add(textBox1);

}

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

{

Calculator obj = new Calculator();

if (resultExist == true)

{

messageOperation = Convert.ToString(result);

}

dot = false;

messageOperation = messageOperation + "-";

oper = true;

if (oper == true)

{

btnAdd.Enabled = false;

btnDivide.Enabled = false;

btnSubtract.Enabled = false;

btnMultiply.Enabled = false;

btndecimal.Enabled = true;

// btnResult.Enabled = true;

}

textBox1.Text = messageOperation;

sign = '-';

allow = true;

}

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

{

Calculator obj = new Calculator();

if (resultExist == true)

{

messageOperation = Convert.ToString(result);

}

dot = false;

messageOperation = messageOperation + "\*";

oper = true;

if (oper == true)

{

btnAdd.Enabled = false;

btnDivide.Enabled = false;

btnSubtract.Enabled = false;

btnMultiply.Enabled = false;

btndecimal.Enabled = true;

// btnResult.Enabled = true;

}

textBox1.Text = messageOperation;

sign = '\*';

allow = true;

//textBox1.Text = messageOperation + (Convert.ToString(result));

}

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

{

Calculator obj = new Calculator();

if (resultExist == true)

{

messageOperation = Convert.ToString(result);

}

dot = false;

messageOperation = messageOperation + "/";

oper = true;

if (oper == true)

{

btnAdd.Enabled = false;

btnDivide.Enabled = false;

btnSubtract.Enabled = false;

btnMultiply.Enabled = false;

button10.Enabled = false;

btndecimal.Enabled = true;

// btnResult.Enabled = true;

}

textBox1.Text = messageOperation;

sign = '/';

allow = true;

}

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

{

Calculator obj = new Calculator();

//oper = true;

allow = false;

if (oper == true)

{

btnAdd.Enabled = true;

btnDivide.Enabled = true;

btnSubtract.Enabled = true;

btnMultiply.Enabled = true;

}

btnResult.Enabled = false;

button10.Enabled = true;

if (sign == '+')

{

result = obj.Add(textBox1);

}

if (sign == '-')

{

result = obj.Subtract(textBox1);

}

if (sign == '\*')

{

result = obj.Multiply(textBox1);

}

if (sign == '/')

{

result = obj.Divide(textBox1);

}

messageOperation = messageOperation + "=" + (Convert.ToString(result));

textBox1.Text = messageOperation;

textBox2.Text = Convert.ToString(result);

resultExist = true;

messageOperation = "";

try

{

fs = new FileStream(path, FileMode.Append, FileAccess.Write);

StreamWriter textOut = new StreamWriter(fs);

textOut.Write(textBox1.Text + "\n");

textOut.Close();

}

catch (FileNotFoundException)

{

MessageBox.Show(path + " not found.", "File Not Found");

}

catch (DirectoryNotFoundException)

{

MessageBox.Show(path + " not found.", "Directory Not Found");

}

catch (IOException ex)

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

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

retour = true;

}

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

{

btnResult.Enabled = false;

Calculator obj = new Calculator();

if (start == true)

{

btnResult.Enabled = true;

}

btnAdd.Enabled = false;

btnDivide.Enabled = false;

btnSubtract.Enabled = false;

btnMultiply.Enabled = false;

btnResult.Enabled = false;

allow = false;

if (!Directory.Exists(dir))

Directory.CreateDirectory(dir);

}

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

{

messageOperation = "";

textBox1.Text = messageOperation;

textBox2.Text = messageOperation;

btnAdd.Enabled = false;

btnDivide.Enabled = false;

btnSubtract.Enabled = false;

btnMultiply.Enabled = false;

btnResult.Enabled = false;

allow = false;

clearStart = true;// a boolean to help enabling the operator button only if we click number

}

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

{

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

{

this.Close();

}

}

}

}

**Calculator.Cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data.SqlTypes;

using System.Drawing;

using System.Linq;

using System.Reflection.Emit;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Final\_Project

{

internal class Calculator

{

private decimal currentValue;

private decimal operand1 = 0;

private decimal operand2 = 0;

private string op;

public decimal CurrentValue { get { return currentValue; } }

public decimal Operand1 { get { return operand1; } }

public decimal Operand2 { get { return operand2; } }

public string OP { get { return op; } }

public Calculator() { }

public Calculator(decimal currentValue, decimal operand1, decimal operand2, string op)

{

currentValue = this.currentValue;

operand1 = this.operand1;

operand2 = this.operand2;

op = this.op;

}

public decimal Add(TextBox txt)

{

string display = txt.Text;

string[] numbers = display.Split('+');

this.operand1 = Convert.ToDecimal(numbers[0]);

this.operand2 = Convert.ToDecimal(numbers[1]);

this.currentValue = this.operand1 + this.operand2;

this.operand1 = this.currentValue;

return currentValue;

}

public decimal Divide(TextBox txt)

{

string display = txt.Text;

string[] numbers = display.Split('/');

this.operand1 = Convert.ToDecimal(numbers[0]);

this.operand2 = Convert.ToDecimal(numbers[1]);

this.currentValue = this.operand1 / this.operand2;

this.operand1 = this.currentValue;

return currentValue;

}

public decimal Subtract(TextBox txt)

{

string display = txt.Text;

string[] numbers = display.Split('-');

this.operand1 = Convert.ToDecimal(numbers[0]);

this.operand2 = Convert.ToDecimal(numbers[1]);

this.currentValue = this.operand1 - this.operand2;

this.operand1 = this.currentValue;

return currentValue;

}

public decimal Multiply(TextBox txt)

{

string display = txt.Text;

string[] numbers = display.Split('\*');

this.operand1 = Convert.ToDecimal(numbers[0]);

this.operand2 = Convert.ToDecimal(numbers[1]);

this.currentValue = this.operand1 \* this.operand2;

this.operand1 = this.currentValue;

return currentValue;

}

public decimal Equals(TextBox txt)

{

char operation = Convert.ToChar(this.op);

string display = txt.Text;

string[] numbers = display.Split(operation);

this.operand2 = Convert.ToDecimal(numbers[1]);

if (operation == '+')

{

currentValue = currentValue + operand2;

}

if (operation == '-')

{

currentValue = currentValue - operand2;

}

if (operation == '\*')

{

currentValue = currentValue \* operand2;

}

if (operation == '/')

{

currentValue = currentValue / operand2;

}

numbers[0] = currentValue.ToString();

return currentValue;

}

public void Clear()

{

operand1 = 0;

operand2 = 0;

currentValue = 0;

op = " ";

}

}

}

**IP Validator**

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;

using System.Text.RegularExpressions;

using System.IO;

//Tsiory Rakotoarimanana

namespace Final\_Project

{

public partial class IPValidator : Form

{

string dir = @".\Final Project\";

string path = @".\Final Project\IP.txt";

FileStream fs = null;

Regex ipV4;

Regex ipV6;

public IPValidator()

{

InitializeComponent();

}

private void label1\_Click(object sender, EventArgs e)

{

}

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

{

DateTime now= DateTime.Today;

label1.Text = "Today : "+ now.ToLongDateString();

if (!Directory.Exists(dir))

Directory.CreateDirectory(dir);

}

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

{

ipV4 = 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}$");

string v4 = txtV4.Text.Trim();

v4 = v4.Replace(" ","");

if (ipV4.IsMatch(v4))

{

MessageBox.Show(txtV4.Text + "\n The IP V4 is correct", "Valid IP");

ipV6 = new Regex(@"^([0-9a-fA-F]{1,4}:){7}([0-9a-fA-F]){1,4}$");

string v6= txtV6.Text.Trim();

v6 = v6.Replace(" ","");

if (ipV6.IsMatch(v6))

{

MessageBox.Show(txtV4.Text + "\n The IP V6 is correct", "Valid IP");

try

{

fs = new FileStream(path, FileMode.Append, FileAccess.Write);

StreamWriter textOut = new StreamWriter(fs);

textOut.Write("IPV4: " + v4 + "\nIPV6" + v6 +"\n");

textOut.Close();

}

catch (FileNotFoundException)

{

MessageBox.Show(path + " not found.", "File Not Found");

}

catch (DirectoryNotFoundException)

{

MessageBox.Show(path + " not found.", "Directory Not Found");

}

catch (IOException ex)

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

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

}

else

{

MessageBox.Show("Invalid format\nPlease enter this format xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx\nx is an hexadecimal number (0-9)or(A-F)","Error");

}

}

else

{

MessageBox.Show(txtV4.Text + "\nThe IP must have 4 bytes\n Integer number between 0 to 255\nseparated by a dot(255.255.255.255)", "Error");

txtV4.Focus();

}

}

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

{

try

{

fs = new FileStream(path, FileMode.OpenOrCreate, FileAccess.Read);

StreamReader textIn = new StreamReader(fs);

string textToPrint = "";

while (textIn.Peek() != -1)

{

string row = textIn.ReadLine().Trim();

textToPrint += row + "\n";

}

MessageBox.Show(textToPrint, "IP Validator");

textIn.Close();

}

catch (FileNotFoundException)

{

MessageBox.Show(path + " not found.", "File Not Found");

}

catch (DirectoryNotFoundException)

{

MessageBox.Show(path + " not found.", "Directory Not Found");

}

catch (IOException ex)

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

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

}

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

{

txtV4.Text = "";

txtV6.Text= string.Empty;

}

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

{

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

{

this.Close();

}

}

}

}

1. **Present the classes and/or methods that you create or you did use in the project.**

|  |  |
| --- | --- |
| **Class/Method Name** | **Description** |
| 1. Class Lotto | It’s a class who has the properties and the methods to use to display and generate method |
| 1. string WinningNumbers(int numberWithoutExtra,int minLimitOfRandom,int maxLimitOfRandom) | It creates random winning number , and has as argument the number without extra, the minimum limit for the random number to display and the max limit |
| 1. void Display (string toDisplay, TextBox winningNumbers)) | It displays the number generated inside the textbox, |
| 1. Class filestream | Allows to work with files |
| 1. Class StreamReader | Allows to read characters or bytes from a stream |
| 1. Class MessageBox | Helps display a message using windows form |
| 1. Class Random | Helps generate random number |
| 1. Class DateTime | Helps access the year,month,day |
| 1. .Split() | Helps to separate and put in an array of strings |
| 1. .Next(,) | Create random number with minimum and maximum as argument |
| 1. Write() | To Write in the textfile |
| 1. Show() | To display the messageBox |
| 1. Class Calculator | Class for doing simple calculation |
| 1. Add(textbox) | It is doing addition |
| 1. Subtract(textbox) | Perform subtraction |
| 1. Multiply(textbox) | Perform multiplication |
| 1. Divide(textbox) | Perform division |
| 1. Equal() | Return the value of the operation |
| 1. .Close() | To close the windows |
| 1. Clear() | Clears the textboxes |
| 1. Convert.ToDecimal | Convert a different type into decimal |
| 1. .Split() | Create an array of string by splitting a set of strings at a certain position |
| 1. Trim() | Remove the first and last space of a set of string |
| 1. Replace(,) | Replace a string by another one |
| 1. IsMatch() | Check if a set of string matches certain condition |
| 1. .Peek() | o return the object at the top of the Stack without removing it. |
| 1. .Exist() | Check if an element exist |
| 1. CreateDirectory() | Creates a directory |

1. **Present the difficulties that you have, what was the hardest and the easiest part of your project.**

The easiest part was to design,

The hardest part was to code the textfile.

the conversion was fun to do , the hard part was to respect the different conditions presented by the teacher(17/07/2023)

I also had a problem with my exit button but the teacher helped me fixed it (12-07-2023)

The hardest part was coding the calculator because I needed to satisfy a lot of condition since I used a lot of Boolean , I believe I can improve my coding but I m ready to keep improving (24-07-2023)

The last section was the most challenging for me (24-07-2023)