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

|  |
| --- |
| Your name: Bruno Rafael Ferreira Lopes  4/13/2023  Version: 4.2 |

1. **Start by adding a short description of your project, and the languages (technologies) used:**
2. Language: C#
3. Tools (IDE): Visual Studio 2022
4. **Present the print screens of yours forms and have a detailed description of the functionalities (step by step).**

Graphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generated

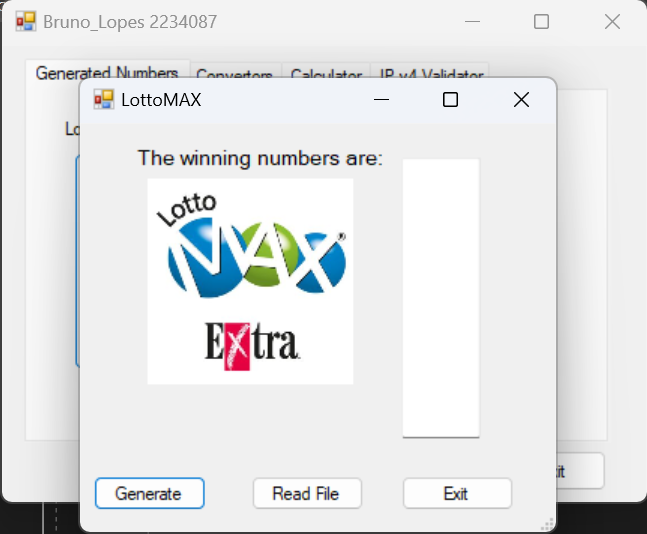
Graphical user interface, application

Description automatically generatedGraphical user interface, application, Word

Description automatically generated

1. If you click on button Lotto Max:

Graphical user interface, application, Word

Description automatically generated

1. If you click on Generate, the 8 unique random numbers will be generated in the textbox, and 7 extra random numbers will be generated in the label below the figure:

Graphical user interface, application

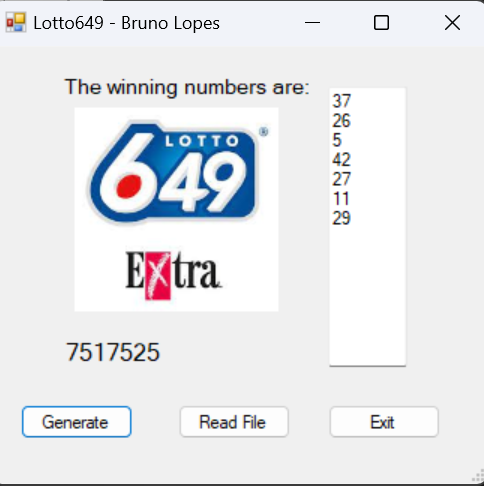
Description automatically generated

1. If you click on button Lotto 649

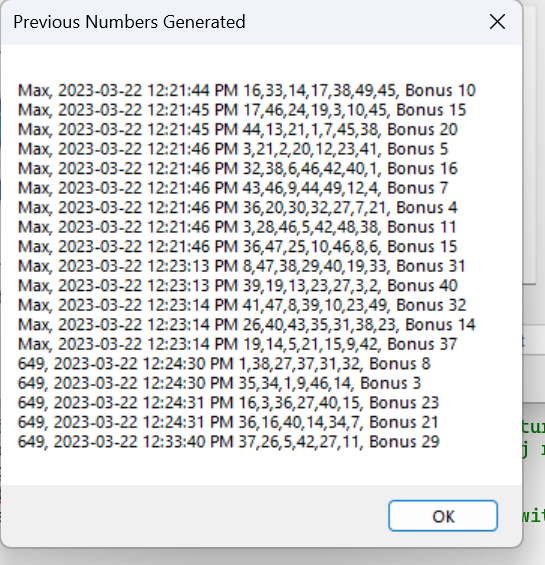
Graphical user interface, application

Description automatically generated

1. If you click on Generate, the 7 unique random numbers will be generated in the textbox, and 7 extra random numbers will be generated in the label below the figure:



1. If you click in the button Read File, this message box will show up:



1. If you click in the button Convertors, this box will show up:

Graphical user interface, application

Description automatically generated

1. If you click on the button Money Exchange, this box will show up, where you can choose the currencies to make the conversion. Also clicking the button Read File we can see the history of the conversions:

Graphical user interface, application

Description automatically generatedText

Description automatically generated

In the textBox1 the user inserts the value he wants to convert and select the currency “From” and “To” to convert, the result will be displayed in the textBox2 (read-only).

When we press the button Exit a message box show the time we spent in this application:

Graphical user interface, application

Description automatically generated

1. If you click in the button Temperature Convert, this box will show up:

Graphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generated

Depending on the option selected the labels in textbox1 and textbox2 will change.

Graphical user interface, text, application

Description automatically generatedGraphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generated

Typing the temperature in textbox1 and clicking in Convert, the result will be displayed in the label 3 with some messages in different colors. Clicking in the button Read File we can see the Message Box with the history of the conversions:

Graphical user interface, text, application

Description automatically generated

1. If you click in the button IP v4 Validator, this box will show up:

Graphical user interface, application, Word

Description automatically generatedGraphical user interface, text, application

Description automatically generated

When the form is loaded, the date is shown on the label on the top.

When we type something in the textbox and click in validate IP, If the input is not in IP format the message will show up:

Graphical user interface, text, application, chat or text message

Description automatically generated

If the IP typed is in a correct format the following message will show up:

Graphical user interface, text, application

Description automatically generated

And the value of the IP and the date will be stored in a binary (.dat) file.

1. If you click in the button Calculator, this box will show up:

A screenshot of a computer

Description automatically generated with medium confidenceA screenshot of a computer

Description automatically generated with medium confidence

If you click in the buttons, they will form a text in the Textbox that when the user click in some operator, the string in the textbox will be converted to a decimal operand1 and the operators buttons will be blocked, after type the second number and click in the button equal, the operation will be performed and the buttons will be liberated to perform another operation. The button clear, clear all the operation and allow to perform other operations.

All operations are stored in a txt file in the file of the project.

1. The file /Files will storage all .txt and .dat from the forms:

Graphical user interface, text, application

Description automatically generated

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

**Lotto Max**

public partial class LottoMAX : Form

{

public LottoMAX()

{

InitializeComponent();

}

string path = @".\Files\LottoNbrs.txt";

FileStream fs = null;

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

{

//creation of extra 7 numbers generated bellow the picture

Random random1 = new Random(); //constructor of the obj random1

int tempNumber;

string tempString = "";

int[] randomNumber1 = new int[7]; //creating an array with random numbers

for (int i = 0; i < 7; i++) {

tempNumber = random1.Next(1,9);

randomNumber1[i] = tempNumber; //variable randomNumber1 is equal to the obj random1 with a random number between 0 and 9

tempString += randomNumber1[i].ToString(); //variable randomNumber1 converted to string

}

label2.Text = tempString; //display the variable at label 2

tempString = ""; //clear the string for the next generation of numbers

//creation of the main 8 random numbers

Random random = new Random(); //constructor of the obj random

string tempLoto = "";

List<int> randomNumber = new List <int>(); //creating a list with random numbers

for (int i = 0; i < 7; i++)

{

int rand = random.Next(1,50); //generate the random number

if (!randomNumber.Contains(rand)) //if the random number is not in the list already

{

randomNumber.Add(rand); //add the random number in the list

tempLoto += randomNumber[i].ToString() + "\t"; //store the number in the variable that will be printed in the textbox

}

else

{

i = i - 1; //if the number generated already exists in the list, the counter back to try again

}

}

//bonus number

string tempBonus = "";

label1: int numBonus = random.Next(1, 50);

if (!randomNumber.Contains(numBonus)) //compare the bonus number with all other 7 numbers in the list

{

randomNumber.Add(numBonus);

tempBonus = numBonus.ToString();

}

else

{

goto label1;

}

textBox1.Text = tempLoto+tempBonus; //display the numbers at the textbox 1

string txtFile = tempLoto + $" Bonus {tempBonus}"; //line of text that will be printed in a txt file

tempLoto = ""; //clear the string for the next generation of numbers

tempBonus = "";

//save the numbers in a text file

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

StreamWriter txt1 = new StreamWriter(fs);

string print = txtFile.Replace("\t",","); //change the text in the txt file, replacing the tabs for ,

txt1.Write("Max, "+DateTime.Now+" "+print+"\n"); //printing in the txt file

txt1.Close();

txt1.Dispose();

}

private void button2\_Click(object sender, EventArgs e)

{

string title = "Previous Numbers Generated";

string textToPrint = "";

FileStream stream = null;

byte counter = 0;

try

{

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

StreamReader txt = new StreamReader(stream);

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

{

string prevNumbers = txt.ReadLine();

textToPrint += prevNumbers + "\n";

counter++;

if (counter == 12)

{

MessageBox.Show(textToPrint, title);

textToPrint = "";

counter = 0;

}

}

if (counter > 0) { MessageBox.Show(textToPrint, title); }

txt.Close();

}

catch(IOException ex)

{

MessageBox.Show("IO Exception\n" + ex.Message);

}

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

}

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

{

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

{

this.Close();

}

}

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

{

}

}

**Lotto 649**

public partial class Lotto649 : Form

{

public Lotto649()

{

InitializeComponent();

}

string path = @".\Files\LottoNbrs.txt";

FileStream fs = null;

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

{

//creation of extra 7 numbers generated bellow the picture

Random random1 = new Random(); //constructor of the obj random1

int tempNumber;

string tempString = "";

int[] randomNumber1 = new int[7]; //creating an array with random numbers

for (int i = 0; i < 7; i++)

{

tempNumber = random1.Next(1, 9);

randomNumber1[i] = tempNumber; //variable randomNumber1 is equal to the obj random1 with a random number between 0 and 9

tempString += randomNumber1[i].ToString(); //variable randomNumber1 converted to string

}

label2.Text = tempString; //display the variable at label 2

tempString = ""; //clear the string for the next generation of numbers

//creation of the main 7 random numbers

Random random = new Random(); //constructor of the obj random

string tempLoto = "";

List<int> randomNumber = new List<int>(); //creating a list with random numbers

for (int i = 0; i < 6; i++)

{

int rand = random.Next(1, 49); //generate the random number

if (!randomNumber.Contains(rand)) //if the random number is not in the list already

{

randomNumber.Add(rand); //add the random number in the list

tempLoto += randomNumber[i].ToString() + "\t"; //store the number in the variable that will be printed in the textbox

}

else

{

i = i - 1; //if the number generated already exists in the list, the counter back to try again

}

}

//bonus number

string tempBonus = "";

label1: int numBonus = random.Next(1, 49);

if (!randomNumber.Contains(numBonus)) //compare the bonus number with all other 7 numbers in the list

{

randomNumber.Add(numBonus);

tempBonus = numBonus.ToString();

}

else

{

goto label1;

}

textBox1.Text = tempLoto + tempBonus; //display the numbers at the textbox 1

string txtFile = tempLoto + $" Bonus {tempBonus}"; //line of text that will be printed in a txt file

tempLoto = ""; //clear the string for the next generation of numbers

tempBonus = "";

//save the numbers in a text file

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

StreamWriter txt2 = new StreamWriter(fs);

string print = txtFile.Replace("\t", ","); //change the text in the txt file, replacing the tabs for ,

txt2.Write("649, " + DateTime.Now + " " + print + "\n"); //printing in the txt file

txt2.Close();

txt2.Dispose();

}

private void button2\_Click(object sender, EventArgs e)

{

string title = "Previous Numbers Generated";

string textToPrint = "";

FileStream stream = null;

byte counter = 0;

try

{

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

StreamReader txt = new StreamReader(stream);

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

{

string prevNumbers = txt.ReadLine();

textToPrint += prevNumbers + "\n";

counter++;

if (counter == 12)

{

MessageBox.Show(textToPrint, title);

textToPrint = "";

counter = 0;

}

}

if (counter > 0) { MessageBox.Show(textToPrint, title); }

txt.Close();

}

catch (IOException ex)

{

MessageBox.Show("IO Exception\n" + ex.Message);

}

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

}

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

{

if (MessageBox.Show("Do you want to close the Lotto649 ? ", "Exit", MessageBoxButtons.YesNo, MessageBoxIcon.Question).ToString() == "Yes")

{

this.Close();

}

}

}

**Money Exchange**

public partial class MoneyEx : Form

{

public MoneyEx()

{

InitializeComponent();

}

string path = @".\Files\MoneyConv.txt";

FileStream fs = null;

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

{

double money\_in = 0.00;

try

{

money\_in = Convert.ToDouble(textBox1.Text); // try and catch to check if the user input a number in the text box.

}

catch(Exception ex0)

{

MessageBox.Show(ex0.Message);

}

double money1 = money\_in;

double money2 = 0.00;

string currency1 = "";

string currency2 = "";

if (radioButton1.Checked)

{

double rate = 1;

currency1 = "CAD";

if (radioButton6.Checked)

{

rate = 1;

currency2 = "CAD";

}

if (radioButton7.Checked)

{

rate = 0.73;

currency2 = "USD";

}

if (radioButton8.Checked)

{

rate = 0.68;

currency2 = "EUR";

}

if (radioButton9.Checked)

{

rate = 0.59;

currency2 = "GBP";

}

if (radioButton10.Checked)

{

rate = 3.82;

currency2 = "BRL";

}

money2 = Math.Round(money1 \* rate, 2);

textBox2.Text = $" {currency2} {money2.ToString()}";

}

if (radioButton2.Checked)

{

double rate = 1;

currency1 = "USD";

if (radioButton6.Checked)

{

rate = 1.37;

currency2 = "CAD";

}

if (radioButton7.Checked)

{

rate = 1;

currency2 = "USD";

}

if (radioButton8.Checked)

{

rate = 0.93;

currency2 = "EUR";

}

if (radioButton9.Checked)

{

rate = 0.82;

currency2 = "GBP";

}

if (radioButton10.Checked)

{

rate = 5.25;

currency2 = "BRL";

}

money2 = Math.Round(money1 \* rate, 2);

textBox2.Text = $" {currency2} {money2.ToString()}";

}

if (radioButton3.Checked)

{

double rate = 1;

currency1 = "EUR";

if (radioButton6.Checked)

{

rate = 1.48;

currency2 = "CAD";

}

if (radioButton7.Checked)

{

rate = 1.08;

currency2 = "USD";

}

if (radioButton8.Checked)

{

rate = 1;

currency2 = "EUR";

}

if (radioButton9.Checked)

{

rate = 0.88;

currency2 = "GBP";

}

if (radioButton10.Checked)

{

rate = 5.65;

currency2 = "BRL";

}

money2 = Math.Round(money1 \* rate, 2);

textBox2.Text = $" {currency2} {money2.ToString()}";

}

if (radioButton4.Checked)

{

double rate = 1;

currency1 = "GBP";

if (radioButton6.Checked)

{

rate = 1.68;

currency2 = "CAD";

}

if (radioButton7.Checked)

{

rate = 1.22;

currency2 = "USD";

}

if (radioButton8.Checked)

{

rate = 1.14;

currency2 = "EUR";

}

if (radioButton9.Checked)

{

rate = 1;

currency2 = "GBP";

}

if (radioButton10.Checked)

{

rate = 6.43;

currency2 = "BRL";

}

money2 = Math.Round(money1 \* rate, 2);

textBox2.Text = $" {currency2} {money2.ToString()}";

}

if (radioButton5.Checked)

{

double rate = 1;

currency1 = "BRL";

if (radioButton6.Checked)

{

rate = 0.26;

currency2 = "CAD";

}

if (radioButton7.Checked)

{

rate = 0.19;

currency2 = "USD";

}

if (radioButton8.Checked)

{

rate = 0.18;

currency2 = "EUR";

}

if (radioButton9.Checked)

{

rate = 0.16;

currency2 = "GBP";

}

if (radioButton10.Checked)

{

rate = 1;

currency2 = "BRL";

}

money2 = Math.Round(money1 \* rate, 2);

textBox2.Text = $" {currency2} {money2.ToString()}";

}

//save the numbers in a text file

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

StreamWriter txt = new StreamWriter(fs);

string print = $"{money1} {currency1} = {money2} {currency2}"; //change the text in the txt file, replacing the tabs for ,

txt.Write($"{print}, { DateTime.Now}\n"); //printing in the txt file

txt.Close();

txt.Dispose();

}

private void button2\_Click(object sender, EventArgs e)

{

string title = "Previous Conversions";

string textToPrint = "";

FileStream stream = null;

byte counter = 0;

try

{

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

StreamReader txt = new StreamReader(stream);

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

{

string prevConv = txt.ReadLine();

textToPrint += prevConv + "\n";

counter++;

if (counter == 12)

{

MessageBox.Show(textToPrint, title);

textToPrint = "";

counter = 0;

}

}

if (counter > 0) { MessageBox.Show(textToPrint, title); }

txt.Close();

}

catch (IOException ex)

{

MessageBox.Show("IO Exception\n" + ex.Message);

}

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

}

DateTime dateOn; //variable that store the time the form is loaded

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

{

TimeSpan timeSpent = new TimeSpan();

timeSpent = DateTime.Now.Subtract(dateOn); //interval between the time the form was loaded and the form required to close.

string ts = $"{timeSpent.Minutes:D2}min:{timeSpent.Seconds:D2}s";

if (MessageBox.Show($"You spent {ts} in this application.\nDo you want to quit the application Money Exchange?", "Exit ?", MessageBoxButtons.YesNo, MessageBoxIcon.Question).ToString() == "Yes")

{

this.Close();

}

}

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

{

DateTime time\_on = new DateTime();

dateOn = DateTime.Now;

}

}

**Temperature Conversion**

public partial class TempConv : Form

{

public TempConv()

{

InitializeComponent();

}

string path = @".\Files\TempConv.txt";

FileStream fs = null;

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

{

double temp1=0; //create a variable for the first temperature

double temp2=0; //create a variable for the temperature converter

label5.Text = "";

try

{

temp1 = Convert.ToDouble(textBox1.Text); //store the input in the first temperature

}

catch(Exception ex)

{

MessageBox.Show("Type a Temperature in correct format",ex.Message);

textBox2.Text = "";

textBox1.Text = "";

}

//creation of a dictionary for the message descriptions

Dictionary<double, string> Coments = new Dictionary<double, string>();

Coments.Add(100, "Water boils");

Coments.Add(40, "Hot Bath");

Coments.Add(37, "Body temperature");

Coments.Add(30, "Beach weather");

//Coments.Add(21, "Room temperature");

Coments.Add(10, "Cool Day");

Coments.Add(0, "Freezing point of water");

//Coments.Add(-18, "Very Cold Day");

Coments.Add(-40, "Extremely Cold Day\r\n(and the same number!)");

if (textBox1.Text!="" && radioButton1.Checked)

{

temp2 = Math.Round((temp1 \* 9/5)+32,2); //convertion from C to F

textBox2.Text = temp2.ToString(); //display the result on the textbox 2

if (temp1 >= 20 && temp1 < 24)

{

Coments.Add(temp1, "Room temperature");

}

else if (temp1 >= -39 && temp1 < -15)

{

Coments.Add(temp1, "Very Cold Day");

}

foreach (KeyValuePair<double,string> t in Coments)

{

if(temp1 == t.Key)

{

if (temp1 >= 40)

{

label5.ForeColor = Color.Red;

label5.Text = t.Value.ToString();

}

else if (temp1 >= 21)

{

label5.ForeColor = Color.Green;

label5.Text = t.Value.ToString();

}

else if (temp1 <= 10)

{

label5.ForeColor = Color.Blue;

label5.Text = t.Value.ToString();

}

}

}

}

if (textBox1.Text != "" && radioButton2.Checked)

{

temp2 = Math.Round((temp1 - 32)\*5/9,2); //convertion from C to F

textBox2.Text = temp2.ToString(); //display the result on the textbox 2

if (temp2>=20 && temp2 < 24)

{

Coments.Add(temp2, "Room temperature");

}

else if (temp2>=-20 && temp2 < -15)

{

Coments.Add(temp2, "Very Cold Day");

}

foreach (KeyValuePair<double, string> t in Coments)

{

if (temp2 == t.Key)

{

if (temp2 >= 40)

{

label5.ForeColor = Color.Red;

label5.Text = t.Value.ToString();

}

else if (temp2 >= 21)

{

label5.ForeColor = Color.Green;

label5.Text = t.Value.ToString();

}

else if (temp2 <= 10)

{

label5.ForeColor = Color.Blue;

label5.Text = t.Value.ToString();

}

}

}

}

//save the conversions in a text file

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

StreamWriter txt = new StreamWriter(fs);

string print = $"{temp1} {label1.Text} = {temp2} {label2.Text}"; //change the text in the txt file, replacing the tabs for ,

txt.Write($"{print}, {DateTime.Now} {label5.Text}\n"); //printing in the txt file

txt.Close();

txt.Dispose();

}

private void radioButton1\_CheckedChanged(object sender, EventArgs e)

{

label1.Text = "C";

label2.Text = "F";

textBox2.Text = "";

label5.Text = "";

}

private void radioButton2\_CheckedChanged(object sender, EventArgs e)

{

label1.Text = "F";

label2.Text = "C";

textBox2.Text = "";

label5.Text = "";

}

private void button2\_Click(object sender, EventArgs e)

{

string title = "Previous Conversions";

string textToPrint = "";

FileStream stream = null;

byte counter = 0;

try

{

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

StreamReader txt = new StreamReader(stream);

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

{

string prevConv = txt.ReadLine();

textToPrint += prevConv + "\n";

counter++;

if (counter == 12)

{

MessageBox.Show(textToPrint, title);

textToPrint = "";

counter = 0;

}

}

if (counter > 0) { MessageBox.Show(textToPrint, title); }

txt.Close();

}

catch (IOException ex)

{

MessageBox.Show("IO Exception\n" + ex.Message);

}

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

}

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

{

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

{

this.Close();

}

}

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

{

}

}

**IP Validation**

public partial class IP4Valid : Form

{

public IP4Valid()

{

InitializeComponent();

}

private bool validIP(string ip)

{

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}$");

return myRegex.IsMatch(ip);

}

DateTime load;

private void IP4Valid\_Load(object sender, EventArgs e) //form loaded

{

DateTime today = DateTime.Now; //time that the form is loaded

label1.Text = $"Today: {today}";

load = DateTime.Now;

if (Directory.Exists(@".\Files\") == false) //if the file does not exist it is created

{

Directory.CreateDirectory(@".\Files\");

}

}

string pathBinary = @".\Files\BIP.dat"; //global variable with the path for the file

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

{

if (validIP(textBox1.Text) == false)

{

MessageBox.Show($"{textBox1.Text}\nThe IP must have 4 bytes\ninteger number between 0 to 255\nseparated by a dot(255.255.255.255)","Error");

textBox1.Focus();

}

else

{

MessageBox.Show($"{textBox1.Text}\nThe IP is correct", "Valid IP");

FileStream fs = null; //declaration of the file

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

// create the output stream for a binary file

BinaryWriter bw = new BinaryWriter(fs);

string ipValid = textBox1.Text.Trim();

bw.Write($"{ipValid}, {DateTime.Now}"); //write the IP validated and the time in a dat file

bw.Close();

fs.Close();

}

}

private void button2\_Click(object sender, EventArgs e)

{

textBox1.Text = "";

textBox1.Focus();

}

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

{

TimeSpan timeSpent = new TimeSpan();

timeSpent = DateTime.Now.Subtract(load); //interval between the time the form was loaded and the form required to close.

string ts = $"{timeSpent.Minutes:D2}min:{timeSpent.Seconds:D2}s";

if (MessageBox.Show($"You spent {ts} in this application.\nDo you want to quit the application IP Validator?", "Exit ?", MessageBoxButtons.YesNo, MessageBoxIcon.Question).ToString() == "Yes")

{

this.Close();

}

}

}

**Calculation**

In order to perform the calculations, I created a new class called Calculator with the following code:

internal class Calculator

{

//private fields

private decimal currentValue;

private decimal operand1;

private decimal operand2;

private string op;

//properties

public decimal CurrentValue

{

get { return currentValue; }

set { currentValue = value; }

}

public decimal Operand1

{

get { return operand1; }

set { operand1 = value; }

}

public decimal Operand2

{

get { return operand2; }

set { operand2 = value; }

}

public string Op

{

get { return op; }

set { op = value; }

}

//Methods

public decimal Add()

{

currentValue = operand1 + operand2;

return currentValue;

}

public decimal Subtract()

{

currentValue = operand1 - operand2;

return currentValue;

}

public decimal Multiply()

{

currentValue = operand1 \* operand2;

return currentValue;

}

public decimal Divide()

{

currentValue = Math.Round(operand1 / operand2, 4);

return currentValue;

}

public void Equals()

{

switch (op)

{

case "+":

{

Add();

break;

}

case "-":

{

Subtract();

break;

}

case "\*":

{

Multiply();

break;

}

case "/":

{

Divide();

break;

}

}

}

}

**Code in the form SimpleCalc.cs:**

public partial class SimpleCalc : Form

{

public SimpleCalc()

{

InitializeComponent();

}

string path = @".\Files\Calculator.txt";

FileStream fs = null;

//Constructor

Calculator calculator = new Calculator();

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

{

textBox1.Text = $"{textBox1.Text}1";

}

private void button2\_Click(object sender, EventArgs e)

{

textBox1.Text = $"{textBox1.Text}2";

}

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

{

textBox1.Text = $"{textBox1.Text}3";

}

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

{

textBox1.Text = $"{textBox1.Text}4";

}

private void button5\_Click(object sender, EventArgs e)

{

textBox1.Text = $"{textBox1.Text}5";

}

private void button6\_Click(object sender, EventArgs e)

{

textBox1.Text = $"{textBox1.Text}6";

}

private void button7\_Click(object sender, EventArgs e)

{

textBox1.Text = $"{textBox1.Text}7";

}

private void button8\_Click(object sender, EventArgs e)

{

textBox1.Text = $"{textBox1.Text}8";

}

private void button9\_Click(object sender, EventArgs e)

{

textBox1.Text = $"{textBox1.Text}9";

}

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

{

textBox1.Text = $"{textBox1.Text}0";

}

private void button11\_Click(object sender, EventArgs e)

{

bool hasPoint = Regex.IsMatch(textBox1.Text, @"\."); //regex to verify if already exist some "." in the number. It is allowed only one "." for each number.

if(hasPoint == false)

{

textBox1.Text = $"{textBox1.Text}.";

}

}

private void button12\_Click(object sender, EventArgs e)

{

if (textBox1.Text == "" || textBox1.Text == ".")

{

MessageBox.Show("Enter with a number first");

}

else

{

calculator.Op = "+";

calculator.Operand1 = Convert.ToDecimal(textBox1.Text);

label1.Text = calculator.Operand1.ToString() + " " + calculator.Op;

textBox1.Clear();

button12.Enabled = false;

button13.Enabled = false;

button14.Enabled = false;

button15.Enabled = false;

}

}

private void button13\_Click(object sender, EventArgs e)

{

if (textBox1.Text == "" || textBox1.Text == ".")

{

MessageBox.Show("Enter with a number first");

}

else

{

calculator.Op = "-";

calculator.Operand1 = Convert.ToDecimal(textBox1.Text);

label1.Text = calculator.Operand1.ToString() + " " + calculator.Op;

textBox1.Clear();

button12.Enabled = false;

button13.Enabled = false;

button14.Enabled = false;

button15.Enabled = false;

}

}

private void button14\_Click(object sender, EventArgs e)

{

if (textBox1.Text == "" || textBox1.Text == ".")

{

MessageBox.Show("Enter with a number first");

}

else

{

calculator.Op = "\*";

calculator.Operand1 = Convert.ToDecimal(textBox1.Text);

label1.Text = calculator.Operand1.ToString() + " " + calculator.Op;

textBox1.Clear();

button12.Enabled = false;

button13.Enabled = false;

button14.Enabled = false;

button15.Enabled = false;

}

}

private void button15\_Click(object sender, EventArgs e)

{

if (textBox1.Text == "" || textBox1.Text == ".")

{

MessageBox.Show("Enter with a number first");

}

else

{

calculator.Op = "/";

calculator.Operand1 = Convert.ToDecimal(textBox1.Text);

label1.Text = calculator.Operand1.ToString() + " " + calculator.Op;

textBox1.Clear();

button12.Enabled = false;

button13.Enabled = false;

button14.Enabled = false;

button15.Enabled = false;

}

}

private void button16\_Click(object sender, EventArgs e)

{

calculator.Operand2 = Convert.ToDecimal(textBox1.Text);

switch (calculator.Op)

{

case "+":

{

textBox1.Text = calculator.Add().ToString();

label1.Text = $"{calculator.Operand1} {calculator.Op} {calculator.Operand2} = {calculator.CurrentValue}";

textBox1.Text = calculator.CurrentValue.ToString();

break;

}

case "-":

{

textBox1.Text = calculator.Subtract().ToString();

label1.Text = $"{calculator.Operand1} {calculator.Op} {calculator.Operand2} = {calculator.CurrentValue}";

textBox1.Text = calculator.CurrentValue.ToString();

break;

}

case "\*":

{

textBox1.Text = calculator.Multiply().ToString();

label1.Text = $"{calculator.Operand1} {calculator.Op} {calculator.Operand2} = {calculator.CurrentValue}";

textBox1.Text = calculator.CurrentValue.ToString();

break;

}

case "/":

{

if(calculator.Operand2 != 0)

{

textBox1.Text = calculator.Divide().ToString();

label1.Text = $"{calculator.Operand1} {calculator.Op} {calculator.Operand2} = {calculator.CurrentValue}";

textBox1.Text = calculator.CurrentValue.ToString();

}

else

{

label1.Text = $"{calculator.Operand1} {calculator.Op} {calculator.Operand2} = NaN";

textBox1.Clear();

}

break;

}

}

//liberate the operation numbers

button12.Enabled = true;

button13.Enabled = true;

button14.Enabled = true;

button15.Enabled = true;

//save the conversions in a text file

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

StreamWriter txt = new StreamWriter(fs);

//StreamWriter txt = new StreamWriter(@".\Files\Calculator.txt", true); //path where will be generated the text file

string print = $"{calculator.Operand1} {calculator.Op} {calculator.Operand2} = {calculator.CurrentValue}"; //creating the string to save in the txt file,

txt.Write($"{print}\n"); //printing in the txt file

txt.Close();

txt.Dispose();

}

private void button17\_Click(object sender, EventArgs e)

{

label1.Text = "";

textBox1.Clear();

calculator.Operand1 = 0;

calculator.Operand2 = 0;

calculator.CurrentValue = 0;

}

private void button18\_Click(object sender, EventArgs e)

{

if (MessageBox.Show("Do you want to quit the application Simple Calculator?", "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.**

Follow some of the Classes, Structs and Methods that I used to create this project:

|  |  |
| --- | --- |
| **Class/Method Name** | **Description** |
| 1. Random | Class to create random numbers for the Lotto |
| 1. FileStream | Class to manipulate a file, write, read, open and close files. |
| 1. .Contains() | Method to identify if a list has or not some specific number |
| 1. .Add() | Method to add some number in a list |
| 1. .ToString() | Method to convert other type of variable into a string |
| 1. StreamReader | Class to read characters from a file |
| 1. StreamWriter | Class to write characters in a file |
| 1. Dictionary | Class that represent a collection of keys and values |
| 1. DateTime | Struct to manage date and time information |
| 1. TimeSpan | Struct to manage interval of time information |
| 1. Regex | Class to deal with regular expressions |
| 1. Calculator | Class created receive the numbers and store the methods for calculations |
| 1. Add(), Subtract(), Multiply(), Divide(), Equals() | Methods inside the class Calculator, to perform the calculations |

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

* The generation of the unique random numbers was an issue, but I could handle it, putting the numbers in a list and using the method Contains(), to verify if the random number already exists in the previous numbers.
* The generation of many Message Box was necessary to read all the information from the text box. When the amount of information is too long, the Message Box will occupy all the screen, so it was better create a Message Box for each 12 lines.
* To calculate and display the time the app stayed opened was a challenge for me, I found the way to display TimeSpan type using a string in the format that I wanted.