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

|  |
| --- |
| Your name: María Angélica González Aragón  11/9/2023  Version 1.0 |

1. **Start by adding a short description of your project, and the languages (technologies) used:**
2. Language: C#
3. Used tool(s): Visual Studio, MS Word, MS PowerPoint (flag images).
4. **Present the print screens of yours forms and have a detailed description of the functionalities (step by step).**
5. When you select the run option, you will find the application dashboard:

A screenshot of a computer

Description automatically generated

Here, you will find 4 tabs and the Exit button.

1. The first tab “Generated Numbers” contains two Lottery options.

A screenshot of a computer

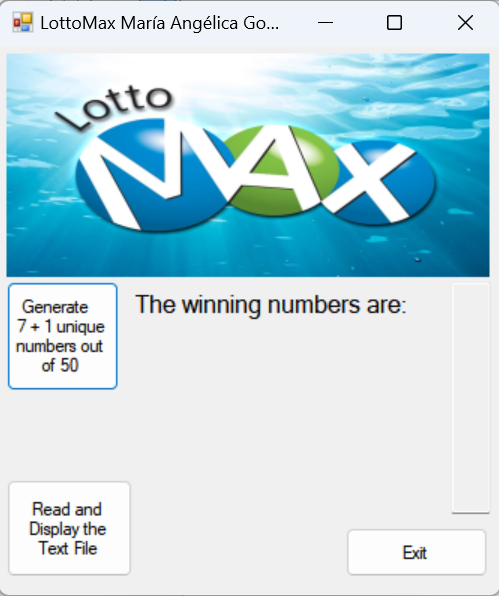
Description automatically generated

1. If you click on the first option, Lotto MAX:

A screenshot of a computer

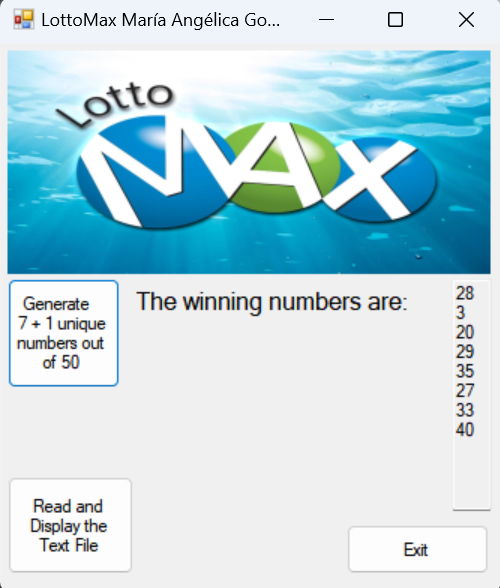
Description automatically generated

1. This will open a new Window:

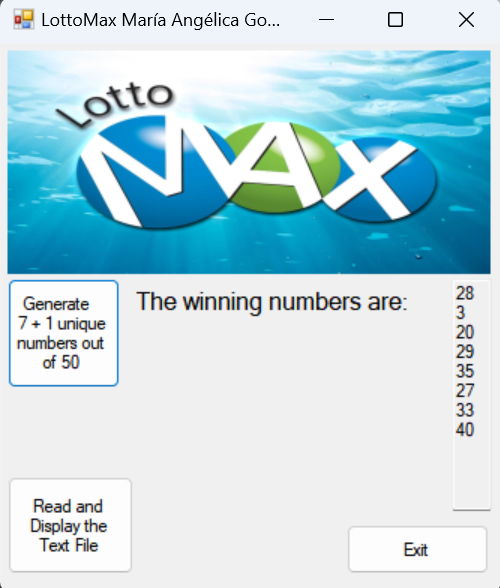
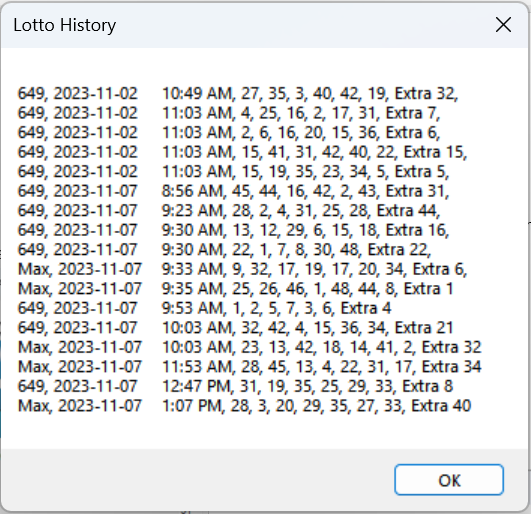


In this Window you have three options.

1. If you click on “Generate 7 + 1 unique numbers out of 50” button, the application will return the winning numbers in the text box, which consists in 8 random unique numbers between 1 and 50, the last number is the extra number.



1. If you click on “Read and Display the Text File” button, the application will open a new message box, presenting all the Historical information of the winning numbers generated for both lottery “Max” and “Lotto649”. Here you can see the lottery, the date and time, as well as the regular numbers and the extra number. At the end you can close the message box, by pressing “Ok” Button or by clicking on the closing option on the top.

1. If you click on the Exit button, a message box will be displayed, asking if you are sure and the time you spend on this lottery window:

A screenshot of a game

Description automatically generatedA screenshot of a computer

Description automatically generated

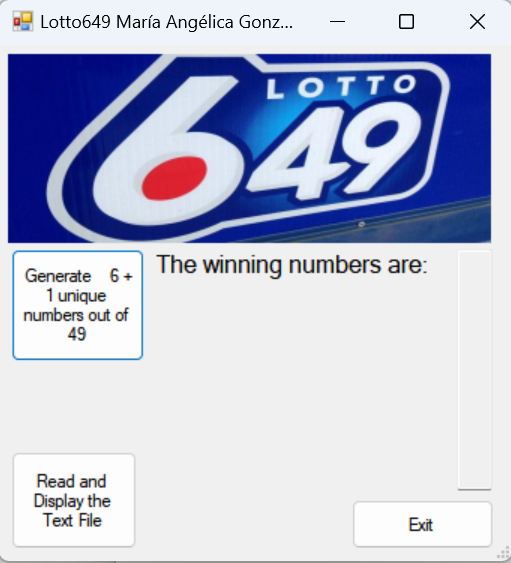
Here you must click on “Yes” to exit LottoMax window and “No” to continue in this Window.

1. Similarly, If you click on the second option, Lotto649:

A screenshot of a computer

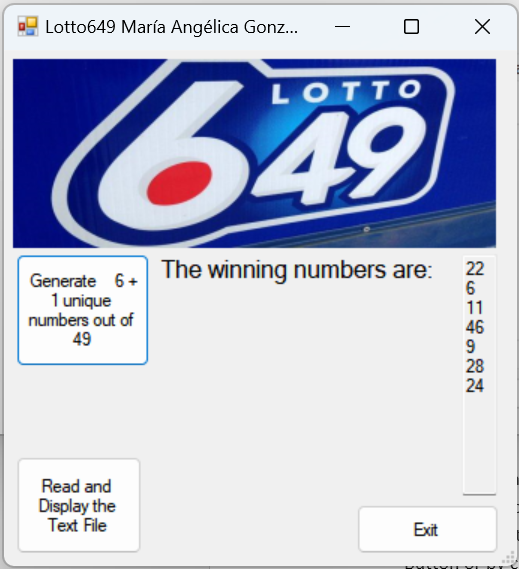
Description automatically generated

1. This will open the corresponding new Window:

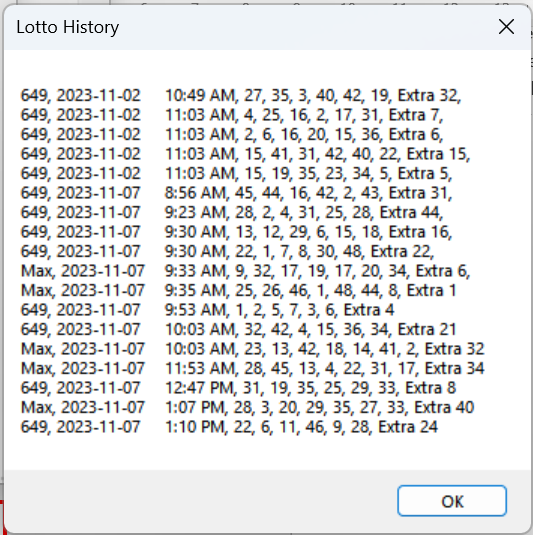
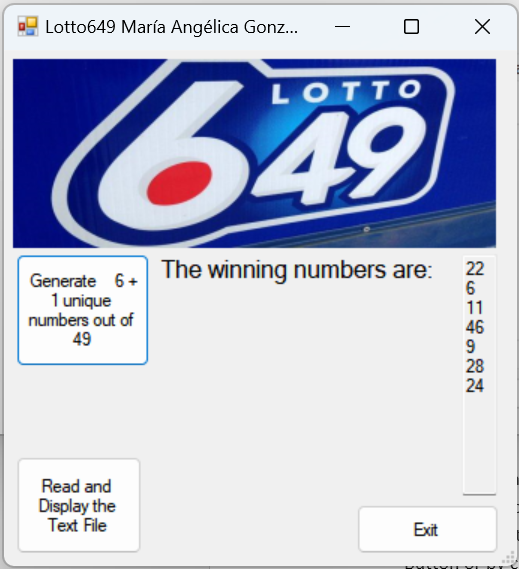


In this Window you have three options.

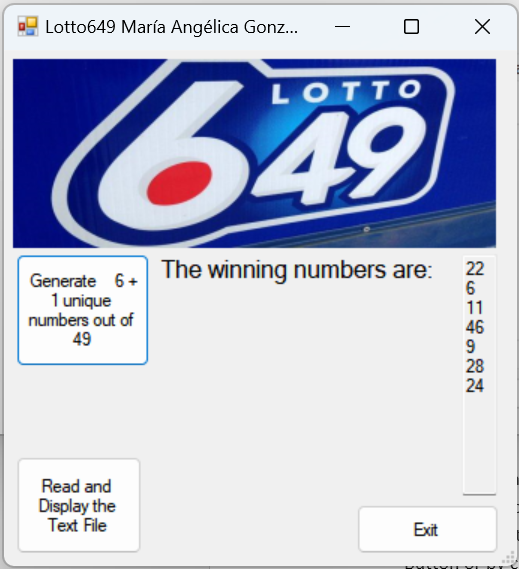
1. If you click on “Generate 6 + 1 unique numbers out of 49” button, the application will return the winning numbers in the text box, which consists in 7 random unique numbers between 1 and 49, the last number is the extra number.



1. If you click on “Read and Display the Text File” button, the application will open a new message box, presenting all the Historical information of the winning numbers generated for both lottery “Max” and “Lotto649”. Here you can see the lottery, the date and time, as well as the regular numbers and the extra number. At the end you can close the message box, by pressing “Ok” Button or by clicking on the closing option on the top.



1. If you click on the Exit button, a message box will be displayed, asking if you are sure and the time you spend on this lottery window:

A screenshot of a computer

Description automatically generated

Here you must click on “Yes” to exit Lotto649 window and “No” to continue in this Window.

1. The second tab “Conversions” contains two converter options.

A screenshot of a computer

Description automatically generated

1. If you click on the first option, Money Exchange:

A screenshot of a computer

Description automatically generated

1. This will open a money exchange application, where you can select the origin currency and register the value to exchange, in the “from” area:

A screenshot of a computer

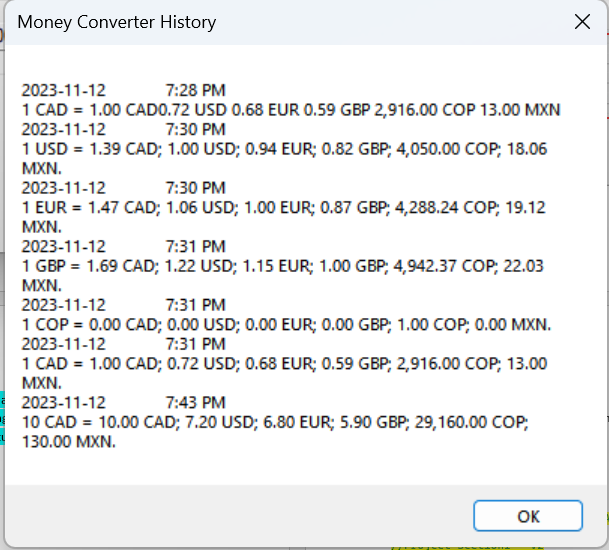
Description automatically generated

1. By clicking on the “Convert from (choose a country)” button, you will find the value in the different currencies, displayed in the “To” area:

A screenshot of a computer

Description automatically generated

1. If you click on “Read and Display the Text File” button, the application will open a new message box, presenting all the Historical information of the Exchange conversion generated. Here you can see the date and time, as well as the origin value and currency and the converted values and currencies. . At the end you can close the message box, by pressing “Ok” Button or by clicking on the closing option on the top:



1. If you click on the Exit button, a message box will be displayed, asking if you are sure and the time you spend (minutes and seconds) on this exchange money Window:

A screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated

Here you must click on “Yes” to exit Money Exchange application and “No” to continue in this Window.

1. If you click on the second option, Temperature Convert:

A screenshot of a computer

Description automatically generated

1. This will open a temperature converter application, where you can choose between two options: C to F or F to C. By default, C to F option will be selected, if you do not select the other radio button, you will continue in the same window and you can convert from Celsius to Fahrenheit:

A screenshot of a computer

Description automatically generated

1. By filling the value to convert above “C” and clicking on the “Convert” button, you will find the value in Fahrenheit, displayed in the “F” field, and a message and color associated with this temperature:

A screenshot of a computer

Description automatically generated

1. If you click on “Read File” button, the application will open a new message box, presenting all the Historical information of the Temperature conversions generated. Here you can find the original value and the result converted value, the date and time, as well as the message displayed. At the end you can close the message box, by pressing “Ok” Button or by clicking on the closing option on the top:

A screenshot of a computer

Description automatically generated

1. If you select the option F to C:

A screenshot of a computer

Description automatically generated

1. This will open a new window that has the same functionality as the previous window: A screenshot of a computer

   Description automatically generated
2. If you click on “Read File” button, the application will also open a new message box, presenting all the same Historical information of the Temperature conversions generated. A screenshot of a computer

   Description automatically generated
3. If you click on the Exit Button of both windows, a message box will be displayed, asking if you are sure and the time you spend (minutes and seconds) on this temperature converter Window: A screenshot of a computer

   Description automatically generatedA screenshot of a computer

   Description automatically generated
4. The forth tab “IP v4 & v6 Validator” contains one IP validator.

A screenshot of a computer

Description automatically generated

1. If you click it, this will open an IP validator for IPv4 and IPv6:

A screenshot of a computer

Description automatically generated

1. Here you can insert and validate an ip v4 or v6 and get the result of your validation by clicking on “Validate IP” button, after inserting the value of the IP to validate in the corresponding field.

A screenshot of a computer error message

Description automatically generated

If the IP is invalid, it will give you the indication to correct it for IPV4 or IPV6 according to the one that was filled out. If both fields were filled out, it will display the value for the IPv4 input:

A screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated

1. By clicking on “Reset” button the value for both fields will be erased.
2. If you click on “Read and Display” button, the application will open a new message box, presenting the Historical information of the IPS validated. At the end you can close the message box, by pressing “Ok” Button or by clicking on the closing option on the top:

A screenshot of a computer

Description automatically generatedA screenshot of a computer error message

Description automatically generated

1. If you click on the Exit Button, a message box will be displayed, asking if you are sure o quit and the time you spend (minutes and seconds) on this IP Validator Window: A screenshot of a computer

   Description automatically generatedA screenshot of a computer

   Description automatically generated
2. Finally, you can exit the application by clicking on the exit button, that will open a new message box, that will ask if you are sure and the time you spend on the application.

A screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated

Here you must click on “Yes” to exit the app and “No” to continue using it.

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

**Dash board**

using System;

using System.Diagnostics;

using System.Windows.Forms;

//Name: María Angélica González Aragón

//Date: November 7th

//Project Section1 - V2

namespace ProjectSection123

{

public partial class frm23Dashboard : Form

{

private Stopwatch stopwatch = new Stopwatch();

public frm23Dashboard()

{

InitializeComponent();

stopwatch.Start();

}

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

{

if (MessageBox.Show($"Do you want to quit this app? You have been here {Math.Floor(stopwatch.Elapsed.TotalMinutes)} minutes", "Exit", MessageBoxButtons.YesNo) == DialogResult.Yes)

{

stopwatch.Stop();

this.Close();

}

}

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

{

LottoMax obj = new LottoMax();

obj.Show();

}

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

{

Lotto649 obj = new Lotto649();

obj.ShowDialog();

}

private void button2\_Click\_1(object sender, EventArgs e)

{

Lotto649 obj = new Lotto649();

obj.ShowDialog();

}

}

}

**LotoMAX**

using System;

using System.Diagnostics;

using System.IO;

using System.Windows.Forms;

//Name: María Angélica González Aragón

//Date: November 7th

//Project Section1 - V2

namespace ProjectSection123

{

public partial class LottoMax : Form

{

private Stopwatch stopwatch = new Stopwatch();

public LottoMax()

{

InitializeComponent();

stopwatch.Start();

}

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

{

string tempo = "", tempoText = "Max, " +

DateTime.Now.ToShortDateString() + "\t" +

DateTime.Now.ToShortTimeString() + ", ";

Random random = new Random();

bool[] numbersGenerated = new bool[50];

for (int i = 0; i < 8; i++)

{

int randomNumber;

do

{

randomNumber = random.Next(1, 50);

} while (numbersGenerated[randomNumber]);

numbersGenerated[randomNumber] = true;

tempo += randomNumber.ToString() + "\t";

if (i == 7)

{

tempoText += "Extra " + randomNumber.ToString();

}

else

{

tempoText += randomNumber.ToString() + ", ";

}

}

textBox1.Text = tempo;

FileStream fs = new FileStream(@"./LottoNumbers.txt", FileMode.Append, FileAccess.Write);

StreamWriter objW = new StreamWriter(fs);

objW.WriteLine(tempoText);

objW.Close();

fs.Close();

}

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

{

if (MessageBox.Show($"Do you want to quit lotto Max Window? You have been here {Math.Floor(stopwatch.Elapsed.TotalMinutes)} minutes", "Exit", MessageBoxButtons.YesNo) == DialogResult.Yes)

{

stopwatch.Stop();

this.Close();

}

}

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

{

string tempoText;

FileStream fs = new FileStream(@"./LottoNumbers.txt", FileMode.Open, FileAccess.Read);

StreamReader objR = new StreamReader(fs);

tempoText = objR.ReadToEnd();

objR.Close();

fs.Close();

MessageBox.Show(tempoText, "Lotto History");

}

}

}

**Loto649**

using System;

using System.Diagnostics;

using System.IO;

using System.Windows.Forms;

//Name: María Angélica González Aragón

//Date: November 7th

//Project Section1 - V2

namespace ProjectSection123

{

public partial class Lotto649 : Form

{

private Stopwatch stopwatch = new Stopwatch();

public Lotto649()

{

InitializeComponent();

stopwatch.Start();

}

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

{

string tempo = "", tempoText = "649, " +

DateTime.Now.ToShortDateString() + "\t" +

DateTime.Now.ToShortTimeString() + ", ";

Random random = new Random();

bool[] numbersGenerated = new bool[49];

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

{

int randomNumber;

do

{

randomNumber = random.Next(1, 49);

} while (numbersGenerated[randomNumber]);

numbersGenerated[randomNumber] = true;

tempo += randomNumber.ToString() + "\t";

if (i == 6)

{

tempoText += "Extra " + randomNumber.ToString();

}

else

{

tempoText += randomNumber.ToString() + ", ";

}

}

textBox1.Text = tempo;

FileStream fs = new FileStream(@"./LottoNumbers.txt", FileMode.Append, FileAccess.Write);

StreamWriter objW = new StreamWriter(fs);

objW.WriteLine(tempoText);

objW.Close();

fs.Close();

}

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

{

string tempoText;

FileStream fs = new FileStream(@"./LottoNumbers.txt", FileMode.Open, FileAccess.Read);

StreamReader objR = new StreamReader(fs);

tempoText = objR.ReadToEnd();

objR.Close();

fs.Close();

MessageBox.Show(tempoText,"Lotto History");

}

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

{

if (MessageBox.Show($"Do you want to quit lotto 649 Window? You have been here {Math.Floor(stopwatch.Elapsed.TotalMinutes)} minutes", "Exit", MessageBoxButtons.YesNo) == DialogResult.Yes)

{

stopwatch.Stop();

this.Close();

}

}

}

}

Conversion

**Money Exchange**

using System;

using System.Diagnostics;

using System.Drawing;

using System.IO;

using System.Windows.Forms;

//Name: María Angélica González Aragón

//Date: November 16th

//Project Section2 - V1

namespace ProjectSection123

{

public partial class MoneyExchange : Form

{

private Stopwatch stopwatch = new Stopwatch();

public MoneyExchange()

{

InitializeComponent();

stopwatch.Start();

}

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

{

double value = double.Parse(textBox1.Text);

try

{

//Convert factor "To CAD" - November 12

//1 CAD = 0.72 USD

//1 CAD = 0.68 EUR

//1 CAD = 0.59 GBP

//1 CAD = 2916 COP

//1 CAD = 13 MXN

double curCAD = 1.0;

double curUSD = 0.72;

double curEUR = 0.68;

double curGBP = 0.59;

double curCOP = 2916;

double curMXN = 13;

double origin = 0.0;

if (radioButton1.Checked)

origin = curCAD;

else if (radioButton3.Checked)

origin = curUSD;

else if (radioButton2.Checked)

origin = curEUR;

else if (radioButton4.Checked)

origin = curGBP;

else if (radioButton5.Checked)

origin = curCOP;

string originCurrency = "";

if (radioButton1.Checked)

originCurrency = " CAD";

else if (radioButton3.Checked)

originCurrency = " USD";

else if (radioButton2.Checked)

originCurrency = " EUR";

else if (radioButton4.Checked)

originCurrency = " GBP";

else if (radioButton5.Checked)

originCurrency = " COP";

double resultCAD = value \* (curCAD / origin);

double resultUSD = value \* (curUSD / origin);

double resultEUR = value \* (curEUR / origin);

double resultGBP = value \* (curGBP / origin);

double resultCOP = value \* (curCOP / origin);

double resultMXN = value \* (curMXN / origin);

textBox2.Text = resultCAD.ToString("N2") + " CAD";

textBox4.Text = resultUSD.ToString("N2") + " USD";

textBox3.Text = resultEUR.ToString("N2") + " EUR";

textBox5.Text = resultGBP.ToString("N2") + " GBP";

textBox7.Text = resultCOP.ToString("N2") + " COP";

textBox6.Text = resultMXN.ToString("N2") + " MXN";

string exchangeText = DateTime.Now.ToShortDateString() + "\t" + DateTime.Now.ToShortTimeString() + "\n" +

value + originCurrency + " = " + resultCAD.ToString("N2") + " CAD;" + " " +

resultUSD.ToString("N2") + " USD;" + " " +

resultEUR.ToString("N2") + " EUR;" + " " +

resultGBP.ToString("N2") + " GBP;" + " " +

resultCOP.ToString("N2") + " COP;" + " " +

resultMXN.ToString("N2") + " MXN.";

using (FileStream fs = new FileStream(@"./MoneyConversions.txt", FileMode.Append, FileAccess.Write))

using (StreamWriter objW = new StreamWriter(fs))

{

objW.WriteLine(exchangeText);

}

}

catch (FormatException)

{

MessageBox.Show("Please, type a valid numerical number", "Format error");

}

catch (Exception ex)

{

MessageBox.Show($"Error: {ex.Message}", "Error");

}

}

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

{

string exchangeText;

FileStream fs = new FileStream(@"./MoneyConversions.txt", FileMode.Open, FileAccess.Read);

StreamReader objR = new StreamReader(fs);

exchangeText = objR.ReadToEnd();

objR.Close();

fs.Close();

MessageBox.Show(exchangeText, "Money Converter History");

}

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

{

double totalMinutes = stopwatch.Elapsed.TotalMinutes;

int minutes = (int)Math.Floor(totalMinutes);

int seconds = (int)Math.Floor((totalMinutes - minutes) \* 60);

if (MessageBox.Show($"Do you want to quit this app? You have been here {minutes} minutes {seconds} seconds.", "Exit", MessageBoxButtons.YesNo) == DialogResult.Yes)

{

stopwatch.Stop();

this.Close();

}

}

}

}

**Temperature Converter (C to F)**

using System;

using System.Diagnostics;

using System.Drawing;

using System.IO;

using System.Windows.Forms;

//Name: María Angélica González Aragón

//Date: November 16th

//Project Section2 - V1

namespace ProjectSection123

{

public partial class temperatureConverter : Form

{

private Stopwatch stopwatch = new Stopwatch();

public temperatureConverter()

{

InitializeComponent();

stopwatch.Start();

}

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

{

double celsius;

double fahrenheit;

string message;

try

{

celsius = Convert.ToDouble(textBox1.Text);

fahrenheit = (celsius \* 9 / 5) + 32;

textBox2.Text = $"{fahrenheit:F2} °F";

message = GetMessageDescription(celsius);

if (celsius >= 40)

{

textBox3.BackColor = Color.Red;

textBox1.ForeColor = Color.Red;

}

else if (celsius < 40 && celsius >= 30)

{

textBox3.BackColor = Color.Yellow;

textBox1.ForeColor = Color.Yellow;

}

else if (celsius < 30 && celsius >= 21)

{

textBox3.BackColor = Color.Green;

textBox1.ForeColor = Color.Green;

}

else if (celsius < 21 && celsius >= 0)

{

textBox3.BackColor = Color.Blue;

textBox1.ForeColor = Color.Blue;

}

else if (celsius == -40)

{

textBox3.Font = new Font(textBox3.Font, FontStyle.Bold);

}

textBox3.Text = message;

string temperatureText = celsius + " C = " + fahrenheit.ToString("F2") + " F," +

"\t" + DateTime.Now.ToShortDateString() + "\t" +

DateTime.Now.ToShortTimeString() + ", " +

message;

using (FileStream fs = new FileStream(@"./TempConversions.txt", FileMode.Append, FileAccess.Write))

using (StreamWriter objW = new StreamWriter(fs))

{

objW.WriteLine(temperatureText);

}

}

catch (FormatException)

{

MessageBox.Show("Please, type a valid numerical number", "Format error");

}

catch (Exception ex)

{

MessageBox.Show($"Error: {ex.Message}", "Error");

}

}

private string GetMessageDescription(double celsius)

{

if (celsius >= 100)

return " Water boils";

else if (celsius >= 40)

return " Hot Bath";

else if (celsius >= 37)

return " Body temperature";

else if (celsius >= 30)

return "Beach weather";

else if (celsius >= 21)

return "Room temperature";

else if (celsius >= 10)

return "Cool Day";

else if (celsius >= 0)

return "Freezing point of water";

else if (celsius >= -18)

return "Very Cold Day";

else if (celsius >= -40)

return "Extremely Cold Day \r\n(and the same number!) ";

else

return "Non valid value";

}

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

{

string temperatureText;

FileStream fs = new FileStream(@"./TempConversions.txt", FileMode.Open, FileAccess.Read);

StreamReader objR = new StreamReader(fs);

temperatureText = objR.ReadToEnd();

objR.Close();

fs.Close();

MessageBox.Show(temperatureText, "Temperature Converter History");

}

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

{

double totalMinutes = stopwatch.Elapsed.TotalMinutes;

int minutes = (int)Math.Floor(totalMinutes);

int seconds = (int)Math.Floor((totalMinutes - minutes) \* 60);

if (MessageBox.Show($"Do you want to quit this app? You have been here {minutes} minutes {seconds} seconds.", "Exit", MessageBoxButtons.YesNo) == DialogResult.Yes)

{

stopwatch.Stop();

this.Close();

}

}

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

{

this.Close();

if (radioButton2.Checked)

{

temperatureConverter2 obj = new temperatureConverter2();

obj.StartPosition = FormStartPosition.CenterParent;

obj.ShowDialog(this);

}

}

}

}

**Temperature Converter (F to C)**

using System;

using System.Diagnostics;

using System.Drawing;

using System.IO;

using System.Windows.Forms;

//Name: María Angélica González Aragón

//Date: November 16th

//Project Section2 - V1

namespace ProjectSection123

{

public partial class temperatureConverter2 : Form

{

private Stopwatch stopwatch = new Stopwatch();

public temperatureConverter2()

{

InitializeComponent();

stopwatch.Start();

}

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

{

}

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

{

this.Close();

temperatureConverter obj = new temperatureConverter();

obj.ShowDialog();

}

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

{

double celsius;

double fahrenheit;

string message;

try

{

fahrenheit = Convert.ToDouble(textBox1.Text);

celsius = (fahrenheit - 32)\* 5 / 9;

textBox2.Text = $"{celsius:F2} °C";

message = GetMessageDescription(fahrenheit);

if (fahrenheit >= 104)

{

textBox3.BackColor = Color.Red;

textBox1.ForeColor = Color.Red;

}

else if (fahrenheit < 104 && fahrenheit >= 86)

{

textBox3.BackColor = Color.Yellow;

textBox1.ForeColor = Color.Yellow;

}

else if (fahrenheit < 86 && fahrenheit >= 70)

{

textBox3.BackColor = Color.Green;

textBox1.ForeColor = Color.Green;

}

else if (fahrenheit < 70 && fahrenheit >= 32)

{

textBox3.BackColor = Color.Blue;

textBox1.ForeColor = Color.Blue;

}

else if (fahrenheit == -40)

{

textBox3.Font = new Font(textBox3.Font, FontStyle.Bold);

}

textBox3.Text = message;

string temperatureText = fahrenheit + " F = " + celsius.ToString("F2") + " C," +

"\t" + DateTime.Now.ToShortDateString() + "\t" +

DateTime.Now.ToShortTimeString() + ", " +

message;

using (FileStream fs = new FileStream(@"./TempConversions.txt", FileMode.Append, FileAccess.Write))

using (StreamWriter objW = new StreamWriter(fs))

{

objW.WriteLine(temperatureText);

}

}

catch (FormatException)

{

MessageBox.Show("Please, type a valid numerical number", "Format error");

}

catch (Exception ex)

{

MessageBox.Show($"Error: {ex.Message}", "Error");

}

}

private string GetMessageDescription(double fahrenheit)

{

if (fahrenheit >= 212)

return " Water boils";

else if (fahrenheit >= 104)

return " Hot Bath";

else if (fahrenheit >= 98.6)

return " Body temperature";

else if (fahrenheit >= 86)

return "Beach weather";

else if (fahrenheit >= 70)

return "Room temperature";

else if (fahrenheit >= 50)

return "Cool Day";

else if (fahrenheit >= 32)

return "Freezing point of water";

else if (fahrenheit >= 0)

return "Very Cold Day";

else if (fahrenheit >= -40)

return "Extremely Cold Day \r\n(and the same number!) ";

else

return "Non valid value";

}

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

{

string temperatureText;

FileStream fs = new FileStream(@"./TempConversions.txt", FileMode.Open, FileAccess.Read);

StreamReader objR = new StreamReader(fs);

temperatureText = objR.ReadToEnd();

objR.Close();

fs.Close();

MessageBox.Show(temperatureText, "Temperature Converter History");

}

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

{

double totalMinutes = stopwatch.Elapsed.TotalMinutes;

int minutes = (int)Math.Floor(totalMinutes);

int seconds = (int)Math.Floor((totalMinutes - minutes) \* 60);

if (MessageBox.Show($"Do you want to quit this app? You have been here {minutes} minutes {seconds} seconds.", "Exit", MessageBoxButtons.YesNo) == DialogResult.Yes)

{

stopwatch.Stop();

this.Close();

}

}

private void temperatureConverter2\_Load(object sender, EventArgs e)

{

}

}

}

**IPv4 & IP**

using System;

using System.Diagnostics;

using System.Text.RegularExpressions;

using System.Windows.Forms;

using System.IO;

using System.Net;

//Name: María Angélica González Aragón

//Date: November 16th

//Project Section2 - V2

namespace ProjectSection123

{

public partial class IP\_Validator : Form

{

private Stopwatch stopwatch = new Stopwatch();

public IP\_Validator()

{

InitializeComponent();

DisplayCurrentDate();

stopwatch.Start();

}

private void DisplayCurrentDate()

{

DateTime currentDate = DateTime.Now;

string formattedDate = currentDate.ToLongDateString();

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

}

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

{

double totalMinutes = stopwatch.Elapsed.TotalMinutes;

int minutes = (int)Math.Floor(totalMinutes);

int seconds = (int)Math.Floor((totalMinutes - minutes) \* 60);

if (MessageBox.Show($"Do you want to quit this app? You have been here {minutes} minutes {seconds} seconds.", "Exit", MessageBoxButtons.YesNo) == DialogResult.Yes)

{

stopwatch.Stop();

this.Close();

}

}

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

{

textBox1.Clear();

textBox2.Clear();

}

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

{

ValidateTextBoxes();

}

private void ValidateTextBoxes()

{

try

{

if (!string.IsNullOrEmpty(textBox1.Text))

{

ValidateTextBox1();

}

else if (!string.IsNullOrEmpty(textBox2.Text))

{

ValidateTextBox2();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Validation Error");

}

}

private void ValidateTextBox1()

{

string ipAddressString = textBox1.Text.Trim();

try

{

if (IsValidIPv4(ipAddressString))

{

WriteToBinaryFile(ipAddressString);

MessageBox.Show(ipAddressString + " The IP is correct", "Success");

}

else

{

MessageBox.Show(ipAddressString + " The IP must have 4 bytes, integer number, between 0 to 255, separated by a dot (255.255.255.255)", "Error");

}

}

catch (Exception ex)

{

MessageBox.Show("Error: " + ex.Message, "Error");

}

}

private bool IsValidIPv4(string ipAddressString)

{

string pattern = @"^(25[0-5]|2[0-4][0-9]|[0-1]?[0-9]?[0-9])\.(25[0-5]|2[0-4][0-9]|[0-1]?[0-9]?[0-9])\.(25[0-5]|2[0-4][0-9]|[0-1]?[0-9]?[0-9])\.(25[0-5]|2[0-4][0-9]|[0-1]?[0-9]?[0-9])$";

Regex regex = new Regex(pattern);

return regex.IsMatch(ipAddressString);

}

private void ValidateTextBox2()

{

string textBox2Value = textBox2.Text.Trim();

try

{

if (IsValidIPv6(textBox2Value))

{

WriteToBinaryFile(textBox2Value);

MessageBox.Show(textBox2Value + " The IP is correct", "Success");

}

else

{

MessageBox.Show(textBox2Value + ", The IP must have 8 quartets, hexadecimal numbers of 4 positions, separated by a colon (0123:4567:89ab:cdef:0123:4567:89ab:cdef)", "Error");

}

}

catch (Exception ex)

{

MessageBox.Show("Error: " + ex.Message, "Error");

}

}

private bool IsValidIPv6(string textBox2Value)

{

string pattern = @"^([0-9a-fA-F]{4}:){7}[0-9a-fA-F]{4}$";

Regex regex = new Regex(pattern);

return regex.IsMatch(textBox2Value);

}

private void WriteToBinaryFile(string dataToWrite)

{

string pathBinary = @"C:\Users\maria\OneDrive\Desktop\Maria Universidad\2nd Term\OOP\Project\ProjectSection123\ProjectSection123\bin\Debug\IPAddressData.dat";

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

using (BinaryWriter binaryOut = new BinaryWriter(fs))

{

// Write data followed by the date and time

string dataWithTimestamp = $"{dataToWrite} {DateTime.Now:yyyy-MM-dd HH:mm:ss}";

binaryOut.Write(dataWithTimestamp);

}

}

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

{

string exchangeText;

FileStream fs = new FileStream(@"./IPAddressData.dat", FileMode.Open, FileAccess.Read);

StreamReader objR = new StreamReader(fs);

exchangeText = objR.ReadToEnd();

objR.Close();

fs.Close();

MessageBox.Show(exchangeText, "IP Validator History");

}

}

}

Calculator

… your code goes here

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

|  |  |
| --- | --- |
| **Class/Method Name** | **Description** |
| 1. Method private Stopwatch stopwatch = new Stopwatch(); | It handles the time elapsed between the start and stop instruction. |
| 1. Try/catch   FormatException  Exception | Handles the format and value errors of input values. |
| 1. **Class: frm23Dashboard** | This class represents form for the main dashboard window of the application |
| 1. Constructor : public frm23Dashboard() | Initializes an instance of the class and starts a stopwatch when an instance of the class is created. |
| 1. Void Method : private void button1\_Click(object sender, EventArgs e) | Handles the click event of the "button1" control. It creates an instance of the LottoMax class and shows it. |
| 1. Void Method: private void button2\_Click(object sender, EventArgs e) | Handles the click event of the "button2" control. It creates an instance of the Lotto649 class and shows it. |
| 1. Void Method: private void button3\_Click(object sender, EventArgs e) | Handles the click event of the "button3" control, which the exit option. It displays a confirmation message to exit the application, showing the time elapsed. |
| 1. **Class: LottoMax** | This class represents the entire Lotto Max window. |
| 1. Constructor public LottoMax() | Initializes an instance of the class and starts a stopwatch when an instance of the class is created. |
| 1. Void Method private void button1\_Click(object sender, EventArgs e) | Handles the click event of the "button1" control : « Generate 7 + 1 unique numbers out of 50 ». It generates random numbers, ensuring they do not repeat, and writes the results into a text file. |
| 1. Void Method private void button3\_Click(object sender, EventArgs e) | Handles the click event of the "button3" control : exit. It displays a confirmation message to exit the Lotto Max window, showing the time elapsed. |
| 1. Void Method private void button2\_Click(object sender, EventArgs e) | Handles the click event of the "button2" control : « Read and Display the Text File ». It reads the text file (LottoNbrs.txt, where all random numbers are stored) and displays the Lotto history from a text file, generated from both clases : LottoMax and Lotto649. |
| 1. **Class: Lotto649** | This class represents the entire Lotto 649 window. |
| 1. Constructor public Lotto649() | Initializes an instance of the class and starts a stopwatch when an instance of the class is created. |
| 1. Void Method private void button1\_Click(object sender, EventArgs e) | Handles the click event of the "button1" control: « Generate 6 + 1 unique numbers out of 49 ». It generates random numbers for a Lotto 649, ensuring they do not repeat, and writes the results into a text file. |
| 1. Void Method private void button2\_Click(object sender, EventArgs e) | Handles the click event of the "button2" control : « Read and Display the Text File ». It reads the text file (LottoNbrs.txt, where all random numbers are stored) and displays the Lotto history from a text file, generated from both clases : LottoMax and Lotto649. |
| 1. Void Method private void button3\_Click(object sender, EventArgs e) | Handles the click event of the "button3" control : exit. It displays a confirmation message to exit the Lotto Max window, showing the time elapsed. |
| 1. **Class: MoneyExchange** | This class represents the entire MoneyExchange window. |
| 1. Void Method private void button1\_Click(object sender, EventArgs e) | Handles the click event of the "button1" control: «Convert from ». It reads the selected option of currency of origin, and the input value, calculates and displayed the converted values for all currencies. |
| 1. Void Method private void button2\_Click(object sender, EventArgs e) | Handles the click event of the "button2" control : « Read and Display the Text File ». It reads the text file (MoneyConversions.txt, where conversions are stored) and displays the history from a text file. |
| 1. Void Method private void Exit\_Click(object sender, EventArgs e) | Handles the click event of the "btnExit" button. It displays a confirmation message to exit the application and the time elapsed. |
| 1. **Class: temperatureConverter /temperatureConverter2** | This classes represents the temperatureConverter and temperatureConverter2 window. Convert from C to F temperature degrees and F to C respectively. |
| 1. Void Method private void button1\_Click(object sender, EventArgs e) | Handles the click event of the "button1" control: «Convert ». It reads the degrees to convert, calculates and display the converted value and message asociated. It uses the function to get the corresponding value of the message. |
| 1. Void Method private void button2\_Click(object sender, EventArgs e) | Handles the click event of the "button2" control : « Read File ». It reads the text file (TempConversions.txt, where conversions are stored) and displays the history from the text file. |
| 1. Void Method private void Exit\_Click(object sender, EventArgs e) | Handles the click event of the "btnExit" button. It displays a confirmation message to exit the application and the time elapsed. |
| 1. Void Method private void radioButton2\_CheckedChanged(object sender, EventArgs e) | Handles the event when the user select radio button2 (F to C). It opens a new form temperatureConverter2. With the same feature as the first one, but the conversion would be from Fahrenheit to Celsius. |
| 1. **Class: IP\_Validator** | This class represents the entire IP Validator windows form. |
| 1. Constructor : IP\_Validator() | Initializes an instance of the class, display the current date (according to the folloing function) and starts a stopwatch when an instance of the class is created. |
| 1. Void Method private void DisplayCurrentDate() | Handles the creation, formating and display for the current date. |
| 1. Void Method private void button2\_Click(object sender, EventArgs e) | Handles the click event of the "button2" control: «Reset». It clears the content of both, textBox1 and textBox2 by the use of .Clear(). |
| 1. Void Method private void button1\_Click(object sender, EventArgs e) | This is the most complet for IP Validator form, as it calls ValidateTextBoxes();function in which it calls ValidateTextBox1(); or ValidateTextBox2(); depending on the one that has any content by the use of an if statement, and try and catch to handle the error.  In ValidateTextBox1() and ValidateTextBox2() correspondingly, it contains an if statement to display whether a success message and write to binary function or a correction file with the parameters for a IPV4 or IPV6 correspondingly, by the use of the private bool IsValidIPv4(string ipAddressString) or private bool IsValidIPv6(string textBox2Value) accordingly that is explained in the next field. |
| 1. boolean Method private bool IsValidIPv4(string ipAddressString) or private bool IsValidIPv6(string textBox2Value) | Validate if the IPv4 or IPv6 is correct, by the use or regex expressions |
| 1. Void Method private void WriteToBinaryFile(string dataToWrite) | It is activated when the IsValidIPv6() or IsValidIPv4() is true accordingly.  It write the IP into a binary file. |
| 1. Void Method private void button3\_Click(object sender, EventArgs e) | Handles the click event of the "button3" control : « Read and display ». It reads the binary file (IPAddressData.dat, where validations are stored) and displays the history from the text file. |

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

* For me the easiest part is always the design part, as using Windows forms is very straight forward and you can visualize the results at the same time you are creating the design of the application.
* The hardest part was to make sure that the numbers does not repeat. For this particular challenge I ended up using an array of Boolean numbers to validate if the number was already selected in the random generation, using a do while.

bool[] numbersGenerated = new bool[50];

do

{

randomNumber = random.Next(1, 50);

} while (numbersGenerated[randomNumber]);

numbersGenerated[randomNumber] = true;

* Another challenge was to display the time spent in the application. In order to solve this, I used a Stopwatch Method, which starts after Initialize Component and stop when the exit button is selected in the private void button3\_Click(object sender, EventArgs e).
* The hardest part of section two was to handle all the options of origin currency and the final values. In order to solve this, I define different variables that represents the rate from all currencies to CAD, Then, a variable that defines the origin value (rate, previously defined) using a conditional validation. Then I was able to calculate each result currency value by using the following formula:

resultCurrency (each corresponding variable) = input value \* (currency (each corresponding variable) / origin (rate previously find by the use of conditional)).

double resultCAD = value \* (curCAD / origin);

double resultUSD = value \* (curUSD / origin);

double resultEUR = value \* (curEUR / origin);

double resultGBP = value \* (curGBP / origin);

double resultCOP = value \* (curCOP / origin);

double resultMXN = value \* (curMXN / origin);

* Another challenge was that at some point some buttons were not doing anything when clicking on them. For some reason, the event on the property section were disabled. The solution was to click on the events section (the one with a thunder icon) on the properties section. In the click section, there was nothing selected, so I click on it, and select the option nameOfTheButton\_Click.