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

|  |
| --- |
| Your name: Alena Belova  4/17/2023  Version: 1.0 |

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).**

* **Generated Numbers**

**Graphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generated**

**Graphical user interface, text, application

Description automatically generatedGraphical user interface, application

Description automatically generated**

**Graphical user interface, application

Description automatically generatedGraphical user interface, text, application

Description automatically generated**

* **Conversions**

**Graphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generated**

**Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generated**

**Graphical user interface, application

Description automatically generatedGraphical user interface, text, application, chat or text message

Description automatically generatedGraphical user interface, application, Word

Description automatically generated**

* **Simple Calculator**

**Graphical user interface

Description automatically generatedGraphical user interface, application

Description automatically generated**

**Graphical user interface, application

Description automatically generated**

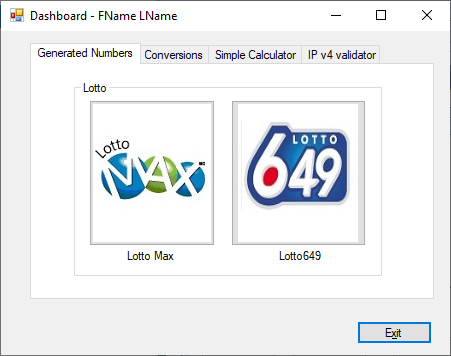
* **IP v4 Validator**

**Graphical user interface, application, Word

Description automatically generatedGraphical user interface

Description automatically generatedGraphical user interface, application, Teams

Description automatically generated**



Main feature is that the exit button will ask the user twice before closing the program, and it won’t just close everything but only the active window. That means the user leaves the application chosen but not

the dashboard.

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

* **Dashboard**

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;

namespace Final\_Project

{

public partial class frmDashboard : Form

{

public frmDashboard()

{

InitializeComponent();

}

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

{

// Create a new instance of LottoMaxForm

LottoMaxForm lottoMaxForm = new LottoMaxForm();

// Show the LottoMaxForm as a dialog

lottoMaxForm.ShowDialog();

}

private void btnLotto649\_Click(object sender, EventArgs e)

{

// Create a new instance of Lotto649Form

Lotto649Form lotto649Form = new Lotto649Form();

// Show the Lotto649Form as a dialog

lotto649Form.ShowDialog();

}

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

{

// Create a new instance of MoneyExForm

MoneyExForm moneyexForm = new MoneyExForm();

// Show the MoneyExForm as a dialog

moneyexForm.ShowDialog();

}

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

{

// Create a new instance of TempAppForm

TempAppForm tempappForm = new TempAppForm();

// Show the TempAppForm as a dialog

tempappForm.ShowDialog();

}

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

{

// Create a new instance of CalculatorForm

CalculatorForm calculatorForm = new CalculatorForm();

// Show the CalculatorForm as a dialog

calculatorForm.ShowDialog();

}

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

{

// Create a new instance of IPForm

IPForm ipForm = new IPForm();

// Show the IPForm as a dialog

ipForm.ShowDialog();

}

private void btn\_exit\_dashboard\_Click(object sender, EventArgs e)

{

// Display a custom message box asking if the user wants to close the form

DialogResult result = MessageBox.Show("Do you want to quit this aplication?", "Exit?", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (result == DialogResult.Yes)

{

// Close the form

this.Close();

}

else if (result == DialogResult.No)

{

// Do nothing and return to the form

}

}

}

}

* **Lotto Max**
* 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;
* namespace Final\_Project
* {
* public partial class LottoMaxForm : Form
* {
* public LottoMaxForm()
* {
* InitializeComponent();
* }
* private void btn\_Generate\_Max\_Click(object sender, EventArgs e)
* {
* // Create a new instance of Random
* Random random = new Random();
* // Create a list to store unique numbers
* List<int> uniqueNumbers = new List<int>();
* // Generate 7 unique random numbers between 1 and 50
* while (uniqueNumbers.Count < 7)
* {
* int randomNumber = random.Next(1, 51);
* if (!uniqueNumbers.Contains(randomNumber))
* {
* uniqueNumbers.Add(randomNumber);
* }
* }
* // Generate a unique bonus number between 1 and 50
* int bonusNumber;
* do
* {
* bonusNumber = random.Next(1, 51);
* } while (uniqueNumbers.Contains(bonusNumber));
* // Create a string to store the formatted content
* string content = "Max, " + DateTime.Now.ToString() + ", " + string.Join(", ", uniqueNumbers) + " Bonus: " + bonusNumber;
* // Write the content to the text file and overwrite existing content
* File.WriteAllText("LottoNbrs.txt", content);
* // Clear the existing text in the textBox\_Max
* textBox\_Max.Clear();
* // Display the 7 unique random numbers and the bonus number in the textBox\_Max
* foreach (int number in uniqueNumbers)
* {
* textBox\_Max.AppendText(number.ToString() + Environment.NewLine);
* }
* textBox\_Max.AppendText(bonusNumber.ToString());
* // Create a list to store unique digits
* List<int> digits = new List<int>();
* // Create a random number generator
* Random randomDigits = new Random();
* // Generate and store 7 unique digits
* while (digits.Count < 7)
* {
* int digit = randomDigits.Next(10); // Generate a random digit from 0 to 9
* // Add the digit to the list if it is not already present
* if (!digits.Contains(digit))
* {
* digits.Add(digit);
* }
* }
* // Update the text in the textBox\_Max\_digits with the generated digits
* textBox\_Max\_digits.Text = string.Join(" ", digits);
* }
* private void btn\_Readfile\_Max\_Click(object sender, EventArgs e)
* {
* try
* {
* // Set the file path
* string filePath = Path.Combine(AppDomain.CurrentDomain.BaseDirectory, "LottoNbrs.txt");
* // Check if the file exists
* if (File.Exists(filePath))
* {
* // Create FileStream and StreamReader objects to read the file
* using (FileStream fileStream = new FileStream(filePath, FileMode.Open, FileAccess.Read))
* {
* using (StreamReader reader = new StreamReader(fileStream))
* {
* // Read the content of the file
* string fileContent = reader.ReadToEnd();
* // Show the content in a MessageBox
* MessageBox.Show(fileContent, "Winning Numbers - Alena Belova", MessageBoxButtons.OK, MessageBoxIcon.Information);
* }
* }
* }
* else
* {
* MessageBox.Show("File not found.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
* }
* }
* catch (Exception ex)
* {
* MessageBox.Show($"An error occurred while reading the file: {ex.Message}", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
* }
* }
* private void btn\_Exit\_Max\_Click(object sender, EventArgs e)
* {
* // Display a custom message box asking if the user wants to close the form
* DialogResult result = MessageBox.Show("Do you want to quit this aplication?", "Exit?", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
* if (result == DialogResult.Yes)
* {
* // Close the form
* this.Close();
* }
* else if (result == DialogResult.No)
* {
* // Do nothing and return to the form
* }
* }
* }
* }
* **Lotto 649**
* 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;
* namespace Final\_Project
* {
* public partial class Lotto649Form : Form
* {
* public Lotto649Form()
* {
* InitializeComponent();
* }
* private void btn\_Generate\_649\_Click(object sender, EventArgs e)
* {
* // Create a new instance of Random
* Random random = new Random();
* // Create a list to store unique numbers
* List<int> uniqueNumbers = new List<int>();
* // Generate 6 unique random numbers between 1 and 49
* while (uniqueNumbers.Count < 6)
* {
* int randomNumber = random.Next(1, 50);
* if (!uniqueNumbers.Contains(randomNumber))
* {
* uniqueNumbers.Add(randomNumber);
* }
* }
* // Generate a unique bonus number between 1 and 49
* int bonusNumber;
* do
* {
* bonusNumber = random.Next(1, 50);
* } while (uniqueNumbers.Contains(bonusNumber));
* // Create a string to store the formatted content
* string content = "649, " + DateTime.Now.ToString() + ", " + string.Join(", ", uniqueNumbers) + " Bonus: " + bonusNumber;
* // Write the content to the text file and overwrite existing content
* File.WriteAllText("LottoNbrs.txt", content);
* // Clear the existing text in the textBox\_649
* textBox\_649.Clear();
* // Display the 6 unique random numbers and the bonus number in the textBox\_649
* foreach (int number in uniqueNumbers)
* {
* textBox\_649.AppendText(number.ToString() + Environment.NewLine);
* }
* textBox\_649.AppendText(bonusNumber.ToString());
* // Create a list to store unique digits
* List<int> digits = new List<int>();
* // Create a random number generator
* Random randomDigits = new Random();
* // Generate and store 7 unique digits
* while (digits.Count < 7)
* {
* int digit = randomDigits.Next(10); // Generate a random digit from 0 to 9
* // Add the digit to the list if it is not already present
* if (!digits.Contains(digit))
* {
* digits.Add(digit);
* }
* }
* // Update the text in the textBox\_649\_digits with the generated digits
* textBox\_649\_digits.Text = string.Join(" ", digits);
* }
* private void btn\_Readfile\_649\_Click(object sender, EventArgs e)
* {
* try
* {
* // Set the file path
* string filePath = Path.Combine(AppDomain.CurrentDomain.BaseDirectory, "LottoNbrs.txt");
* // Check if the file exists
* if (File.Exists(filePath))
* {
* // Create FileStream and StreamReader objects to read the file
* using (FileStream fileStream = new FileStream(filePath, FileMode.Open, FileAccess.Read))
* {
* using (StreamReader reader = new StreamReader(fileStream))
* {
* // Read the content of the file
* string fileContent = reader.ReadToEnd();
* // Show the content in a MessageBox
* MessageBox.Show(fileContent, "Winning Numbers - Alena Belova", MessageBoxButtons.OK, MessageBoxIcon.Information);
* }
* }
* }
* else
* {
* MessageBox.Show("File not found.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
* }
* }
* catch (Exception ex)
* {
* MessageBox.Show($"An error occurred while reading the file: {ex.Message}", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
* }
* }
* private void btn\_Exit\_649\_Click(object sender, EventArgs e)
* {
* // Display a custom message box asking if the user wants to close the form
* DialogResult result = MessageBox.Show("Do you want to quit this aplication?", "Exit?", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
* if (result == DialogResult.Yes)
* {
* // Close the form
* this.Close();
* }
* else if (result == DialogResult.No)
* {
* // Do nothing and return to the form
* }
* }
* }
* }
* **Money Exchange**

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;

namespace Final\_Project

{

public partial class Lotto649Form : Form

{

public Lotto649Form()

{

InitializeComponent();

}

private void btn\_Generate\_649\_Click(object sender, EventArgs e)

{

// Create a new instance of Random

Random random = new Random();

// Create a list to store unique numbers

List<int> uniqueNumbers = new List<int>();

// Generate 6 unique random numbers between 1 and 49

while (uniqueNumbers.Count < 6)

{

int randomNumber = random.Next(1, 50);

if (!uniqueNumbers.Contains(randomNumber))

{

uniqueNumbers.Add(randomNumber);

}

}

// Generate a unique bonus number between 1 and 49

int bonusNumber;

do

{

bonusNumber = random.Next(1, 50);

} while (uniqueNumbers.Contains(bonusNumber));

// Create a string to store the formatted content

string content = "649, " + DateTime.Now.ToString() + ", " + string.Join(", ", uniqueNumbers) + " Bonus: " + bonusNumber;

// Write the content to the text file and overwrite existing content

File.WriteAllText("LottoNbrs.txt", content);

// Clear the existing text in the textBox\_649

textBox\_649.Clear();

// Display the 6 unique random numbers and the bonus number in the textBox\_649

foreach (int number in uniqueNumbers)

{

textBox\_649.AppendText(number.ToString() + Environment.NewLine);

}

textBox\_649.AppendText(bonusNumber.ToString());

// Create a list to store unique digits

List<int> digits = new List<int>();

// Create a random number generator

Random randomDigits = new Random();

// Generate and store 7 unique digits

while (digits.Count < 7)

{

int digit = randomDigits.Next(10); // Generate a random digit from 0 to 9

// Add the digit to the list if it is not already present

if (!digits.Contains(digit))

{

digits.Add(digit);

}

}

// Update the text in the textBox\_649\_digits with the generated digits

textBox\_649\_digits.Text = string.Join(" ", digits);

}

private void btn\_Readfile\_649\_Click(object sender, EventArgs e)

{

try

{

// Set the file path

string filePath = Path.Combine(AppDomain.CurrentDomain.BaseDirectory, "LottoNbrs.txt");

// Check if the file exists

if (File.Exists(filePath))

{

// Create FileStream and StreamReader objects to read the file

using (FileStream fileStream = new FileStream(filePath, FileMode.Open, FileAccess.Read))

{

using (StreamReader reader = new StreamReader(fileStream))

{

// Read the content of the file

string fileContent = reader.ReadToEnd();

// Show the content in a MessageBox

MessageBox.Show(fileContent, "Winning Numbers - Alena Belova", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

}

else

{

MessageBox.Show("File not found.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

catch (Exception ex)

{

MessageBox.Show($"An error occurred while reading the file: {ex.Message}", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void btn\_Exit\_649\_Click(object sender, EventArgs e)

{

// Display a custom message box asking if the user wants to close the form

DialogResult result = MessageBox.Show("Do you want to quit this aplication?", "Exit?", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (result == DialogResult.Yes)

{

// Close the form

this.Close();

}

else if (result == DialogResult.No)

{

// Do nothing and return to the form

}

}

}

}

* **Temp conversion**

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;

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

namespace Final\_Project

{

public partial class TempAppForm : Form

{

public TempAppForm()

{

InitializeComponent();

}

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

{

labelFrom.Text = "C";

labelTo.Text = "F";

}

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

{

//c to f

if (radioButtonCtoF.Checked)

{

string content = "";

DateTime d = DateTime.Now;

double num1 = 0, num2 = 0;

num1 = Convert.ToDouble(textBoxFrom.Text);

num2 = num1 \* (9 / 5) + 32;

textBoxTo.Text = num2.ToString();

try

{

if (num1 == 100)

{

textBoxMessage.Text = "Water boils";

}

else if (num1 == 40)

{

textBoxMessage.Text = "Hot Bath";

}

else if (num1 == 37)

{

textBoxMessage.Text = "Body temperature";

}

else if (num1 == 30)

{

textBoxMessage.Text = "Beach weather";

}

else if (num1 == 21)

{

textBoxMessage.Text = "Room temperature";

}

else if (num1 == 10)

{

textBoxMessage.Text = "Cool Day";

}

else if (num1 == 0)

{

textBoxMessage.Text = "Freezing point of water";

}

else if (num1 == -18)

{

textBoxMessage.Text = "Very Cold Day";

}

else if (num1 == -40)

{

textBoxMessage.Text = "Extremely Cold Day \n(and the same number!)";

}

}

catch (Exception ex) { Console.WriteLine(ex.Message); }

content = textBoxFrom.Text + "C " + " = " + textBoxTo.Text + "F " + " " + d.ToString();

FileStream fs = new FileStream(@".\TempConv.txt", FileMode.Append);

StreamWriter obj = new StreamWriter(fs);

obj.WriteLine(content);

obj.Close();

fs.Close();

}

//from f to c

else if (radioButtonFtoC.Checked)

{

string content = "";

DateTime d = DateTime.Now;

double num1 = 0, num2 = 0;

num1 = Convert.ToDouble(textBoxFrom.Text);

num2 = (num1 - 32) \* 5 / 9;

textBoxTo.Text = num2.ToString();

try

{

if (num1 == 212)

{

textBoxMessage.Text = "Water boils";

}

else if (num1 == 104)

{

textBoxMessage.Text = "Hot Bath";

}

else if (num1 == 98.6)

{

textBoxMessage.Text = "Body temperature";

}

else if (num1 == 86)

{

textBoxMessage.Text = "Beach weather";

}

else if (num1 == 70)

{

textBoxMessage.Text = "Room temperature";

}

else if (num1 == 50)

{

textBoxMessage.Text = "Cool Day";

}

else if (num1 == 32)

{

textBoxMessage.Text = "Freezing point of water";

}

else if (num1 == 0)

{

textBoxMessage.Text = "Very Cold Day";

}

else if (num1 == -40)

{

textBoxMessage.Text = "Extremely Cold Day \n(and the same number!)";

}

}

catch (Exception ex) { Console.WriteLine(ex.Message); }

content = textBoxFrom.Text + "F " + " = " + textBoxTo.Text + "C " + " " + d.ToString();

FileStream fs = new FileStream(@".\TempConv.txt", FileMode.Append);

StreamWriter obj = new StreamWriter(fs);

obj.WriteLine(content);

obj.Close();

fs.Close();

}

}

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

{

labelFrom.Text = "F";

labelTo.Text = "C";

}

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

{

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

FileStream fs = null;

try

{

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

StreamReader textIn = new StreamReader(fs);

string textToPrint = textBoxFrom.Text + " " + labelFrom.Text + " = " + labelTo.Text + " " + textBoxTo.Text;

MessageBox.Show(textToPrint);

textIn.Close();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

fs.Close();

}

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

{

// Display a custom message box asking if the user wants to close the form

DialogResult result = MessageBox.Show("Do you want to quit this aplication?", "Exit?", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (result == DialogResult.Yes)

{

// Close the form

this.Close();

}

else if (result == DialogResult.No)

{

// Do nothing and return to the form

}

}

}

}

* **Simple Calculator**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

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

namespace Final\_Project

{

public partial class CalculatorForm : Form

{

static string dir = @".\";

String filePath = dir + "Calculator.txt";

double FirNum, Secnum, result;

string Operation;

public CalculatorForm()

{

InitializeComponent();

}

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

{

if (Directory.Exists(dir) == false)

{

Directory.CreateDirectory(dir);

}

}

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

{

textBoxOutput.Text += "1";

}

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

{

textBoxOutput.Text += "2";

}

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

{

textBoxOutput.Text += "3";

}

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

{

textBoxOutput.Text += "4";

}

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

{

textBoxOutput.Text += "5";

}

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

{

textBoxOutput.Text += "6";

}

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

{

textBoxOutput.Text += "7";

}

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

{

textBoxOutput.Text += "8";

}

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

{

textBoxOutput.Text += "9";

}

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

{

textBoxOutput.Text += "0";

}

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

{

int c = textBoxOutput.TextLength;

int flag = 0;

string text = textBoxOutput.Text;

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

{

if (text[i].ToString() == ".")

{

flag = 1;

break;

}

else

{

flag = 0;

}

}

if (flag == 0)

{

textBoxOutput.Text = textBoxOutput.Text + ".";

}

}

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

{

Operation = "+";

FirNum = double.Parse(textBoxOutput.Text);

textBoxOutput.Clear();

}

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

{

Operation = "-";

FirNum = double.Parse(textBoxOutput.Text);

textBoxOutput.Clear();

}

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

{

Operation = "\*";

FirNum = double.Parse(textBoxOutput.Text);

textBoxOutput.Clear();

}

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

{

Operation = "/";

FirNum = double.Parse(textBoxOutput.Text);

textBoxOutput.Clear();

}

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

{

Secnum = double.Parse(textBoxOutput.Text);

if (Operation == "+") { result = FirNum + Secnum; }

if (Operation == "-") { result = FirNum - Secnum; }

if (Operation == "\*") { result = FirNum \* Secnum; }

if (Operation == "/")

{

if (Secnum == 0)

{

MessageBox.Show("Cannot divide by zero!");

}

else

{

result = FirNum / Secnum;

}

}

textBoxOutput.Text = result + "";

FileStream fs = null;

try

{

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

StreamWriter textOut = new StreamWriter(fs);

if (Operation == "/" && Secnum == 0)

{

textOut.WriteLine(FirNum.ToString() + " " + Operation + " " + Secnum.ToString() + " = undefined");

}

else

{

textOut.WriteLine(FirNum.ToString() + " " + Operation + " " + Secnum.ToString() + " = " +

result.ToString() + " ");

}

textOut.Close();

}

catch (IOException ex)

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

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

}

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

{

textBoxOutput.Clear();

}

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

{

// Display a custom message box asking if the user wants to close the form

DialogResult result = MessageBox.Show("Do you want to quit this aplication?", "Exit?", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (result == DialogResult.Yes)

{

// Close the form

this.Close();

}

else if (result == DialogResult.No)

{

// Do nothing and return to the form

}

}

}

}

* **IP v4 Validator**

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 System.Text.RegularExpressions;

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

namespace Final\_Project

{

public partial class IPForm : Form

{

static string dir = @".\";

String pathBinary = dir + "IP4-Validator.dat";

public IPForm()

{

InitializeComponent();

}

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

{

DateTime d = DateTime.Now;

labelDate.Text = d.ToString();

if (Directory.Exists(dir) == false)

{

Directory.CreateDirectory(dir);

}

}

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

{

//validation

string ip = textBoxIP.Text;

Regex regex = new Regex(@"^(?:[0-9]{1,3}\.){3}[0-9]{1,3}$");

string[] byteNbrs = new string[4];

byteNbrs = ip.Split('.');

if (byteNbrs.Length != 4 || !regex.IsMatch(ip))

{

MessageBox.Show($"{ip}\nThe IP must have 4 bytes\ninteger must be between 0 to 255\n" +

$" Separated by a dot (255.255.255.255)\nReset to try again");

textBoxIP.ReadOnly = true;

return;

}

foreach (string bytes in byteNbrs)

{

string bytes1 = bytes.Trim();

int value = int.Parse(bytes1);

if (value < 0 || value > 255)

{

MessageBox.Show($"{ip}\nThe IP must have 4 bytes\ninteger must be between 0 to 255" +

$" Separated by a dot (255.255.255.255) Reset to try again");

textBoxIP.ReadOnly = true;

return;

}

}

textBoxIP.ReadOnly = true;

MessageBox.Show($"{ip} The IP address is correct");

//write in binary file

FileStream fs = null;

try

{

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

BinaryWriter binaryOut = new BinaryWriter(fs);

// write the fields into text file

binaryOut.Write(textBoxIP.Text.Trim());

// close the output stream for the text file

binaryOut.Close();

//fs.Close();

string textToPrint = "ip\n";

}

catch (IOException ex)

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

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

}

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

{

textBoxIP.Text = string.Empty;

textBoxIP.ReadOnly = false;

}

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

{

// Display a custom message box asking if the user wants to close the form

DialogResult result = MessageBox.Show("Do you want to quit this aplication?", "Exit?", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (result == DialogResult.Yes)

{

// Close the form

this.Close();

}

else if (result == DialogResult.No)

{

// Do nothing and return to the form

}

}

}

}

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

|  |  |
| --- | --- |
| **Class/Method Name** | **Description** |
| 1. Class1 | Enter a short description of the class or of the method ………………. |
| 1. Class2 | Enter a short description of the class or of the method ……………….. |
| 1. void Method(int total) | Enter a short description of the class or of the method ……….. |
|  |  |
| 1. Constructor() | Enter a short description of the class or of the method ………… |