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

|  |
| --- |
| Your name: Thai Duy Roan  7/12/2023  Version 1.0 |

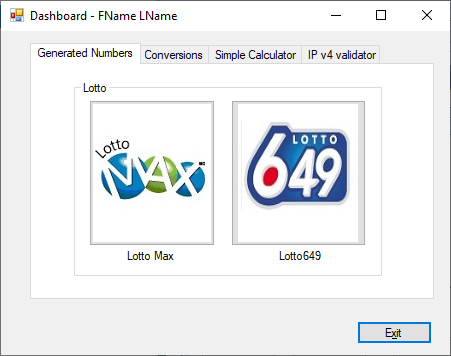
1. **Start by adding a short description of your project, and the languages (technologies) used:**
2. Language A, B, C

* I’m using C# to create the project

1. Used tool(s): Visual Studio version 2019 or 2022

* I used Visual Studio version 2022
* Inside Visual Studio, I chose

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



1. If you click on tab LottoMax, it will lead you to another page of the project. In that page, you can generate random number and it will be shown in the label in the right side. With the Read and Display button, you can see all the numbers you pressed previously stored in another file and displayed in a message box
2. As same as the tab LottoMax. If you click on tab Lotto649, it will generate randomly the numbers and store it in another file
3. If you click on the Exit button, it will ask you if you really want to exit that program and will close the program if you confirmed.
4. **Present the code of your application (forms).**

**Main Form**

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 projectForOOP

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void label2\_Click(object sender, EventArgs e)

{

}

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

{

if (MessageBox.Show("You want to Exit?", "Exit", MessageBoxButtons.OKCancel).ToString() == "OK")

{

this.Close();

}

}

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

{

frmMax obj = new frmMax();

obj.Show();

}

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

{

Form649 obj = new Form649();

obj.Show();

}

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

{

MoneyConvertor obj = new MoneyConvertor();

obj.Show();

}

private void tabPage2\_Click(object sender, EventArgs e)

{

}

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

{

TempConvertor obj = new TempConvertor();

obj.Show();

}

private void tabPage3\_Click(object sender, EventArgs e)

{

}

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

{

Simple\_Calculator obj = new Simple\_Calculator();

obj.Show();

}

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

{

IPValidator iPValidator = new IPValidator();

iPValidator.Show();

}

private void Form1\_Load(object sender, EventArgs e)

{

}

}

}

**LotoMAX**

string dir = @"..\Test\";

string path = @"..\Test\Test1.txt";

FileStream pr = null;

try

{

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

string dateTime = DateTime.Now.ToString("yyyy-MM-dd hh:mm:ss");

StreamWriter textOut = new StreamWriter(pr);

string lotto649 = "649";

textOut.Write(dateTime + "|");

textOut.WriteLine(textBox1.Text.Trim() + "|" + lotto649 + dateTime);

textOut.Close();

}

catch (FileNotFoundException)

{

MessageBox.Show(path + " unable to find.", "File Not Found");

}

catch (DirectoryNotFoundException)

{

MessageBox.Show(path + " unable to find.", "Directory Not Found");

}

catch (IOException ex)

{

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

}

finally

{

if (pr != null) pr.Close();

}

try

{

pr = new FileStream(path, FileMode.Open, FileAccess.Read);

StreamReader textIn = new StreamReader(pr);

string printText = "";

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

{

string row = textIn.ReadLine();

string[] columns = row.Split('|');

printText += columns[0] + ",\t" + columns[1] + "\n";

}

MessageBox.Show(printText);

textIn.Close();

}

catch (FileNotFoundException)

{

MessageBox.Show(path + " unable to find.", "File Not Found");

}

catch (DirectoryNotFoundException)

{

MessageBox.Show(path + " unable to find.", "Directory Not Found");

}

catch (IOException ex)

{

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

}

finally

{

if (pr != null) pr.Close();

}

**Loto649**

string dir = @"..\Test\";

string path = @"..\Test\Test2.txt";

FileStream pr = null;

try

{

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

string dateTime = DateTime.Now.ToString("yyyy-MM-dd hh:mm:ss");

StreamWriter textOut = new StreamWriter(pr);

string lotto649 = "649";

textOut.Write(dateTime + "|");

textOut.WriteLine(textBox1.Text.Trim() + "|" + lotto649 + dateTime);

textOut.Close();

}

catch (FileNotFoundException)

{

MessageBox.Show(path + " unable to find.", "File Not Found");

}

catch (DirectoryNotFoundException)

{

MessageBox.Show(path + " unable to find.", "Directory Not Found");

}

catch (IOException ex)

{

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

}

finally

{

if (pr != null) pr.Close();

}

try

{

pr = new FileStream(path, FileMode.Open, FileAccess.Read);

StreamReader textIn = new StreamReader(pr);

string printText = "";

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

{

string row = textIn.ReadLine();

string[] columns = row.Split('|');

printText += columns[0] + ",\t" + columns[1] + "\n";

}

MessageBox.Show(printText);

textIn.Close();

}

catch (FileNotFoundException)

{

MessageBox.Show(path + " unable to find.", "File Not Found");

}

catch (DirectoryNotFoundException)

{

MessageBox.Show(path + " unable to find.", "Directory Not Found");

}

catch (IOException ex)

{

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

}

finally

{

if (pr != null) pr.Close();

}

**MoneyExchange**

**using System;**

**using System.Collections.Generic;**

**using System.ComponentModel;**

**using System.Data;**

**using System.Drawing;**

**using System.IO;**

**using System.Linq;**

**using System.Text;**

**using System.Threading.Tasks;**

**using System.Windows.Forms;**

**namespace projectForOOP**

**{**

**public partial class MoneyConvertor : Form**

**{**

**public class MoneyExchange**

**{**

**private double num;**

**public double Num**

**{**

**get { return num; }**

**set { num = value; }**

**}**

**public MoneyExchange()**

**{**

**}**

**public MoneyExchange(double num)**

**{**

**this.num = num;**

**}**

**internal double cadToUSD(double num)**

**{**

**return num \* 0.7579602;**

**}**

**internal double cadToEUR(double num)**

**{**

**return num \* 0.68059079;**

**}**

**internal double cadToGBP(double num)**

**{**

**return num \* 0.58336193;**

**}**

**internal double cadToVND(double num)**

**{**

**return num \* 17870.657;**

**}**

**internal double cadToYEN(double num)**

**{**

**return num \* 104.87025;**

**}**

**internal double usdToCAD(double num)**

**{**

**return num \* 1.3194704;**

**}**

**internal double usdToEUR(double num)**

**{**

**return num \* 0.89811112;**

**}**

**internal double usdToGBP(double num)**

**{**

**return num \* 0.76980937;**

**}**

**internal double usdToVND(double num)**

**{**

**return num \* 23581.454;**

**}**

**internal double usdToYen(double num)**

**{**

**return num \* 138.38844;**

**}**

**internal double eurToCAD(double num)**

**{**

**return num \* 1.4692594;**

**}**

**internal double eurToUSD(double num)**

**{**

**return num \* 1.113592;**

**}**

**internal double eurToGBP(double num)**

**{**

**return num \* 0.85724961;**

**}**

**internal double eurToVND(double num)**

**{**

**return num \* 26262.402;**

**}**

**internal double eurToYEN(double num)**

**{**

**return num \* 154.12659;**

**}**

**internal double gbpToCAD(double num)**

**{**

**return num \* 1.7136804;**

**}**

**internal double gbpToUSD(double num)**

**{**

**return num \* 1.299135;**

**}**

**internal double gbpToEUR(double num)**

**{**

**return num \* 1.1665408;**

**}**

**internal double gbpToVND(double num)**

**{**

**return num \* 30667.487;**

**}**

**internal double gbpToYEN(double num)**

**{**

**return num \* 179.80221;**

**}**

**internal double vndToCAD(double num)**

**{**

**return num \* 0.000055936878;**

**}**

**internal double vndToUSD(double num)**

**{**

**return num \* 0.000042368297;**

**}**

**internal double vndToEUR(double num)**

**{**

**return num \* 0.00003808186;**

**}**

**internal double vndToGBP(double num)**

**{**

**return num \* 0.000032646175;**

**}**

**internal double vndToYEN(double num)**

**{**

**return num \* 0.0058698007;**

**}**

**}**

**public MoneyConvertor()**

**{**

**InitializeComponent();**

**}**

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

**{**

**Application.Exit();**

**}**

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

**{**

**}**

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

**{**

**}**

**static string dirPath = @"..\Test\";**

**string path = dirPath + "MoneyConversions.txt";**

**FileStream fs = null;**

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

**{**

**string chosenCurrency = "";**

**if (!string.IsNullOrWhiteSpace(textBox7.Text))**

**{**

**if (radioButton1.Checked)**

**{**

**chosenCurrency = radioButton1.Text;**

**MoneyExchange converNum = new MoneyExchange();**

**int i;**

**double[] num = new double[6];**

**double[] num2 = new double[6];**

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

**{**

**num[i] = Convert.ToDouble(textBox7.Text);**

**for (int j = 0; j < 6; j++)**

**{**

**switch (j)**

**{**

**case 0:**

**textBox1.Text = num[i].ToString();**

**break;**

**case 1:**

**{**

**num2[i] = converNum.cadToUSD(num[i]);**

**textBox2.Text = num2[i].ToString();**

**break;**

**}**

**case 2:**

**{**

**num2[i] = converNum.cadToEUR(num[i]);**

**textBox3.Text = num2[i].ToString();**

**break;**

**}**

**case 3:**

**{**

**num2[i] = converNum.cadToGBP(num[i]);**

**textBox4.Text = num2[i].ToString();**

**break;**

**}**

**case 4:**

**{**

**num2[i] = converNum.cadToVND(num[i]);**

**textBox5.Text = num2[i].ToString();**

**break;**

**}**

**case 5:**

**{**

**num2[i] = converNum.cadToYEN(num[i]);**

**textBox6.Text = num2[i].ToString();**

**break;**

**}**

**}**

**}**

**}**

**}**

**else if (radioButton2.Checked)**

**{**

**chosenCurrency = radioButton2.Text;**

**MoneyExchange converNum = new MoneyExchange();**

**int i;**

**double[] num = new double[6];**

**double[] num2 = new double[6];**

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

**{**

**num[i] = Convert.ToDouble(textBox7.Text);**

**for (int j = 0; j < 6; j++)**

**{**

**switch (j)**

**{**

**case 0:**

**textBox2.Text = num[i].ToString();**

**break;**

**case 1:**

**{**

**num2[i] = converNum.usdToCAD(num[i]);**

**textBox1.Text = num2[i].ToString();**

**break;**

**}**

**case 2:**

**{**

**num2[i] = converNum.usdToEUR(num[i]);**

**textBox3.Text = num2[i].ToString();**

**break;**

**}**

**case 3:**

**{**

**num2[i] = converNum.usdToGBP(num[i]);**

**textBox4.Text = num2[i].ToString();**

**break;**

**}**

**case 4:**

**{**

**num2[i] = converNum.usdToVND(num[i]);**

**textBox5.Text = num2[i].ToString();**

**break;**

**}**

**case 5:**

**{**

**num2[i] = converNum.usdToYen(num[i]);**

**textBox6.Text = num2[i].ToString();**

**break;**

**}**

**}**

**}**

**}**

**}**

**else if (radioButton3.Checked)**

**{**

**chosenCurrency = radioButton2.Text;**

**MoneyExchange converNum = new MoneyExchange();**

**int i;**

**double[] num = new double[6];**

**double[] num2 = new double[6];**

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

**{**

**num[i] = Convert.ToDouble(textBox7.Text);**

**for (int j = 0; j < 6; j++)**

**{**

**switch (j)**

**{**

**case 0:**

**textBox3.Text = num[i].ToString();**

**break;**

**case 1:**

**{**

**num2[i] = converNum.eurToCAD(num[i]);**

**textBox1.Text = num2[i].ToString();**

**break;**

**}**

**case 2:**

**{**

**num2[i] = converNum.eurToUSD(num[i]);**

**textBox2.Text = num2[i].ToString();**

**break;**

**}**

**case 3:**

**{**

**num2[i] = converNum.eurToGBP(num[i]);**

**textBox4.Text = num2[i].ToString();**

**break;**

**}**

**case 4:**

**{**

**num2[i] = converNum.eurToVND(num[i]);**

**textBox5.Text = num2[i].ToString();**

**break;**

**}**

**case 5:**

**{**

**num2[i] = converNum.eurToYEN(num[i]);**

**textBox6.Text = num2[i].ToString();**

**break;**

**}**

**}**

**}**

**}**

**}**

**else if (radioButton4.Checked)**

**{**

**chosenCurrency = radioButton3.Text;**

**MoneyExchange converNum = new MoneyExchange();**

**int i;**

**double[] num = new double[6];**

**double[] num2 = new double[6];**

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

**{**

**num[i] = Convert.ToDouble(textBox7.Text);**

**for (int j = 0; j < 6; j++)**

**{**

**switch (j)**

**{**

**case 0:**

**textBox4.Text = num[i].ToString();**

**break;**

**case 1:**

**{**

**num2[i] = converNum.gbpToCAD(num[i]);**

**textBox1.Text = num2[i].ToString();**

**break;**

**}**

**case 2:**

**{**

**num2[i] = converNum.gbpToUSD(num[i]);**

**textBox2.Text = num2[i].ToString();**

**break;**

**}**

**case 3:**

**{**

**num2[i] = converNum.gbpToEUR(num[i]);**

**textBox3.Text = num2[i].ToString();**

**break;**

**}**

**case 4:**

**{**

**num2[i] = converNum.gbpToVND(num[i]);**

**textBox5.Text = num2[i].ToString();**

**break;**

**}**

**case 5:**

**{**

**num2[i] = converNum.gbpToYEN(num[i]);**

**textBox6.Text = num2[i].ToString();**

**break;**

**}**

**}**

**}**

**}**

**}**

**else if (radioButton5.Checked)**

**{**

**chosenCurrency = radioButton4.Text;**

**MoneyExchange converNum = new MoneyExchange();**

**int i;**

**double[] num = new double[6];**

**double[] num2 = new double[6];**

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

**{**

**num[i] = Convert.ToDouble(textBox7.Text);**

**for (int j = 0; j < 6; j++)**

**{**

**switch (j)**

**{**

**case 0:**

**textBox5.Text = num[i].ToString();**

**break;**

**case 1:**

**{**

**num2[i] = converNum.vndToCAD(num[i]);**

**textBox1.Text = num2[i].ToString();**

**break;**

**}**

**case 2:**

**{**

**num2[i] = converNum.vndToUSD(num[i]);**

**textBox2.Text = num2[i].ToString();**

**break;**

**}**

**case 3:**

**{**

**num2[i] = converNum.vndToEUR(num[i]);**

**textBox3.Text = num2[i].ToString();**

**break;**

**}**

**case 4:**

**{**

**num2[i] = converNum.vndToGBP(num[i]);**

**textBox4.Text = num2[i].ToString();**

**break;**

**}**

**case 5:**

**{**

**num2[i] = converNum.vndToYEN(num[i]);**

**textBox6.Text = num2[i].ToString();**

**break;**

**}**

**}**

**}**

**}**

**}**

**else**

**{**

**MessageBox.Show("Please enter a valid amount.");**

**}**

**}**

**try**

**{**

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

**StreamWriter sw = new StreamWriter(fs);**

**sw.Write($"{DateTime.Now.ToString("yyyy/MM/dd hh:mm:ss tt")}\n{textBox7.Text} {chosenCurrency} = {textBox1.Text} CAD; {textBox2.Text} USD; {textBox3.Text} EUR; {textBox4.Text} GBP; {textBox5.Text} VND; {textBox6.Text} YEN\n");**

**sw.Close();**

**}**

**catch (FileNotFoundException)**

**{**

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

**}**

**catch (DirectoryNotFoundException)**

**{**

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

**}**

**catch (IOException ex)**

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

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

**}**

**private void MoneyConvertor\_Load(object sender, EventArgs e)**

**{**

**if (!Directory.Exists(dirPath))**

**Directory.CreateDirectory(dirPath);**

**}**

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

**{**

**try**

**{**

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

**StreamReader sr = new StreamReader(fs);**

**string textToPrint = "";**

**while (sr.Peek() != -1)**

**{**

**string row = sr.ReadLine().Trim();**

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

**}**

**MessageBox.Show(textToPrint);**

**sr.Close();**

**}**

**catch (FileNotFoundException)**

**{**

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

**}**

**catch (DirectoryNotFoundException)**

**{**

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

**}**

**catch (IOException ex)**

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

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

**}**

**}**

**}**

**Temperature**

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 projectForOOP

{

public partial class TempConvertor : Form

{

public TempConvertor()

{

InitializeComponent();

}

public class TempConvertors

{

private float num;

public float Num

{

get { return num; }

set { num = value; }

}

public TempConvertors()

{

}

public TempConvertors(float num)

{

this.num = num;

}

internal float CtoF(float num)

{

return (num \* 9 / 5) + 32;

}

internal float FtoC(float num)

{

return (num-32) \*5 /9;

}

}

static string dirPath = @"..\Test\";

string path = dirPath + "TempConversions.txt";

FileStream fs = null;

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

{

string chosenTemp = "";

try

{

TempConvertors tempConverted = new TempConvertors();

float num = Convert.ToSingle(textBox1.Text);

if (radioButton1.Checked)

{

chosenTemp = radioButton1.Text;

num = tempConverted.CtoF(num);

textBox2.Text = Convert.ToString(num);

textBox3.Text = GetTemperatureDescription(num, "F");

}

else if (radioButton2.Checked)

{

chosenTemp = radioButton2.Text;

num = tempConverted.FtoC(num);

textBox2.Text = Convert.ToString(num);

textBox3.Text = GetTemperatureDescription(Convert.ToSingle(textBox2.Text), "C");

}

}

catch (Exception ex2)

{

Console.WriteLine(ex2.Message);

}

try

{

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

StreamWriter sw = new StreamWriter(fs);

sw.Write($"{DateTime.Now.ToString("yyyy/MM/dd hh:mm:ss tt")}\n{textBox1.Text} {chosenTemp} = {textBox1.Text} {label3.Text}");

sw.Close();

}

catch (FileNotFoundException)

{

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

}

catch (DirectoryNotFoundException)

{

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

}

catch (IOException ex)

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

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

}

private string GetTemperatureDescription(float temperature, string unit)

{

switch (unit)

{

case "C":

{

textBox3.Text = GetCelsiusDescription(temperature);

break;

}

case "F":

{

textBox3.Text = GetFahrenheitDescription(temperature);

break;

}

default:

{

textBox3.Text = "Invalid unit.";

break;

}

}

return textBox3.Text;

}

private string GetCelsiusDescription(float temperature)

{

if (temperature > 40 && temperature <= 100)

{

textBox3.Text = "Water boils";

}

else if (temperature <= 40 && temperature > 37)

{

textBox3.Text = "Hot Bath";

}

else if (temperature <= 37 && temperature > 30)

{

textBox3.Text = "Body temperature";

}

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

{

textBox3.Text = "Beach Weather";

}

else if (temperature <= 21 && temperature > 10)

{

textBox3.Text = "Room Temperature";

}

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

{

textBox3.Text = "Cool Day";

}

else if (temperature <= 0 && temperature > -18)

{

textBox3.Text = "Freezing point of water";

}

else if (temperature <= -18 && temperature > -40)

{

textBox3.Text = "Very Cold Day";

}

else if (temperature == -40)

{

textBox3.Text = "Extremly Cold Day";

}

return textBox3.Text;

}

private string GetFahrenheitDescription(float temperature)

{

if (temperature > 212 && temperature <= 104)

{

textBox3.Text = "Water boils";

}

else if (temperature <= 104 || temperature > 98.6)

{

textBox3.Text = "Hot Bath";

}

else if (temperature <= 98.6 && temperature > 86)

{

textBox3.Text = "Body temperature";

}

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

{

textBox3.Text = "Beach Weather";

}

else if (temperature <= 70 && temperature > 50)

{

textBox3.Text = "Room Temperature";

}

else if (temperature <= 50 && temperature > 32)

{

textBox3.Text = "Cool Day";

}

else if (temperature <= 32 && temperature > 0)

{

textBox3.Text = "Freezing point of water";

}

else if (temperature <= 0 && temperature > -40)

{

textBox3.Text = "Very Cold Day";

}

else if (temperature == -40)

{

textBox3.Text = "Extremly Cold Day";

}

return textBox3.Text;

}

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

{

Application.Exit();

}

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

{

radioButton1.Checked = true;

if (!Directory.Exists(dirPath))

Directory.CreateDirectory(dirPath);

}

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

{

label2.Text = "C";

label3.Text = "F";

}

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

{

label2.Text = "F";

label3.Text = "C";

}

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

{

try

{

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

StreamReader sr = new StreamReader(fs);

string textToPrint = "";

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

{

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

textToPrint += row + "\n";

}

MessageBox.Show(textToPrint);

sr.Close();

}

catch (FileNotFoundException)

{

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

}

catch (DirectoryNotFoundException)

{

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

}

catch (IOException ex)

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

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

}

}

}

**IPv4 and IPv6**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading.Tasks;

using System.Windows.Forms;

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

namespace projectForOOP

{

public partial class IPValidator : Form

{

public IPValidator()

{

InitializeComponent();

}

Regex obj;

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

{

Application.Exit();

}

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

{

textBox1.Text = "";

textBox2.Text = "";

}

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

{

}

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

{

label1.Text = DateTime.Now.ToString();

}

static string dirPath = @"..\Test\";

string path = dirPath + "IPv4.txt";

FileStream fs = null;

static string dirPath2 = @"..\Test\";

string path2 = dirPath2 + "IPv6.txt";

FileStream fs2 = null;

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

{

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

{

obj = new Regex(@"^(\d{1,3}\.){3}\d{1,3}$");

if (obj.IsMatch(textBox1.Text) == true)

{

MessageBox.Show("Correct IP v4 form");

}

else

{

MessageBox.Show("Incorrect IP v4 form");

}

}

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

{

obj = new Regex(@"^(?:[\w]{1,4}:){6}(:|[\w]{1,4})$");

if (obj.IsMatch(textBox2.Text) == true)

{

MessageBox.Show("Correct IP v6 form");

}

else

{

MessageBox.Show("Incorrect IP v6 form");

}

}

try

{

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

StreamWriter sw = new StreamWriter(fs);

sw.Write($"{DateTime.Now.ToString("yyyy/MM/dd hh:mm:ss tt")}\n{textBox1.Text}");

sw.Close();

}

catch (FileNotFoundException)

{

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

}

catch (DirectoryNotFoundException)

{

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

}

catch (IOException ex)

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

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

try

{

fs2 = new FileStream(path2, FileMode.Append, FileAccess.Write);

StreamWriter sw2 = new StreamWriter(fs2);

sw2.Write($"{DateTime.Now.ToString("yyyy/MM/dd hh:mm:ss tt")}\n{textBox2.Text}");

sw2.Close();

}

catch (FileNotFoundException)

{

MessageBox.Show(path2 + " not found.", "File Not Found");

}

catch (DirectoryNotFoundException)

{

MessageBox.Show(dirPath2 + " not found.", "Directory Not Found");

}

catch (IOException ex)

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

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

}

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

{

try

{

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

StreamReader sr = new StreamReader(fs);

string textToPrint = "";

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

{

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

textToPrint += row + "\n";

}

MessageBox.Show(textToPrint);

sr.Close();

}

catch (FileNotFoundException)

{

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

}

catch (DirectoryNotFoundException)

{

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

}

catch (IOException ex)

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

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

try

{

fs2 = new FileStream(path2, FileMode.Open, FileAccess.Read);

StreamReader sr1 = new StreamReader(fs2);

string textToPrint1 = "";

while (sr1.Peek() != -1)

{

string row1 = sr1.ReadLine().Trim();

textToPrint1 += row1 + "\n";

}

MessageBox.Show(textToPrint1);

sr1.Close();

}

catch (FileNotFoundException)

{

MessageBox.Show(path2 + " not found.", "File Not Found");

}

catch (DirectoryNotFoundException)

{

MessageBox.Show(dirPath2 + " not found.", "Directory Not Found");

}

catch (IOException ex)

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

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

}

}

}

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

|  |  |
| --- | --- |
| **Class/Method Name** | **Description** |
| 1. Class MoneyExchange | Class MoneyExchange is used to contain all the formula, formula to convert from CAD dollars to USD dollars and so on. In main, I called out the class to do the calculate |
| 1. Class SimpleCalculator | Class SimpleCalculator is used to contain function too calculate entered numbers. Inside SimpleCalculator, when 2 values is entered, it will send into SimpleCalculator, send the function got called out into and execute |
| 1. Random random = new Random(); | In FormMax and Form649 is used to generate random number from 1 to 50 and 1 to 49 |
| 1. Class TempConvertors | Same with other classes, contains 2 functions, one is to convert from C degrees to F degrees and vice visa |

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

I had difficulty in creating a Read and Display button but after many hours, I was able to fix it.

For SimpleCalculator, I can’t sum, minus, multiply or devide multiple numbers.

For me, the easiest part is the MoneyExchange, I created the .cs without any problems.