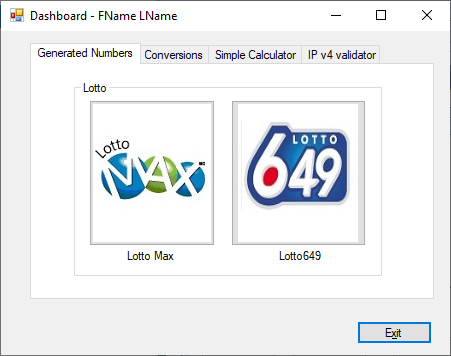
|  |
| --- |
| College LaSalle |
| Project - Oriented Object Programming User and Technical Manual |
|  |
| Presented to: Mihai Maftei. |

|  |
| --- |
| Your name: Jose Antonio Bellorin Julio  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): Microsoft visual studio
4. **Present the print screens of yours forms, and have a detailed description of the functionalities (step by step).**



1. If you click on tab Generated Number you going to see the windows above
2. If you click on the button Lotto Max the next window you will see

A screenshot of a computer

Description automatically generated

1. …
2. If you click on th5- when you click on generating 7+1 unique number out 49. The system going to generated the number in the textbox

A screenshot of a computer

Description automatically generated

If you click on the button

A white rectangular object with black text

Description automatically generated

You will see

A screenshot of a computer

Description automatically generated

If you click Exit button,

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

If you click on the button Lotto 649 you will see

A screenshot of a computer

Description automatically generated

**If you click generate 6+1 unique number out of 49**

A screenshot of a computer

Description automatically generated

**If you click Read and Display the Text File.**

**A screenshot of a computer

Description automatically generated**

**If you click in the button exit.**

**A screenshot of a computer

Description automatically generated**

**If you click in the tab Conversions you will see this window**

A screenshot of a computer

Description automatically generated

**If you click in the tab Money Exchanged**

A screenshot of a computer

Description automatically generated

**You must select the type of change that you need to convert and after put the amount in the text box**

A screenshot of a computer

Description automatically generated

**And you will see all the type of change saved pressing the button convert from .**

**And then you can read all your conversions stored in the app**

A screenshot of a computer

Description automatically generated

**If you want to close the app you can press the button exit and you will see the next window**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**In this windows if it selected Temperature Convert you will see**

A screenshot of a computer

Description automatically generated

**You can select what kind of conversion you want to do.**

**If you click the option button from C to F you going to convert Celsius grades to Fahrenheit Grades.**

**If you click the option button from F to C you going to convert Fahrenheit to Celsius grades.**

**In the text box C or F depending of your choice you going to put your value to convert.**

**Depending of the temperature in the box text Message you will see a message telling you how about it the Temperature.**

**After you can click the button convert to se the conversion.**

**In the button Read File you can read every conversion you made.**

**The button exit will close the windows showing you how many time you were in the app.**

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

Dash board

… public partial class Form1 : Form

{

DateTime startTime;

public Form1()

{

InitializeComponent();

startTime = DateTime.Now;

}

private void tabPage1\_Click(object sender, EventArgs e)

{

}

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

{

LottoMax obj = new LottoMax();

obj.Show(this);

}

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

{

Lotto69 lotto69 = new Lotto69();

lotto69.Show();

}

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

{

}

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

{

DateTime closeTime = DateTime.Now;

TimeSpan loginTime = closeTime - startTime;

int minutes = loginTime.Minutes;

int seconds = loginTime.Seconds;

if (MessageBox.Show($"Do you wanna exit? You have been here for {minutes} minutes and {seconds} seconds ", " Exit", MessageBoxButtons.YesNo, MessageBoxIcon.Question).ToString().Equals("Yes"))

{

this.Close();

}

}

}

}

**LotoMAX**

public partial class LottoMax : Form

{

//name:jose bellorin

//description:version section 1 Lotto Max of final project

//date:23/11/03

DateTime startTime;

static string dir = @"./Files/";

string path = dir + "LottoNumbers.txt";

Data data = new Data();

static string[] numbers = new string[8];

static FileStream fs = null;

public LottoMax()

{

InitializeComponent();

}

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

{

startTime=DateTime.Now;

}

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

{

}

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

{

string dir = @"C:\Files\";

if (!Directory.Exists(dir))

Directory.CreateDirectory(dir);

//create a instance of the clase Data to generate de random number

Data data = new Data();

data.GenerateMaxNumber();

textBox1.Text = data.NumberTodisplay;

try

{

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

StreamWriter objW = new StreamWriter(fs);

objW.WriteLine(data.NumberToPrint);

objW.Close();

}

catch (FileNotFoundException)

{

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

}

catch (DirectoryNotFoundException)

{

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

}

catch (IOException ex)

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

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

}

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

{

try

{

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

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

// create the object for the input stream for a text file

StreamReader textIn = new StreamReader(fs);

string textToPrint = "Numbers \n";

// read the data from the file and store it in the list

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

{

string row = textIn.ReadLine();

textToPrint += row+"\n";

}

MessageBox.Show(textToPrint);

// close the input stream for the text file

textIn.Close();

}

catch (FileNotFoundException)

{

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

}

catch (DirectoryNotFoundException)

{

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

}

catch (IOException ex)

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

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

}

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

{

DateTime closeTime = DateTime.Now;

TimeSpan loginTime = closeTime - startTime;

int minutes = loginTime.Minutes;

int seconds = loginTime.Seconds;

if (MessageBox.Show($"Do you wanna exit? You have been here for {minutes} minutes and {seconds} seconds ", " Exit", MessageBoxButtons.YesNo, MessageBoxIcon.Question).ToString().Equals("Yes"))

{

// MessageBox.Show("you decided to quit de app", $"{DateTime.Now.ToShortDateString()} at {DateTime.Now.ToShortTimeString()}");

this.Close();

}

}

}

Loto649

public partial class Lotto69 : Form

{

static string dir = @"./Files/";

string path = dir + "LottoNumbers.txt";

FileStream fs = null;

DateTime startTime;

public Lotto69()

{

InitializeComponent();

}

private void Lotto69\_Load(object sender, EventArgs e)

{

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

{

Directory.CreateDirectory(dir);

}

startTime = DateTime.Now;

}

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

{

string dir = @"C:\Files\";

if (!Directory.Exists(dir))

Directory.CreateDirectory(dir);

//create a instance of the clase Data to generate de random number

Data data = new Data();

data.GenerateLottoNumber();

textBox1.Text = data.NumberTodisplay.ToString();

try

{

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

StreamWriter objW = new StreamWriter(fs);

objW.WriteLine(data.NumberToPrint);

objW.Close();

}

catch (FileNotFoundException)

{

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

}

catch (DirectoryNotFoundException)

{

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

}

catch (IOException ex)

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

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

}

private void maxNumber()

{

}

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

{

try

{

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

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

// create the object for the input stream for a text file

StreamReader textIn = new StreamReader(fs);

string textToPrint = "Numbers \n";

// read the data from the file and store it in the list

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

{

string row = textIn.ReadLine();

textToPrint += row + "\n";

}

MessageBox.Show(textToPrint);

// close the input stream for the text file

textIn.Close();

}

catch (FileNotFoundException)

{

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

}

catch (DirectoryNotFoundException)

{

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

}

catch (IOException ex)

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

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

}

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

{

}

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

{

}

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

{

}

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

{

DateTime closeTime= DateTime.Now;

TimeSpan loginTime = closeTime - startTime;

int minutes=loginTime.Minutes;

int seconds=loginTime.Seconds;

if (MessageBox.Show($"Do you wanna exit? You have been here for {minutes} minutes and {seconds} seconds " , " Exit", MessageBoxButtons.YesNo, MessageBoxIcon.Question).ToString().Equals("Yes"))

{

// MessageBox.Show("you decided to quit de app", $"{DateTime.Now.ToShortDateString()} at {DateTime.Now.ToShortTimeString()}");

this.Close();

}

}

}

}

Money Conversion

public partial class Exchange : Form

{

string from="";

static string dirFile = @".\Files\";

string path = dirFile + "MoneyConversions.txt";

FileStream fs = null;

DateTime iniTime;

public Exchange()

{

InitializeComponent();

iniTime=DateTime.Now;

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

{

Directory.CreateDirectory(dirFile);

}

}

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

{

}

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

{

string pattern = @"^[0-9]+$";

if (textValue.Text=="")

{

MessageBox.Show("sorry, the field cannot be empty");

textValue.Focus();

}

else

{

if (Regex.IsMatch(textValue.Text,pattern))

{

MakeCalculations();

SaveDataTextFile();

SaveDataBinaryFile();

}

else

{

MessageBox.Show("sorry, the value entered is invalid. Enter a valid number");

}

}

}

//functions

private void ConvertFromCad(float cad)

{

Calcs calcs = new Calcs(cad);

textCanada.Text = calcs.FromCad().Cad + " CAD";

textUsa.Text = calcs.FromCad().Usd.ToString() + " USD";

textUK.Text = calcs.FromCad().Gpb.ToString() + " GPB";

textEuro.Text = calcs.FromCad().Euro.ToString() + " Eur";

textJapan.Text = calcs.FromCad().Jpy.ToString() + " JPY";

textChina.Text = calcs.FromCad().Cny.ToString() + " CNY";

}

private void ConvertFromUsd(float usd)

{

Calcs calcs = new Calcs(usd);

textCanada.Text = calcs.FromUSD().Cad + " CAD";

textUsa.Text = calcs.FromUSD().Usd.ToString() + " USD";

textUK.Text = calcs.FromUSD().Gpb.ToString() + " GPB";

textEuro.Text = calcs.FromUSD().Euro.ToString() + " Eur";

textJapan.Text = calcs.FromUSD().Jpy.ToString() + " JPY";

textChina.Text = calcs.FromUSD().Cny.ToString() + " CNY";

}

private void ConvertFromGpb(float gpb)

{

Calcs calcs = new Calcs(gpb);

textCanada.Text = calcs.FromGpb().Cad + " CAD";

textUsa.Text = calcs.FromGpb().Usd.ToString() + " USD";

textUK.Text = calcs.FromGpb().Gpb.ToString() + " GPB";

textEuro.Text = calcs.FromGpb().Euro.ToString() + " Eur";

textJapan.Text = calcs.FromGpb().Jpy.ToString() + " JPY";

textChina.Text = calcs.FromGpb().Cny.ToString() + " CNY";

}

private void ConvertFromEur(float eur)

{

Calcs calcs = new Calcs(eur);

textCanada.Text = calcs.FromEur().Cad + " CAD";

textUsa.Text = calcs.FromEur().Usd.ToString() + " USD";

textUK.Text = calcs.FromEur().Gpb.ToString() + " GPB";

textEuro.Text = calcs.FromEur().Euro.ToString() + " Eur";

textJapan.Text = calcs.FromEur().Jpy.ToString() + " JPY";

textChina.Text = calcs.FromEur().Cny.ToString() + " CNY";

}

private void ConvertFromJpy(float eur)

{

Calcs calcs = new Calcs(eur);

textCanada.Text = calcs.FromJpy().Cad + " CAD";

textUsa.Text = calcs.FromJpy().Usd.ToString() + " USD";

textUK.Text = calcs.FromJpy().Gpb.ToString() + " GPB";

textEuro.Text = calcs.FromJpy().Euro.ToString() + " Eur";

textJapan.Text = calcs.FromJpy().Jpy.ToString() + " JPY";

textChina.Text = calcs.FromJpy().Cny.ToString() + " CNY";

}

private void MakeCalculations()

{

float value = 0;

if (rBCanada.Checked)

{

from = rBCanada.Text;

try

{

value = float.Parse(textValue.Text.Trim());

ConvertFromCad(value);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

else if (rBUsa.Checked)

{

from = rBUsa.Text;

try

{

value = float.Parse(textValue.Text.Trim());

ConvertFromUsd(value);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

else if (rBGbp.Checked)

{

from = rBGbp.Text;

try

{

value = float.Parse(textValue.Text.Trim());

ConvertFromGpb(value);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

else if (rBEuro.Checked)

{

from = rBEuro.Text;

try

{

value = float.Parse(textValue.Text.Trim());

ConvertFromEur(value);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

else if (rBJapan.Checked)

{

from = rBJapan.Text;

try

{

value = float.Parse(textValue.Text.Trim());

ConvertFromJpy(value);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

}

private void SaveDataTextFile()

{

Currencies currencies = new Currencies();

string cad, usd, eur, gpb, jpy, cny, textToSave,date,formattedDate;

date= DateTime.Now.ToString()+"\n" ;

textToSave = textValue.Text + from + " = " + textCanada.Text + "; " + textUsa.Text + "; " + textEuro.Text + "; " + textUK.Text + "; " + textJapan.Text + ";" + textChina.Text + ";\n";

formattedDate = date.Replace('-', '/');

try

{

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

StreamWriter objSw = new StreamWriter (fs);

if (textValue.Text!="")

{

objSw.Write (formattedDate);

objSw.Write( textToSave+"\n\n");

objSw.Close();

}

}

catch (FileNotFoundException)

{

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

}

catch (DirectoryNotFoundException)

{

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

}

catch (IOException ex)

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

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

}

private void SaveDataBinaryFile()

{

Currencies currencies = new Currencies();

string textToSave, date, formattedDate;

path = dirFile + "MoneyConversionsB.txt";

date = DateTime.Now.ToString() ;

textToSave = textValue.Text + from + " = " + textCanada.Text + "; " + textUsa.Text + "; " + textEuro.Text + "; " + textUK.Text + "; " + textJapan.Text + ";" + textChina.Text ;

formattedDate = date.Replace('-', '/');

try

{

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

BinaryWriter objWb = new BinaryWriter(fs);

if (textValue.Text != "")

{

objWb.Write(formattedDate);

objWb.Write(textToSave);

objWb.Close();

}

}

catch (FileNotFoundException)

{

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

}

catch (DirectoryNotFoundException)

{

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

}

catch (IOException ex)

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

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

}

private void ReadTxtField()

{

string path = dirFile + "MoneyConversions.txt";

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

string row = "";

StreamReader objR = new StreamReader(fs);

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

{

row+=objR.ReadLine()+"\n";

}

MessageBox.Show(row);

objR.Close();

}

// MessageBox.Show(textToSave);

//events radiobuttons

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

{

textValue.Focus();

}

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

{

textValue.Focus();

}

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

{

textValue.Focus();

}

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

{

textValue.Focus();

}

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

{

textValue.Focus();

}

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

{

TimeSpan currenttime=DateTime.Now- iniTime;

int minutes=currenttime.Minutes;

int seconds=currenttime.Seconds;

if (MessageBox.Show($"Do you wanna quit? you decided to quit de app? you have been here {minutes} sec {seconds}", " Exit", MessageBoxButtons.YesNo,MessageBoxIcon.Question).ToString().Equals("Yes"))

{

Application.Exit();

}

}

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

{

ReadTxtField();

}

}

}

public class Calcs

{

//private varibles

float value,cad, usd, eur, gpb, jpy, cny;

//constructor with 1 parameter

public Calcs(float value) {

this.Value = value;

}

public float Cad { get => cad; set => cad = value; }

public float Usd { get => usd; set => usd = value; }

public float Euro { get => eur; set => eur = value; }

public float Gpb { get => gpb; set => gpb = value; }

public float Jpy { get => jpy; set => jpy = value; }

public float Cny { get => cny; set => cny = value; }

public float Value { get => value; set => this.value = value; }

public Currencies FromCad()

{

Currencies currencies = new Currencies();

this.Cad = Value;

currencies.Cad = Value;

currencies.Usd = Cad\* 0.73f;

currencies.Gpb = Cad \* 0.59f;

currencies.Euro = Cad \* 0.67f;

currencies.Jpy = Cad \* 110.54f;

currencies.Cny = Cad \* 5.30f;

return currencies;

}

public Currencies FromUSD()

{

Currencies currencies = new Currencies();

this.Usd = Value;

currencies.Usd = Usd;

currencies.Cad = Usd \* 1.37f;

currencies.Gpb = Usd \* 0.81f;

currencies.Euro = Usd \* 0.92f;

currencies.Jpy = Usd \* 151.29f;

currencies.Cny = Usd \* 7.26f;

return currencies;

}

public Currencies FromGpb()

{

Currencies currencies = new Currencies();

currencies.Cad = Value \* 1.70f;

currencies.Usd = Value \* 1.24f;

currencies.Gpb = Value;

currencies.Euro = Value \* 1.14f;

currencies.Jpy = Value \* 187.41f;

currencies.Cny = Value \* 8.99f;

return currencies;

}

public Currencies FromEur()

{

Currencies currencies = new Currencies();

currencies.Cad = Value \* 1.48f;

currencies.Usd = Value \* 1.08f;

currencies.Gpb = Value\* 0.87f;

currencies.Euro = Value;

currencies.Jpy = Value \* 163.90f;

currencies.Cny = Value \* 7.87f;

return currencies;

}

public Currencies FromJpy()

{

Currencies currencies = new Currencies();

currencies.Cad = Value \* 0.0091f;

currencies.Usd = Value \* 0.0066f;

currencies.Gpb = Value \* 0.0053f;

currencies.Euro = Value\* 0.0061f;

currencies.Jpy = Value;

currencies.Cny = Value \* 0.048f;

return currencies;

}

}

}

public class Currencies

{

float cad, usd, eur, gpb, jpy, cny;

public Currencies() {

}

public float Cad { get => cad; set => cad = value; }

public float Usd { get => usd; set => usd = value; }

public float Euro { get => eur; set => eur = value; }

public float Gpb { get => gpb; set => gpb = value; }

public float Jpy { get => jpy; set => jpy = value; }

public float Cny { get => cny; set => cny = value; }

}

}

Temperature

public partial class The\_Temperature\_application : Form

{

//variables

DateTime iniTime;

static string dirFile = @".\Files\";

string path = dirFile + "TempConversions.txt";

FileStream fs = null;

public The\_Temperature\_application()

{

if (!Directory.Exists(dirFile))

Directory.CreateDirectory(path);

InitializeComponent();

iniTime=DateTime.Now;

}

private void The\_Temperature\_application\_Load(object sender, EventArgs e)

{

}

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

{

}

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

{

TimeSpan currenttime = DateTime.Now - iniTime;

int minutes = currenttime.Minutes;

int seconds = currenttime.Seconds;

if (MessageBox.Show($"Do you wanna quit? you decided to quit de app? you have been here {minutes} sec {seconds}", " Exit", MessageBoxButtons.YesNo, MessageBoxIcon.Question).ToString().Equals("Yes"))

{

System.Windows.Forms.Application.Exit();

}

}

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

{

bool empty = true;

try

{

if (!validateData(textBox1.Text.Trim()) && textBox1.Text!="")

{

MessageBox.Show("Sorry, the field must contain only numbers for C or F.");

empty = false;

// return;

}

if (radioBFromC.Checked && textBox1.Text.Trim() != "")

{

float celsius = float.Parse(textBox1.Text);

Temperature obj = new CtoF();

textBox2.Text = obj.ConvertTemp(celsius).ToString("0.00");

setMessasge();

ChangeCelsiusColor();

SaveDataTextFile();

SaveDataBinaryFile();

empty = false;

}

if (radioBFromF.Checked && textBox1.Text.Trim() != "")

{

int Fahrenheit = int.Parse(textBox1.Text);

Temperature obj2 = new FtoC();

textBox2.Text = obj2.ConvertTemp(Fahrenheit).ToString("0.00");

setMessasge();

ChangeFarenheitColor();

SaveDataTextFile();

empty = false;

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

if (empty)

{

MessageBox.Show("Sorry, the field C or F cannot be empty");

}

}

private void setMessasge()

{

string temp="",message="";

temp=textBox1.Text;

if (textBox1.Text!="")

{

switch (temp)

{

case "100":

case "212":

message = "Water boils";

break;

case "104":

case "40":

message = "Hot Bath";

break;

case "37":

case "98.6":

message = "Body temperature";

break;

case "30":

case "86":

message = "Beach weather";

break;

case "21":

case "70":

message = "Room temperature";

break;

case "-18":

case "0":

message = "Very Cold Day";

break;

case "-40":

message = "Extremely Cold Day";

break;

}

textMessage.Text = message;

}

}

private void SaveDataTextFile()

{

// Currencies currencies = new Currencies();

path = dirFile + "TempConversions.txt";

string textToSave="", date, formattedDate,farenheit,celsius,time,message;

time = DateTime.Now.ToShortTimeString() + "\t\t";

date = DateTime.Now.ToShortDateString();

formattedDate = date.Replace('-', '/') + "\t";

//100 C = 212 F, 2023/7/22 01:01:33 PM Water Boils

// 104 F = 40 C, 2023 / 7 / 22 10:07:03 PM Hot Bath

try

{

celsius = textBox1.Text.Trim() + "C =";

farenheit = textBox2.Text.Trim() + "F," + "\t";

message = textMessage.Text.Trim() + "\t";

if (radioBFromC.Checked)

{

textToSave = $"{celsius} {farenheit} {formattedDate} {time} {message} \n";

}

else

{

celsius = textBox2.Text.Trim();

farenheit = textBox1.Text.Trim();

textToSave = $"{farenheit}F = {celsius}C , {formattedDate} {time} {message} \n";

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

try

{

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

StreamWriter objSw = new StreamWriter(fs);

if (textBox1.Text != "")

{

objSw.Write(textToSave + "\n\n");

objSw.Close();

}

}

catch (FileNotFoundException)

{

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

}

catch (DirectoryNotFoundException)

{

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

}

catch (IOException ex)

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

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

}

private void SaveDataBinaryFile()

{

path = dirFile + "TempConversionsB.txt.";

string textToSave, date, formattedDate, farenheit, celsius, time, message;

celsius = textBox1.Text.Trim();

farenheit = textBox2.Text.Trim();

message = textMessage.Text.Trim() + "\t";

time = DateTime.Now.ToShortTimeString() + "\t\t";

date = DateTime.Now.ToShortDateString();

formattedDate = date.Replace('-', '/');

try

{

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

BinaryWriter objWb = new BinaryWriter(fs);

if (textBox1.Text != "")

{

if (radioBFromC.Checked)

{

textToSave = $"{celsius} C = {farenheit} F , {formattedDate} {time} {message} \n";

}

else

{

celsius = textBox2.Text.Trim();

farenheit = textBox1.Text.Trim();

textToSave = $"{farenheit} F = {celsius} C , {formattedDate} {time} {message} \n";

}

objWb.Write(textToSave);

objWb.Close();

}

}

catch (FileNotFoundException)

{

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

}

catch (DirectoryNotFoundException)

{

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

}

catch (IOException ex)

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

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

}

private void ChangeCelsiusColor()

{

float value=float.Parse(textBox1.Text.Trim());

if (radioBFromC.Checked)

{

if (value>=40 )

{

textBox1.ForeColor=Color.Red;

}

if (value>+30 && value <= 39)

{

textBox1.ForeColor = Color.Orange;

}

if (value > 10 && value <= 21)

{

textBox1.ForeColor = Color.Green;

}

if (value >= 0 && value <= 10)

{

textBox1.ForeColor = Color.Blue;

}

if (value < 0 )

{

textBox1.ForeColor = Color.Black;

}

}

}

private bool validateData(string data)

{

string patter = @"^-?[0-9]+(\.[0-9]+)?$";

Regex regex = new Regex(patter);

if (regex.IsMatch(data))

{

return true;

}

return false;

}

private void ChangeFarenheitColor()

{

float value = float.Parse(textBox1.Text.Trim());

if (radioBFromF.Checked)

{

if (value >= 104)

{

textBox1.ForeColor = Color.Red;

}

if (value > 85 && value < 99)

{

textBox1.ForeColor = Color.Orange;

}

if (value >= 31 && value <= 85)

{

textBox1.ForeColor = Color.Green;

}

if (value >= 32 && value <= 50)

{

textBox1.ForeColor = Color.Blue;

}

if (value < 32)

{

textBox1.ForeColor = Color.Black;

}

}

}

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

{

label2.Text = "F";

label3.Text = "C";

}

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

{

label2.Text = "C";

label3.Text = "F";

}

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

{

string title = "Temperature" + "\t" + " Date" + "\t" + "\t" + "Time" + "\t\t" + "Message" + "\n";

string name = "Jose Bellorin";

try

{

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

string row = "";

StreamReader objR = new StreamReader(fs);

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

{

row += objR.ReadLine() + "\n";

}

MessageBox.Show(name+"\n\n\n"+title+"\n"+row);

}

catch (FileNotFoundException)

{

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

}

catch (DirectoryNotFoundException)

{

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

}

catch (IOException ex)

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

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

}

}

}

IPv4 & IP

… your code goes here

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. Class Data | I use to generate the randoms number and use those properties to set the value |
| 1. Class Calcs | It used to make all the calculations related to the money change |
| 1. Class Currencies | I use to have all the currencies and that help me when I want to return some values at the same time |
| 1. Class Temperature | Was used like class base with the function ConvertTemp() to apply polymorphism |
| 1. Class C to F | Use with the override to use this method that come from the class temperature to make de conversion |
| 1. Class F to C | Use with the override to use this method that come from the class temperature to make de conversion |

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

The problem I had was how use the text files but after the explanation in class I understand that better.