|  |
| --- |
| Collège LaSalle |
| Projet P25 - Introduction à la Programmation objet - Manuel Utilisateur et Technique |
|  |
| Présenté a : Mihai Maftei |

|  |
| --- |
| Auteur: Lyes HADJ AISSA  7/27/2017 |

1. **Description du projet :**

* Application : Menu principal qui contient 5 application : Deux application qui converties les devises et les températures.  
  Une application de calculatrice, et deux applications de Lotto : Lotto 6-49 et Lotto max.
* Application de convertisseur de devise : application qui convertie 5 devises : CAD, USD, Euro, CHF, GBP et il enregistre les résultats des conversions avec date et heure dans un fichier texte et il donne la possibilité de lire le fichier texte.
* Application de Calculatrice : Une application avec des boutons qui fait le calcul.
* Application de convertisseur de température : Application qui convertie la température de fahrenheit en Celsius ou de Celsius en fahrenheit, en utilisant une classe avec des méthodes et des constructeurs et il enregistre les résultats des conversions avec date et heure dans un fichier texte et il donne la possibilité de lire le fichier texte.
* Lotto Max : Application de Lotto qui génère huit chiffres et il enregistre avec date et heure dans un fichier texte et il donne la possibilité de lire le fichier texte.
* Lotto 6\_49 : Application de Lotto qui génère sept chiffres et il enregistre dans un fichier texte avec date et heure et il donne la possibilité de lire le fichier texte.

1. **Langage utilisé :**

* Windows Visual Studio 2015, Application Form avec C#.

1. **Les écrans d’impression que j’ai créée avec une description détaillée du fonctionnement :**

L’application Dash Board avec Prénom+ nom :

Une image contenant capture d’écran

Description générée avec un niveau de confiance élevé

La message box qui ferme Dash Board :

Une image contenant capture d’écran

Description générée avec un niveau de confiance très élevé

Application échange de l’argent :

Une image contenant capture d’écran

Description générée avec un niveau de confiance très élevé

Fichier texte d’échange de l’argent :

Une image contenant capture d’écran

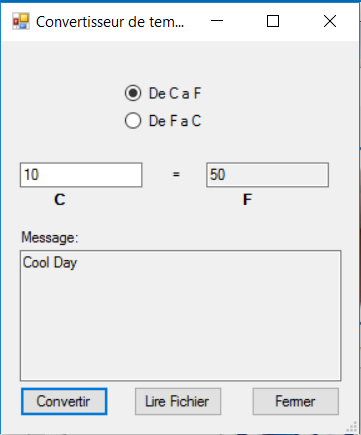
Description générée avec un niveau de confiance très élevé

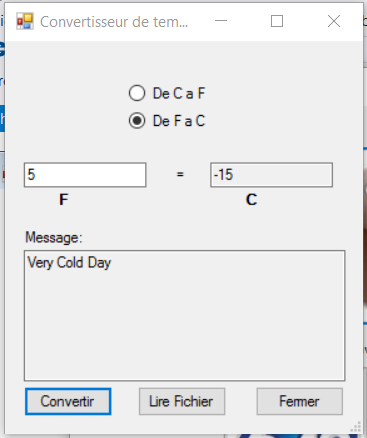
La message box pour fermer l’application fermer échange de l’argent :

Une image contenant capture d’écran

Description générée avec un niveau de confiance très élevé

L’application de convertisseur de température :





Fichier texte de convertisseur de température :

Une image contenant capture d’écran

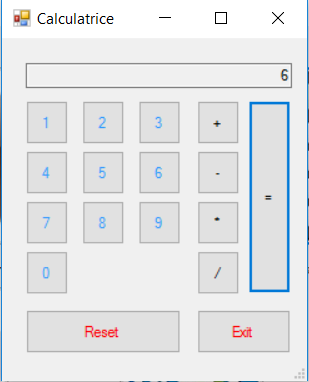
Description générée avec un niveau de confiance très élevé

La message box pour quitter l’application convertisseur de température :

Une image contenant capture d’écran

Description générée avec un niveau de confiance très élevé

La Calculatrice :

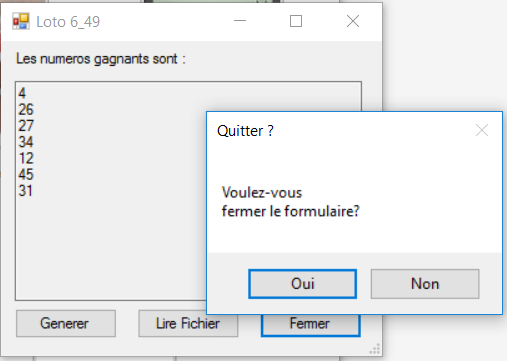


L’application Lotto 4\_49 :

Une image contenant capture d’écran

Description générée avec un niveau de confiance très élevé

Message box pour fermer Lotto 6\_49 :



Le fichier texte du Lotto 6\_49 :

Une image contenant capture d’écran

Description générée avec un niveau de confiance très élevé

Lotto Max :

Une image contenant capture d’écran

Description générée avec un niveau de confiance très élevé

Le fichier texte du Lotto Max :

Une image contenant capture d’écran

Description générée avec un niveau de confiance très élevé

Message box pour fermer le Lotto Max :

Une image contenant capture d’écran

Description générée avec un niveau de confiance très élevé

1. **Les codes que j’ai créés :**

* **Dash Board:**

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

{

string ex;

ex = Convert.ToString(MessageBox.Show("Êtes-vous sûr de vouloir de quitter l'application?", "Quitter?", MessageBoxButtons.YesNo));

if (ex == "Yes")

this.Close();

}

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

{

Loto\_Max LotoMx = new Loto\_Max();

LotoMx.ShowDialog();

}

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

{

Loto6\_49Form Loto649 = new Loto6\_49Form();

Loto649.ShowDialog();

}

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

{

EchangeArgentForm EchangeArg = new EchangeArgentForm();

EchangeArg.ShowDialog();

}

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

{

ConverTemperatureForm ConvTem = new ConverTemperatureForm();

ConvTem.ShowDialog();

}

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

{

CalculatriceForm Calcul = new CalculatriceForm();

Calcul.ShowDialog();

}

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

{

}

* **Calculatrice:**

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

{

txtDisplay.Text = ""; // initialise la textbox

}

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

{

string ex;

ex = Convert.ToString(MessageBox.Show("Voulez vous fermer la calculatrice?", "Quitter?", MessageBoxButtons.YesNo));

if (ex == "Yes")

this.Close();

}

double num1, num2;

double res;

string operateur;

private void btn1\_Click(object sender, EventArgs e)

{

txtDisplay.Text += "1";

}

private void btn2\_Click(object sender, EventArgs e)

{

txtDisplay.Text += "2";

}

private void btn3\_Click(object sender, EventArgs e)

{

txtDisplay.Text += "3";

}

private void btn4\_Click(object sender, EventArgs e)

{

txtDisplay.Text += "4";

}

private void btn5\_Click(object sender, EventArgs e)

{

txtDisplay.Text += "5";

}

private void btn6\_Click(object sender, EventArgs e)

{

txtDisplay.Text += "6";

}

private void btn7\_Click(object sender, EventArgs e)

{

txtDisplay.Text += "7";

}

private void btn8\_Click(object sender, EventArgs e)

{

txtDisplay.Text += "8";

}

private void btn9\_Click(object sender, EventArgs e)

{

txtDisplay.Text += "9";

}

private void btn0\_Click(object sender, EventArgs e)

{

txtDisplay.Text += "0";

}

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

{

if (txtDisplay.Text != "") // Faire les calcules

{

num2 = Convert.ToDouble(txtDisplay.Text);

}

if (operateur == "+")

{

Calculate c1 = new Calculate(num1, num2);

res = c1.Add();

txtDisplay.Text = res.ToString();

}

if (operateur == "-")

{

Calculate c1 = new Calculate(num1, num2);

res = c1.Sub();

txtDisplay.Text = res.ToString();

}

if (operateur == "\*")

{

Calculate c1 = new Calculate(num1, num2);

res = c1.Mul();

txtDisplay.Text = res.ToString();

}

if (operateur == "/")

{

Calculate c1 = new Calculate(num1, num2);

res = c1.Div();

txtDisplay.Text = res.ToString();

}

}

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

{

operateur = "-";

num1 = Convert.ToDouble(txtDisplay.Text);

txtDisplay.Text = "";

txtDisplay.Focus();

}

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

{

operateur = "\*";

num1 = Convert.ToDouble(txtDisplay.Text);

txtDisplay.Text = "";

txtDisplay.Focus();

}

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

{

operateur = "/";

num1 = Convert.ToDouble(txtDisplay.Text);

txtDisplay.Text = "";

txtDisplay.Focus();

}

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

{

}

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

{

operateur = "+";

num1 = Convert.ToDouble(txtDisplay.Text);

txtDisplay.Text = "";

txtDisplay.Focus();

}

}

}

class Calculate

{

private double val1;

private double val2;

private double Res;

public double Va1 // Les propriétés

{

get { return val1; }

set { val1 = value; }

}

public double Val2

{

get { return val2; }

set { val2 = value; }

}

public Calculate() { } // les constructeurs

public Calculate(double Val1,double Val2)

{

this.val1 = Val1;

this.val2 = Val2;

}

public double Add() { // les méthodes pour le calcule

Res =val1+val2;

return Res;

}

public double Sub()

{

Res = val1 - val2;

return Res;

}

public double Mul()

{

Res = val1 \* val2;

return Res;

}

public double Div()

{

Res = val1 / val2;

return Res;

}

}

* **Convertisseur de température :**

public partial class ConverTemperatureForm : Form

{

public ConverTemperatureForm()

{

InitializeComponent();

}

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

{

string ex;

ex = Convert.ToString(MessageBox.Show("Voulez vous \nfermer le formulaire?", "Quitter?", MessageBoxButtons.YesNo));

if (ex == "Yes")

this.Close();

}

private void radioBtn1\_CheckedChanged(object sender, EventArgs e)

{

if (radioBtn1.Checked == true)

{

lbl1.Text = "C";

lbl2.Text = "F";

}

}

private void radioBtn2\_CheckedChanged(object sender, EventArgs e)

{

if (radioBtn2.Checked == true)

{

lbl1.Text = "F";

lbl2.Text = "C";

}

}

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

{

double temperature;

double resultat;

string temp = @"^(\-?\d?\d?\d)$";

if (radioBtn1.Checked == true)

{

try

{

if (Regex.IsMatch(textBox1.Text, temp)) // Convertir de Celsuis a Fahrenheit

{

// MessageBox.Show("Temperature valide");

temperature = Convert.ToDouble(textBox1.Text);

ConverTemp c1 = new ConverTemp(temperature);

resultat = c1.CelsToFarnh();

textBox2.Text = resultat.ToString();

textBox2.ReadOnly = true;

if (resultat >= 212)

{

txtMessage.Text = "Water Boils";

}

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

{

txtMessage.Text = "Hot Bath";

}

if ((resultat >= 98.6) && (resultat < 104))

{

txtMessage.Text = "Body temperature";

}

if ((resultat >= 86) && (resultat < 98.6))

{

txtMessage.Text = "Beach weather";

}

if ((resultat >= 70) && (resultat < 86))

{

txtMessage.Text = "Room temperature";

}

if ((resultat >= 50) && (resultat < 70))

{

txtMessage.Text = "Cool Day";

}

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

{

txtMessage.Text = "Freezing point of water";

}

if ((resultat >= 0) && (resultat < 32))

{

txtMessage.Text = "Very Cold Day";

}

if (resultat <= -40)

{

txtMessage.Text = "Extremly Cold Day";

}

}

else

{

MessageBox.Show("Temperature non valide!! \nEntrer une temperature valide!", "Erreur!");

textBox1.Text = "";

textBox1.Focus();

}

// Ecrire dans un fichier text

string dossier = @"..\FichiersTexte\Fichier\";

if (!Directory.Exists(dossier))

Directory.CreateDirectory(dossier);

string fichier = dossier + "TempConversions.txt";

FileStream FichierTex = null;

try

{

FichierTex = new FileStream(fichier, FileMode.Append, FileAccess.Write);

StreamWriter EcrireFT = new StreamWriter(FichierTex);

EcrireFT.Write(textBox1.Text + " Celsius " + " = " + textBox2.Text + " Fahrenheit" + " | ");

EcrireFT.WriteLine(DateTime.Now);

EcrireFT.Close();

}

catch (FileNotFoundException)

{

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

}

catch (DirectoryNotFoundException)

{

MessageBox.Show(dossier + " not found.");

}

catch (IOException ex)

{

MessageBox.Show(ex.Message, "IO EXception");

}

finally

{

if (FichierTex != null)

FichierTex.Close();

}

}

catch (Exception ex1)

{

MessageBox.Show("Veuillez entrer une temperature valide!!", "Erreur!", MessageBoxButtons.OK, MessageBoxIcon.Error);

textBox1.Text = "";

textBox1.Focus();

}

}

if (radioBtn2.Checked == true) // Convertir de Fahrenheit a Celsuis

{

try

{

if (Regex.IsMatch(textBox1.Text, temp))

{

// MessageBox.Show("Temperature valide");

temperature = Convert.ToDouble(textBox1.Text);

ConverTemp c1 = new ConverTemp(temperature);

resultat = c1.FarnhToCels();

textBox2.Text = resultat.ToString();

textBox2.ReadOnly = true;

if (resultat >= 100)

{

txtMessage.Text = "Water Boils";

}

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

{

txtMessage.Text = "Hot Bath";

}

if ((resultat >= 37) && (resultat < 40))

{

txtMessage.Text = "Body temperature";

}

if ((resultat >= 30) && (resultat < 37))

{

txtMessage.Text = "Beach weather";

}

if ((resultat >= 21) && (resultat < 30))

{

txtMessage.Text = "Room temperature";

}

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

{

txtMessage.Text = "Cool Day";

}

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

{

txtMessage.Text = "Freezing point of water";

}

if ((resultat >= -18) && (resultat < 0))

{

txtMessage.Text = "Very Cold Day";

}

if (resultat <= -40)

{

txtMessage.Text = "Extremly Cold Day";

}

}

else

{

MessageBox.Show("Temperature non valide !!","Erreur!");

}

//Ecrire dans un fichier texte

string dossier = @"..\FichiersTexte\Fichier\";

if (!Directory.Exists(dossier))

Directory.CreateDirectory(dossier);

string fichier = dossier + "TempConversions.txt";

FileStream FichierTex = null;

try

{

FichierTex = new FileStream(fichier, FileMode.Append, FileAccess.Write);

StreamWriter EcrireFT = new StreamWriter(FichierTex);

EcrireFT.Write(textBox1.Text + " Fahrenheit " + " = " + textBox2.Text + " Celsius " + " | ");

EcrireFT.WriteLine(DateTime.Now);

EcrireFT.Close();

}

catch (FileNotFoundException)

{

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

}

catch (DirectoryNotFoundException)

{

MessageBox.Show(dossier + " not found.");

}

catch (IOException ex)

{

MessageBox.Show(ex.Message, "IO EXception");

}

finally

{

if (FichierTex != null)

FichierTex.Close();

}

}

catch (Exception ex1)

{

MessageBox.Show("Temperature non valide!! \nEntrer une temperature valide!!", "Erreur!", MessageBoxButtons.OK, MessageBoxIcon.Error);

textBox1.Text = "";

textBox1.Focus();

}

}

}

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

{

}

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

{ // Lire le fichier text

string dossier = @"..\FichiersTexte\Fichier\";

if (!Directory.Exists(dossier))

Directory.CreateDirectory(dossier);

string fichier = dossier + "TempConversions.txt";

FileStream FichierTex = null;

try

{

FichierTex = new FileStream(fichier, FileMode.OpenOrCreate, FileAccess.Read);

StreamReader LireFT = new StreamReader(FichierTex);

string lirefichier = "";

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

{

string row = LireFT.ReadLine();

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

//lirefichier += ConvArg[0]+ ConvArg[1] + "\n";

lirefichier = File.ReadAllText(@"..\FichiersTexte\Fichier\TempConversions.txt");

}

MessageBox.Show("Convertisseur de temperature : \n" + lirefichier, "Lyes Hadj Aissa");

LireFT.Close();

}

catch (FileNotFoundException)

{

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

}

catch (DirectoryNotFoundException)

{

MessageBox.Show(dossier + " not found.");

}

catch (IOException ex)

{

MessageBox.Show(ex.Message, "IO EXception");

}

finally

{

if (FichierTex != null)

FichierTex.Close();

}

}

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

{

}

}

}

class ConverTemp

{

double temp;

double res;

public double Temp

{

get { return temp; }

set { temp = value; }

}

public ConverTemp() { } // Les constructeurs

public ConverTemp(double T)

{

this.temp = T;

}

public double CelsToFarnh() // Methode 1

{

res = temp \* 9 / 5;

res = res + 32;

return res;

}

public double FarnhToCels() // Methode 2

{

res = temp - 32;

res = res \* 5 / 9;

return res;

}

}

* **Convertisseur de devises :**

namespace PROJET\_PROGRAM\_ORIENTE\_OBJET

{

public partial class EchangeArgentForm : Form

{

public EchangeArgentForm()

{

InitializeComponent();

}

private void groupBox1\_Enter(object sender, EventArgs e)

{

}

private void radioButton7\_CheckedChanged(object sender, EventArgs e)

{

}

private void radioButton8\_CheckedChanged(object sender, EventArgs e)

{

}

private void radioButton9\_CheckedChanged(object sender, EventArgs e)

{

}

private void radioButton6\_CheckedChanged(object sender, EventArgs e)

{

}

private void radioButton10\_CheckedChanged(object sender, EventArgs e)

{

}

private void radioButton5\_CheckedChanged(object sender, EventArgs e)

{

}

private void radioButton4\_CheckedChanged(object sender, EventArgs e)

{

}

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

{

}

private void radioButton3\_CheckedChanged(object sender, EventArgs e)

{

}

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

{

}

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

{

string ex;

ex = Convert.ToString(MessageBox.Show("Voulez vous \nfermer le formulaire \nÉchange de l'argent?", "Quitter?", MessageBoxButtons.YesNo));

if (ex == "Yes")

this.Close();

}

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

{

double d, res ;

string argent = @"^(\d?\d?\d?\d)$" , txt = "";

if (Regex.IsMatch(txtBox1.Text, argent)) // Utilistation de regex

{

txt = txtBox1.Text;

// MessageBox.Show("Argent valide!!");

// Faire le calcul pour toutes les devises et ecrire dans un fichier texte

if ((radioBtn1.Checked == true) && (radioBtn6.Checked == true))

{

d = Convert.ToDouble(txt);

ConvertMoney c2 = new ConvertMoney(d) { };

res = c2.CadToCad();

txtBox2.Text = res.ToString();

txtBox2.ReadOnly = true;

string dossier = @"..\FichiersTexte\Fichier\";

if (!Directory.Exists(dossier))

Directory.CreateDirectory(dossier);

string fichier = dossier + "MoneyConversions.txt";

FileStream FichierTex = null;

try

{

FichierTex = new FileStream(fichier, FileMode.Append, FileAccess.Write);

StreamWriter EcrireFT = new StreamWriter(FichierTex);

EcrireFT.Write(txtBox1.Text + " CAD " + " = " + txtBox2.Text + " CAD" + " | ");

EcrireFT.WriteLine(DateTime.Now);

// EcrireFT.WriteLine(DateTime.Now + " | ");

EcrireFT.Close();

}

catch (FileNotFoundException)

{

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

}

catch (DirectoryNotFoundException)

{

MessageBox.Show(dossier + " not found.");

}

catch (IOException ex)

{

MessageBox.Show(ex.Message, "IO EXception");

}

finally

{

if (FichierTex != null)

FichierTex.Close();

}

}

if ((radioBtn1.Checked == true) && (radioBtn7.Checked == true))

{

d = Convert.ToDouble(txtBox1.Text);

ConvertMoney c1 = new ConvertMoney(d) { };

res = c1.CadToUsd();

txtBox2.Text = res.ToString();

txtBox2.ReadOnly = true;

string dossier = @"..\FichiersTexte\Fichier\";

if (!Directory.Exists(dossier))

Directory.CreateDirectory(dossier);

string fichier = dossier + "MoneyConversions.txt";

FileStream FichierTex = null;

try

{

FichierTex = new FileStream(fichier, FileMode.Append, FileAccess.Write);

StreamWriter EcrireFT = new StreamWriter(FichierTex);

EcrireFT.Write(txtBox1.Text + " CAD " + " = " + txtBox2.Text + " USD" + " | ");

EcrireFT.WriteLine(DateTime.Now);

EcrireFT.Close();

}

catch (FileNotFoundException)

{

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

}

catch (DirectoryNotFoundException)

{

MessageBox.Show(dossier + " not found.");

}

catch (IOException ex)

{

MessageBox.Show(ex.Message, "IO EXception");

}

finally

{

if (FichierTex != null)

FichierTex.Close();

}

}

if ((radioBtn1.Checked == true) && (radioBtn8.Checked == true))

{

try

{

d = Convert.ToDouble(txtBox1.Text);

ConvertMoney c3 = new ConvertMoney(d) { };

res = c3.CadToEuro();

txtBox2.Text = res.ToString();

txtBox2.ReadOnly = true;

string dossier = @"..\FichiersTexte\Fichier\";

if (!Directory.Exists(dossier))

Directory.CreateDirectory(dossier);

string fichier = dossier + "MoneyConversions.txt";

FileStream FichierTex = null;

try

{

FichierTex = new FileStream(fichier, FileMode.Append, FileAccess.Write);

StreamWriter EcrireFT = new StreamWriter(FichierTex);

EcrireFT.Write(txtBox1.Text + " CAD " + " = " + txtBox2.Text + " EURO" + " | ");

EcrireFT.WriteLine(DateTime.Now);

EcrireFT.Close();

}

catch (FileNotFoundException)

{

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

}

catch (DirectoryNotFoundException)

{

MessageBox.Show(dossier + " not found.");

}

catch (IOException ex)

{

MessageBox.Show(ex.Message, "IO EXception");

}

finally

{

if (FichierTex != null)

FichierTex.Close();

}

}catch (Exception ex) {

MessageBox.Show(ex.Message+" Erreur");

}

}

if ((radioBtn1.Checked == true) && (radioBtn9.Checked == true))

{

d = Convert.ToDouble(txtBox1.Text);

ConvertMoney c4 = new ConvertMoney(d) { };

res = c4.CadToGbp();

txtBox2.Text = res.ToString();

txtBox2.ReadOnly = true;

string dossier = @"..\FichiersTexte\Fichier\";

if (!Directory.Exists(dossier))

Directory.CreateDirectory(dossier);

string fichier = dossier + "MoneyConversions.txt";

FileStream FichierTex = null;

try

{

FichierTex = new FileStream(fichier, FileMode.Append, FileAccess.Write);

StreamWriter EcrireFT = new StreamWriter(FichierTex);

EcrireFT.Write(txtBox1.Text + " CAD " + " = " + txtBox2.Text + " GBP" + " | ");

EcrireFT.WriteLine(DateTime.Now);

EcrireFT.Close();

}

catch (FileNotFoundException)

{

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

}

catch (DirectoryNotFoundException)

{

MessageBox.Show(dossier + " not found.");

}

catch (IOException ex)

{

MessageBox.Show(ex.Message, "IO EXception");

}

finally

{

if (FichierTex != null)

FichierTex.Close();

}

}

if ((radioBtn1.Checked == true) && (radioBtn10.Checked == true))

{

d = Convert.ToDouble(txtBox1.Text);

ConvertMoney c5 = new ConvertMoney(d) { };

res = c5.CadToChf();

txtBox2.Text = res.ToString();

txtBox2.ReadOnly = true;

string dossier = @"..\FichiersTexte\Fichier\";

if (!Directory.Exists(dossier))

Directory.CreateDirectory(dossier);

string fichier = dossier + "MoneyConversions.txt";

FileStream FichierTex = null;

try

{

FichierTex = new FileStream(fichier, FileMode.Append, FileAccess.Write);

StreamWriter EcrireFT = new StreamWriter(FichierTex);

EcrireFT.Write(txtBox1.Text + " CAD " + " = " + txtBox2.Text + " CHF" + " | ");

EcrireFT.WriteLine(DateTime.Now);

EcrireFT.Close();

}

catch (FileNotFoundException)

{

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

}

catch (DirectoryNotFoundException)

{

MessageBox.Show(dossier + " not found.");

}

catch (IOException ex)

{

MessageBox.Show(ex.Message, "IO EXception");

}

finally

{

if (FichierTex != null)

FichierTex.Close();

}

}

if ((radioBtn2.Checked == true) && (radioBtn6.Checked == true))

{

d = Convert.ToDouble(txtBox1.Text);

ConvertMoney c5 = new ConvertMoney(d) { };

res = c5.UsdToCad();

txtBox2.Text = res.ToString();

txtBox2.ReadOnly = true;

string dossier = @"..\FichiersTexte\Fichier\";

if (!Directory.Exists(dossier))

Directory.CreateDirectory(dossier);

string fichier = dossier + "MoneyConversions.txt";

FileStream FichierTex = null;

try

{

FichierTex = new FileStream(fichier, FileMode.Append, FileAccess.Write);

StreamWriter EcrireFT = new StreamWriter(FichierTex);

EcrireFT.Write(txtBox1.Text + " USD " + " = " + txtBox2.Text + " CAD" + " | ");

EcrireFT.WriteLine(DateTime.Now);

EcrireFT.Close();

}

catch (FileNotFoundException)

{

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

}

catch (DirectoryNotFoundException)

{

MessageBox.Show(dossier + " not found.");

}

catch (IOException ex)

{

MessageBox.Show(ex.Message, "IO EXception");

}

finally

{

if (FichierTex != null)

FichierTex.Close();

}

}

if ((radioBtn2.Checked == true) && (radioBtn7.Checked == true))

{

d = Convert.ToDouble(txtBox1.Text);

ConvertMoney c5 = new ConvertMoney(d) { };

res = c5.UsdToUsd();

txtBox2.Text = res.ToString();

txtBox2.ReadOnly = true;

string dossier = @"..\FichiersTexte\Fichier\";

if (!Directory.Exists(dossier))

Directory.CreateDirectory(dossier);

string fichier = dossier + "MoneyConversions.txt";

FileStream FichierTex = null;

try

{

FichierTex = new FileStream(fichier, FileMode.Append, FileAccess.Write);

StreamWriter EcrireFT = new StreamWriter(FichierTex);

EcrireFT.Write(txtBox1.Text + " USD " + " = " + txtBox2.Text + " USD" + " | ");

EcrireFT.WriteLine(DateTime.Now);

EcrireFT.Close();

}

catch (FileNotFoundException)

{

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

}

catch (DirectoryNotFoundException)

{

MessageBox.Show(dossier + " not found.");

}

catch (IOException ex)

{

MessageBox.Show(ex.Message, "IO EXception");

}

finally

{

if (FichierTex != null)

FichierTex.Close();

}

}

if ((radioBtn2.Checked == true) && (radioBtn8.Checked == true))

{

d = Convert.ToDouble(txtBox1.Text);

ConvertMoney c5 = new ConvertMoney(d) { };

res = c5.UsdToEuro();

txtBox2.Text = res.ToString();

txtBox2.ReadOnly = true;

string dossier = @"..\FichiersTexte\Fichier\";

if (!Directory.Exists(dossier))

Directory.CreateDirectory(dossier);

string fichier = dossier + "MoneyConversions.txt";

FileStream FichierTex = null;

try

{

FichierTex = new FileStream(fichier, FileMode.Append, FileAccess.Write);

StreamWriter EcrireFT = new StreamWriter(FichierTex);

EcrireFT.Write(txtBox1.Text + " USD " + " = " + txtBox2.Text + " EURO" + " | ");

EcrireFT.WriteLine(DateTime.Now);

EcrireFT.Close();

}

catch (FileNotFoundException)

{

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

}

catch (DirectoryNotFoundException)

{

MessageBox.Show(dossier + " not found.");

}

catch (IOException ex)

{

MessageBox.Show(ex.Message, "IO EXception");

}

finally

{

if (FichierTex != null)

FichierTex.Close();

}

}

if ((radioBtn2.Checked == true) && (radioBtn9.Checked == true))

{

d = Convert.ToDouble(txtBox1.Text);

ConvertMoney c5 = new ConvertMoney(d) { };

res = c5.UsdToGpb();

txtBox2.Text = res.ToString();

txtBox2.ReadOnly = true;

string dossier = @"..\FichiersTexte\Fichier\";

if (!Directory.Exists(dossier))

Directory.CreateDirectory(dossier);

string fichier = dossier + "MoneyConversions.txt";

FileStream FichierTex = null;

try

{

FichierTex = new FileStream(fichier, FileMode.Append, FileAccess.Write);

StreamWriter EcrireFT = new StreamWriter(FichierTex);

EcrireFT.Write(txtBox1.Text + " USD " + " = " + txtBox2.Text + " GBP" + " | ");

EcrireFT.WriteLine(DateTime.Now);

EcrireFT.Close();

}

catch (FileNotFoundException)

{

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

}

catch (DirectoryNotFoundException)

{

MessageBox.Show(dossier + " not found.");

}

catch (IOException ex)

{

MessageBox.Show(ex.Message, "IO EXception");

}

finally

{

if (FichierTex != null)

FichierTex.Close();

}

}

if ((radioBtn2.Checked == true) && (radioBtn10.Checked == true))

{

d = Convert.ToDouble(txtBox1.Text);

ConvertMoney c5 = new ConvertMoney(d) { };

res = c5.UsdToChf();

txtBox2.Text = res.ToString();

txtBox2.ReadOnly = true;

string dossier = @"..\FichiersTexte\Fichier\";

if (!Directory.Exists(dossier))

Directory.CreateDirectory(dossier);

string fichier = dossier + "MoneyConversions.txt";

FileStream FichierTex = null;

try

{

FichierTex = new FileStream(fichier, FileMode.Append, FileAccess.Write);

StreamWriter EcrireFT = new StreamWriter(FichierTex);

EcrireFT.Write(txtBox1.Text + " USD " + " = " + txtBox2.Text + " CHF" + " | ");

EcrireFT.WriteLine(DateTime.Now);

EcrireFT.Close();

}

catch (FileNotFoundException)

{

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

}

catch (DirectoryNotFoundException)

{

MessageBox.Show(dossier + " not found.");

}

catch (IOException ex)

{

MessageBox.Show(ex.Message, "IO EXception");

}

finally

{

if (FichierTex != null)

FichierTex.Close();

}

}

if ((radioBtn3.Checked == true) && (radioBtn6.Checked == true))

{

d = Convert.ToDouble(txtBox1.Text);

ConvertMoney c5 = new ConvertMoney(d) { };

res = c5.EuroToCad();

txtBox2.Text = res.ToString();

txtBox2.ReadOnly = true;

string dossier = @"..\FichiersTexte\Fichier\";

if (!Directory.Exists(dossier))

Directory.CreateDirectory(dossier);

string fichier = dossier + "MoneyConversions.txt";

FileStream FichierTex = null;

try

{

FichierTex = new FileStream(fichier, FileMode.Append, FileAccess.Write);

StreamWriter EcrireFT = new StreamWriter(FichierTex);

EcrireFT.Write(txtBox1.Text + " EURO " + " = " + txtBox2.Text + " CAD" + " | ");

EcrireFT.WriteLine(DateTime.Now);

EcrireFT.Close();

}

catch (FileNotFoundException)

{

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

}

catch (DirectoryNotFoundException)

{

MessageBox.Show(dossier + " not found.");

}

catch (IOException ex)

{

MessageBox.Show(ex.Message, "IO EXception");

}

finally

{

if (FichierTex != null)

FichierTex.Close();

}

}

if ((radioBtn3.Checked == true) && (radioBtn7.Checked == true))

{

d = Convert.ToDouble(txtBox1.Text);

ConvertMoney c5 = new ConvertMoney(d) { };

res = c5.EuroToUsd();

txtBox2.Text = res.ToString();

txtBox2.ReadOnly = true;

string dossier = @"..\FichiersTexte\Fichier\";

if (!Directory.Exists(dossier))

Directory.CreateDirectory(dossier);

string fichier = dossier + "MoneyConversions.txt";

FileStream FichierTex = null;

try

{

FichierTex = new FileStream(fichier, FileMode.Append, FileAccess.Write);

StreamWriter EcrireFT = new StreamWriter(FichierTex);

EcrireFT.Write(txtBox1.Text + " EURO " + " = " + txtBox2.Text + " USD" + " | ");

EcrireFT.WriteLine(DateTime.Now);

EcrireFT.Close();

}

catch (FileNotFoundException)

{

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

}

catch (DirectoryNotFoundException)

{

MessageBox.Show(dossier + " not found.");

}

catch (IOException ex)

{

MessageBox.Show(ex.Message, "IO EXception");

}

finally

{

if (FichierTex != null)

FichierTex.Close();

}

}

if ((radioBtn3.Checked == true) && (radioBtn8.Checked == true))

{

d = Convert.ToDouble(txtBox1.Text);

ConvertMoney c5 = new ConvertMoney(d) { };

res = c5.EuroToEuro();

txtBox2.Text = res.ToString();

txtBox2.ReadOnly = true;

string dossier = @"..\FichiersTexte\Fichier\";

if (!Directory.Exists(dossier))

Directory.CreateDirectory(dossier);

string fichier = dossier + "MoneyConversions.txt";

FileStream FichierTex = null;

try

{

FichierTex = new FileStream(fichier, FileMode.Append, FileAccess.Write);

StreamWriter EcrireFT = new StreamWriter(FichierTex);

EcrireFT.Write(txtBox1.Text + " EURO " + " = " + txtBox2.Text + " EURO" + " | ");

EcrireFT.WriteLine(DateTime.Now);

EcrireFT.Close();

}

catch (FileNotFoundException)

{

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

}

catch (DirectoryNotFoundException)

{

MessageBox.Show(dossier + " not found.");

}

catch (IOException ex)

{

MessageBox.Show(ex.Message, "IO EXception");

}

finally

{

if (FichierTex != null)

FichierTex.Close();

}

}

if ((radioBtn3.Checked == true) && (radioBtn9.Checked == true))

{

d = Convert.ToDouble(txtBox1.Text);

ConvertMoney c5 = new ConvertMoney(d) { };

res = c5.EuroToGbp();

txtBox2.Text = res.ToString();

txtBox2.ReadOnly = true;

string dossier = @"..\FichiersTexte\Fichier\";

if (!Directory.Exists(dossier))

Directory.CreateDirectory(dossier);

string fichier = dossier + "MoneyConversions.txt";

FileStream FichierTex = null;

try

{

FichierTex = new FileStream(fichier, FileMode.Append, FileAccess.Write);

StreamWriter EcrireFT = new StreamWriter(FichierTex);

EcrireFT.Write(txtBox1.Text + " EURO " + " = " + txtBox2.Text + " GBP" + " | ");

EcrireFT.WriteLine(DateTime.Now);

EcrireFT.Close();

}

catch (FileNotFoundException)

{

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

}

catch (DirectoryNotFoundException)

{

MessageBox.Show(dossier + " not found.");

}

catch (IOException ex)

{

MessageBox.Show(ex.Message, "IO EXception");

}

finally

{

if (FichierTex != null)

FichierTex.Close();

}

}

if ((radioBtn3.Checked == true) && (radioBtn10.Checked == true))

{

d = Convert.ToDouble(txtBox1.Text);

ConvertMoney c5 = new ConvertMoney(d) { };

res = c5.EuroToChf();

txtBox2.Text = res.ToString();

txtBox2.ReadOnly = true;

string dossier = @"..\FichiersTexte\Fichier\";

if (!Directory.Exists(dossier))

Directory.CreateDirectory(dossier);

string fichier = dossier + "MoneyConversions.txt";

FileStream FichierTex = null;

try

{

FichierTex = new FileStream(fichier, FileMode.Append, FileAccess.Write);

StreamWriter EcrireFT = new StreamWriter(FichierTex);

EcrireFT.Write(txtBox1.Text + " EURO " + " = " + txtBox2.Text + " CHF" + " | ");

EcrireFT.WriteLine(DateTime.Now);

EcrireFT.Close();

}

catch (FileNotFoundException)

{

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

}

catch (DirectoryNotFoundException)

{

MessageBox.Show(dossier + " not found.");

}

catch (IOException ex)

{

MessageBox.Show(ex.Message, "IO EXception");

}

finally

{

if (FichierTex != null)

FichierTex.Close();

}

}

if ((radioBtn4.Checked == true) && (radioBtn6.Checked == true))

{

d = Convert.ToDouble(txtBox1.Text);

ConvertMoney c5 = new ConvertMoney(d) { };

res = c5.GbpToCad();

txtBox2.Text = res.ToString();

txtBox2.ReadOnly = true;

string dossier = @"..\FichiersTexte\Fichier\";

if (!Directory.Exists(dossier))

Directory.CreateDirectory(dossier);

string fichier = dossier + "MoneyConversions.txt";

FileStream FichierTex = null;

try

{

FichierTex = new FileStream(fichier, FileMode.Append, FileAccess.Write);

StreamWriter EcrireFT = new StreamWriter(FichierTex);

EcrireFT.Write(txtBox1.Text + " GBP " + " = " + txtBox2.Text + " CAD" + " | ");

EcrireFT.WriteLine(DateTime.Now);

EcrireFT.Close();

}

catch (FileNotFoundException)

{

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

}

catch (DirectoryNotFoundException)

{

MessageBox.Show(dossier + " not found.");

}

catch (IOException ex)

{

MessageBox.Show(ex.Message, "IO EXception");

}

finally

{

if (FichierTex != null)

FichierTex.Close();

}

}

if ((radioBtn4.Checked == true) && (radioBtn7.Checked == true))

{

d = Convert.ToDouble(txtBox1.Text);

ConvertMoney c5 = new ConvertMoney(d) { };

res = c5.GbpToUsd();

txtBox2.Text = res.ToString();

txtBox2.ReadOnly = true;

string dossier = @"..\FichiersTexte\Fichier\";

if (!Directory.Exists(dossier))

Directory.CreateDirectory(dossier);

string fichier = dossier + "MoneyConversions.txt";

FileStream FichierTex = null;

try

{

FichierTex = new FileStream(fichier, FileMode.Append, FileAccess.Write);

StreamWriter EcrireFT = new StreamWriter(FichierTex);

EcrireFT.Write(txtBox1.Text + " GBP " + " = " + txtBox2.Text + " USD" + " | ");

EcrireFT.WriteLine(DateTime.Now);

EcrireFT.Close();

}

catch (FileNotFoundException)

{

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

}

catch (DirectoryNotFoundException)

{

MessageBox.Show(dossier + " not found.");

}

catch (IOException ex)

{

MessageBox.Show(ex.Message, "IO EXception");

}

finally

{

if (FichierTex != null)

FichierTex.Close();

}

}

if ((radioBtn4.Checked == true) && (radioBtn8.Checked == true))

{

d = Convert.ToDouble(txtBox1.Text);

ConvertMoney c5 = new ConvertMoney(d) { };

res = c5.GbpToEuro();

txtBox2.Text = res.ToString();

txtBox2.ReadOnly = true;

string dossier = @"..\FichiersTexte\Fichier\";

if (!Directory.Exists(dossier))

Directory.CreateDirectory(dossier);

string fichier = dossier + "MoneyConversions.txt";

FileStream FichierTex = null;

try

{

FichierTex = new FileStream(fichier, FileMode.Append, FileAccess.Write);

StreamWriter EcrireFT = new StreamWriter(FichierTex);

EcrireFT.Write(txtBox1.Text + " GBP " + " = " + txtBox2.Text + " EURO" + " | ");

EcrireFT.WriteLine(DateTime.Now);

EcrireFT.Close();

}

catch (FileNotFoundException)

{

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

}

catch (DirectoryNotFoundException)

{

MessageBox.Show(dossier + " not found.");

}

catch (IOException ex)

{

MessageBox.Show(ex.Message, "IO EXception");

}

finally

{

if (FichierTex != null)

FichierTex.Close();

}

}

if ((radioBtn4.Checked == true) && (radioBtn9.Checked == true))

{

d = Convert.ToDouble(txtBox1.Text);

ConvertMoney c5 = new ConvertMoney(d) { };

res = c5.GbpToGbp();

txtBox2.Text = res.ToString();

txtBox2.ReadOnly = true;

string dossier = @"..\FichiersTexte\Fichier\";

if (!Directory.Exists(dossier))

Directory.CreateDirectory(dossier);

string fichier = dossier + "MoneyConversions.txt";

FileStream FichierTex = null;

try

{

FichierTex = new FileStream(fichier, FileMode.Append, FileAccess.Write);

StreamWriter EcrireFT = new StreamWriter(FichierTex);

EcrireFT.Write(txtBox1.Text + " GBP " + " = " + txtBox2.Text + " GBP" + " | ");

EcrireFT.WriteLine(DateTime.Now);

EcrireFT.Close();

}

catch (FileNotFoundException)

{

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

}

catch (DirectoryNotFoundException)

{

MessageBox.Show(dossier + " not found.");

}

catch (IOException ex)

{

MessageBox.Show(ex.Message, "IO EXception");

}

finally

{

if (FichierTex != null)

FichierTex.Close();

}

}

if ((radioBtn4.Checked == true) && (radioBtn10.Checked == true))

{

d = Convert.ToDouble(txtBox1.Text);

ConvertMoney c5 = new ConvertMoney(d) { };

res = c5.GbpToChf();

txtBox2.Text = res.ToString();

txtBox2.ReadOnly = true;

string dossier = @"..\FichiersTexte\Fichier\";

if (!Directory.Exists(dossier))

Directory.CreateDirectory(dossier);

string fichier = dossier + "MoneyConversions.txt";

FileStream FichierTex = null;

try

{

FichierTex = new FileStream(fichier, FileMode.Append, FileAccess.Write);

StreamWriter EcrireFT = new StreamWriter(FichierTex);

EcrireFT.Write(txtBox1.Text + " GBP " + " = " + txtBox2.Text + " CHF" + " | ");

EcrireFT.WriteLine(DateTime.Now);

EcrireFT.Close();

}

catch (FileNotFoundException)

{

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

}

catch (DirectoryNotFoundException)

{

MessageBox.Show(dossier + " not found.");

}

catch (IOException ex)

{

MessageBox.Show(ex.Message, "IO EXception");

}

finally

{

if (FichierTex != null)

FichierTex.Close();

}

}

if ((radioBtn5.Checked == true) && (radioBtn6.Checked == true))

{

d = Convert.ToDouble(txtBox1.Text);

ConvertMoney c5 = new ConvertMoney(d) { };

res = c5.ChfToCad();

txtBox2.Text = res.ToString();

txtBox2.ReadOnly = true;

string dossier = @"..\FichiersTexte\Fichier\";

if (!Directory.Exists(dossier))

Directory.CreateDirectory(dossier);

string fichier = dossier + "MoneyConversions.txt";

FileStream FichierTex = null;

try

{

FichierTex = new FileStream(fichier, FileMode.Append, FileAccess.Write);

StreamWriter EcrireFT = new StreamWriter(FichierTex);

EcrireFT.Write(txtBox1.Text + " CHF " + " = " + txtBox2.Text + " CAD" + " | ");

EcrireFT.WriteLine(DateTime.Now);

EcrireFT.Close();

}

catch (FileNotFoundException)

{

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

}

catch (DirectoryNotFoundException)

{

MessageBox.Show(dossier + " not found.");

}

catch (IOException ex)

{

MessageBox.Show(ex.Message, "IO EXception");

}

finally

{

if (FichierTex != null)

FichierTex.Close();

}

}

if ((radioBtn5.Checked == true) && (radioBtn7.Checked == true))

{

d = Convert.ToDouble(txtBox1.Text);

ConvertMoney c5 = new ConvertMoney(d) { };

res = c5.ChfToUsd();

txtBox2.Text = res.ToString();

txtBox2.ReadOnly = true;

string dossier = @"..\FichiersTexte\Fichier\";

if (!Directory.Exists(dossier))

Directory.CreateDirectory(dossier);

string fichier = dossier + "MoneyConversions.txt";

FileStream FichierTex = null;

try

{

FichierTex = new FileStream(fichier, FileMode.Append, FileAccess.Write);

StreamWriter EcrireFT = new StreamWriter(FichierTex);

EcrireFT.Write(txtBox1.Text + " CHF " + " = " + txtBox2.Text + " USD" + " | ");

EcrireFT.WriteLine(DateTime.Now);

EcrireFT.Close();

}

catch (FileNotFoundException)

{

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

}

catch (DirectoryNotFoundException)

{

MessageBox.Show(dossier + " not found.");

}

catch (IOException ex)

{

MessageBox.Show(ex.Message, "IO EXception");

}

finally

{

if (FichierTex != null)

FichierTex.Close();

}

}

if ((radioBtn5.Checked == true) && (radioBtn8.Checked == true))

{

d = Convert.ToDouble(txtBox1.Text);

ConvertMoney c5 = new ConvertMoney(d) { };

res = c5.ChfToEuro();

txtBox2.Text = res.ToString();

txtBox2.ReadOnly = true;

string dossier = @"..\FichiersTexte\Fichier\";

if (!Directory.Exists(dossier))

Directory.CreateDirectory(dossier);

string fichier = dossier + "MoneyConversions.txt";

FileStream FichierTex = null;

try

{

FichierTex = new FileStream(fichier, FileMode.Append, FileAccess.Write);

StreamWriter EcrireFT = new StreamWriter(FichierTex);

EcrireFT.Write(txtBox1.Text + " CHF " + " = " + txtBox2.Text + " EURO" + " | ");

EcrireFT.WriteLine(DateTime.Now);

EcrireFT.Close();

}

catch (FileNotFoundException)

{

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

}

catch (DirectoryNotFoundException)

{

MessageBox.Show(dossier + " not found.");

}

catch (IOException ex)

{

MessageBox.Show(ex.Message, "IO EXception");

}

finally

{

if (FichierTex != null)

FichierTex.Close();

}

}

if ((radioBtn5.Checked == true) && (radioBtn9.Checked == true))

{

d = Convert.ToDouble(txtBox1.Text);

ConvertMoney c5 = new ConvertMoney(d) { };

res = c5.ChfToGbp();

txtBox2.Text = res.ToString();

txtBox2.ReadOnly = true;

string dossier = @"..\FichiersTexte\Fichier\";

if (!Directory.Exists(dossier))

Directory.CreateDirectory(dossier);

string fichier = dossier + "MoneyConversions.txt";

FileStream FichierTex = null;

try

{

FichierTex = new FileStream(fichier, FileMode.Append, FileAccess.Write);

StreamWriter EcrireFT = new StreamWriter(FichierTex);

EcrireFT.Write(txtBox1.Text + " CHF " + " = " + txtBox2.Text + " GBP" + " | ");

EcrireFT.WriteLine(DateTime.Now);

EcrireFT.Close();

}

catch (FileNotFoundException)

{

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

}

catch (DirectoryNotFoundException)

{

MessageBox.Show(dossier + " not found.");

}

catch (IOException ex)

{

MessageBox.Show(ex.Message, "IO EXception");

}

finally

{

if (FichierTex != null)

FichierTex.Close();

}

}

if ((radioBtn5.Checked == true) && (radioBtn10.Checked == true))

{

d = Convert.ToDouble(txtBox1.Text);

ConvertMoney c5 = new ConvertMoney(d) { };

res = c5.ChfToChf();

txtBox2.Text = res.ToString();

txtBox2.ReadOnly = true;

string dossier = @"..\FichiersTexte\Fichier\";

if (!Directory.Exists(dossier))

Directory.CreateDirectory(dossier);

string fichier = dossier + "MoneyConversions.txt";

FileStream FichierTex = null;

try

{

FichierTex = new FileStream(fichier, FileMode.Append, FileAccess.Write);

StreamWriter EcrireFT = new StreamWriter(FichierTex);

EcrireFT.Write(txtBox1.Text + " CHF " + " = " + txtBox2.Text + " CHF" + " | ");

EcrireFT.WriteLine(DateTime.Now);

EcrireFT.Close();

}

catch (FileNotFoundException)

{

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

}

catch (DirectoryNotFoundException)

{

MessageBox.Show(dossier + " not found.");

}

catch (IOException ex)

{

MessageBox.Show(ex.Message, "IO EXception");

}

finally

{

if (FichierTex != null)

FichierTex.Close();

}

}

}

else

{

MessageBox.Show("Entrer un nombre valide entre 0 et 9999.");

}

}

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

{ // Lire le fichier text

string dossier = @"..\FichiersTexte\Fichier\";

if (!Directory.Exists(dossier))

Directory.CreateDirectory(dossier);

string fichier = dossier + "MoneyConversions.txt";

FileStream FichierTex = null;

try

{

FichierTex = new FileStream(fichier, FileMode.OpenOrCreate, FileAccess.Read);

StreamReader LireFT = new StreamReader(FichierTex);

string lirefichier = "";

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

{

string row = LireFT.ReadLine();

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

//lirefichier += ConvArg[0]+ ConvArg[1] + "\n";

lirefichier = File.ReadAllText(@"..\FichiersTexte\Fichier\MoneyConversions.txt");

}

MessageBox.Show("Resultat d'echange d'argent : \n" + lirefichier, "Lyes Hadj Aissa");

LireFT.Close();

}

catch (FileNotFoundException)

{

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

}

catch (DirectoryNotFoundException)

{

MessageBox.Show(dossier + " not found.");

}

catch (IOException ex)

{

MessageBox.Show(ex.Message, "IO EXception");

}

finally

{

if (FichierTex != null)

FichierTex.Close();

}

}

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

{

}

}

class ConvertMoney

{

private double mon;

private double res;

public double Mon

{

get { return mon; }

set { mon = value; }

}

public ConvertMoney() { } // Les constructeurs

public ConvertMoney(double mon)

{

this.mon = mon;

}

public double CadToCad() // 25 Méthodes pour convertir de devise A à devise B.

{

res = mon \* 1;

return res;

}

public double CadToUsd()

{

res = mon \* 0.771;

return res;

}

public double CadToEuro()

{

res = mon \* 0.675;

return res;

}

public double CadToGbp()

{

res = mon \* 0.594;

return res;

}

public double CadToChf()

{

res = mon \* 0.741;

return res;

}

public double UsdToUsd()

{

res = mon \* 1;

return res;

}

public double UsdToCad()

{

res = mon \* 1.296;

return res;

}

public double UsdToEuro()

{

res = mon \* 0.875;

return res;

}

public double UsdToGpb()

{

res = mon \* 0.770;

return res;

}

public double UsdToChf()

{

res = mon \* 0.963;

return res;

}

public double EuroToEuro()

{

res = mon \* 1;

return res;

}

public double EuroToCad()

{

res = mon \* 1.4808;

return res;

}

public double EuroToUsd()

{

res = mon \* 1.142;

return res;

}

public double EuroToGbp()

{

res = mon \* 0.8805;

return res;

}

public double EuroToChf()

{

res = mon \* 1.097;

return res;

}

public double GbpToGbp()

{

res = mon \* 1;

return res;

}

public double GbpToCad()

{

res = mon \* 1.681;

return res;

}

public double GbpToUsd()

{

res = mon \* 1.296;

return res;

}

public double GbpToEuro()

{

res = mon \* 1.135;

return res;

}

public double GbpToChf()

{

res = mon \* 1.246;

return res;

}

public double ChfToChf()

{

res = mon \* 1;

return res;

}

public double ChfToCad()

{

res = mon \* 1.349;

return res;

}

public double ChfToUsd()

{

res = mon \* 1.0407;

return res;

}

public double ChfToGbp()

{

res = mon \* 0.802;

return res;

}

public double ChfToEuro()

{

res = mon \* 0.911;

return res;

}

}

}

* **Lotto Max:**

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

{

Random LotoMax = new Random();

int[] lotto = new int[8];

int NumeroLoto;

for (int i = 0; i < lotto.Length; i++)

{

NumeroLoto = LotoMax.Next(1, 49);

if (!lotto.Contains(NumeroLoto)) // méthode pour qu'il repete pas deux chiffre.

{

lotto[i] = NumeroLoto;

}

if (lotto.Contains(0))

{

lotto[i] = NumeroLoto;

}

}

txtBox1.Text = lotto[0].ToString() + "\r\n" + lotto[1].ToString()

+ "\r\n" + lotto[2].ToString() + "\r\n" + lotto[3].ToString() + "\r\n" + lotto[4].ToString()

+ "\r\n" + lotto[5].ToString() + "\r\n" + lotto[6].ToString() + "\r\n" + lotto[7].ToString();

txtBox1.ReadOnly = true;

// Ecrire dans un fichier texte

string dossier = @"..\FichiersTexte\Fichier\";

if (!Directory.Exists(dossier))

Directory.CreateDirectory(dossier);

string fichier = dossier + "LottoNumbers.txt";

FileStream Fichiertxt = null;

try

{

Fichiertxt = new FileStream(fichier, FileMode.Append, FileAccess.Write);

StreamWriter EcrireFT = new StreamWriter(Fichiertxt);

EcrireFT.Write("Max," + DateTime.Now + ", ");

EcrireFT.Write(lotto[0] + ", " + lotto[1] + ", " + lotto[2] + ", " + lotto[3] + ", " + lotto[4] + ", " + lotto[5] + ", " + lotto[6]);

EcrireFT.WriteLine(" Extra " + lotto[7] + ".");

EcrireFT.Close();

}

catch (FileNotFoundException)

{

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

}

catch (DirectoryNotFoundException)

{

MessageBox.Show(dossier + " not found.");

}

catch (IOException ex)

{

MessageBox.Show(ex.Message, "IO EXception");

}

finally

{

if (Fichiertxt != null)

Fichiertxt.Close();

}

}

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

{

string quitt;

quitt = Convert.ToString(MessageBox.Show("Voulez-vous \nfermer le formulaire?", "Quitter ?", MessageBoxButtons.YesNo));

if (quitt == "Yes")

{

this.Close();

}

}

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

{// Lire le fichier texte.

string dossier = @"..\FichiersTexte\Fichier\";

string fichier = dossier + "LottoNumbers.txt";

FileStream Fichiertxt = null;

try

{

Fichiertxt = new FileStream(fichier, FileMode.OpenOrCreate, FileAccess.Read);

StreamReader LireFT = new StreamReader(Fichiertxt);

string text = "";

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

{

string row = LireFT.ReadLine();

// MessageBox.Show(row);

string[] Lire = row.Split(',');

//MessageBox.Show(Lire.Length.ToString());

text += Lire[0] + ", " + Lire[1] + ", " + Lire[2] + ", " + Lire[3] + ", " + Lire[4] + ", " + Lire[5] + ", " + Lire[6] + ", " + Lire[7] + ", " + Lire[8] + "\n";

}

MessageBox.Show("Resultat du Lotto!! \n"+text,"Lyes Hadj Aissa" );

LireFT.Close();

}

catch (FileNotFoundException)

{

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

}

catch (DirectoryNotFoundException)

{

MessageBox.Show(dossier + " not found.");

}

catch (IOException ex)

{

MessageBox.Show(ex.Message, "IO EXception");

}

finally

{

if (Fichiertxt != null)

Fichiertxt.Close();

}

}

private void Loto\_Max\_Load(object sender, EventArgs e)

{

}

}

}

* **Lotto 6\_49 :**

namespace PROJET\_PROGRAM\_ORIENTE\_OBJET

{

public partial class Loto6\_49Form : Form

{

public Loto6\_49Form()

{

InitializeComponent();

}

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

{

string quitt;

quitt = Convert.ToString(MessageBox.Show("Voulez-vous \nfermer le formulaire?", "Quitter ?", MessageBoxButtons.YesNo));

if (quitt == "Yes")

{

this.Close();

}

}

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

{

Random Loto6\_49 = new Random();

int[] lotto = new int[7];

int NumeroLoto;// Loto6\_49.Next(1, 49);

for (int i = 0; i < lotto.Length; i++) {

NumeroLoto = Loto6\_49.Next(1, 49);

if (!lotto.Contains(NumeroLoto)) // Pour qu'il repete pas deux nombres

{

lotto[i] = NumeroLoto;

}

if (lotto.Contains(0))

{

lotto[i] = NumeroLoto;

}

}

txtBox1.Text= lotto[0].ToString()+"\r\n" + lotto[1].ToString()

+ "\r\n" + lotto[2].ToString() + "\r\n" + lotto[3].ToString() + "\r\n" + lotto[4].ToString()

+ "\r\n" + lotto[5].ToString() + "\r\n" + lotto[6].ToString();

txtBox1.ReadOnly = true;

// Ecrire dans le fichier txt.

string dossier = @"..\FichiersTexte\Fichier\";

if (!Directory.Exists(dossier))

Directory.CreateDirectory(dossier);

string fichier = dossier + "LottoNumbers.txt";

FileStream Fichiertxt = null;

try

{

Fichiertxt = new FileStream(fichier, FileMode.Append, FileAccess.Write);

StreamWriter EcrireFT = new StreamWriter(Fichiertxt);

EcrireFT.Write("649," + DateTime.Now + ", ");

EcrireFT.Write(lotto[0] + ", " + lotto[1] + ", " + lotto[2] + ", " + lotto[3] + ", " +

lotto[4] + ", " + lotto[5] + ", ");

EcrireFT.WriteLine(" Extra " + lotto[6]+"." );

EcrireFT.Close();

}

catch (FileNotFoundException)

{

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

}

catch (DirectoryNotFoundException)

{

MessageBox.Show(dossier + " not found.");

}

catch (IOException ex)

{

MessageBox.Show(ex.Message, "IO EXception");

}

finally

{

if (Fichiertxt != null)

Fichiertxt.Close();

}

}

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

{ // Lire le fichier texte

string dossier = @"..\FichiersTexte\Fichier\";

string fichier = dossier + "LottoNumbers.txt";

FileStream Fichiertxt = null;

try

{

Fichiertxt = new FileStream(fichier, FileMode.OpenOrCreate, FileAccess.Read);

StreamReader LireFT = new StreamReader(Fichiertxt);

string text = "";

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

{

string row = LireFT.ReadLine();

string[] Lire = row.Split(',');

text += Lire[0]+"," + Lire[1]+"," + " " + Lire[2] +", "+ Lire[3] + ", " + Lire[4] + ", " + Lire[5] + "," + Lire[6] + "," + Lire[7] + "," + " " + Lire[8]+"\n";

}

MessageBox.Show("Resultat du Lotto!! \n"+text,"Lyes Hadj Aissa");

LireFT.Close();

}

catch (FileNotFoundException)

{

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

}

catch (DirectoryNotFoundException)

{

MessageBox.Show(dossier + " not found.");

}

catch (IOException ex)

{

MessageBox.Show(ex.Message, "IO EXception");

}

finally

{

if (Fichiertxt != null)

Fichiertxt.Close();

}

}

private void Loto6\_49Form\_Load(object sender, EventArgs e)

{

}

}

}

1. **Les fonctions que j’ai créés et utilisées dans le projet.**

|  |  |
| --- | --- |
| **Nom Class/Méthode** | **Description** |
| 1. Class ConvertMoney() | Une classe qui contient des constructeurs et des propriétés. De plus, cette classe contient 25 méthodes, chaque méthode fait une conversion de devise A à devise B. |
| 1. Méthode doube CadToCad() | Cette méthode convertie l’argent de dollar canadien en dollar canadien. |
| 1. Méthode double CadToUsd() | Cette méthode convertie l’argent de dollar canadien en dollar américain. |
| 1. Méthode EuroToGbp() | Cette méthode convertie l’argent d’Euro en livre sterling. |
| 1. Les autres méthodes (25 méthodes) de la classe Convertmoney() | Tous les autres méthodes font la conversion de la devise A à une autre devise B. |
| 1. Class Calculate() | Une classe qui contient des constructeurs et des propriétés. Elle a quatre méthodes. |
| 1. Méthode public double Add() | Cette méthode additionne deux nombres et retourne le résultat. |
| 1. Méthode public double Sub() | Cette méthode soustrait deux nombres et retourne le résultat. |
| 1. Méthode public double Mul() | Cette méthode fait la multiplication de deux nombres et retourne le résultat. |
| 1. Méthode public double Div() | Cette méthode fait la division de deux chiffres et retourne le résultat. |
| 1. Class ConverTemp() | Une classe qui contient des constructeurs et des propriétés. Elle contient deux méthodes. |
| 1. Méthode double FarnhToCels() de la classe ConverTemp() | Cette méthode convertie la température de Fahrenheit à Celsius et retourne le résultat. |
| 1. Méthode double CelsToFarnh() de la classe ConverTemp() | Cette méthode convertie la température de Celsius à Fahrenheit et retourne le résultat. |