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

|  |
| --- |
| Your name: Kathleen Forgiarini da Silva  3/18/2023  Version: 1.4 |

1. **Start by adding a short description of your project, and the languages (technologies) used:**
2. Language: C#
3. Tools (IDE): Visual Studio 2022
4. **Present the print screens of yours forms, and have a detailed description of the functionalities (step by step).**
5. When initiate the application, you will see the dashboard. Interface gráfica do usuário, Aplicativo

   Descrição gerada automaticamente
6. If you click on Exit button, it will show a message to confirm if you want to exit.

Interface gráfica do usuário, Aplicativo, Site

Descrição gerada automaticamente

1. On tab “Generated Numbers”, it has two buttons with images, one button for Lotto Max and another one for Lotto649.
2. When click on Lotto Max button, it will open a dialog.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

1. Clicking on Generate button, it will generate 7 unique random numbers from 0-9 under the image, and another 8 random unique numbers from 1-49 in the textbox. In addition, a data will be written to a .txt file, with the name of the lottery, the date, and the numbers.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

1. Clicking on Read File button, it will read the file generated and show the data in a Message Box.

Texto

Descrição gerada automaticamente

1. Clicking on Exit button, it will ask if you want to exit this form. If yes, it will return to the dashboard.

Interface gráfica do usuário, Aplicativo, Site

Descrição gerada automaticamente

1. When click on Lotto649 button, it will open a dialog.

Interface gráfica do usuário, Aplicativo, Word

Descrição gerada automaticamente

1. Clicking on Generate button, it will generate 7 unique random numbers from 0-9 under the image, and another 8 random unique numbers from 1-49 in the textbox. In addition, a data will be written to a .txt file, with the name of the lottery, the date, and the numbers.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

1. Clicking on Read File button, it will read the file generated and show the data in a Message Box.

Texto

Descrição gerada automaticamente

1. Clicking on Exit button, it will ask if you want to exit this form. If yes, it will return to the dashboard.

Interface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto, Site

Descrição gerada automaticamente

1. On tab “Conversions”, it has two buttons with images, one button for Money Exchange and another one for Temperature Convert.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

1. Clicking on Money Exchange button, it will open the form to convert money.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

1. Clicking on Exit button, it will ask if you want to quit. If yes, it will show how much time you spent in this form.

Interface gráfica do usuário, Texto, Aplicativo, Site

Descrição gerada automaticamente

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

1. Clicking on Convert button it will validate the value entered and the selected currencies, and display the result. In addition, it will save the conversions into a .txt file.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

1. Clicking on Read File button, it will read the file generated and show the data in a Message Box.

Texto

Descrição gerada automaticamente

1. On tab “Simple Calculator”, it has one button with image to open the calculator.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

1. On tab “IP v4 validator”, it has one button with image to open the validator of IPs.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

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

**MAIN.CS**

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

{

if (MessageBox.Show("Are you sure you want to exit?", "Exit", MessageBoxButtons.YesNo).ToString() == "Yes")

{

Application.Exit();

}

}

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

{

LottoMax obj = new LottoMax();

obj.ShowDialog();

}

private void lotto649\_Click(object sender, EventArgs e)

{

Lotto649 obj = new Lotto649();

obj.ShowDialog();

}

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

{

MoneyExchange obj = new MoneyExchange();

obj.ShowDialog();

}

**LOTTOMAX.CS**

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

{

if (MessageBox.Show("Do you want to quit?", "Exit Lotto Max", MessageBoxButtons.YesNo).ToString() == "Yes")

{

this.Close();

}

}

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

{

Random labelRandom = new Random();

int labelRandomNumber = 0;

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

string labelNumbers = "";

while (labelUniqueNumbers.Count < 7)

{

labelRandomNumber = labelRandom.Next(0, 10);

if (!labelUniqueNumbers.Contains(labelRandomNumber))

{

labelUniqueNumbers.Add(labelRandomNumber);

labelNumbers += labelRandomNumber;

}

}

maxLabel.Text = labelNumbers;

Random boxRandom = new Random();

int boxRandomNumber = 0;

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

string boxNumbers = "";

while (boxUniqueNumbers.Count < 8)

{

boxRandomNumber = boxRandom.Next(1, 50);

if (!boxUniqueNumbers.Contains(boxRandomNumber))

{

boxUniqueNumbers.Add(boxRandomNumber);

boxNumbers += boxRandomNumber.ToString() + "\t\t";

}

}

maxTextbox.Text = boxNumbers;

//Text File

string fileName = @"LottoNbrs.txt";

using (FileStream fileStream = new FileStream(fileName, FileMode.Append))

using (StreamWriter writer = new StreamWriter(fileStream))

{

string lotteryName = "Max";

string dateTimeString = DateTime.Now.ToString("yyyy/MM/dd h:mm:ss tt");

int bonusNumber = boxUniqueNumbers[7];

writer.Write(lotteryName + ", " + dateTimeString + ", ");

for (int i = 0; i < boxUniqueNumbers.Count-1; i++)

{

writer.Write(boxUniqueNumbers[i]);

if (i != boxUniqueNumbers.Count - 1)

{

writer.Write(",");

}

}

writer.Write(" Bonus " + bonusNumber);

writer.WriteLine();

}

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

{

string fileName = "LottoNbrs.txt";

string fileContent = "";

using (StreamReader reader = new StreamReader(fileName))

{

fileContent = reader.ReadToEnd();

}

string message = fileContent;

string title = "Lottery Numbers by Kathleen Forgiarini";

MessageBox.Show(message, title);

}

}

**LOTTO649.CS**

private void l649Exit\_Click(object sender, EventArgs e)

{

if (MessageBox.Show("Do you want to quit?", "Exit Lotto649", MessageBoxButtons.YesNo).ToString() == "Yes")

{

this.Close();

}

}

private void l649Generate\_Click(object sender, EventArgs e)

{

Random labelRandom = new Random();

int labelRandomNumber = 0;

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

string labelNumbers = "";

while (labelUniqueNumbers.Count < 7)

{

labelRandomNumber = labelRandom.Next(0, 10);

if (!labelUniqueNumbers.Contains(labelRandomNumber))

{

labelUniqueNumbers.Add(labelRandomNumber);

labelNumbers += labelRandomNumber;

}

}

l649Label.Text = labelNumbers;

Random boxRandom = new Random();

int boxRandomNumber = 0;

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

string boxNumbers = "";

while (boxUniqueNumbers.Count < 7)

{

boxRandomNumber = boxRandom.Next(1, 49);

if (!boxUniqueNumbers.Contains(boxRandomNumber))

{

boxUniqueNumbers.Add(boxRandomNumber);

boxNumbers += boxRandomNumber.ToString() + "\t\t";

}

}

l649Textbox.Text = boxNumbers;

//Text File

string fileName = @"LottoNbrs.txt";

using (FileStream fileStream = new FileStream(fileName, FileMode.Append))

using (StreamWriter writer = new StreamWriter(fileStream))

{

string lotteryName = "649";

string dateTimeString = DateTime.Now.ToString("yyyy/MM/dd h:mm:ss tt");

int bonusNumber = boxUniqueNumbers[6];

writer.Write(lotteryName + ", " + dateTimeString + ", ");

for (int i = 0; i < boxUniqueNumbers.Count - 1; i++)

{

writer.Write(boxUniqueNumbers[i]);

if (i != boxUniqueNumbers.Count - 1)

{

writer.Write(",");

}

}

writer.Write(" Bonus " + bonusNumber);

writer.WriteLine();

}

}

private void l649Read\_Click(object sender, EventArgs e)

{

string fileName = "LottoNbrs.txt";

string fileContent = "";

using (StreamReader reader = new StreamReader(fileName))

{

fileContent = reader.ReadToEnd();

}

string message = fileContent;

string title = "Lottery Numbers by Kathleen Forgiarini";

MessageBox.Show(message, title);

}

**MONEYEXCHANGE.CS**

public partial class MoneyExchange : Form

{

private const double cadRate = 1.0;

private const double usdRate = 0.7289;

private const double eurRate = 0.6845;

private const double gbpRate = 0.5997;

private const double brlRate = 3.8265;

private DateTime startTime;

private DateTime endTime;

public MoneyExchange()

{

InitializeComponent();

}

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

{

if (MessageBox.Show("Do you want \nto quit the application \nMoney Exchange?", "Exit ?", MessageBoxButtons.YesNo).ToString() == "Yes")

{

this.Close();

}

}

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

{

double amount;

double result;

string amountStr = fromTextbox.Text.Trim();

if (Regex.IsMatch(amountStr, @"^\d{1,3}(,\d{3})\*(\.\d+)?$"))

{

try

{

amount = Convert.ToDouble(fromTextbox.Text);

}

catch (FormatException)

{

MessageBox.Show("Invalid input amount.");

return;

}

string from = GetSelectedCurrency(cad, usd, eur, gbp, brl);

if (from == null)

{

MessageBox.Show("Please select a FROM currency.");

return;

}

string to = GetSelectedCurrency(toCad, toUsd, toEur, toGbp, toBrl);

if (to == null)

{

MessageBox.Show("Please select a TO currency.");

return;

}

switch (to)

{

case "CAD":

result = amount \* cadRate / GetCurrencyRate(from);

break;

case "USD":

result = amount \* usdRate / GetCurrencyRate(from);

break;

case "EUR":

result = amount \* eurRate / GetCurrencyRate(from);

break;

case "GBP":

result = amount \* gbpRate / GetCurrencyRate(from);

break;

case "BRL":

result = amount \* brlRate / GetCurrencyRate(from);

break;

default:

MessageBox.Show("Invalid currency selection.");

return;

}

toTextbox.Text = result.ToString("0.00");

//Text File

string fileName = @"MoneyConv.txt";

using (FileStream fileStream = new FileStream(fileName, FileMode.Append))

using (StreamWriter writer = new StreamWriter(fileStream))

{

string dateTimeString = DateTime.Now.ToString("yyyy/MM/dd h:mm:ss tt");

writer.Write($"{amount} {from} = {Math.Round(result)} {to}, {dateTimeString}");

writer.WriteLine();

}

}

else

{

MessageBox.Show("Invalid amount entered. Please enter a valid amount with a comma or dot as decimal separator.");

}

}

private string GetSelectedCurrency(params RadioButton[] radioButtons)

{

foreach (RadioButton rb in radioButtons)

{

if (rb.Checked)

{

return rb.Text;

}

}

return null;

}

private double GetCurrencyRate(string currency)

{

switch (currency)

{

case "CAD":

return cadRate;

case "USD":

return usdRate;

case "EUR":

return eurRate;

case "GBP":

return gbpRate;

case "BRL":

return brlRate;

default:

return 0.0;

}

}

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

{

string fileName = "MoneyConv.txt";

string fileContent = "";

using (StreamReader reader = new StreamReader(fileName))

{

fileContent = reader.ReadToEnd();

}

string message = fileContent;

string title = "Money Exchange by Kathleen Forgiarini";

MessageBox.Show(message, title);

}

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

{

startTime = DateTime.Now;

}

private void MoneyExchange\_FormClosing(object sender, FormClosingEventArgs e)

{

endTime = DateTime.Now;

TimeSpan interval = endTime - startTime;

int totalSeconds = Convert.ToInt32(interval.TotalSeconds);

int totalMinutes = totalSeconds / 60;

totalSeconds %= 60;

if (totalMinutes >= 2)

{

MessageBox.Show($"You used the money exchange form for {totalMinutes} minutes and {totalSeconds} seconds", "Time Spent");

} else if (totalMinutes == 0)

{

MessageBox.Show($"You used the money exchange form for {totalSeconds} seconds", "Time Spent");

} else

{

MessageBox.Show($"You used the money exchange form for {totalMinutes} minute and {totalSeconds} seconds", "Time Spent");

}

}

}

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

|  |  |
| --- | --- |
| **Class/Method Name** | **Description** |
| 1. Main | This class has the dashboard |
| 1. mainExit\_Click | A method inside Main class to close the application |
| 1. lottoMax\_Click | A method inside Main class to open Lotto Max form |
| 1. lotto649\_Click | A method inside Main class to open Lotto649 form |
| 1. money\_Click | A method inside Main class to open MoneyExchange form |
| 1. LottoMax | This class has the Lotto Max form |
| 1. maxExit\_Click | A method inside LottoMax class to exit the form |
| 1. maxGenerate\_Click | A method inside LottoMax to generate random numbers and save it to a .txt file |
| 1. maxRead\_Click | A method inside LottoMax to read the .txt file generated and show the data in a Message Box |
| 1. Lotto649 | This class has the Lotto649 form |
| 1. l649Exit\_Click | A method inside the Lotto649 class to exit the form |
| 1. l649Generate\_Click | A method inside the Lotto649 to generate random numbers and save it to a .txt file |
| 1. l649Read\_Click | A method inside Lotto649 to read the .txt file generated and show the data in a Message Box |
| 1. MoneyExchange | This class has the Money Exchange form |
| 1. exit\_Click | A method inside MoneyExchange class to exit the form |
| 1. convert\_Click | A method inside MoneyExchange class to convert the amount entered into the selected currency and save the conversions in a txt file |
| 1. GetSelectedCurrency | A method inside MoneyExchange class, to get the value of the selected radio button |
| 1. GetCurrencyRate | A method inside MoneyExchange class, to get the exchange factor depending on which radio button is selected |
| 1. read\_Click | A method inside MoneyExchange class, to read the .txt file generated and show the data in a Message Box |
| 1. MoneyExchange\_Load | A method inside MoneyExchange class, to get the current time when the form is loaded |
| 1. MoneyExchange\_FormClosing | A method inside MoneyExchange class, to get the current time and calculate the difference between the time the form was loaded and closed, and display the result to the user |

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

* In the generated numbers forms, I had a little bit of hard time figuring out how to get only unique numbers to display. But it was easy to design the layout and to the display.
* I had some difficult to create the text file and read it, because I didn’t know about this function and it was my first time using the objects FileStream, StreamReader and StreamWriter.
* In money exchange, I had some difficult to get the value of the selected radio buttons because I have never used this input before. But it was ok to do the layout, the math conversions and display the results.