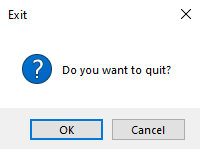
|  |
| --- |
| College LaSalle |
| Project - Oriented Object Programming User and Technical Manual |
|  |
| Presented to: Mihai Maftei |

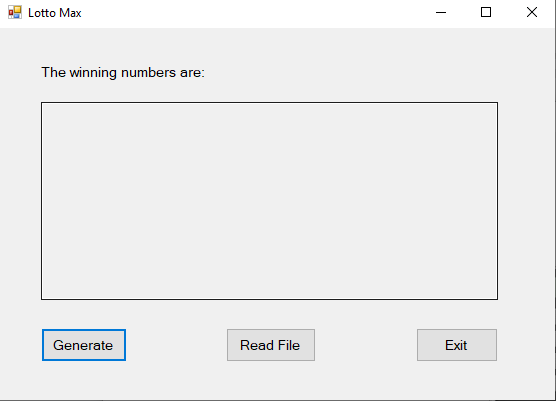
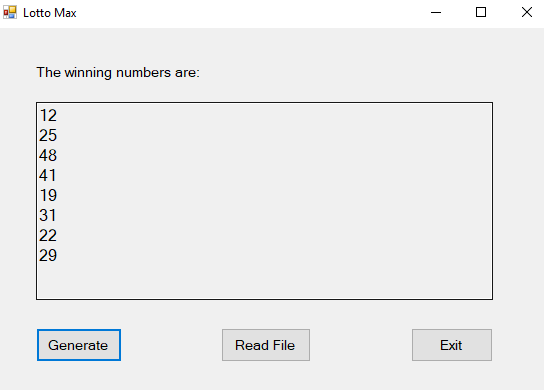
|  |
| --- |
| Hoang Nhan Duyen, Nguyen  4/15/2020 |

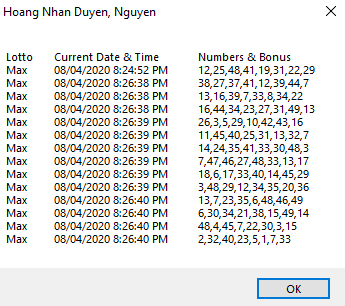


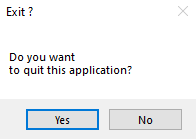
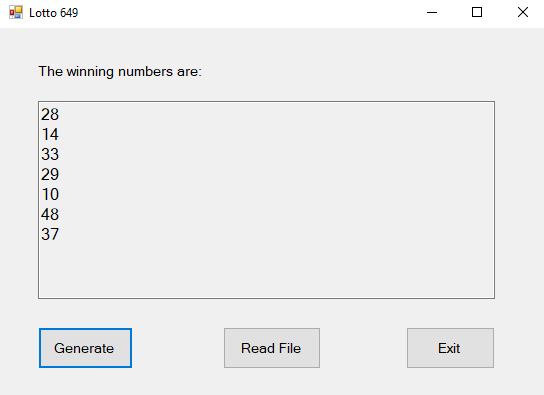
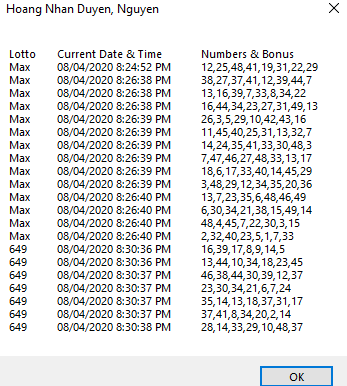
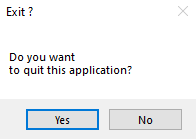
1. Start by adding a short description of your project, and the languages (technologies) used:
2. Language: C#
3. Tool: Visual Studio 2019
4. Present the print screens of yours forms, and have a detailed description of the functionalities (step by step).



1. On the **Dashboard menu**, if click on the **Exit** button, a message box will show up and ask whether the user want to quit this application or not. If the “OK” button is clicked, the application will disappear, if the “Cancel” button is clicked, the message box will disappear, and the application remains working.
2. On the **Dashboard menu**, if click on the **Lotto Max button** (the first button from the left), another form will show up.
3. If click on the Generate button, **8 unique** numbers which are less than **50** will be presented into a multiline read only text box above the button and all the numbers also be saved into a text file **LottoNbrs.txt** in a folder Lotto. This folder is in the same folder as the user’s execute file.

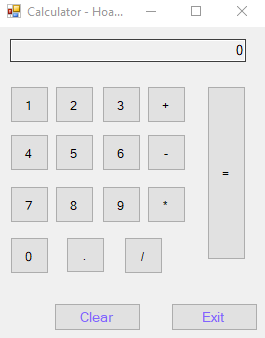




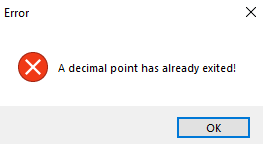
1. If click on the button Read File, the text file content which includes the name of lotto, the current date and time, **7 digits and a bonus one** will be showed by using a message box of maximum 40 lines.
2. If click on the Exit button, a message box will show up and ask whether the user want to quit this form or not. If the “Yes” button is clicked, the Lotto Max form will disappear but not the Dashboard, if the “No” button is clicked, the message box will disappear, and Lotto Max form remains working.
3. On the **Dashboard menu**, if click on the **Lotto 649 button** (the second button from the left), another form will show up.
4. If click on the Generate button, **7 unique numbers** which are less than **49** will be presented into a multiline read only text box above the button and all the numbers also be saved into a text file **LottoNbrs.txt**. This folder is in the same folder as the user’s execute file.
5. If click on the button Read File, the text file content which includes the name of lotto, the current date and time**, 6 digits and a bonus one** will be showed by using a message box.
6. If click on the Exit button, a message box will show up and ask whether the user want to quit this application or not. If the “Yes” button is clicked, the Lotto 649 form will disappear but not the Dashboard, if the “No” button is clicked, the message box will

disappear, and **Lotto 649** application remains working.

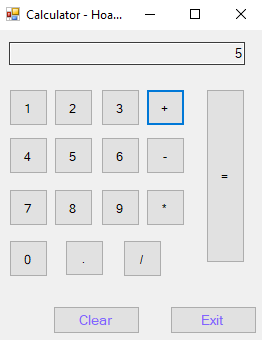
1. On the **Dashboard menu**, if click on the **Calculator button** (the third button), another form will show up.



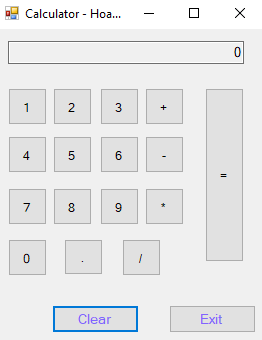
1. When the user clicks on a button from 0 to 9, the number(s) will be presented on the **text box**.



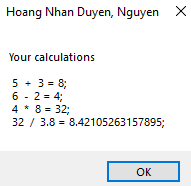
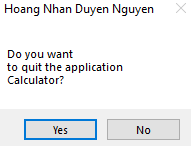
1. The “.” button presents the decimal button, if a decimal button is already existed, a message box will show up an error.



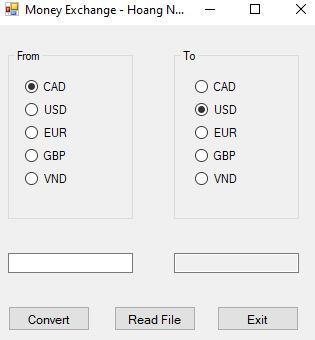
1. The four operator buttons “+,-,\*,/” will store the operand(s) in appropriate variable(s) of the Calculator class. If the result of previous has not been showed on the text box yet, the buttons will present the result on the text box and store the operator in the variable of the class.
2. If the Equal button “=” is clicked, the result appears on the **text box**. The calculation will be also saved into a text file **Calculator.txt**.

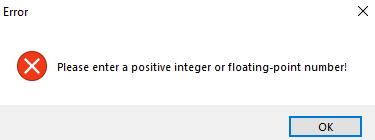


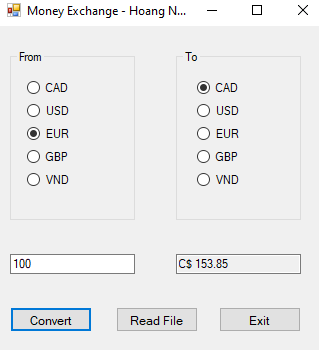
1. The Clear button will clear all the fields’ members of the Calculator class, and the text box.
2. The Exit button is clicked, all the arithmetical operations and theirs results appear on the message box(s). If the “Ok” or “x” **button** is clicked, another message box will show up and ask whether the user want to quit this form or not. If the “Yes” button is clicked, the **Calculator** application will disappear but not the Dashboard, if the “No” button is clicked, the message box will disappear, and **Calculator** application remains working.

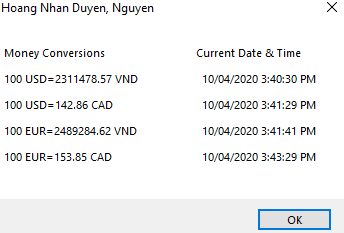


1. On the **Dashboard menu**, if click on the **Money Exchange** another form will show up.
2. The application allows the user to select on one Radio buttonin **FROM** and one Radio button in **TO**.

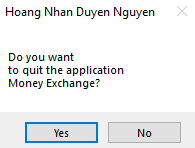


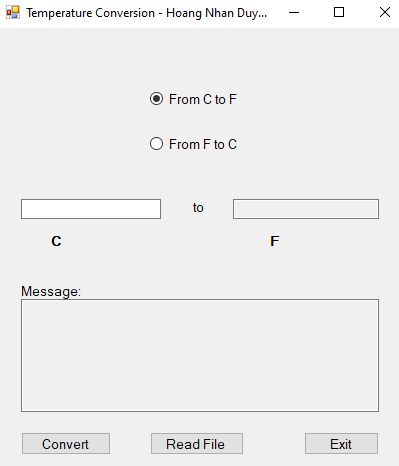
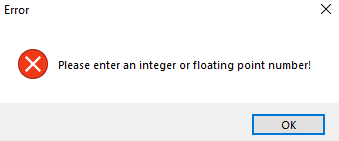
1. Then, the user types the amount of money into the **left text box**.
2. If a letter is typed, an error will show up on a message box when the Convert button is clicked.
3. When the user enters the proper value into the **left text box**, the **right text box** will display the value, which is converted.

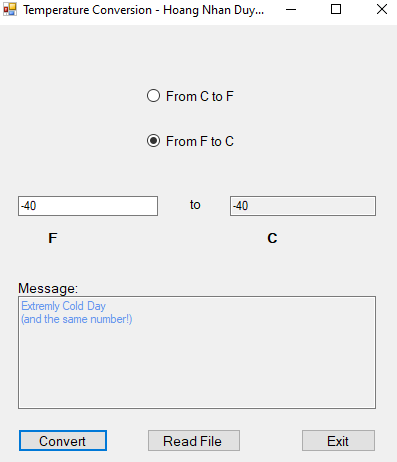


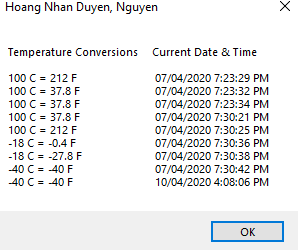
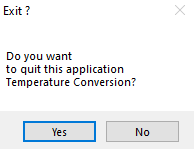
1. If the **Read File** button is pressed, the user can see the list of exchanging money that all of transactions have been converted since the application activated.

The message box will show the currency from / to and the current date and time and press **OK** or **“x”** button to close the box.

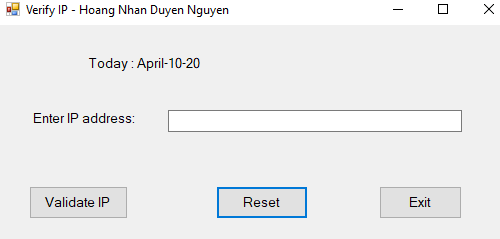


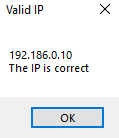
1. If click on the Exit button, a message box will show up and ask whether the user want to quit this application or not. If the “Yes” button is clicked, the **Money Exchange** will disappear but not the Dashboard, if the “No” button is clicked, the message box will disappear, and **Money Exchange** form remains working.
2. On the **Dashboard menu**, if click the **Temperature Convert** another form will show up.
   1. To use the application, the user needs to select one of two Radio Button to decide which degree the user wants to convert (C 🡪 F or F 🡪 C).
   2. Then, type the degree into the **text box** and press **Convert button.** If the user types the degree which includes **any special characters (A 🡪 Z, °C,** **°F …)** the form will show an error message:

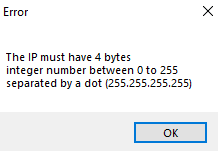


* 1. When the user enters the proper value into the **left text box**, the **right text box** will display the value, which is converted. Then, it will show a message in the Read only message box.
  2. Press **Read File button,** the user can see **the list of** all the conversion have been saved since the form activated. The message box will show the conversion from / to and the current date and time and press **OK or “x” button** to close the box.
  3. If click on the Exit button, a message box will show up and ask whether the user wants to quit this application or not. If the “Yes” button is clicked, the **Temperature Conversion** will disappear but not the Dashboard, if the “No” button is clicked, the message box will disappear, and **Temperature Conversion** form remains working.

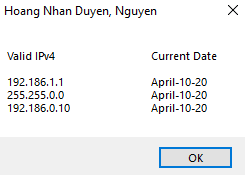
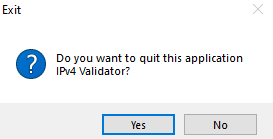
1. On the **Dashboard menu**, if click the **IPv4 Validator** another form will show up.



* 1. To use this application, the user types an IPv4 address into the text box and click on the **Validate IP button**.
  2. If the **input IP** is valid, a message box shows up to inform that the IP is correct.



* 1. If the **input IP** is invalid or the text box is empty, an error message box appears.
  2. If the **Reset button** is clicked, the text box will be clear.
  3. The **Exit button** is clicked, all the date and the correct IPv4 addresses appear on the message box(s). When the “Ok” or “x” button is clicked, another message box will show up and ask whether the user want to quit this application or not. If the “Yes” button is clicked, the **IPv4 Validator** application will disappear but not the Dashboard, if the “No” button is clicked, the message box will disappear, and **IPv4 Validator** form remains working.

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

**FORM 0: The Dash board**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Project\_420\_CT2\_AS

{

public partial class Form0 : Form

{

public Form0()

{

InitializeComponent();

}

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

{

if (MessageBox.Show("Do you want to quit?", "Exit",

MessageBoxButtons.OKCancel, MessageBoxIcon.Question).ToString() == "OK")

{

Application.Exit();

}

}

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

{

var lottoMax = new Form1();

lottoMax.Show();

}

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

{

var lotto649 = new Form2();

lotto649.Show();

}

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

{

var cal = new Form3();

cal.Show();

}

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

{

var moneyEx = new Form4();

moneyEx.Show();

}

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

{

var temCon = new Form5();

temCon.Show();

}

private void btnIPv4Val\_Click(object sender, EventArgs e)

{

var ipv4Val = new Form6();

ipv4Val.Show();

}

}

}

**Form 1: The Lotto Max**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.IO;

using System.Text.RegularExpressions;

namespace Project\_420\_CT2\_AS

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

if (MessageBox.Show("Do you want \nto quit this application? ", "Exit ?",

MessageBoxButtons.YesNo).ToString() == "Yes")

{

this.Close();

}else { btnGen.Focus(); }

}

Lotto obj; //Global object

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

{

obj = new Lotto();

}

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

{

string newLine = Environment.NewLine; // Create a new line in multiple text box

obj.LottoMax("Max");

string[] arrNum = obj.ArrNum;

txtResult.Text = arrNum[0];

for (int i = 1; i <= 6; i++) // For loop to generate and print out the

second, third, ... sixth random numbers

{

txtResult.Text = txtResult.Text + newLine + arrNum[i]; // Print them with

the new line

}

txtResult.Text = txtResult.Text + newLine + arrNum[7];

}

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

{

string dirPath = @".\Lotto\";

FileStream fs = null;

string filePath = dirPath + "LottoNbrs.txt";

try

{

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

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

StreamReader textIn = new StreamReader(fs);

byte count = 0;

string textToPrint = "Lotto\tCurrent Date & Time\tNumbers & Bonus\n";

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

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

{

string row = textIn.ReadLine();

row = row.Replace("Bonus ", ",");

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

textToPrint += columns[0] + "\t" + columns[1] + "\t" + columns[2] +

columns[3] + "\n";

count += 1;

if (count == 40)

{

MessageBox.Show(textToPrint);

count = 0;

textToPrint = "";

}

}

if (count != 0)

{

MessageBox.Show(textToPrint, "Hoang Nhan Duyen, Nguyen");

}

// close the input stream for the text file

textIn.Close();

}

catch (IOException ex)

{

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

}

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

}

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

{

}

}

}

**Form 2: The Lotto 649**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.IO;

using System.Text.RegularExpressions;

namespace Project\_420\_CT2\_AS

{

public partial class Form2 : Form

{

public Form2()

{

InitializeComponent();

}

Lotto obj;

private void Form2\_Load(object sender, EventArgs e)

{

obj = new Lotto();

}

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

{

string newLine = Environment.NewLine;//Create a new line in multiple text box

obj.Lotto649("649",49);

string[] arrNum = obj.ArrNum;

txtResult.Text = arrNum[0];

for (int i = 1; i <= 5; i++) // For loop to generate and print out the

second, third, ... sixth random numbers

{

txtResult.Text = txtResult.Text + newLine + arrNum[i]; // Print them with

the new line

}

txtResult.Text = txtResult.Text + newLine + arrNum[6];

}

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

{

if (MessageBox.Show("Do you want \nto quit this application? ", "Exit ?",

MessageBoxButtons.YesNo).ToString() == "Yes")

{

this.Close();

}

}

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

{

string dirPath = @".\Lotto\";

FileStream fs = null;

string filePath = dirPath + "LottoNbrs.txt";

try

{

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

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

StreamReader textIn = new StreamReader(fs);

byte count = 0;

string textToPrint = "Lotto\tCurrent Date & Time\tNumbers & Bonus\n";

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

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

{

string row = textIn.ReadLine();

row = row.Replace("Bonus ", ",");

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

textToPrint += columns[0] + "\t" + columns[1] + "\t" + columns[2] +

columns[3] +"\n";

count += 1;

if (count == 40)

{

MessageBox.Show(textToPrint);

count = 0;

textToPrint = "";

}

}

if (count != 0)

{

MessageBox.Show(textToPrint, "Hoang Nhan Duyen, Nguyen");

}

// close the input stream for the text file

textIn.Close();

}

catch (IOException ex)

{

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

}

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

}

}

}

**Form 3: Simple calculator application**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.IO;

using System.Text.RegularExpressions;

namespace Project\_420\_CT2\_AS

{

public partial class Form3 : Form

{

Boolean reset = false; // check if the whole calculation is done and user want to

do another calculation without clicking on clear button

public Form3()

{

InitializeComponent();

}

Calculator obj; // Gobal variable

private void btn\_click(object sender, EventArgs e)

{

if (txtCal.Text == "0" || reset ||obj.Fini) // if the textbox show 0 or user

want to reset the calculator or all the operands are not null

{

txtCal.Clear();

reset = false;

obj.Fini = false;

}

Button btn = (Button)sender;

txtCal.Text = txtCal.Text + btn.Text.Substring(1,1); // print the numbers

from the buttons without "&"

}

private void op\_click(object sender, EventArgs e)

{

Button btn = (Button)sender;

obj.CurrentVal = Convert.ToDouble(txtCal.Text); // set the first value for

operand1

obj.Setup(); // set the values for operand1 or operand2

if (obj.Fini) // check if the two operands are different from 0

{

txtCal.Text = obj.Equals(Convert.ToDouble(txtCal.Text)).ToString();

//pass the value to operand2 and calculate the result

obj.Op = btn.Text; //set the value for the operation

reset = true;

}

else

{

obj.Op = btn.Text; //if obj.Fini = false, operand2 has not been passed

the value so just pass the value of the button to the operation

txtCal.Clear();

}

}

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

{

txtCal.Text = "0";

obj.Clear();

}

private void Form3\_Load(object sender, EventArgs e)

{

obj = new Calculator();

}

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

{

string dirPath = @".\Calculator\";

FileStream fs = null;

string filePath = dirPath + "Calculator.txt";

try

{

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

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

StreamReader textIn = new StreamReader(fs);

byte count = 0;

string textToPrint = "Your calculations\n\n";

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

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

{

string row = textIn.ReadLine();

textToPrint += row + "\n";

count += 1;

if (count == 30)

{

MessageBox.Show(textToPrint);

count = 0;

textToPrint = "";

}

}

if (count != 0)

{

MessageBox.Show(textToPrint, "Hoang Nhan Duyen, Nguyen");

}

// close the input stream for the text file

textIn.Close();

}

catch (IOException ex)

{

MessageBox.Show(ex.Message, "IOException", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

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

if (MessageBox.Show("Do you want\nto quit the application\nCalculator?",

"Hoang Nhan Duyen Nguyen", MessageBoxButtons.YesNo).ToString() == "Yes")

{

this.Close();

}

else

{

txtCal.Focus();

}

}

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

{

if (!txtCal.Text.Contains(".")) //check if the decimal point already exits

{

if (txtCal.Text == "") //if the textbox is empty

{

txtCal.Text = "0" + "."; // 0. will appear in the textbox

}else

{

txtCal.Text = txtCal.Text + "."; //if the text box has numbers, only add "."

}

}else if (reset && txtCal.Text.Contains(".")) // if the previous calculation

has been done and the user want to start with 0. but the user does not click on 0

{

txtCal.Text = "0" + ".";

}else

{

MessageBox.Show("A decimal point has already exited!","Error",MessageBoxButtons.OK,MessageBoxIcon.Error);// wrong input message

}

}

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

{

txtCal.Text = obj.Equals(Convert.ToDouble(txtCal.Text)).ToString(); // do the

calculation

obj.Clear(); //then clear the value of operands and op

reset = true; // reset to get ready for the new calcultion

}

}

}

**Form 4: The Money Exchange application**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.IO;

using System.Text.RegularExpressions;

using System.Runtime.Serialization;

namespace Project\_420\_CT2\_AS

{

public partial class Form4 : Form

{

public string inTo = “CAD”, inFrom = “CAD” , toSign = “C$”;

public Form4()

{

InitializeComponent();

}

Exchange obj;

private void Form4\_Load(object sender, EventArgs e)

{

obj = new Exchange();

}

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

{

if (MessageBox.Show("Do you want\nto quit the application\nMoney Exchange?",

"Hoang Nhan Duyen Nguyen", MessageBoxButtons.YesNo).ToString() == "Yes")

{

this.Close();

}

else

{

txtEnVal.Focus();

}

}

private void From\_click(object sender, EventArgs e)

{

RadioButton click = (RadioButton)sender;

inFrom = click.Text;

txtEnVal.Clear();

txtConVal.Clear();

}

private void To\_click(object sender, EventArgs e)

{

RadioButton click = (RadioButton)sender;

inTo = click.Text;

switch (inTo)

{

case "CAD":

toSign = "C$";

break;

case "USD":

toSign = "$";

break;

case "EUR":

toSign = "€";

break;

case "GBP":

toSign = "£";

break;

case "VND":

toSign = "₫";

break;

}

}

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

{

if (inTo == inFrom)

{

MessageBox.Show("Same choice!","Error", MessageBoxButtons.OK,

MessageBoxIcon.Error);

}

if (inTo is null)

{

MessageBox.Show("Please choose the currency you want to exchange!",

"Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else

{

try

{

Regex reg = new Regex(@"^([0-9]+)([\.]?)([0-9]\*)$");

if (reg.IsMatch(txtEnVal.Text) == true)

{

txtConVal.Text = toSign + " " + obj.MonExchange(txtEnVal.Text,

inFrom, inTo).ToString();

}

else

{

throw new NotAPositiveIntegerOrFloatingPointNumberException

("Please enter a positive integer or floating-point number!");

}

}

catch (NotAPositiveIntegerOrFloatingPointNumberException ex)

{

MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK,

MessageBoxIcon.Error);

}

}

}

private void raBtnCAD1\_CheckedChanged(object sender, EventArgs e)

{

}

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

{

string dirPath = @".\MoneyExchange\";

FileStream fs = null;

string filePath = dirPath + "MoneyConversions.txt";

try

{

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

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

StreamReader textIn = new StreamReader(fs);

byte count = 0;

string textToPrint = "Money Conversions\t\tCurrent Date & Time\n\n";

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

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

{

string row = textIn.ReadLine();

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

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

count += 1;

if (count == 30)

{

MessageBox.Show(textToPrint);

count = 0;

textToPrint = "";

}

}

if (count != 0)

{

MessageBox.Show(textToPrint, "Hoang Nhan Duyen, Nguyen");

}

// close the input stream for the text file

textIn.Close();

}

catch (IOException ex)

{

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

}

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

}

}

[Serializable]

internal class NotAPositiveIntegerOrFloatingPointNumberException : Exception

{

public NotAPositiveIntegerOrFloatingPointNumberException()

{

}

public NotAPositiveIntegerOrFloatingPointNumberException(string message) : base(message)

{

}

public NotAPositiveIntegerOrFloatingPointNumberException(string message, Exception innerException) : base(message, innerException)

{

}

protected NotAPositiveIntegerOrFloatingPointNumberException(SerializationInfo info, StreamingContext context) : base(info, context)

{

}

}

}

**Form 5: The Temperature application**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Runtime.Serialization;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.IO;

using System.Text.RegularExpressions;

namespace Project\_420\_CT2\_AS

{

public partial class Form5 : Form

{

string choice = “C to F”;

Temperature obj;

public Form5()

{

InitializeComponent();

}

private void Form5\_Load(object sender, EventArgs e)

{

obj = new Temperature();

}

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

{

try

{

double enVal;

txtMess.Clear();

Regex reg = new Regex(@"^[\-]?[0-9]+[\.]?[0-9]\*$");

if (reg.IsMatch(txtFrom.Text) == true)

{

enVal = Convert.ToDouble(txtFrom.Text.Trim());

obj.EnVal = enVal;

txtTo.Text = obj.TemCon(choice).ToString();

if ((enVal == 100 && choice =="From C to F") || (enVal == 212 && choice == "From F to C"))

{

txtMess.Text = "Water boils";

txtMess.ForeColor = Color.Red;

txtMess.BackColor = txtMess.BackColor;

}

if ((enVal == 40 && choice == "From C to F")|| (enVal == 104 && choice == "From F to C"))

{

txtMess.Text = "Hot Bath";

txtMess.ForeColor = Color.Red;

txtMess.BackColor = txtMess.BackColor;

}

if ((enVal == 37 && choice == "From C to F") || (enVal == 98.6 && choice == "From F to C"))

{

txtMess.Text = "Body temperature";

txtMess.ForeColor = Color.DarkGreen;

txtMess.BackColor = txtMess.BackColor;

}

if ((enVal == 30 && choice == "From C to F")|| (enVal == 86 && choice == "From F to C"))

{

txtMess.Text = "Beach weather";

txtMess.ForeColor = Color.DarkGreen;

txtMess.BackColor = txtMess.BackColor;

}

if (((Math.Abs(enVal - 21) <= 0.01) && choice == "From C to F") || ((Math.Abs(enVal - 70) <= 0.01) && choice == "From F to C"))

{

txtMess.Text = "Room temperature";

txtMess.ForeColor = Color.DarkGreen;

txtMess.BackColor = txtMess.BackColor;

}

if ((enVal == 10 && choice == "From C to F") || (enVal == 50 && choice == "From F to C"))

{

txtMess.Text = "Cool Day";

txtMess.ForeColor = Color.CornflowerBlue;

txtMess.BackColor = txtMess.BackColor;

}

if ((enVal == 0 && choice == "From C to F")|| (enVal == 32 && choice == "From F to C"))

{

txtMess.Text = "Freezing point of water";

txtMess.ForeColor = Color.CornflowerBlue;

txtMess.BackColor = txtMess.BackColor;

}

if (((Math.Abs(enVal - (-18)) <= 0.01) && choice == "From C to F") || ((Math.Abs(enVal - 0) <= 0.01) && choice == "From F to C"))

{

txtMess.Text = "Very Cold Day";

txtMess.ForeColor = Color.CornflowerBlue;

txtMess.BackColor = txtMess.BackColor;

}

if (enVal == -40)

{

txtMess.Text = "Extremly Cold Day" + Environment.NewLine + "(and the same number!)";

txtMess.ForeColor = Color.CornflowerBlue;

txtMess.BackColor = txtMess.BackColor;

}

}

else

{

throw new NotAnIntegerOrFloatingPointNumberException("Please enter an integer or floating point number!");

}

}

catch (NotAnIntegerOrFloatingPointNumberException ex)

{

MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

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

{

string dirPath = @".\Temperature\";

FileStream fs = null;

string filePath = dirPath + "TempConvert.txt";

try

{

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

StreamReader textIn = new StreamReader(fs);

byte count = 0;

string textToPrint = "Temperature Conversions\tCurrent Date & Time\n\n";

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

{

string row = textIn.ReadLine();

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

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

count += 1;

if (count == 20)

{

MessageBox.Show(textToPrint);

count = 0;

textToPrint = "";

}

}

if (count != 0)

{

MessageBox.Show(textToPrint, "Hoang Nhan Duyen, Nguyen");

}

textIn.Close();

}

catch (IOException ex)

{

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

}

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

}

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

{

if (MessageBox.Show("Do you want\nto quit this application \nTemperature Conversion?", "Exit ?", MessageBoxButtons.YesNo).ToString() == "Yes")

{

this.Close();

}

}

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

{

leftLable.Text = "C";

rightLable.Text = "F";

RadioButton btn = (RadioButton)sender;

choice = btn.Text.Trim();

}

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

{

leftLable.Text = "F";

rightLable.Text = "C";

RadioButton btn = (RadioButton)sender;

choice = btn.Text.Trim();

}

}

[Serializable]

internal class NotAnIntegerOrFloatingPointNumberException : Exception

{

public NotAnIntegerOrFloatingPointNumberException()

{

}

public NotAnIntegerOrFloatingPointNumberException(string message) : base(message)

{

}

public NotAnIntegerOrFloatingPointNumberException(string message, Exception innerException) : base(message, innerException)

{

}

protected NotAnIntegerOrFloatingPointNumberException(SerializationInfo info, StreamingContext context) : base(info, context)

{

}

}

}

**Form 6: The IP4 Validator application**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Runtime.Serialization;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.IO;

using System.Text.RegularExpressions;

namespace Project\_420\_CT2\_AS

{

public partial class Form6 : Form

{

IPv4 obj;

public Form6()

{

InitializeComponent();

}

private void Form6\_Load(object sender, EventArgs e)

{

obj = new IPv4();

lblDate.Text = "Today : " + obj.Today();

}

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

{

try

{

obj.EnIPv4 = txtIpv4.Text.Trim();

if (obj.Validation() == "correct")

{

MessageBox.Show(obj.EnIPv4 + "\nThe IP is correct", "Valid IP");

}

else

{

throw new ErrorInvalidIP("The IP must have 4 bytes\ninteger number

between 0 to 255" + "\nseparated by a dot (255.255.255.255)");

}

}catch (ErrorInvalidIP ex)

{

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

}

}

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

{

txtIpv4.Clear();

}

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

{

string dirPath = @".\IPv4\";

FileStream fs = null;

string filePath = dirPath + "IPv4Validation.txt";

try

{

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

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

StreamReader textIn = new StreamReader(fs);

byte count = 0;

string textToPrint = "Valid IPv4\t\tCurrent Date\n\n";

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

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

{

string row = textIn.ReadLine();

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

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

count += 1;

if (count == 40)

{

MessageBox.Show(textToPrint);

count = 0;

textToPrint = "";

}

}

if (count != 0)

{

MessageBox.Show(textToPrint, "Hoang Nhan Duyen, Nguyen");

}

// close the input stream for the text file

textIn.Close();

}

catch (IOException ex)

{

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

}

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

if(MessageBox.Show("Do you want to quit this application\nIPv4 Validator?",

"Exit", MessageBoxButtons.YesNo, MessageBoxIcon.Question).ToString() == "Yes")

{

this.Close();

}

}

}

[Serializable]

internal class ErrorInvalidIP : Exception

{

public ErrorInvalidIP()

{

}

public ErrorInvalidIP(string message) : base(message)

{

}

public ErrorInvalidIP

(string message, Exception innerException) : base(message, innerException)

{

}

protected ErrorInvalidIP

(SerializationInfo info, StreamingContext context) : base(info, context)

{

}

}

}

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

|  |  |
| --- | --- |
| **Class/Method Name** | **Description** |
| 1. class Lotto | A class with fields, constructor, overloaded methods to generate 7 + 1 random unique numbers out of 50 for the game Lotto Max, and 6 + 1 random unique numbers out of 49 for the game Lotto 649. Then write them in a text file. |
| 1. public void LottoMax (string name) | Overloading method which genrate and write into the text file LottoNbrs.txt for MAX lottery 7+1 random unique numbers out of 50. |
| 1. public void Lotto649(string name, int nbr) | Overloading methods which generate, then write to the text file LottoNbrs.txt for 649 lottery 6+1 random unique numbers out of 49. |
| 1. class Calculator | Perform 4 basic operations, and save all the arithmetical operations and theirs results into a text file Calculator.txt. |
| 1. public void Setup() | Decide where the value should pass to operand1 or operand2. |
| 1. public double Equals(double displayVal) | Sets the operand2 field to the value that’s passed to it. Then, performs the operation specified by the op field on the operand1 and operand2 fields, and stores the result in the operand1 field. Write the operations into text file. |
| 1. public void Clear() | Sets the private fields to their default values. |
| 1. class Exchange | Calculate the results for possible conversions between five different currencies, then save the conversion into a text file MoneyConversions.txt. |
| 1. private double Value(string enCur) | Calculate the value of two currencies and return the value. |
| 1. public double MonExchange (string enVal, string inFrom, string inTo) | Calculate the results for the conversions and write them into MoneyConversions.txt text file. |
| 1. class Temperature | Calculate the results for possible temperature conversions. Save the temperature conversion into a text file TempConvert.txt |
| 1. public double Convert(string choice) | Convert the temperature from C to F or from F to C and return the result. |
| 1. public double TemCon(string choice) | Write the temperature conversions into TempConvert.txt text file. |
| 1. class IPv4 | Test the input IP address from the user and validate if it is valid or not, then save the the correct IPv4 addresses into a text file. |
| 1. public string Validation() | Perform the validation process to the input IPv4 and write them into IPv4Validation.txt text file. |

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

I had some difficulties in creating the class for the Calculator application and how to generate unique random number for Lotto application.

The hardest part was making the Calculator class and the easiest one is making the design for the forms.