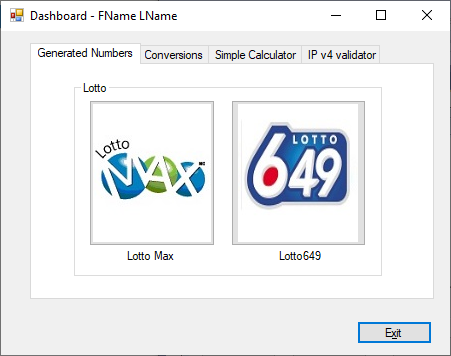
|  |
| --- |
| College LaSalle |
| Project - Oriented Object Programming User and Technical Manual |
|  |
| Presented to: Mihai Maftei. |

Your name: Christian Denis Marcelin

7/24/2023

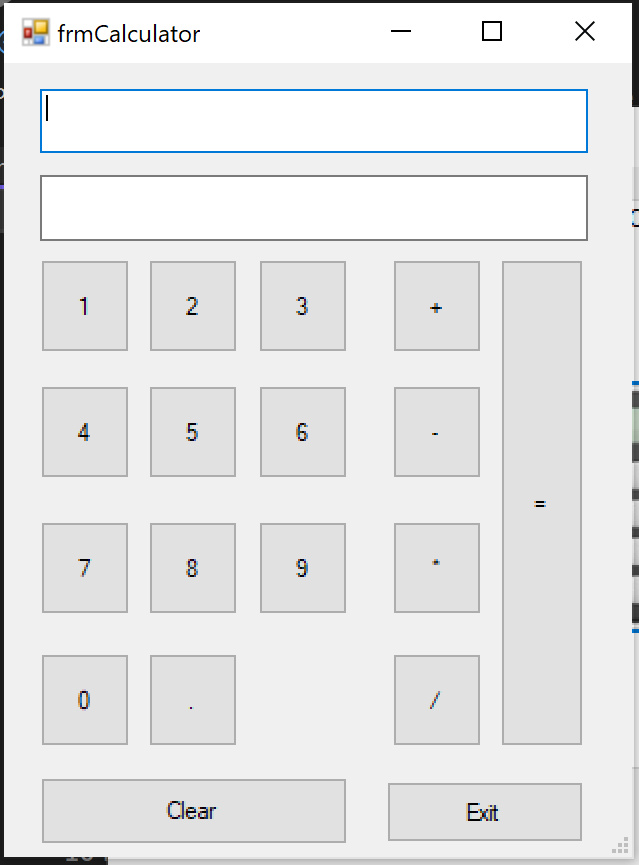
Version 1.0

1. **Start by adding a short description of your project, and the languages (technologies) used:**
2. Language : C#
3. Used tool(s): Visual Studio version 2022
4. **Present the print screens of yours forms, and have a detailed description of the functionalities (step by step).**

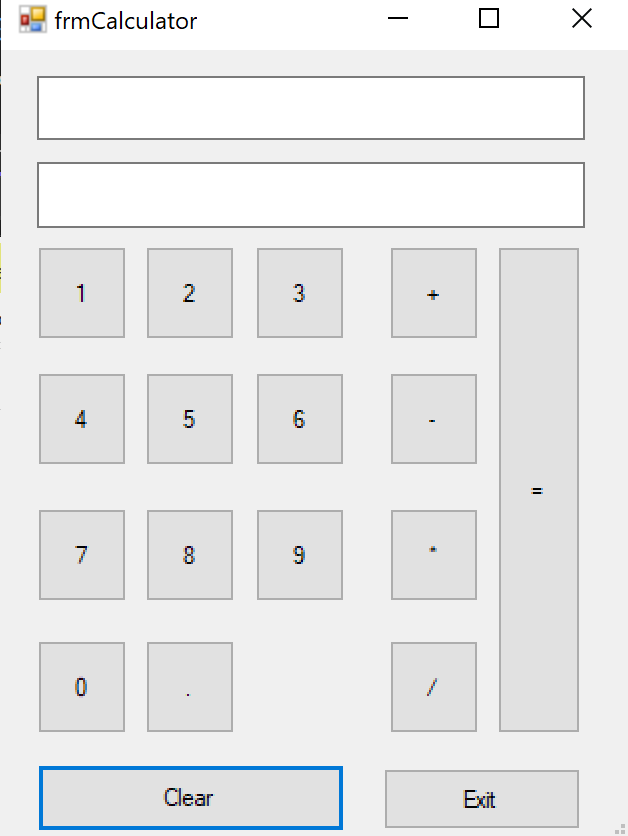


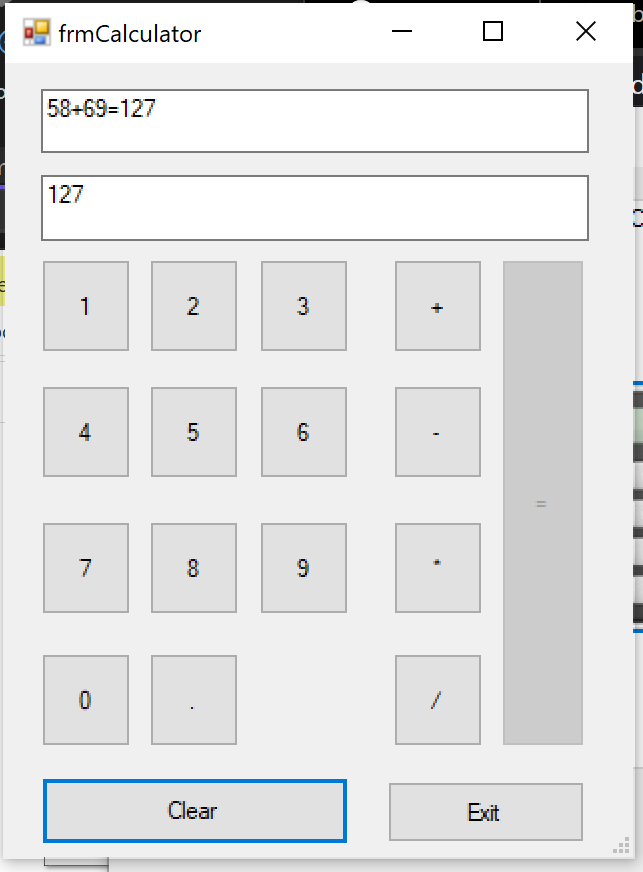
1. If you click on tab…

2.



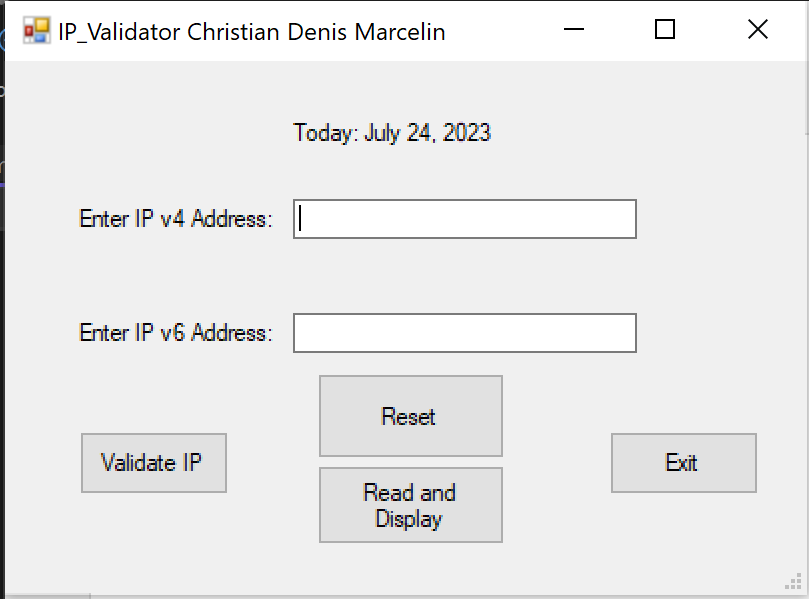
1. If you input an operation to do it does it.
2. If you click on clear the text boxes are cleared

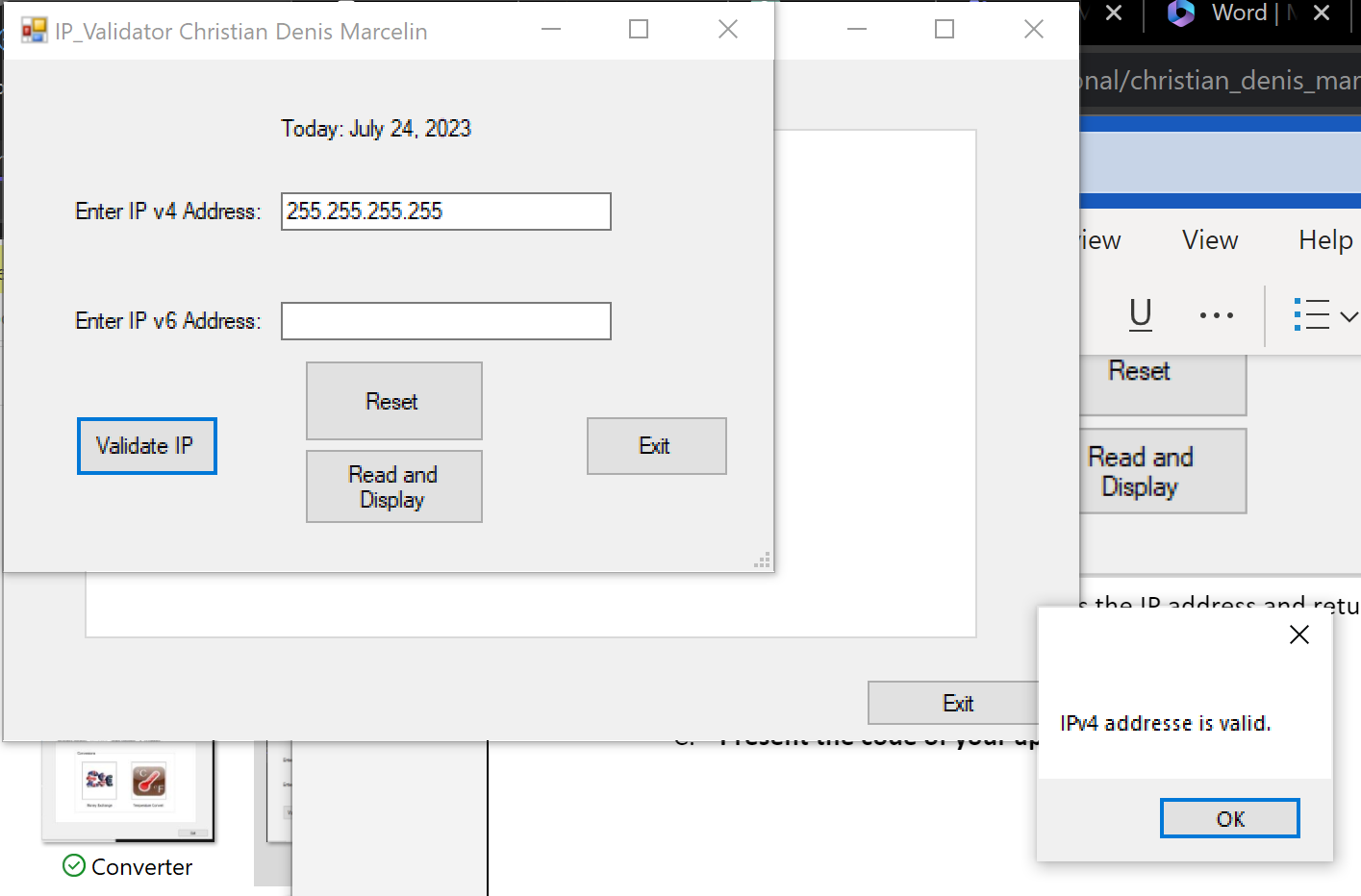


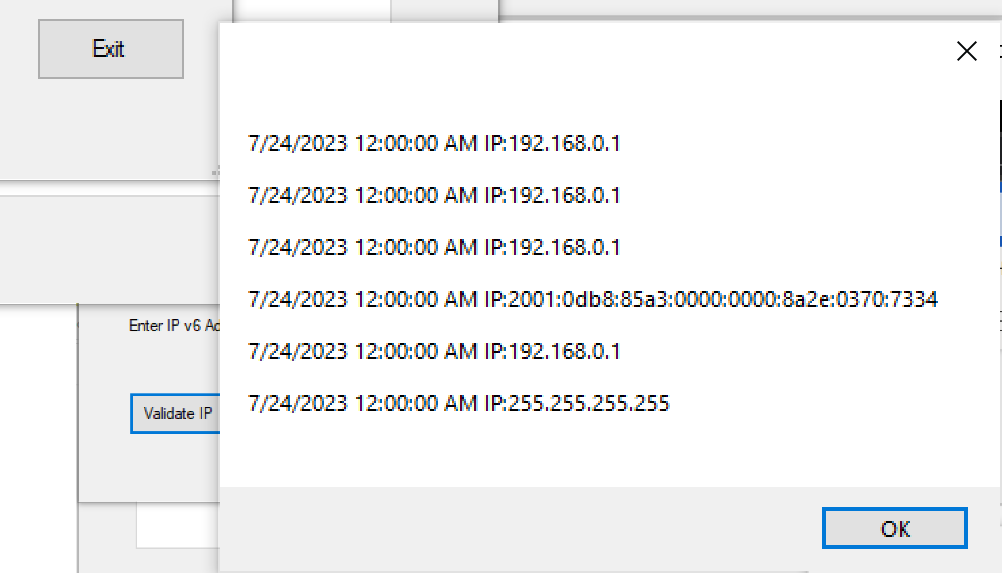


1. If you click on exit you can exit

(IP validator)

  
2. If you click on validate IP it checks the IP address and returns a message if its valid or not



1. Read and display  
   
2. Reset clear the text boxes
3. **Present the code of your application (forms).**

LotoMAX

… your code goes here

**Calculator**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

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

namespace WindowsFormsApp7

{

public partial class frmCalculator : Form

{

public frmCalculator()

{

InitializeComponent();

}

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

{

textBox1.Text += button1.Text;

textBox2.Text += button1.Text;

}

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

{

textBox1.Text += button2.Text;

textBox2.Text += button2.Text;

}

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

{

textBox1.Text += button3.Text;

textBox2.Text += button3.Text;

}

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

{

textBox1.Text += button4.Text;

textBox2.Text += button4.Text;

}

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

{

textBox1.Text += button5.Text;

textBox2.Text += button5.Text;

}

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

{

textBox1.Text += button6.Text;

textBox2.Text += button6.Text;

}

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

{

textBox1.Text += button7.Text;

textBox2.Text += button7.Text;

}

private void button8\_Click(object sender, EventArgs e)

{

textBox1.Text += button8.Text;

textBox2.Text += button8.Text;

}

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

{

textBox1.Text += button9.Text;

textBox2.Text += button9.Text;

}

private void button10\_Click(object sender, EventArgs e)

{

textBox1.Text += button10.Text;

textBox2.Text += button10.Text;

}

double num1 = 0;

double num2 = 0;

private void button12\_Click(object sender, EventArgs e)

{

textBox1.Text += button12.Text;

num1 = Convert.ToDouble(textBox2.Text);

textBox2.Text = "";

}

private void button16\_Click(object sender, EventArgs e)

{

if (textBox1.Text.Contains("+"))

{

textBox1.Text += button16.Text;

num2 = Convert.ToDouble(textBox2.Text);

textBox2.Text = (num1 + num2).ToString();

textBox1.Text += (num1 + num2).ToString();

button16.Enabled = false;

FileStream fs = new FileStream(@"./calc.txt",

FileMode.Append, FileAccess.Write);

StreamWriter objW = new StreamWriter(fs);

objW.WriteLine(textBox1.Text);

objW.Close();

fs.Close();

}

else if(textBox1.Text.Contains("-"))

{

textBox1.Text += button16.Text;

num2 = Convert.ToDouble(textBox2.Text);

textBox2.Text = (num1 - num2).ToString();

textBox1.Text += (num1 - num2).ToString();

button16.Enabled = false;

FileStream fs = new FileStream(@"./calc.txt",

FileMode.Append, FileAccess.Write);

StreamWriter objW = new StreamWriter(fs);

objW.WriteLine(textBox1.Text);

objW.Close();

fs.Close();

}

else if (textBox1.Text.Contains("\*"))

{

textBox1.Text += button16.Text;

num2 = Convert.ToDouble(textBox2.Text);

textBox2.Text = (num1 \* num2).ToString();

textBox1.Text += (num1 \* num2).ToString();

button16.Enabled = false;

FileStream fs = new FileStream(@"./calc.txt",

FileMode.Append, FileAccess.Write);

StreamWriter objW = new StreamWriter(fs);

objW.WriteLine(textBox1.Text);

objW.Close();

fs.Close();

}

else if (textBox1.Text.Contains("/"))

{

textBox1.Text += button16.Text;

num2 = Convert.ToDouble(textBox2.Text);

textBox2.Text = (num1 / num2).ToString();

textBox1.Text += (num1 / num2).ToString();

button16.Enabled = false;

FileStream fs = new FileStream(@"./calc.txt",

FileMode.Append, FileAccess.Write);

StreamWriter objW = new StreamWriter(fs);

objW.WriteLine(textBox1.Text);

objW.Close();

fs.Close();

}

}

private void button17\_Click(object sender, EventArgs e)

{

textBox1.Text = "";

textBox2.Text = "";

num1 = 0;

num2 = 0;

button16.Enabled = true;

}

private void button13\_Click(object sender, EventArgs e)

{

textBox1.Text += button13.Text;

num1 = Convert.ToDouble(textBox2.Text);

textBox2.Text = "";

}

private void button14\_Click(object sender, EventArgs e)

{

textBox1.Text += button14.Text;

num1 = Convert.ToDouble(textBox2.Text);

textBox2.Text = "";

}

private void button15\_Click(object sender, EventArgs e)

{

textBox1.Text += button15.Text;

num1 = Convert.ToDouble(textBox2.Text);

textBox2.Text = "";

}

private void button18\_Click(object sender, EventArgs e)

{

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

{

this.Close();

}

}

private void button11\_Click(object sender, EventArgs e)

{

textBox1.Text += button11.Text;

textBox2.Text += button11.Text;

}

}

}

**IPv4**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Net;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApp7

{

public partial class IP\_Validator : Form

{

public IP\_Validator()

{

InitializeComponent();

}

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

{

}

private void IP\_Validator\_Load(object sender, EventArgs e)

{

label1.Text = $"Today: {DateTime.Now.ToString("MMMM dd, yyyy")}";

}

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

{

string IPv4Pattern =

@"^(25[0-5]|2[0-4][0-9]|[0-1]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[0-1]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[0-1]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[0-1]?[0-9][0-9]?)$";

string IPv6Pattern =

@"^(?:[0-9a-fA-F]{1,4}:){7}[0-9a-fA-F]{1,4}$";

if ((Regex.IsMatch(textBox2.Text, IPv6Pattern)))

{

// Both IPv4 and IPv6 addresses are valid

MessageBox.Show("IPv6 addresse is valid.");

FileStream fs = new FileStream(@"./IP-Addresses.txt", FileMode.Append, FileAccess.Write);

// create the output stream for a text file that exists

BinaryWriter textOut = new BinaryWriter(fs);

// write the fields into text file

DateTime currentDate = DateTime.Now.Date;

textOut.Write($" {currentDate} IP:{textBox2.Text}" + "\n");

//textOut.WriteLine(textBox1.Text);

// close the output stream for the text file

textOut.Close();

}

else

{

// Show message box for invalid address

MessageBox.Show($"{textBox2.Text}\n Invalid IPv4 or IPv6 address.\n eight groups of four hexadecimal digits separated by colons.");

}

if ((Regex.IsMatch(textBox1.Text, IPv4Pattern)) )

{

// Both IPv4 and IPv6 addresses are valid

MessageBox.Show("IPv4 addresse is valid.");

FileStream fs = new FileStream(@"./IP-Addresses.txt", FileMode.Append, FileAccess.Write);

// create the output stream for a text file that exists

BinaryWriter textOut = new BinaryWriter(fs);

// write the fields into text file

DateTime currentDate = DateTime.Now.Date;

textOut.Write($" {currentDate} IP:{textBox1.Text}" + "\n");

//textOut.WriteLine(textBox1.Text);

// close the output stream for the text file

textOut.Close();

}

else

{

// Show message box for invalid address

MessageBox.Show($"{textBox1.Text}\n The IP must have 4 bytes \n integer numbers between 0 and 255 \n separated by a dot. 255.255.255.255.");

}

}

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

{

FileStream fs = new FileStream(@"./IP-Addresses.txt", FileMode.Open, FileAccess.Read);

try

{

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

BinaryReader textIn = new BinaryReader(fs);

string textToPrint = "";

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

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

{

string row = textIn.ReadString();

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

textToPrint += row + "\n";

}

MessageBox.Show(textToPrint);

// close the input stream for the text file

textIn.Close();

}

catch (FileNotFoundException)

{

MessageBox.Show(@".../IP-Addresses.txt" + " not found.", "File Not Found");

}

finally

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

}

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

{

textBox1.Text = "";

textBox2.Text = "";

}

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

{

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

{

this.Close();

}

}

}

}

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

|  |  |
| --- | --- |
| **Class/Method Name** | **Description** |
| 1. BinaryWriter | To write data to binary file |
| 1. BinaryReader | To read data from a binary file |
| 1. Regex | TO check a script for a particular pattern |
| 1. DateTime | TO create a time object |
| 1. FileStream | To Open a file to read or write |

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

The most difficult part of this project was to do the calculator, It was not straightforward as I needed to find the right box to find and display the information.