

BÁO CÁO THỰC HÀNH

Môn học: Lập trình mạng căn bản

Kỳ báo cáo: Buổi 02 (Session 02)

Tên chủ đề:

GVHD: Nghi Hoàng Khoa

Ngày báo cáo: 03/04/2022

Nhóm: NH (ghi số thứ tự nhóm)

1. THÔNG TIN CHUNG:

(Liệt kê tất cả các thành viên trong nhóm)

Lớp: NT106.M21.ATCL

STT	Họ và tên	MSSV	Email
1	Võ Khoa Nam	19521877	19521877@gm.uit.edu.vn
2	Trịnh Minh Hoàng	19521547	19521547@gm.uit.edu.vn
3	Phạm Trung Hiếu	19521512	19521512@gm.uit.edu.vn

2. NỘI DUNG THỰC HIỆN:¹

STT	Công việc	Kết quả tự đánh giá
1	Ghi và đọc file	100%
2	Đọc thông tin một file .txt	100%
3	Đọc và ghi file mở rộng	100%
4	Đọc và ghi file sử dụng BinaryFormatter	100%
5	Duyệt thư mục	100%

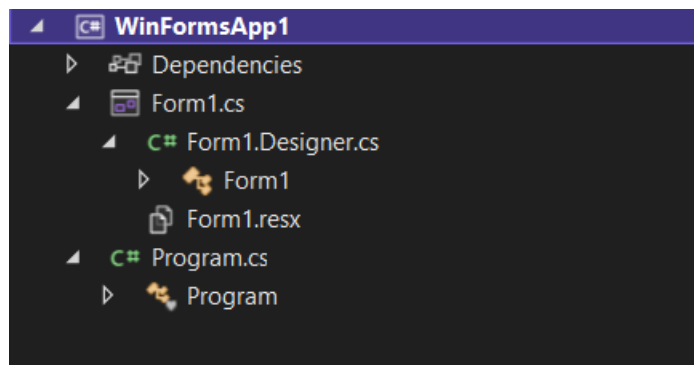
Phần bên dưới của báo cáo này là tài liệu báo cáo chi tiết của nhóm thực hiện.

¹ Ghi nội dung công việc, các kịch bản trong bài Thực hành

BÁO CÁO CHI TIẾT

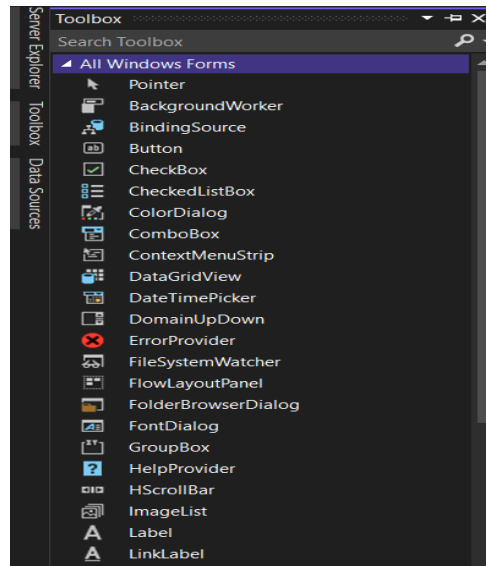
1. Ghi và đọc file

- Tài nguyên:
- Mô tả/mục tiêu:
 - Mô tả: đọc nội dung một file 'input.txt' và xuất ra màn hình nội dung file đó. Sau đó ghi nội dung (chuyển toàn bộ file sang chữ hoa) ghi xuống file 'output.txt'.
 - Mục tiêu: đọc được file, thực hiện được việc in hoa toàn bộ nội dung file và ghi nội dung đã in hoa ra file khác.
- Các bước thực hiện/ Phương pháp thực hiện (Ảnh chụp màn hình, có giải thích)
 - Bước 1: Tạo project trong Visual Code bằng cách chọn File -> New -> Project -> Visual C# -> Windows Form Application. Đặt tên cho project.

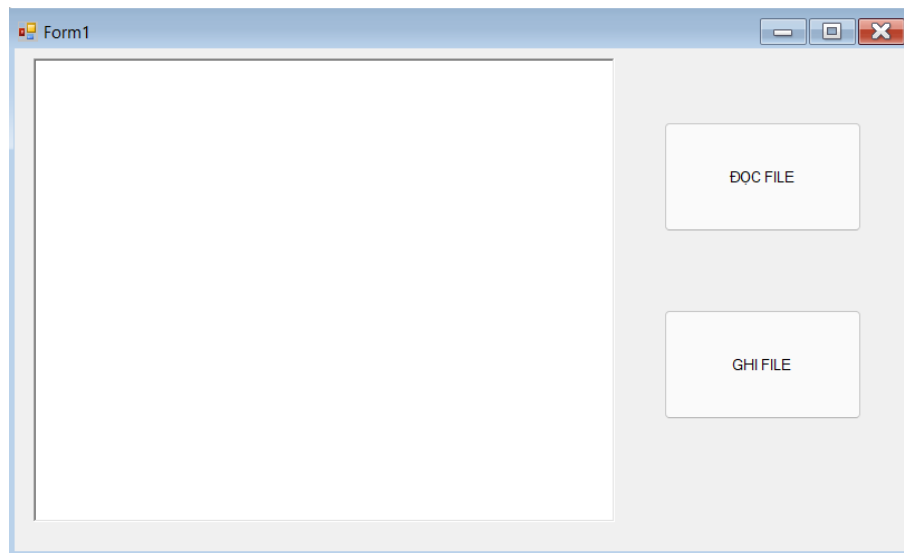


Hình 1: Sau khi tạo project sẽ được tổ chức gồm các file trong hình.

- Bước 1: Tạo form giao diện theo yêu cầu bằng cách sử dụng toolbox trong Visual Studio. Sử dụng button và richTextBox để tạo ra giao. Sử dụng button để làm nút khi nhấn vào sẽ thực hiện việc đọc và ghi file.



Hình 2: Thanh toolbox chứa các thẻ label, input text, button, ... Để tạo giao diện form



Hình 3: Giao diện windowsform tính tổng hai số nguyên sau khi kéo thả các thẻ từ thanh toolbox.

- Bước 2: Lập trình sự kiện và chức năng theo yêu cầu bài toán đọc ghi file và in hoa các nội dung trong file.
 - o Để đọc file cần click vào nút “ĐỌC FILE” trên giao diện. Vì vậy ta phải bắt sự kiện click cho button này.
 - o Nhấp đúp chuột vào button “ĐỌC FILE”. Chương trình sẽ khởi tạo 1 hàm tương ứng với sự kiện click vào button này.

```
private void button1_Click(object sender, EventArgs e)
{
    ...
}
```

Hình 4: Hàm khởi tạo sau khi click vào nút button tính tổng trên giao diện.



- Thêm code để thực hiện yêu cầu của bài toán vào

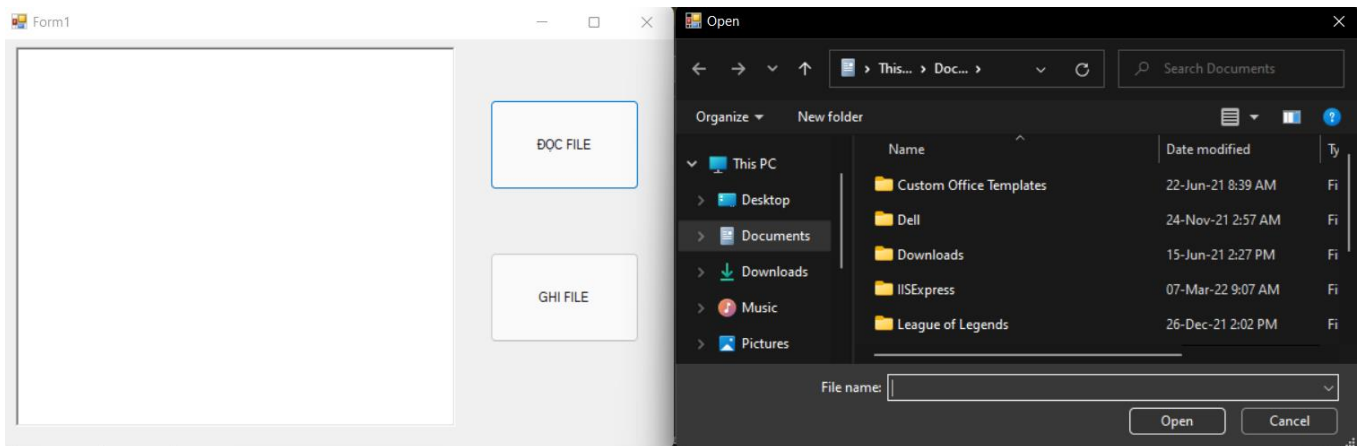
```
private void button1_Click(object sender, EventArgs e)
{
    OpenFileDialog ofd = new OpenFileDialog();
    ofd.ShowDialog();
    FileStream fs = new FileStream(ofd.FileName, FileMode.OpenOrCreate);
    StreamReader sr = new StreamReader(fs);

    string content = sr.ReadToEnd();
    richTextBox1.Text = content;

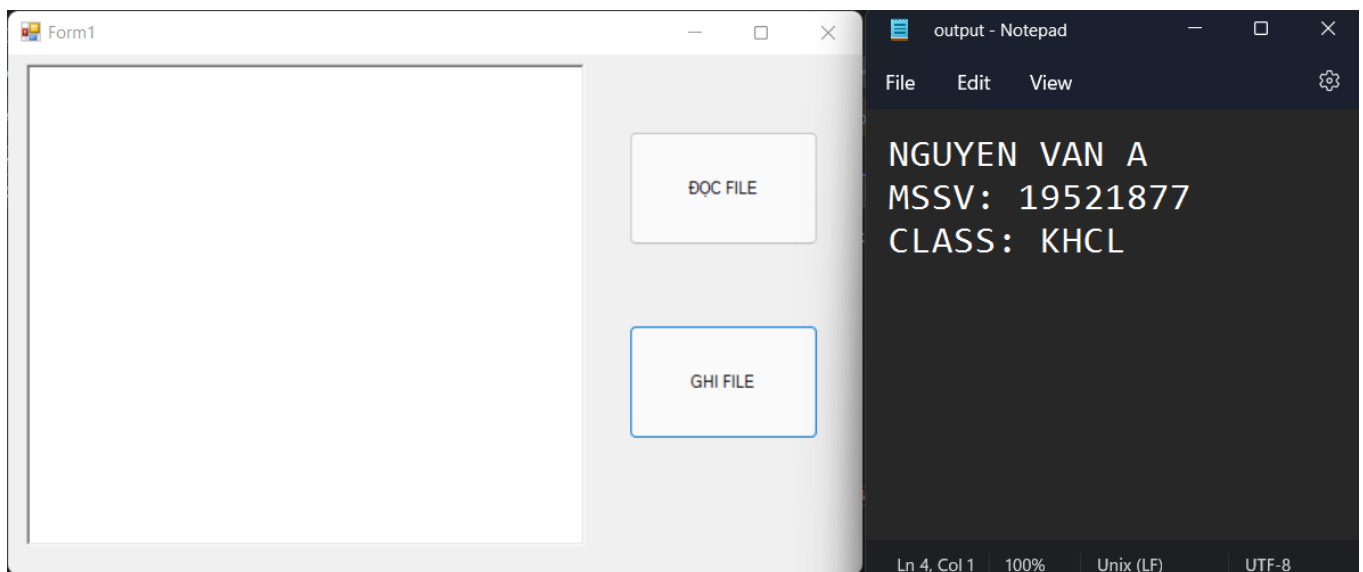
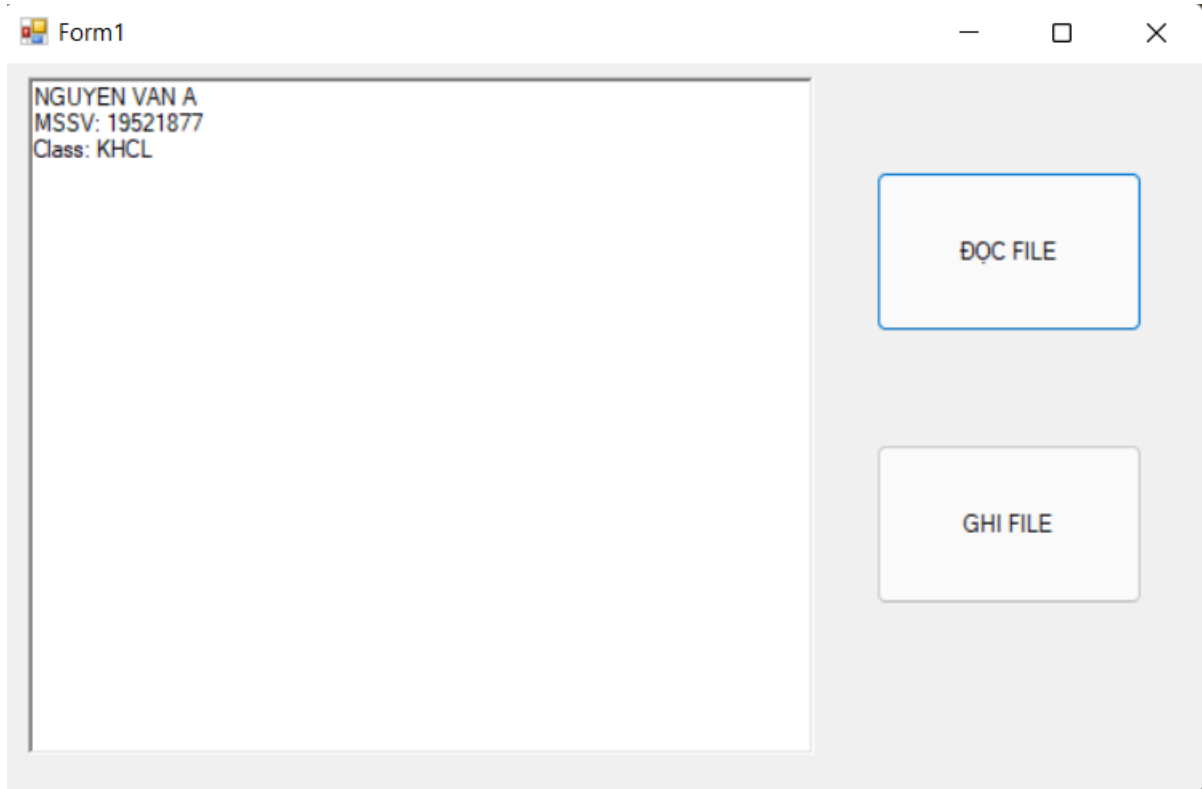
    fs.Close();
}
```

Hình 5: Đoạn mã thực hiện việc mở file.

- Sử dụng OpenFileDialog để mở hộp thoại windows để duyệt và chọn tệp.
- Sau khi sự kiện click để mở OpenFileDialog bằng phương thức ShowDialog. Người dùng sẽ chọn file cần mở và chọn OK.
- Tiến hành đọc file dưới dạng luồng bằng StreamReader. StreamReader sẽ đọc file và hiển thị nội dung của nó trong textBox.



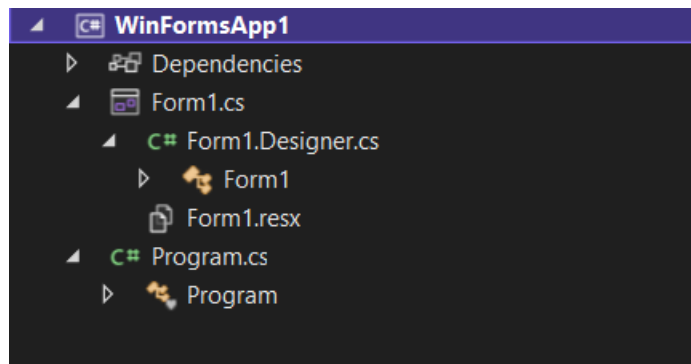
Hình 6: Khi click vào button đọc file OpenFileDialog sẽ mở hộp thoại windows để người dùng lựa chọn file cần mở



Hình 7: Kết quả sau khi mở file input.txt và ghi ra file output

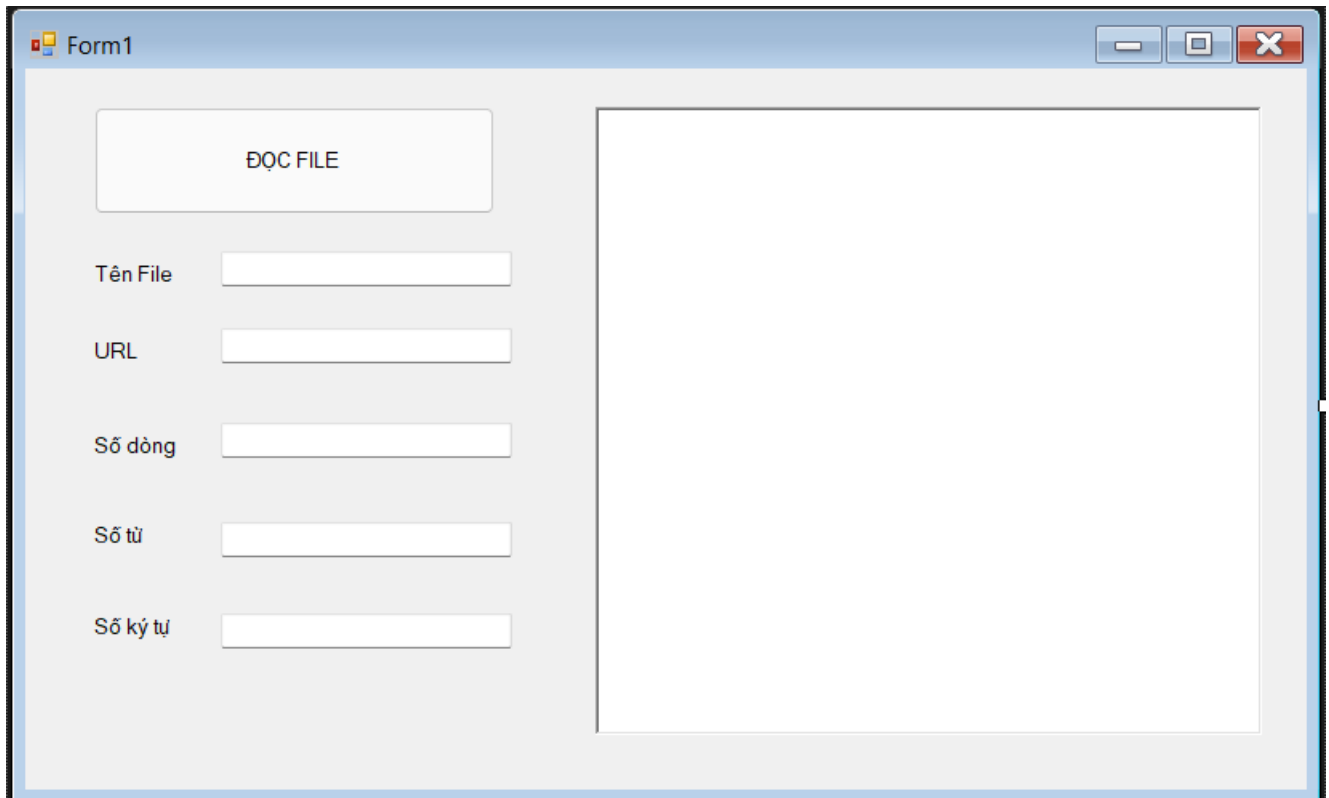
2. Đọc thông tin một file .txt

- Tài nguyên:
- Mô tả/mục tiêu:
 - Mô tả: đọc một file và hiển thị một số thông tin của file như tên file, url, số dòng, số ký tự, số từ, nội dung của file.
 - Mục tiêu: viết được chương trình đọc một file và thực hiện được các yêu cầu trên.
- Các bước thực hiện/ Phương pháp thực hiện (Ảnh chụp màn hình, có giải thích)
 - Bước 1: Tạo project trong Visual Code bằng cách chọn File -> New -> Project -> Visual C# -> Windows Form Application. Đặt tên cho project.



Hình 8: Sau khi tạo project sẽ được tổ chức gồm các file trong hình

- Bước 2: Tạo form giao diện theo yêu cầu bằng cách sử dụng toolbox trong Visual Studio. Sử dụng label và input text để tạo ra giao diện. Sử dụng button để tạo nút “ĐỌC FILE” khi nhấn vào sẽ hiển thị nội dung của file và các thông tin theo yêu cầu ra richTextBox.



Hình 9: Tạo form giao diện theo yêu cầu

- Bước 2: Lập trình sự kiện và chức năng theo yêu cầu
 - Để thực hiện đọc file và hiển thị các nội dung theo yêu cầu thì người dùng phải tiến hành click vào nút “ĐỌC FILE” vì vậy ta tiến hành bắt sự kiện khi người dùng click vào nút này.
 - Nhấp đúp chuột vào button “ĐỌC FILE”. Chương trình sẽ khởi tạo 1 hàm tương ứng với sự kiện click vào button này.

```
private void button1_Click(object sender, EventArgs e)
{
    // ...
}
```

Hình 10: Hàm khởi tạo sao khi click vào nút button tính tổng trên giao diện.

- Thêm code để thực hiện yêu cầu của bài toán vào

```
private void button1_Click(object sender, EventArgs e)
{
    //Đọc File
    OpenFileDialog ofd = new OpenFileDialog();
    ofd.ShowDialog();
    FileStream fs = new FileStream(ofd.FileName, FileMode.OpenOrCreate);
    StreamReader sr = new StreamReader(fs);

    string content = sr.ReadToEnd();
    richTextBox1.Text = content;

    fs.Close();

    //Lấy tên File
    string namefile = ofd.SafeFileName.ToString();
    textBox1.Text = namefile;

    //Lấy URL
    string URL = fs.Name.ToString();
    textBox2.Text = URL;

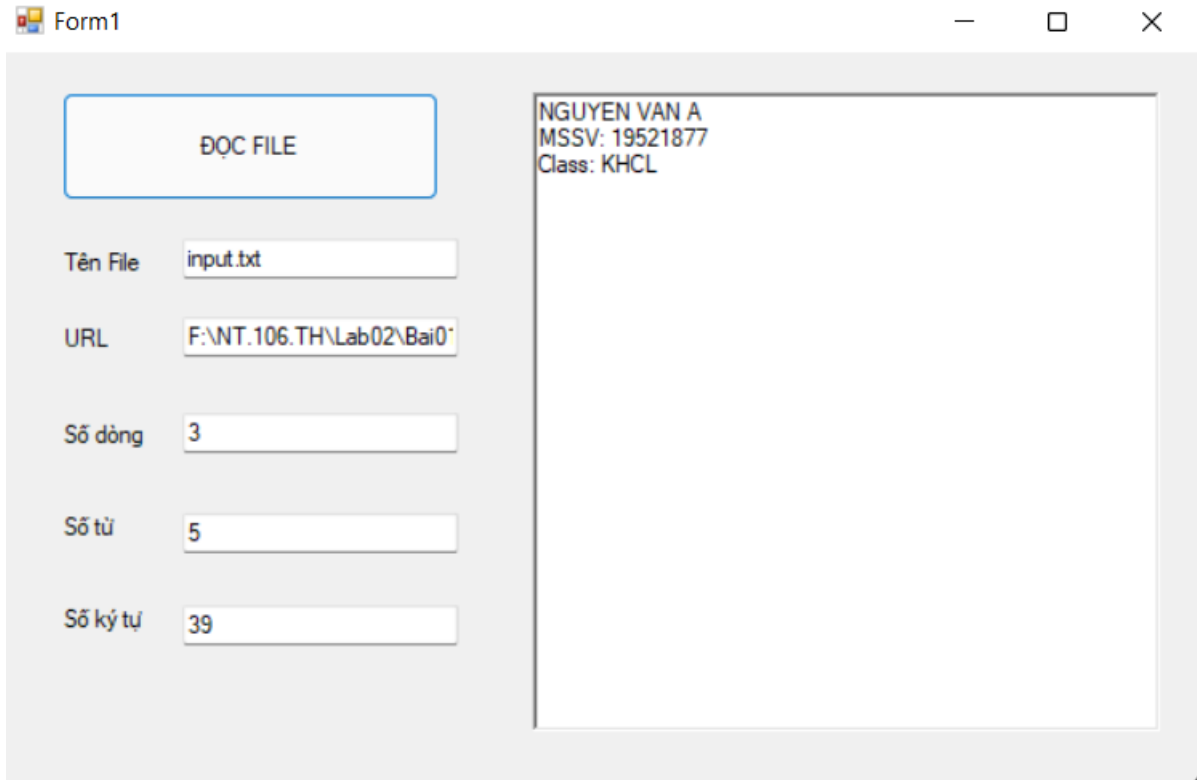
    //Số dòng
    content = content.Replace("\r\n", "\n");
    int LineCount = richTextBox1.Lines.Count();
    content = content.Replace('\r', ' ');
    textBox3.Text = LineCount.ToString();

    //Số ký tự
    int CharacterCount = content.Length;
    textBox5.Text = CharacterCount.ToString();

    //Số từ
    string[] Source = content.Split(new char[] { '.', '?', '!', ' ', '(', ')', ':', ';', ',', '}', StringSplitOptions.RemoveEmptyEntries);
    int WordCount = Source.Length;
    textBox4.Text = WordCount.ToString();
}
```

Hình 11: Đoạn mã thực hiện đọc file và hiển thị các nội dung theo yêu cầu

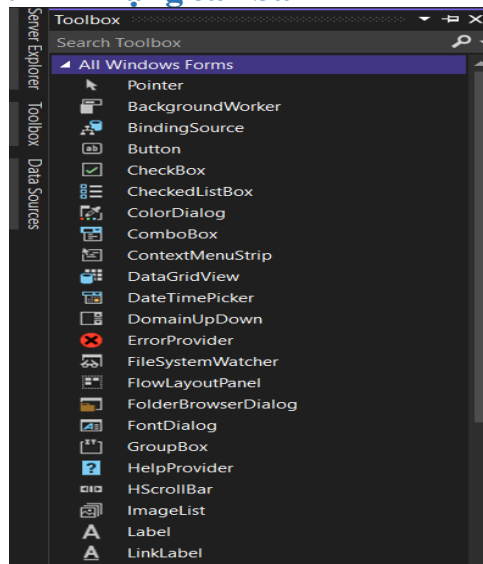
- Sử dụng StreamReader để đọc file giống yêu cầu của bài 1
- Sau khi đọc file sử dụng .SafeFileName, .Name Property của OpenFileDialog để lấy được tên của file, url và hiển thị ra textBox.
- Đối với việc lấy số dòng, đếm số từ số ký tự của file ta dựa vào việc hiển thị nội dung của file ra richTextBox. Sử dụng .Lines.Count() để thực hiện đếm số dòng của file đã mở. Đối với số ký tự của file sử dụng CharacterCount để đếm số ký tự, đối với số từ sử dụng việc tách chuỗi sử dụng hàm split() để tách bỏ các ký tự, dấu câu và khoảng trắng sau đó lưu vào một mảng Source đã khai báo. Sử dụng .Length để đếm số từ của file.



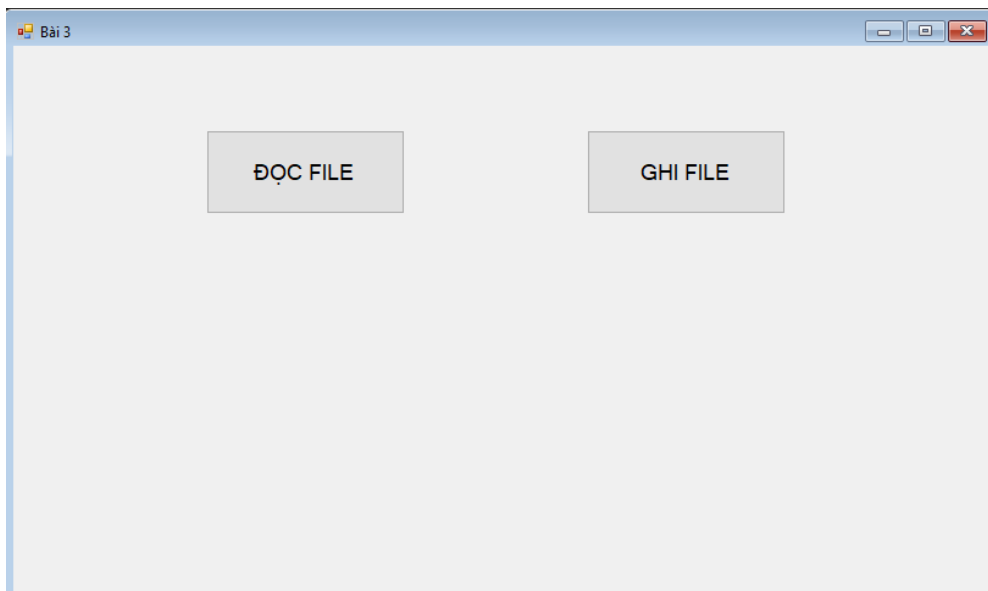
Hình 12: Kết quả hoạt động của chương trình khi thực hiện chức năng đọc file

3. Đọc và ghi file mở rộng

- Tài nguyên:
- Mô tả/mục tiêu:
 - Mô tả: đọc nội dung từ file “input.txt” với nội dung theo định dạng, sau đó thực hiện các phép tính và ghi kết quả xuống file “output.txt”.
 - Mục tiêu: kết quả là file output của phép tính.
- Các bước thực hiện/ Phương pháp thực hiện (Ảnh chụp màn hình, có giải thích)
 - Bước 1: Tạo form giao diện theo yêu cầu bằng cách sử dụng toolbox trong Visual Studio. Sử dụng label và input text để tạo ra giao diện nhập số thứ nhất, số thứ hai và hiển thị kết quả tính tổng hai số. Sử dụng 2 button để làm nút “ĐỌC FILE” và “GHI FILE” khi nhấn vào “GHI FILE” sẽ thực hiện các phép tính của file “input.txt”



Hình 13: Thanh tool box chứa các thành phần để tạo giao diện form



Hình 14: Giao diện windows form khởi tạo theo yêu cầu

- Bước 2: Lập trình sự kiện và chức năng theo yêu cầu bài toán thực hiện các phép tính
 - Để đọc file “input.txt” người dùng cần click vào nút “ĐỌC FILE” trên giao diện. Vì vậy ta phải bắt sự kiện click cho button này.
 - Nhấp đúp chuột vào button “ĐỌC FILE”. Chương trình sẽ khởi tạo 1 hàm tương ứng với sự kiện click vào button này.

```
private void button1_Click(object sender, EventArgs e)
{
    ...
}
```

Hình 15: Hàm được khởi tạo sao khi click vào nút đọc file

- Bắt sự kiện cho nút “ĐỌC FILE”, sử dụng lớp StreamReader.

```

StreamReader sr;
1 reference
private void button1_Click(object sender, EventArgs e)
{
    OpenFileDialog ofd = new OpenFileDialog();
    ofd.ShowDialog();
    FileStream fs = new FileStream(ofd.FileName, FileMode.OpenOrCreate);
    sr = new StreamReader(fs);
    MessageBox.Show("Đọc file thành công");
}

```

Hình 16: Đoạn mã kiểm tra đọc file input.txt

- Bước 3: Đoạn mã thực hiện các phép tính “+, -, *, /” trong file “input.txt”

```

private string bieuThucOut(string line)
{
    String a = line;
    string[] source = a.Split(' ');
    float result = 0, num1 = 0, num2 = 0;
    char pheptinh = source[1][0];
    num1 = Int32.Parse(source[0]);
    string sothu2 = source[1].Substring(1);
    num2 = Int32.Parse(sothu2);

    switch (pheptinh)
    {
        case '+': result = num1 + num2; break;
        case '-': result = num1 - num2; break;
        case '*': result = num1 * num2; break;
        case '/': result = num1 / num2; break;
    }

    return line + " = " + result.ToString();
}

```

Hình 17: Hàm để thực hiện các phép tính

- Bước 4:
 - Để ghi kết quả xuống file “output.txt” người dùng cần click vào nút “GHI FILE” trên giao diện. Vì vậy ta phải bắt sự kiện click cho button này.
 - Nhấp đúp chuột vào button “GHI FILE”. Chương trình sẽ khởi tạo 1 hàm tương ứng với sự kiện click vào button này.

```

private void button1_Click(object sender, EventArgs e)
{
    // ...
}

```

- Bắt sự kiện cho nút “GHI FILE”, sử dụng lớp StreamWriter

```
private void button2_Click(object sender, EventArgs e)
{
    OpenFileDialog ofd = new OpenFileDialog();
    ofd.ShowDialog();
    FileStream fs = new FileStream(ofd.FileName, FileMode.Create);
    string line;
    using (StreamWriter sw = new StreamWriter(fs))
    {
        while ((line = sr.ReadLine()) != null)
        {
            sw.WriteLine(bieuThucOut(line));
        }
    }
}
```

Hình 18: Thực hiện ghi file

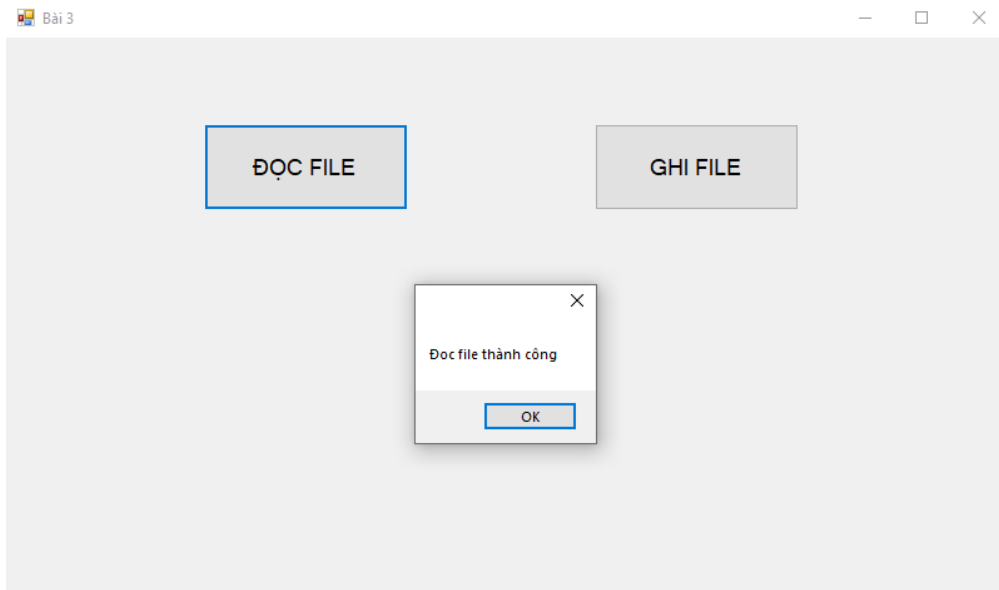
- Khi chương trình hoạt động:
 - Sau khi thực hiện đọc file “input.txt”.

input.txt - Notepad

File Edit Format View Help

```
1 +2
12 -7
10 *20
200 /10
```

Hình 19: Nội dung file input.txt



Hình 20: Xuất hiện thông báo khi đọc file thành công

output.txt - Notepad

File Edit Format View Help

```
1 +2 = 3
12 -7 = 5
10 *20 = 200
200 /10 = 20
```

Hình 21: Kết quả file output

4. Kịch bản 05

- Tài nguyên:
- Mô tả/mục tiêu:
 - Mô tả: Nhập một mảng các học viên ghi xuống file “input.txt” sử dụng BinaryFormatter. Đọc thông tin mảng học viên từ “input.txt” tính điểm trung bình và lưu vào file “output.txt”
 - Mục tiêu: Thu được 2 file “input.txt” và “output.txt” và hiển thị thông tin từ hai file ra màn hình
- Các bước thực hiện/ Phương pháp thực hiện (Ảnh chụp màn hình, có giải thích)
 - Bước 1: Tạo form



Hình 21: Tạo window form theo yêu cầu

- Nhập thông tin học viên bằng TextBox
- Save: lưu thông tin học viên
- Load input: đọc file “input.txt” từ máy tính
- Tính TB và Lưu: tính điểm trung bình từng học viên và lưu vào file “output.txt”
- Load Output: đọc file “output.txt”
- Bước 2: Lập trình chức năng

```

struct SinhVien
{
    public string MSSV;
    public string HoTen;
    public string DienThoai;
    public float DiemToan;
    public float DiemVan;
    public float DiemTrungBinh;
}

```

Hình 21: Tạo kiểu dữ liệu SinhVien để lưu thông tin sinh viên

```
private void btnSave_Click(object sender, EventArgs e)
{
    string[] lSinhVien = txtNhap.Text.Split("\r\n");
    SinhVien[] infoStudent = new SinhVien[lSinhVien.Length];
    int i = 0;
    foreach (string sinhvien in lSinhVien)
    {
        string[] s = sinhvien.Split(",");
        infoStudent[i].MSSV = s[0].Trim();
        infoStudent[i].HoTen = s[1].Trim();
        infoStudent[i].DienThoai = s[2].Trim();
        infoStudent[i].DiemToan = float.Parse(s[3].Trim());
        infoStudent[i].DiemVan = float.Parse(s[4].Trim());
        i++;
    }
    save(infoStudent);
}
```

```
static void save(SinhVien[] info)
{
    SaveFileDialog sfd = new SaveFileDialog();
    sfd.ShowDialog();
    FileStream fs = new FileStream(sfd.FileName, FileMode.OpenOrCreate);
    var formater = new BinaryFormatter();
    formater.Serialize(fs, info);
    fs.Close();
}
```

- Nhấn nút “Save”:
 - Dùng “Split” tách các dòng từ mảng người dùng nhập để lấy được từng học viên. Sau đó lưu thông tin của từng học viên vào mảng “infoStudent” có kiểu dữ liệu SinhVien mới tạo.
 - Sử dụng Serialize của BinaryFormatter để biến đổi mảng “infoStudent” về mảng byte và ghi file.

```
// Lưu trữ thông tin sinh viên ở file output
List<SinhVien> outputs = new List<SinhVien>();
```

```
private void btnInput_Click(object sender, EventArgs e)
{
    OpenFileDialog ofd = new OpenFileDialog();
    ofd.ShowDialog();
    FileStream fs = new FileStream(ofd.FileName, FileMode.OpenOrCreate);
    var formater = new BinaryFormatter();
    SinhVien[] students = formater.Deserialize(fs) as SinhVien[];

    string s = "";
    int i = 0;
    foreach (SinhVien student in students)
    {
        s = s + student.MSSV + "\r\n" + student.HoTen + "\r\n" + student.DienThoai + "\r\n" + student.DiemToan.ToString() + "\r\n" + student.DiemVan.ToString() + "\r\n\r\n\r\n";
        i++;
        SinhVien sv = new SinhVien();
        sv = student;
        sv.DiemTrungBinh = (student.DiemToan + student.DiemVan) / 2;
        outputs.Add(sv);
    }
    txtInput.Text = s;
    fs.Close();
}
```

- Nhấn nút “Load Input”
 - Đọc dữ liệu từ file “input.txt” sử dụng “Deserialize” để khôi phục từ mảng byte về mảng có kiểu dữ liệu SinhVien.
 - Tính điểm trung bình và lưu trữ thông tin sinh viên ở biến toàn cục outputs có kiểu dữ liệu List.

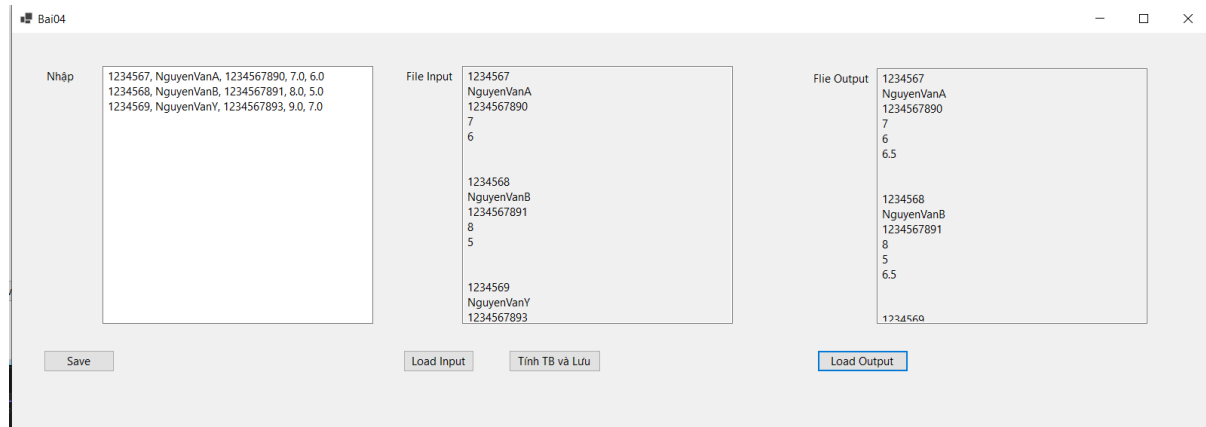
```
1 reference
private void btnTB_Click(object sender, EventArgs e)
{
    ...
    save(outputs.ToArray());
}
```

- Nhấn nút “Tính TB và Lưu”: chuyển đổi “outputs” từ dạng list sang dạng array và lưu vào file “output.txt”.

```
private void btnOutput_Click(object sender, EventArgs e)
{
    OpenFileDialog ofd = new OpenFileDialog();
    ofd.ShowDialog();
    FileStream fs = new FileStream(ofd.FileName, FileMode.OpenOrCreate);
    var formatter = new BinaryFormatter();
    SinhVien[] students = formatter.Deserialize(fs) as SinhVien[];

    string s = "";
    foreach (SinhVien student in students)
    {
        s = s + student.MSSV + "\r\n" + student.HoTen + "\r\n" + student.DienThoai + "\r\n" + student.DiemToan.ToString() + "\r\n" + student.DiemVan.ToString() + "\r\n" + student.DiemTrungBinh + "\r\n\r\n\r\n";
    }
    txtOutput.Text = s;
    fs.Close();
}
```

- Nhấn nút “Load Output”: đọc file “output.txt” và xuất ra màn hình
- Kết quả:



- File Input:

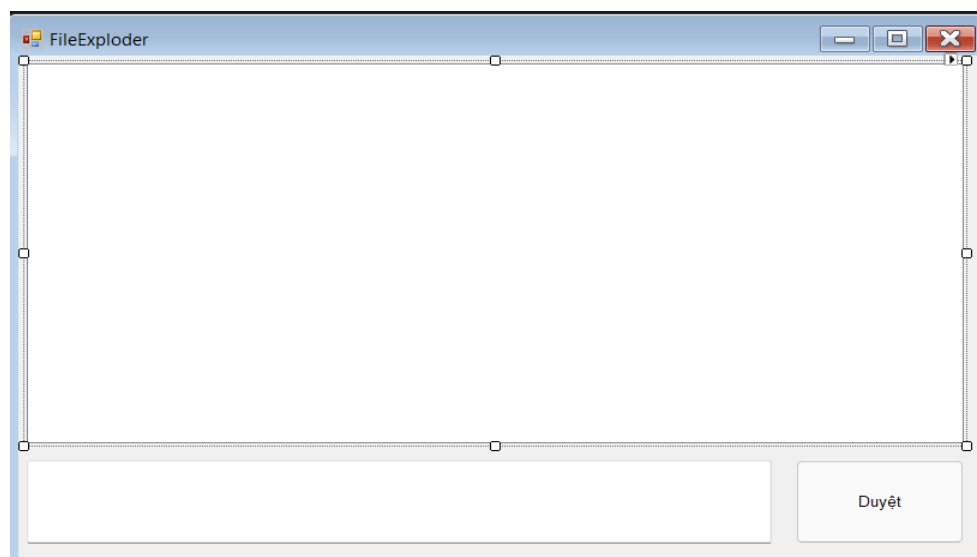


- File Output:



5. Duyệt thư mục

- Tài nguyên:
- Mô tả/mục tiêu:
 - Mô tả: Duyệt thư mục cho tất cả các file có trong thư mục, hiển thị được danh sách các file.
 - Tên file
 - Kích thước
 - Ngày tạo
 - Đuôi mở rộng
 - Mục tiêu: Viết được chương trình duyệt thư mục với các yêu cầu trên
- Các bước thực hiện/ Phương pháp thực hiện (Ảnh chụp màn hình, có giải thích)
 - Bước 1: Tạo project trong Visual Code bằng cách chọn File -> New -> Project -> Visual C# -> Windows Form Application. Đặt tên cho project.
 - Bước 2: Tạo form giao diện theo yêu cầu bằng cách sử dụng toolbox



- Bước 3: Lập trình sự kiện và chức năng theo yêu cầu
 - Tiến hành lập trình để bắt sự kiện khi người dùng click vào button “Duyệt”
 - Khi người dùng click vào button “Duyệt” sẽ khởi tạo một hàm tương ứng.
 - Tiến hành thêm các câu lệnh vào hàm vừa mới khởi tạo để thực hiện yêu cầu
 - Đầu tiên ta tiến hành khởi tạo một hàm LoadListView để có thể tạo ra một listView với các cột có tên lần lượt là “Tên”, “Kích thước”, “Đuôi mở rộng”, “Ngày khởi tạo”

```
void LoadListView()
{
    listView1.View = View.Details;
    listView1.Columns.Add("Tên");
    listView1.Columns.Add("Kích thước");
    listView1.Columns.Add("Đuôi mở rộng");
    listView1.Columns.Add("Ngày khởi tạo");
}
```

Hình 14: Khởi tạo hàm LoadListView và thêm tên các cột vào

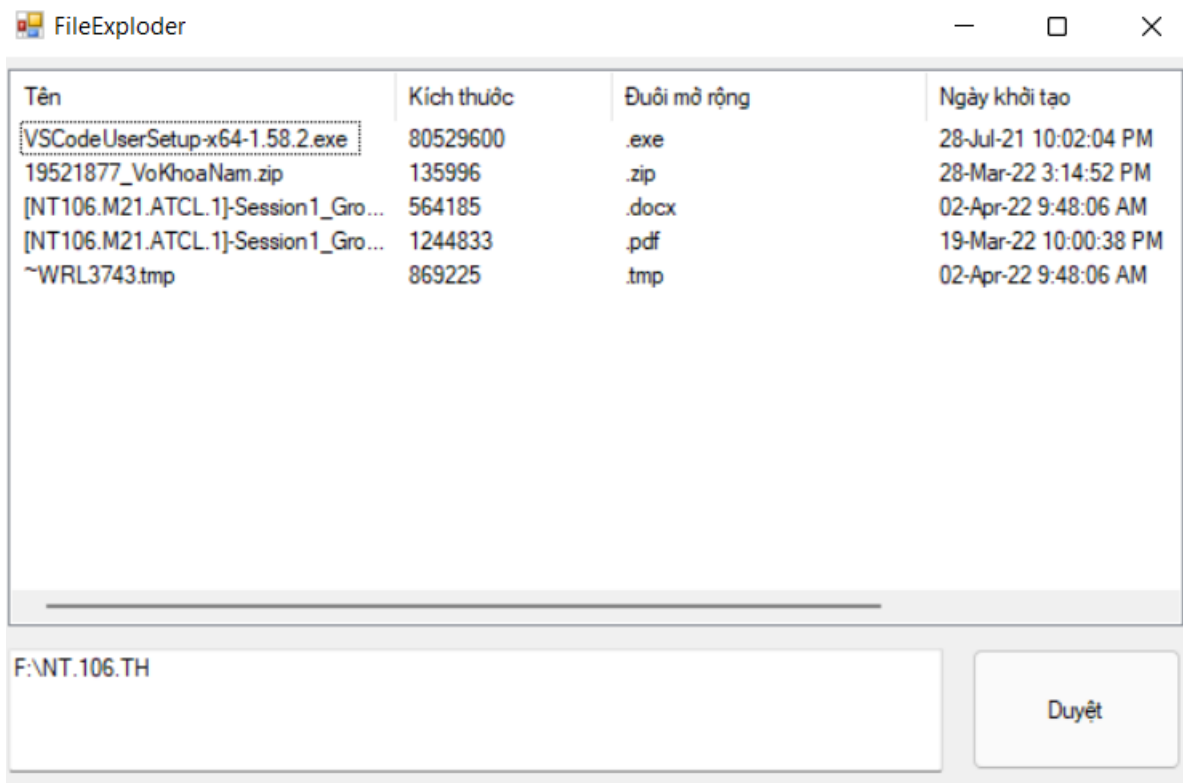
- Tiến hành thêm câu lệnh vào hàm clickButton để thực hiện yêu cầu duyệt thư mục

```
private void button1_Click(object sender, EventArgs e)
{
    FolderBrowserDialog fbd = new FolderBrowserDialog();
    if (fbd.ShowDialog() == DialogResult.OK)
    {
        textBox1.Text = fbd.SelectedPath;
        DirectoryInfo di = new DirectoryInfo(fbd.SelectedPath);
        FileInfo[] fiArr = di.GetFiles();
        foreach (FileInfo file in fiArr)
        {
            ListViewItem item = new ListViewItem();
            item.Text = file.Name;
            item.SubItems.Add(new ListViewItem.ListViewSubItem() { Text = file.Length.ToString() });
            item.SubItems.Add(new ListViewItem.ListViewSubItem() { Text = file.Extension });
            item.SubItems.Add(new ListViewItem.ListViewSubItem() { Text = file.CreationTime.ToString() });
            listView1.Items.Add(item);
        }
    }
}
```

Hình 15: Thêm mã vào hàm button_click để thực hiện yêu cầu

- Sử dụng thành phần của windows form là FolderBrowserDialog để cho phép người dùng chọn 1 thư mục trong hộp thoại windows.
- Tên thư mục cần duyệt được hiển thị trong textBox để người dùng có thể chỉnh sửa lựa chọn.

- Sử dụng các hàm .Name, .Length, .Extension, .Createtime có sẵn để hiển thị các giá trị tên, kích thước, đuôi mở rộng, thời gian tạo file trong thư mục.



Hình 16: Kết quả duyệt thư mục NT.106.TH ở F

LinkGitHub(Source Code):

<https://github.com/khoanamgit/NT106.M21.ATCL-NH/tree/main/Lab02/SourceCode>

YÊU CẦU CHUNG

- Sinh viên tìm hiểu và thực hành theo hướng dẫn.
- Nộp báo cáo kết quả chi tiết những việc (**Report**) bạn đã thực hiện, quan sát thấy và kèm ảnh chụp màn hình kết quả (nếu có); giải thích cho quan sát (nếu có).
- Sinh viên báo cáo kết quả thực hiện và nộp bài.

Báo cáo:

- File **.PDF**. Tập trung vào nội dung, không mô tả lý thuyết.
- Nội dung trình bày bằng Font chữ Times New Romans/ hoặc font chữ của mẫu báo cáo này (UTM Neo Sans Intel/UTM Viet Sach)– cỡ chữ 13. Canh đều (Justify) cho văn bản. Canh giữa (Center) cho ảnh chụp.
- Đặt tên theo định dạng: [Mã lớp]-SessionX_GroupY. (trong đó X là Thứ tự buổi Thực hành, Y là số thứ tự Nhóm Thực hành đã đăng ký với GVHD-TH).

Ví dụ: [NT101.K11.ANTT]-Session1_Group3.

- Nếu báo cáo có nhiều file, nén tất cả file vào file .ZIP với cùng tên file báo cáo.
- Không đặt tên đúng định dạng – yêu cầu, sẽ **KHÔNG** chấm điểm bài Lab.
- Nộp file báo cáo trên theo thời gian đã thống nhất tại courses.uit.edu.vn.

Đánh giá: Sinh viên hiểu và tự thực hiện được bài thực hành. Khuyến khích:

- Chuẩn bị tốt.
- Có nội dung mở rộng, ứng dụng trong kịch bản phức tạp hơn, có đóng góp xây dựng bài thực hành.

Bài sao chép, trễ, ... sẽ được xử lý tùy mức độ vi phạm.

HẾT