BÁO CÁO THỰC HÀNH

**Môn học: Nhập môn mạng máy tính**

**Buổi báo cáo: Lab 02**

**Tên chủ đề:** File và I/O stream trong C#

*GVHD: Nguyễn Văn Bảo*

*Ngày thực hiện: 03/04/2023*

**THÔNG TIN CHUNG:**

*(Liệt kê tất cả các thành viên trong nhóm)*

Lớp: NT106.O23

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ và tên** | **MSSV** | **Email** |
| 1 | Nguyễn Tài Hiếu | 22520442 | 22520442@gm.uit.edu.vn |
|  |  |  |  |

1. **ĐÁNH GIÁ KHÁC:**

|  |  |
| --- | --- |
| **Nội dung** | **Kết quả** |
| Tổng thời gian thực hiện bài thực hành trung bình | 13 ngày |
| Link Video thực hiện  *(nếu có)* | https://github.com/nguyentaihieu004/thuchanhltm.git |
| Ý kiến *(nếu có)*  + Khó khăn  + Đề xuất … |  |
| Điểm tự đánh giá | 7 |

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

BÁO CÁO CHI TIẾT

1. Bài 01- Ghi và Đọc file

A screenshot of a computer

Description automatically generated

* Bố cục: richtextbox hiển thị nội dung file sau khi đọc, button “Đọc File” thực hiện mở Opendialog() để đọc file, button “Ghi File” thực hiện ghi nội dung trong richtextbox vào file .
* Ý tưởng: click nút button “đọc file” hiển thị opendialog() chọn file input1.txt đã có sẵn dữ liệu, hiển thị lên richtextbox, click button “Ghi File” để ghi nội dùng trong richtextbox vào file output1.txt

Code:

A screen shot of a computer program

Description automatically generated

* Tạo biến input tạm để lưu chuỗi đọc từ file.
* Hàm docfile\_Click:

+ tạo đối tượng opdl OpenFileDialog, thực hiện mở opdl.ShowDialog();

+ tạo một luồng tệp tin để đọc được chọn bằng cách sử dụng tên tệp được chọn từ Openfiledialog. File sẽ mở để đọc

+ StreamReadder rs , tạo đối tượng StreamReader để đọc từ luồng tệp tin tạo ra.

+ input=rs.ReadtoEnd() đọc toàn bộ nội dung của tệp tin đucợ chọn bằng phương thức ReadtoEnd , gán vào đối tượng vào biến input.

+ đặt nội dung richtexBox1 = input, hiển thị cho người dùng.

+ rs.Close(); và fs.Close(); : Đóng các luồng tệp tin và giải phóng tài nguyên.

A computer screen with text on it

Description automatically generated

* FileStream fs = new FileStream(opdl.FileName, FileMode.OpenOrCreate, FileAccess.Write) Tạo một luồng tệp tin để ghi dữ liệu vào tệp được chọn. Nếu tệp đã tồn tại, nó sẽ được mở để ghi (FileMode.OpenOrCreate), và chỉ cho phép ghi (FileAccess.Write).
* StreamWriter , tạo đối tượng đẻ ghi dữ liệu vào luồng tệp tin được tạo
* string tmp = input.ToUpper(): Chuyển đổi nội dung của biến input thành chữ hoa và gán vào biến tmp.
* sw.Flush(): Đẩy bất kỳ dữ liệu đang nằm trong bộ đệm của StreamWriter vào luồng tệp tin.
* sw.Write(tmp): Ghi nội dung của biến tmp (đã được chuyển thành chữ hoa) vào luồng tệp tin.
* sw.Close() và fs.Close(): Đóng các luồng tệp tin và giải phóng tài nguyên.

1. Bài 02 – Đọc thông tin một file.txt

A screenshot of a computer

Description automatically generated

* Bố cục: 2 buton để thực hiện event đọc file và thoát chương trình, các textbox tương ứng để hiện yêu cầu
* Ý tưởng: click nút đọc file , OpenFiledialog hiện ra người dùng chọn file txt, hiển thị trên form các nội dung như tên file, kích thước, đường dẫn…

A screen shot of a computer program

Description automatically generated

* A screen shot of a computer program

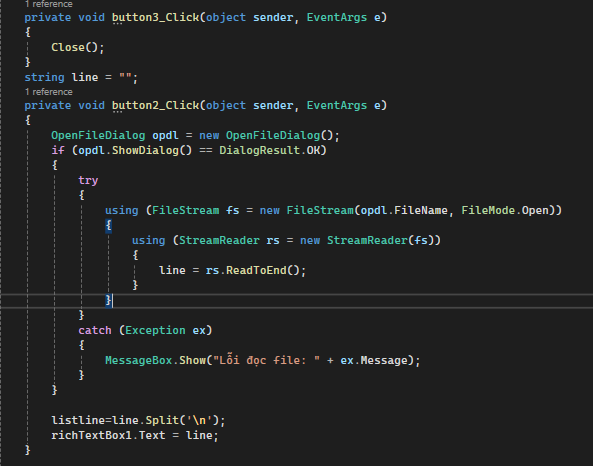
  Description automatically generated
* Các hàm và phương thức để đọc file đã trình bày ở câu 1, tương tự.
* ‘while (rs.ReadLine() != null) { linecount++; }’: Đếm số dòng trong tệp tin bằng cách đọc từng dòng của tệp tin cho đến khi gặp kết thúc tệp null. Mỗi lần đọc một dòng mới, biến linecount sẽ được tăng lên.
* textBox4linecount.Text = linecount.ToString(): Hiển thị số dòng đếm được trong một TextBox
* rs.BaseStream.Seek(0, SeekOrigin.Begin): Đặt con trỏ đọc của luồng tệp tin về đầu tệp tin để đọc toàn bộ nội dung từ đầu.
* string str = rs.ReadToEnd();: Đọc toàn bộ nội dung của tệp tin vào một chuỗi str. Và cho hiển thị ra richtextbox
* textBox1filename.Text = opdl.SafeFileName: Hiển thị tên tệp tin đã chọn trong một textbox
* textBox2size.Text = rs.BaseStream.Length.ToString() + " bytes";: Hiển thị kích thước của tệp tin trong một TextBox
* textBox3url.Text = fs.Name.ToString();: Hiển thị đường dẫn đến tệp tin trong một TextBox có tên textBox3url.
* Đếm số từ trong tệp tin bằng cách xác định các khoảng trắng, dấu xuống dòng và tab, sau đó tăng biến wordscount lên và hiển thị ra text box , đếm từ sẽ sử dụng vòng lặp với mỗi dấu xuống dong, space
* textBox6characcount.Text = str.Length.ToString();: Hiển thị số ký tự trong tệp tin trong một TextBox có tên textBox6characcount.

1. Bài 03 – Đọc và Ghi file và tính toán

A screenshot of a computer

Description automatically generated

* Bố cục : 3 button ứng với các event, tính đọc file, thoát, richtextbox hiện kết quả
* Ý tưởng: đọc file từ tệp txt, tách chuỗi trong file thành các biểu thức phép tính vào 1 mảng, duyệt qua mảng và sử dụng thư viện Ncal để tính toán, in ra richtextbox theo format.



* Hàm thực hiện đọc file từ file txt
* Khởi tạo opendialog, kiểm tra opendialog có mở thành công không.
* Try catch, khởi tạo luồng đọc file, đọc ReadtoEnd() gán nội dung đọc vào biến line.
* Nếu không đọc dược hiển thị thông báo lỗi
* Tách từng dòng của line vào list để tính toán sau đó hiển thị lên richtextbox

A computer screen shot of a program code

Description automatically generated

* Hàm CaculateExpression , NCalc.Expression expr = new NCalc.Expression(expression);: Tạo một đối tượng của lớp NCalc.Expression
* expr.Evaluate(): Gọi phương thức Evaluate() của đối tượng expr. Phương thức này sẽ tính toán giá trị của biểu thức toán học và trả về kết quả dưới dạng một đối tượng của lớp Object.
* Convert.ToDouble(): Chuyển đổi kết quả tính toán từ kiểu Object sang kiểu dữ liệu double.
* return: Trả về kết quả tính toán của biểu thức toán học dưới dạng kiểu dữ liệu double từ phương thức CalculateExpression.
* Hàm buttun 1\_click duyệt các phần tử trong listline, hiển thị chuỗi = và kết quả đã tính toán .

1. Bài 4 - Đọc ghi file sử dụng BinaryFormatter

A screenshot of a computer

Description automatically generated

* Bố cục: Phần bên pahir là các ô textboc để nhập giá trị , button viết nội dung xuống file, và add hiển thị cho người dùng xem, cột bên trái là đọc file và hiển thị các mục đã nhập tương ứng.
* Ý tưởng: người dùng nhập nội dung tương ứng theo form phần bên tría, click add, hiển thị ra richtextbox, click write a file để viết xuống file, phần bên phải có nút để đọc nội dung file , nút chuyển next, back , hiện nội dung từng học sinh.

A computer screen shot of a program

Description automatically generated

* Tạo class Student có các thuộc tính như, tên mssv, Phone, điểm 1 2 3 điểm tủng bình.
* Phương thức khởi taok
* Hàm tính trung bình cộng điểm

A computer screen with text and symbols

Description automatically generated with medium confidence

* Tạo danh sách chứa các đối tượng student, và lastStudent .

A computer screen shot of text

Description automatically generated

* Theo thứ tự sắp xếp của các textbox để người dùng nhập là id,sdt,diem,
* Xét điều kiện mssv phải có 8 kí tự nếu không báo lỗi
* Xét điều kiện số điện thoại phải bắt đầu bằng số 0 và có 10 chữ số,nếu không báo lỗi
* Xét điều kiện điểm >0 <10 nếu không báo lỗi.
* Nếu thảo các điều kiện trên:
* Tạo đối tượng Student tmp để lấy giá trị các thuộc tính từ người dùng nhập vào . add đối tượng vào students.
* Thực hiện + format các thuộc tính vào chuỗi textrich để hiện ra richtextbox
* Cộng biến dem thêm 1 để xác định đối tượng Student tiếp theo.

A screenshot of a computer program

Description automatically generated

* try catch Sử dụng cấu trúc try-catch để xử lý các ngoại lệ có thể xảy ra trong quá trình ghi dữ liệu vào tệp tin.
* using (StreamWriter writer = new StreamWriter(filePath)) Sử dụng câu lệnh using để tạo một đối tượng StreamWriter để ghi dữ liệu vào tệp tin được chỉ định bởi đường dẫn filePath. Việc sử dụng using đảm bảo rằng tài nguyên của StreamWriter sẽ được giải phóng sau khi kết thúc phạm vi của khối using, ngay cả khi có ngoại lệ xảy ra.
* foreach (Student student in students) Duyệt qua mỗi đối tượng Student trong danh sách students.
* writer.WriteLine($"{student.Name},{student.MSSV},{student.Phone},{student.Course1},{student.Course2},{student.Course3}");: Ghi thông tin của mỗi đối tượng Student vào tệp tin dưới dạng một dòng trong định dạng chuỗi. Mỗi thuộc tính của đối tượng Student được phân tách bằng dấu phẩy.
* MessageBox.Show: Hiển thị hộp thoại thông báo cho người dùng khi quá trình ghi dữ liệu hoàn thành mà không có lỗi.
* MessageBox.Show: Hiển thị hộp thoại thông báo cho người dùng khi có lỗi xảy ra trong quá trình ghi dữ liệu vào tệp tin.

A computer screen shot of a program

Description automatically generated

* Khởi tạo hàm kiểu danh sách trả về 1 list là students.
* Sử dụng StreamReader để đọc file.
* Vòng lặp while, điều kiện dừng là đọc hết file reader.ReadLine() !null
* Với format khi write xuống file là các thuộc tính cách nhau bởi dấu ,
* Có 6 thuộc tính, parts.length =6, trong string sẽ là thứ tự các thuộc tính, name,mssv,phone,course1,2,3. Khởi tạo đối tượng đồng thời gán vào list students, đối tượng này sẽ có các thuộc tính đã được gán ở trên.
* Sau khi đọc thành công in thông báo , nếu không thông báo lỗi.
* Trả về chuỗi students.
* Kết quả của hàm là các list gồn các dối tượng từ trong file đã đọc.

A computer screen shot of a program

Description automatically generated

A computer screen shot of a program

Description automatically generated

* Gán danh sách trả về từ hàm trên đọc từ file input4.txt vào danh sách đối tượng lastStudent.
* Hàm button\_click thực hiện đọc file từ input4.txt đồng thời kiểm tra xem danh sách thu được là laststudent có rỗng ko, nếu không thực hiện hiển thị student ở vị trí 0 là student đầu tiên theo format ở hàm Displaystudent()
* Button next, hiển thị đối tượng tiếp thoe trong laststudent và đổi giá trị cảu label hiển thị index đối tượng, tương tự với button back , đi từ lùi từ đối tượng

7. bài 07 – Duyệt thư mục

A screenshot of a computer

Description automatically generated

* Bố cục: treeview hiển thị cây thư mục, picturebox hiển thị hình ảnh.
* Ý tưởng: khi chạy chương trình sẽ thực hiện load cây thư mục lên, khi click vào 1 file ảnh bất kì thì sẽ hiện lên picturebox.

A computer screen shot of code

Description automatically generated

* if (Directory.Exists(path)) Kiểm tra xem đường dẫn path có tồn tại hay không bằng cách sử dụng phương thức Directory.Exists. Nếu đường dẫn tồn tại, các node của TreeView sẽ được tạo ra và cây thư mục sẽ được tải lên.
* TreeNode root = new TreeNode() { Text = path, Tag = path };: Tạo một nút gốc mới cho TreeView với dữ liệu tương ứng với đường dẫn path. Thuộc tính Text được thiết lập là path, là tên của thư mục gốc, và thuộc tính Tag được thiết lập là path, lưu trữ đường dẫn tuyệt đối của thư mục gốc.
* treeView1.Nodes.Add(root);: Thêm nút gốc vào TreeView.
* loadDrive(root);: Gọi phương thức loadDrive để tải thư mục con của thư mục gốc và các thư mục con của chúng vào cây.
* treeView1.AfterSelect += treeView1\_AfterSelect;: Gán sự kiện AfterSelect của TreeView vào phương thức treeView1\_AfterSelect, để xử lý sự kiện khi một nút được chọn trên TreeView.

A computer screen shot of a program

Description automatically generated

* Phương thức DisplayFile(string filepath):
* + Đây là một phương thức private, nó nhận đầu vào là đường dẫn của file (filepath) cần hiển thị.
* + Phương thức này cố gắng mở file hình ảnh từ đường dẫn được cung cấp bằng cách sử dụng phương thức Image.FromFile(filepath). Nếu file hình ảnh tồn tại và có thể mở được, nó sẽ được load vào một đối tượng hình ảnh (Image).
* + phương thức này hiển thị hình ảnh với tỷ lệ zoom thông qua thuộc tính SizeMode của nó.
* + hình ảnh được gán vào pictureBox1 bằng cách sử dụng thuộc tính Image của nó.
* Hàm treeView1\_AfterSelect:
* Đây là một sự kiện được kích hoạt sau khi một node trong treeView1 được chọn.
* Kiểm tra xem node được chọn có chứa dữ liệu (Tag) không (e.Node.Tag != null), và nếu có và đường dẫn của file đó tồn tại (File.Exists(e.Node.Tag.ToString())), thì sẽ gọi phương thức DisplayFile với đường dẫn của file đó.

A screen shot of a computer program

Description automatically generated

* Biến string path = @"D:": Biến này chứa đường dẫn của ổ đĩa hoặc thư mục cần được duyệt.
* Phương thức loadDrive(TreeNode root):
* Đây là một phương thức để tải cây thư mục từ một nút gốc (root).
* Nếu root là null, phương thức sẽ kết thúc.
* Trong phương thức này, một đối tượng DirectoryInfo được tạo để đại diện cho thư mục được chỉ định bởi root.Tag.
* Các file trong thư mục đó được liệt kê bằng phương thức GetFiles() và mỗi file được thêm vào cây dưới dạng một TreeNode với Tag là đường dẫn đầy đủ của file.
* Tương tự, các thư mục con được liệt kê bằng phương thức GetDirectories() và mỗi thư mục con được thêm vào cây dưới dạng một TreeNode với Tag là đường dẫn đầy đủ của thư mục đó.
* Sau đó, phương thức này được gọi đệ quy cho mỗi thư mục con, giúp duyệt qua toàn bộ cây thư mục.
* Nếu có bất kỳ lỗi nào xảy ra trong quá trình này, nó sẽ được bỏ qua.

YÊU CẦU CHUNG

1. Đánh giá

* Chuẩn bị tốt các yêu cầu đặt ra trong bài thực hành.
* Sinh viên hiểu và tự thực hiện được bài thực hành, trả lời đầy đủ các yêu cầu đặt ra.
* Nộp báo cáo kết quả chi tiết những 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.

1. Báo cáo

* File .PDF hoặc .docx. Tập trung vào nội dung, giải thích.
* Nội dung trình bày bằng Font chữ Times New Romans/ hoặc font chữ của mẫu báo cáo này (UTM Avo)– cỡ chữ 13. Canh đều (Justify) cho văn bản. Canh giữa (Center) cho ảnh chụp.
* Đặt tên theo định dạng: LabX\_MSSV1\_MSSV2. (trong đó X là Thứ tự buổi Thực hành).

Ví dụ: Lab01\_21520001\_21520002

* Nộp file báo cáo trên theo thời gian đã thống nhất tại courses.uit.edu.vn.

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

**HẾT**