2

Lab

**BÁO CÁO BÀI THỰC HÀNH SỐ 2**

**File và I/O Stream trong C#**

**Môn học: Lập Trình Mạng Căn Bản**

|  |  |
| --- | --- |
| **Sinh viên thực hiện** | Võ Duy Hiếu (23520498) |
| **Thời gian thực hiện** | 10/10/2025 – 22/10/2025 |
| **Số câu đã hoàn thành** | 7/7 |
| **Link github source code** | <https://github.com/VDHieu0612/23520498-NT106.Q14-labs/tree/main/Lab02> |

**TRẢ LỜI CÁC CÂU HỎI**

**Gợi ý:** *Trình bày, mô tả ý tưởng chính của các bài tập. Cung cấp mã nguồn, hình ảnh kết quả minh họa.*

*Câu 1. Ghi và Đọc file*

***Trả lời:***

Ý tưởng chính: Xây dựng một ứng dụng Windows Forms có chức năng đọc nội dung một file “input1.txt” và xuất ra màn hình. Sau đó người dùng có thể ghi nội dung vào textbox sau đó chương trình chuyển toàn bộ ký tự sang kiểu in hoa và ghi xuống file “output1.txt”.

Các bước thực hiện chính:

1. Nhận sự kiện: Khi người dùng nhấp chuột vào Button "Đọc file” hoặc “Ghi File”, chương trình sẽ thực hiện việc đọc nội dung từ file đã xác định hoặc ghi nội dung vào file.
2. Đọc dữ liệu: Sử dụng hàm ReadToEnd để đọc toàn bộ dữ liệu của file input1.txt (dưới dạng string) và đẩy dữ liệu vào RichTextBox.
3. Ghi dữ liệu: Sử dụng 1 biến dạng string để gán nội dung trong RichTextBox, sau đó sử dụng StreamWritter để ghi nội dung xuống file output1.txt
4. Xác thực dữ liệu: Sử dụng phương thức File.Exists(FilePath) để kiểm tra sự tồn tại của 2 file input1.txt và output1.txt (xem đã gán đúng đường dẫn chưa) và sử dụng string.IsNullOrWhiteSpace(content) để kiểm tra xem nội dung trong RichTextBox để ghi vào file output3.txt có rỗng hay không.
5. Tính toán: Nếu tồn tại 2 file input1.txt và output1.txt thì tiến hành đọc dữ liệu từ file input1.txt và đẩy vào RichTextBox, tương tư việc ghi dữ liệu từ RichTextBox vào file output3.txt (thêm điều kiện nội dung ghi không được rỗng).
6. Hiển thị kết quả:

* Nếu hợp lệ, hiển thị nội dung trong input1.txt vào RichTextBox, hoặc ghi dữ liệu từ RichTextBox vào file output3.txt.
* Nếu dữ liệu không hợp lệ, hiển thị một MessageBox với thông báo lỗi, ví dụ: “Không có nội dung để ghi!”.

***Test case:***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Mã TC | Mô tả | Dữ liệu đầu vào | Kết quả mong đợi | Hình ảnh kết quả |
| TC1.1 | Đọc dữ liệu từ input1.txt | Nguyen Van A  MSSV: 16520001  ATTT | Nguyen Van A  MSSV: 16520001  ATTT |  |
| TC1.2 | Ghi dữ liệu vào output1.txt | Nguyen Van A  MSSV: 16520001  ATTT | Nguyen Van A  MSSV: 16520001  ATTT |  |
| TC1.3 | Đọc dữ liệu từ input1.txt | Võ Duy Hiếu  23520498  ATTT2023.1 | Võ Duy Hiếu  23520498  ATTT2023.1 |  |
| TC1.4 | (Validation) Để trống richtextbox và ghi file |  | Hiển thị MessageBox báo lỗi: " Không có nội dung để ghi!". File output3.txt không thay đổi. |  |
| TC1.5 | (Validation) Đọc file khi file input1.txt không tồn tại |  | Hiển thị MessageBox báo lỗi: " Không tìm thấy file input1.txt!” |  |
| TC1.6 | (Validation) Ghi file khi file output1.txt không tồn tại |  | Hiển thị MessageBox báo lỗi: " Không tìm thấy file output1.txt!” |  |

*Câu 2.* *Đọc thông tin một file .txt*

***Trả lời:***

Ý tưởng chính: Xây dựng một ứng dụng Windows Forms cho phép người dùng chọn file có định dạng .txt và đọc file đó. Khi người dùng chọn xong file txt, chương trình sẽ lấy các thông tin của file như Tên file - Kích thước file - Đường dẫn Url - Số dòng, số từ, số ký tự - Hiển thị nội dung của file ra giao diện. Giao diện cũng có nút “Exit” để Thoát ứng dụng.

Các bước thực hiện chính:

1. Nhận sự kiện: Khi người dùng nhấp chuột vào Button "Read from file” chương trình bắt đầu thực thi.
2. Đọc dữ liệu: Sử dụng FileStream để mở file .txt mà người dùng chọn, sau đó sử dụng phương thức ReadToEnd() để đọc toàn bộ nội dung trong file và hiển thị vào RichTextBox.
3. Tính toán: Sử dụng .SafeFileName của OpenFileDialog để lấy tên file, .Name và .Length của FileStream để lấy lần lượt đường dẫn và kích thước của file cùng các phương thức xử lý khác để lấy số dòng, số từ, số ký tự của file.
4. Hiển thị kết quả: Nếu file đầu vào hợp lệ, hiển thị các thông tin của file (Tên file - Kích thước file - Đường dẫn Url - Số dòng, số từ, số ký tự - Nội dung của file) ra giao diện.

* ***Test case:***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Mã TC | Mô tả | Dữ liệu đầu vào | Kết quả mong đợi | Hình ảnh kết quả |
| TC2.1 | Đọc file input1.txt | Võ Duy Hiếu  23520498  ATTT2023.1 | File name: input1.txt  Size: 41 bytes  URL: D:\LTM\LAB02\Lab02\Lab02\Input-Output\input1.txt  Line count: 3  Words count: 5  Character count: 35 |  |
| TC2.2 | Đọc file output1.txt | NGUYEN VAN A  MSSV: 16520001  ATTT | File name: output1.txt  Size: 34 bytes  URL: D:\LTM\LAB02\Lab02\Lab02\Input-Output\output1.txt  Line count: 3  Words count: 6  Character count: 34 |  |

*Câu 3.* *Đọc và Ghi file và tính toán*

***Trả lời:***

Ý tưởng chính: Xây dựng một ứng dụng Windows Forms cho phép người dùng đọc nội dung từ file “input3.txt” với nội dung theo định dạng, sau đó thực hiện các phép tính và ghi kết quả xuống file “output3.txt”. Các phép tính bao gồm: cộng trừ, nhân, chia, ngoặc đơn.

Các bước thực hiện chính:

1. Nhận sự kiện: Khi người dùng nhấp chuột vào Button "Đọc", thì chương trình bắt đầu thực thi.
2. Đọc dữ liệu: Chương trình sử dụng FileStream và StreamReader để đọc toàn bộ nội dung từ file "input3.txt. Nội dung file được đọc dưới dạng chuỗi (string) và hiển thị vào RichTextBox. Sử dụng phương thức File.ReadAllLines() để đọc từng dòng biểu thức khi thực hiện tính toán.
3. Xác thực dữ liệu: Sử dụng File.Exists() để kiểm tra xem file input3.txt và output3.txt có tồn tại trên hệ thống hay không. Kiểm tra xem nội dung file input có rỗng không bằng cách sử dụng lines.Length == 0 || lines.All(string.IsNullOrWhiteSpace).

Và còn có cả kiểm tra tính hợp lệ của từng biểu thức.

1. Tính toán: Chương trình sử dụng thuật toán Stack-based Expression Evaluation với hai stack:

* Stack values: Lưu trữ các giá trị số (operands).
* Stack ops: Lưu trữ các toán tử (operators).
* Sau đó duyệt qua từng kí tự của từng dòng biểu thức trong file input, nếu hợp lệ thì sử dụng các phương thức hỗ trợ như so sánh độ ưu tiên của các toán tử hay thực hiện phép toán giữa các số cũng như khi gặp dấu đóng mở ngoặc thì xử lí các phép toán trong ngoặc trước.

1. Hiển thị kết quả:

* Nếu dữ liệu hợp lệ, mỗi biểu thức được tính toán thành công sẽ được ghi vào danh sách outputLines với định dạng: "biểu\_thức = kết\_quả".
* Nếu dữ liệu không hợp lệ (biểu thức bị lỗi), ghi vào outputLines với định dạng: "biểu\_thức = Lỗi biểu thức". Đồng thời hiển thị MessageBox với thông báo số lượng biểu thức bị lỗi: "Có [số\_lượng] biểu thức bị lỗi trong file input3.txt!”.
* ***Test case:***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Mã TC | Mô tả | Dữ liệu đầu vào | Kết quả mong đợi | Hình ảnh kết quả |
| TC3.1 | Đọc file input3.txt | 1 + 2 + 3 + 4  12 - 7 - 5 + 2 - 3  2024 - 1 - 2 + 3  222 + 333 - 444 + 1 | 1 + 2 + 3 + 4  12 - 7 - 5 + 2 - 3  2024 - 1 - 2 + 3  222 + 333 - 444 + 1 |  |
| TC3.2 | Tính toán và ghi kết quả vào file output3.txt | 1 + 2 + 3 + 4  12 - 7 - 5 + 2 - 3  2024 - 1 - 2 + 3  222 + 333 - 444 + 1 | 1 + 2 + 3 + 4 = 10  12 - 7 - 5 + 2 - 3 = -1  2024 - 1 - 2 + 3 = 2024  222 + 333 - 444 + 1 = 112 |  |
| TC3.3 | Tính toán và ghi kết quả vào file output3.txt | ( 1 + 3 ) \* 4  5 / 5 + 1 \* 5  2024 - 1 \* 2 + 2  222 + 333.3 - 444.2 + 1.9 | ( 1 + 3 ) \* 4 = 16  5 / 5 + 1 \* 5 = 6  2024 - 1 \* 2 + 2 = 2024  222 + 333.3 - 444.2 + 1.9 = 113 |  |
| TC3.4 | (Validation) Tính toán khi có một biểu thức không hợp lệ | 1 + 2 + 3 + 4  12 - 7 - 5 + 2 - 3  2024 - 1 - 2 + 3  222 + 333 - 444 + a | Hiển thị MessageBox báo lỗi: “Có 1 biểu thức bị lỗi trong file input3.txt”. File output3.txt thực hiện tính toán cho các biểu thức còn lại |  |
| TC3.5 | (Validation) Tính toán khi không có biểu thức nào |  | Hiển thị MessageBox báo lỗi: "File input3.txt rỗng!”.  File output3.txt không thay đổi. |  |

*Câu 4: Đọc và Ghi file sử dụng BinaryFormatter (JsonSerializer)*

***Trả lời:***

Xây dựng một ứng dụng Windows Forms cho phép quản lý thông tin sinh viên bao gồm: nhập thông tin sinh viên (họ tên, MSSV, số điện thoại, điểm 3 môn học), lưu trữ dữ liệu vào file text dạng mảng (input4.txt), sau đó đọc và serialize sang định dạng JSON (output4.txt) sử dụng JsonSerializer, và cho phép đọc lại dữ liệu JSON sử dụng JsonDeserializer để hiển thị. Chương trình tự động tính toán điểm trung bình và hỗ trợ hiển thị danh sách sinh viên cùng chức năng phân trang để xem chi tiết từng sinh viên.

Các bước thực hiện chính:

1. Nhận sự kiện: Chương trình xử lý các sự kiện sau:

* Sự kiện "Thêm sinh viên": Khi người dùng nhấp chuột vào Button "Add", chương trình bắt đầu thực thi chức năng thêm sinh viên vào danh sách tạm (List<Student> trong bộ nhớ) và đồng thời ghi ngay thông tin sinh viên đó vào file input4.txt dưới dạng text với các trường phân cách bởi dấu |.
* Sự kiện "Ghi file": Khi người dùng nhấp chuột vào Button "Write to a file", chương trình đọc toàn bộ dữ liệu từ file input4.txt, tính điểm trung bình cho từng sinh viên, sử dụng JsonSerializer để chuyển đổi danh sách sinh viên thành chuỗi JSON và lưu vào file output4.txt.
* Sự kiện "Đọc file": Khi người dùng nhấp chuột vào Button "Read from a file", chương trình sử dụng JsonSerializer.Deserialize() để chuyển đổi chuỗi JSON thành List<Student> và hiển thị thông tin sinh viên đầu tiên trên giao diện.
* Sự kiện "Phân trang": Khi người dùng nhấp các Button "Back" hoặc "Next", chương trình thay đổi biến currentIndex và hiển thị thông tin sinh viên tương ứng trong danh sách.

1. Đọc dữ liệu:

* Đọc từ TextBox: Chương trình đọc dữ liệu dạng chuỗi (string) từ thuộc tính Text của các TextBox: txtName (họ tên), txtID (MSSV), txtPhone (số điện thoại), txtCourse1, txtCourse2, txtCourse3 (điểm 3 môn học).
* Đọc từ file text (input4.txt): Sử dụng File.ReadAllLines(inputPath, Encoding.UTF8) để đọc tất cả các dòng trong file (mỗi dòng là thông tin của 1 sinh viên) và parse điểm từ string sang float bằng float.Parse().
* Đọc từ file JSON (output4.txt): Sử dụng File.ReadAllText(outputPath, Encoding.UTF8) để đọc toàn bộ nội dung JSON sau đó sử dụng JsonSerializer.Deserialize<List<Student>>(json) để chuyển đổi chuỗi JSON thành đối tượng List.

1. Xác thực dữ liệu:

* Kiểm tra dữ liệu rỗng: Sử dụng string.IsNullOrWhiteSpace() để đảm bảo tất cả các trường (họ tên, MSSV, số điện thoại, điểm 3 môn) đều được nhập đầy đủ.
* Kiểm tra MSSV hợp lệ: Sử dụng phương thức .Text.Length để kiểm tra độ dài và .Text.All(char.IsDigit) để đảm bảo MSSV gồm đúng 8 chữ số.
* Kiểm tra số điện thoại hợp lệ: Sử dụng các điều kiện: Text.Length, Text.All(char.IsDigit), Text[0], để kiểm tra tính hợp lệ của số điện thoại gồm 10 chữ số và bắt đầu từ chữ số 0.
* Kiểm tra điểm hợp lệ:
* Sử dụng float.TryParse() để kiểm tra xem chuỗi nhập vào có thể chuyển đổi thành số thực (float) được không.
* Kiểm tra khoảng giá trị: c1 < 0 || c1 > 10 (tương tự cho c2, c3) để đảm bảo điểm nằm trong khoảng từ 0 đến 10.
* Kiểm tra file tồn tại: Sử dụng File.Exists() để kiểm tra file input4.txt có tồn tại trước khi đọc.
* Kiểm tra danh sách rỗng: Trước khi ghi file, kiểm tra students.Count == 0 để đảm bảo có dữ liệu trước khi xử lý.

1. Tính toán: Chương trình tự động tính điểm trung bình thông qua property Average trong class Student: public float Average => (Course1 + Course2 + Course3) / 3;
2. Hiển thị kết quả:

* Nếu dữ liệu sinh viên hợp lệ, nhấn “Add” sẽ thêm thông tin sinh viên đó vào 1 RichTextBox, đồng thời ghi vào file input4.txt dưới dạng mảng (định dạng: Name|ID|Phone|Course1|Course2|Course3).
* Nếu danh sách sinh viên không trống và hợp lệ, nhấn “Write to file” để ghi danh sách sinh viên vào file output4.txt dưới dạng JSON.
* Nếu file output4.txt tồn tại và có chứa dữ liệu dạng JSON của danh sách sinh viên, nhấn “Read from file” sẽ đọc file output4.txt, sử dụng JsonDeserializer và hiển thị thông tin vào các textbox tương ứng trên giao diện, ngoài ra có nút “Back” và nút “Next” để hiển thị từng sinh viên trong danh sách.
* Nếu không điền đẩy đủ thông tin sinh viên, hiển thị một MessageBox với thông báo lỗi, ví dụ: “Vui lòng nhập đầy đủ thông tin!”
* ***Test case:***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Mã TC | Mô tả | Dữ liệu đầu vào | Kết quả mong đợi | Hình ảnh kết quả |
| TC4.1 | Thêm 1 sinh viên vào danh sách tạm và ghi vào input.txt | Họ tên: Võ Duy Hiếu  ID: 23520498  Phone: 0812375179  Course 1: 9  Course 2: 10  Course 3: 9.5 | Kết quả:  Họ tên: Võ Duy Hiếu  MSSV: 23520498  SĐT: 0812375179  Điểm: 9, 10, 9.5  Trung bình: 9.50  Ghi thông tin vào input4.txt |  |
| TC4.2 | Thêm 2 sinh viên vào danh sách tạm và ghi vào input.txt | Họ tên: Võ Duy Hiếu  ID: 23520498  Phone: 0812375179  Course 1: 9  Course 2: 10  Course 3: 9.5  Họ tên: Võ Hiếu  ID: 23520499  Phone: 0812375178  Course 1: 9  Course 2: 9  Course 3: 9.5 | Kết quả:  Họ tên: Võ Duy Hiếu  MSSV: 23520498  SĐT: 0812375179  Điểm: 9, 10, 9.5  Trung bình: 9.50  Ghi thông tin vào input4.txt |  |
| TC4.3 | Ghi danh sách sinh viên vào file output4.txt | Họ tên: Võ Duy Hiếu  ID: 23520498  Phone: 0812375179  Course 1: 9  Course 2: 10  Course 3: 9.5  Họ tên: Võ Hiếu  ID: 23520499  Phone: 0812375178  Course 1: 9  Course 2: 9  Course 3: 9.5 | Kết quả:  Thông tin của 2 sinh viên dưới dạng Json |  |
| TC4.4 | Đọc danh sách sinh viên từ file | File output4.txt như ở TC4.3 | Kết quả:  Họ tên: Võ Duy Hiếu  ID: 23520498  Phone: 0812375179  Course 1: 9  Course 2: 10  Course 3: 9.5  Họ tên: Võ Hiếu  ID: 23520499  Phone: 0812375178  Course 1: 9  Course 2: 9  Course 3: 9.5 |  |
| TC4.5 | (Validation) Ghi vào file ouput4.txt khi danh sách sinh viên rỗng |  | Hiển thị MessageBox báo lỗi: “Danh sách sinh viên trống. Vui lòng thêm sinh viên trước!”. File output4.txt không thay đổi |  |
| TC4.6 | (Validation) Để trống ô họ tên và thêm sinh viên | Họ tên: ``  ID: 23520498  Phone: 0812375179  Course 1: 9  Course 2: 10  Course 3: 9.5 | Hiển thị MessageBox báo lỗi: “Vui lòng nhập đầy đủ thông tin!”. Danh sách sinh viên không thay đổi |  |
| TC4.7 | (Validation) Nhập sai định dạng ID | Họ tên: Hiếu  ID: 2345  Phone: 0812375179  Course 1: 9  Course 2: 10  Course 3: 9.5 | Hiển thị MessageBox báo lỗi: “MSSV phải gồm 8 chữ số!”. Danh sách sinh viên không thay đổi |  |
| TC4.7 | (Validation) Nhập sai định dạng Phone | Họ tên: Hiếu  ID: 23520498  Phone: 1812375179  Course 1: 9  Course 2: 10  Course 3: 9.5 | Hiển thị MessageBox báo lỗi: “Số điện thoại phải có 10 chữ số và bắt đầu bằng 0!”. Danh sách sinh viên không thay đổi |  |

*Câu 5: Quản lý phòng vé (phiên bản số 2)*

***Trả lời:***

Ý tưởng chính: Xây dựng một ứng dụng Windows Forms hỗ trợ rạp phim trong việc phân phối vé phim, tính toán tiền vé xem phim, lưu trữ dữ liệu giao dịch và thống kê doanh thu. Chương trình cho phép người dùng nhập họ tên, chọn tên phim, phòng chiếu, số ghế và xác nhận thanh toán. Sau đó, chương trình sẽ xuất ra hóa đơn trên màn hình về thông tin khách hàng bao gồm: họ và tên, vé đã chọn, tên phim, phòng chiếu, số tiền cần phải thanh toán. Các vé đã được mua sẽ không thể được mua tiếp và dữ liệu được lưu trữ vào file để sử dụng lại trong các lần chạy chương trình sau. Ngoài ra, ứng dụng còn hỗ trợ xuất file doanh thu phòng vé để giúp người quản lí thống kê.

Các bước thực hiện chính:

1. Nhận sự kiện: Khi người dùng nhấp chuột vào ComboBox và chọn phim cùng với phòng chiếu và số ghế, sau đó nhấp “Thanh toán” chương trình bắt đầu thực thi. Ngoài ra, khi nhấp “Xem thống kê” sẽ xuất ra file thống kê doanh thu phòng vé hiện tại, cũng như khi nhấp “Reset All” sẽ làm mới file tickets (vé đặt) và file output5.txt (doanh thu).
2. Đọc dữ liệu:

* Đọc dữ liệu phim từ file input5.txt: Chương trình đọc thông tin các phim bao gồm tên phim, giá vé cơ bản, và danh sách phòng chiếu từ file văn bản, sau đó nạp danh sách phim vào combobox để lựa chọn.
* Đọc dữ liệu vé đã bán từ file tickets.txt: Load thông tin các ghế đã được đặt từ các phiên làm việc trước để hiển thị trạng thái ghế đã bán.
* Chương trình đọc dữ liệu chuỗi từ thuộc tính Text của Textbox (txtHoTen) và lấy giá trị được chọn từ các ComboBox (cbChonPhim, cbPhongChieu) để xử lý.

1. Xác thực dữ liệu: Chương trình kiểm tra xem người dùng đã nhập đầy đủ thông tin về họ tên, lựa chọn phim và phòng chiếu cũng như chọn số ghế chưa, nếu đã đầy đủ, xuất hoá đơn gồm các thông tin liên quan, nếu chưa thì hiển thị thông báo lỗi. Ngoài ra, các ghế của các phòng chiếu đã được chọn người dùng sẽ không thể chọn được nữa. Và người dùng không thể nhập được phim khác cũng như không chọn được phòng chiếu khác mà không chiếu phim đã chọn. Ngoài ra, còn có phương thức kiểm tra file có tồn tại hay không và phương thức xác thực định dạng phim trong file input có hợp lệ không.
2. Tính toán: Tính giá vé dựa trên loại ghế và giá cơ bản của phim.

* Ghế góc (A1, A5, C1, C5) có giá bằng ¼ giá vé gốc.
* Ghế trung tâm (B2, B3, B4) có giá gấp đôi.
* Các ghế còn lại giữ nguyên giá.
* Tổng tiền được tính bằng cách cộng giá của các ghế đã chọn.

1. Hiển thị kết quả:

* Nếu dữ liệu về phim, phòng vé, giá cơ bản hợp lệ, sẽ đẩy dữ liệu từ file input5.txt lên.
* Nếu dữ liệu đặt vé hợp lệ, hiển thị kết quả hoá đơn vào một TextBox. Ghi dữ liệu vé đã được đặt vào file tickets.txt và xuất ra doanh thu phòng vé ra output5.txt khi cần thống kê.
* Nếu không điền đẩy đủ thông tin, hiển thị một MessageBox với thông báo lỗi, ví dụ: “Vui lòng nhập đủ thông tin và chọn ghế!”.
* ***Test case:***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Mã TC | Mô tả | Dữ liệu đầu vào | Kết quả mong đợi | Hình ảnh kết quả |
| TC4.1 | Nhập họ tên và chọn phim, phòng chiếu, ghế | Họ tên: Võ Hiếu  Phim: Đào, phở và piano  Phòng chiếu: 1  Ghế: A1, A2 | Kết quả:  Họ tên: Võ Hiếu  Phim: Đào, phở và piano  Phòng chiếu: 1  Ghế: A1 – 10000đ  A2 – 40000đ  Tổng tiền: 50000đ |  |
| TC4.2 | Nhập họ tên và chọn phim, phòng chiếu, ghế | Họ tên: Võ Hiếu  Phim: Mai  Phòng chiếu: 2  Ghế: A3 | Kết quả:  Họ tên: Võ Hiếu  Phim: Mai  Phòng chiếu: 2  Ghế:  A3 – 100000đ  Tổng tiền: 100,000đ |  |
| TC4.3 | Nhập họ tên và chọn phim, phòng chiếu, ghế | Họ tên: Võ Hiếu  Phim: Mai  Phòng chiếu: 3  Ghế: A1, B1, B2 | Kết quả:  Họ tên: Võ Hiếu  Phim: Mai  Phòng chiếu: 3  Ghế:  A1 - 25,000đ  B2 - 200,000đ  B1 - 100,000đ  Tổng tiền: 325,000đ |  |
| TC4.4 | Xem thống kê | File output5.txt | Kết quả:  Thống kê doanh thu |  |
| TC4.4 | (Validation) Chọn ghế đã đươc đặt trước | Họ tên: Võ Hiếu  Phim: Mai  Phòng chiếu: 2  Ghế: A1, A2 | Hiển thị các ghế đã được chọn màu đỏ  Không thể nhấp chọn ghế |  |
| TC4.5 | (Validation) Để trống ô họ tên | Họ tên: ``  Phim: Tarot  Phòng chiếu: 3  Ghế: B1, B2 | Hiển thị MessageBox báo lỗi: “Vui lòng nhập đủ thông tin và chọn ghế!”. Ô kết quả và file tickets không thay đổi. |  |
| TC4.6 | (Validation) Không chọn ghế | Họ tên: Hiếu  Phim: Mai  Phòng chiếu: 2  Ghế: `` | Hiển thị MessageBox báo lỗi: “Vui lòng nhập đủ thông tin và chọn ghế!”. Ô kết quả và file tickets không thay đổi. |  |

*Câu 6: Hôm nay ăn gì? (phiên bản số 2)*

***Trả lời:***

Xây dựng một ứng dụng Windows Forms quản lý danh sách món ăn với cơ sở dữ liệu SQLite. Chương trình cho phép người dùng thêm món ăn mới (tên món, hình ảnh, người đóng góp), hiển thị danh sách món ăn đã lưu, và chức năng ngẫu nhiên chọn một món ăn để gợi ý "Hôm nay ăn gì?". Cơ sở dữ liệu bao gồm 2 bảng có quan hệ: NguoiDung (IDNCC, HoVaTen, QuyenHan) và MonAn (IDMA, TenMonAn, HinhAnh, IDNCC - khóa ngoại).

Các bước thực hiện chính:

1. Nhận sự kiện: Sự kiện Load Form: Khi form được tải, chương trình tự động tạo cơ sở dữ liệu (nếu chưa có), tải danh sách người dùng vào ComboBox và hiển thị danh sách món ăn trong ListView. Khi nhấp vào button “Chonj”, mở hộp thoại OpenFileDialog để người dùng chọn file hình ảnh. Khi nhấp vào button “Thêm món”, chương trình validate dữ liệu và thêm món ăn mới vào cơ sở dữ liệu SQLite. Khi nhấp vào button btn “Hiển thị DS”, tải lại danh sách món ăn từ database. Khi nhấp vào button “Random”, chương trình chọn ngẫu nhiên một món ăn và hiển thị thông tin.
2. Đọc dữ liệu:

* Đọc từ TextBox: Lấy tên món ăn từ txtTenMon.Text
* Đọc từ ComboBox: Lấy người đóng góp đã chọn từ cbNguoiDung.SelectedItem
* Đọc từ OpenFileDialog: Lấy đường dẫn hình ảnh từ dlg.FileName sau khi người dùng chọn file hình ảnh
* Đọc từ SQLite: Sử dụng SQLiteDataReader để đọc tuần tự từng dòng dữ liệu
* Sử dụng SQLiteDataAdapter để đổ dữ liệu vào DataTable phục vụ random

1. Xác thực dữ liệu:

* Kiểm tra tên món ăn: Sử dụng string.IsNullOrWhiteSpace(txtTenMon.Text) để đảm bảo người dùng đã nhập tên món
* Kiểm tra hình ảnh: Sử dụng string.IsNullOrEmpty(selectedImagePath) để đảm bảo đã chọn ảnh
* Kiểm tra người đóng góp: Kiểm tra cbNguoiDung.SelectedItem == null để đảm bảo đã chọn người dùng
* Kiểm tra file tồn tại: Sử dụng File.Exists(anh) trước khi load hình ảnh
* Kiểm tra database rỗng: Kiểm tra dt.Rows.Count == 0 trước khi random món ăn

1. Tính toán:

* Tạo cơ sở dữ liệu (phương thức CreateDatabase()):

Tạo file SQLite nếu chưa tồn tại: SQLiteConnection.CreateFile("MonAn.db") -> Tạo bảng NguoiDung với khóa chính IDNCC (AUTO INCREMENT) -> Tạo bảng MonAn với khóa ngoại IDNCC tham chiếu đến bảng NguoiDung -> Thêm dữ liệu mẫu nếu bảng NguoiDung rỗng.

* Thêm món ăn (phương thức btnAdd\_Click()):

Sử dụng parameterized query để tránh SQL Injection -> Thực thi lệnh INSERT bằng cmd.ExecuteNonQuery() -> Làm mới danh sách và xóa form sau khi thêm thành công.

* Xoá món ăn (phương thức btnDelete\_Click()):

Dùng parameterized query để tránh SQL Injection -> Thực thi lệnh xóa bằng cmd.ExecuteNonQuery() -> Sau khi xóa, làm mới danh sách bằng LoadMonAn() để cập nhật giao diện.

* Random món ăn (phương thức btnRandom\_Click()):

Load toàn bộ món ăn vào DataTable -> Sử dụng Random.Next(dt.Rows.Count) để chọn ngẫu nhiên index -> Truy xuất thông tin từ dòng được chọn và hiển thị.

1. Hiển thị kết quả:

* Hiển thị danh sách người dùng trong ComboBox.
* Hiển thị danh sách món ăn trong ListView.
* Hiển thị hình ảnh.
* Hiển thị kết quả random: Hiển thị tên món ăn trong TextBox: txtKetqua.Text = ten, hình ảnh món ăn trong PictureBoxbà thông tin người đóng góp được lấy từ query JOIN.
* ***Test case:***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Mã TC | Mô tả | Dữ liệu đầu vào | Kết quả mong đợi | Hình ảnh kết quả |
| TC8.1 | Nhập và thêm các món ăn hợp lệ | Món ăn: Cơm tấm, Phở bò, Bún riêu | Kết quả:  Danh sách có thêm món “Cơm tấm, Phở bò, Bún riêu” |  |
| TC8.2 | Tìm ngẫu nhiêu món ăn hôm nay từ danh sách món ăn | Danh sách có: Cơm tấm, Phở bò, Bún riêu | Kết quả:  Ngẫu nhiên trong 3 món đã có trong danh sách |  |
| TC8.3 | Chọn và xoá món ăn trong danh sách | Món ăn: Cơm tấm | Kết quả:  Danh sách còn 2 món Phở bò và Bún riêu |  |
| TC8.4 | (Validation) Nhập và thêm món ăn không hợp lệ | Món ăn: C@m1 | Hiển thị MessageBox báo lỗi: “Tên món ăn chỉ được chứa chữ cái, số và khoảng trắng, không có ký tự đặc biệt!”. Danh sách món ăn không thay đổi. |  |
| TC8.5 | (Validation) Để trống ô món ăn | Món ăn: `` | Hiển thị MessageBox báo lỗi: “Vui lòng nhập tên món ăn!". Danh sách món ăn không thay đổi. |  |
| TC8.6 | (Validation) Tìm món ăn khi danh sách trống |  | Hiển thị MessageBox báo lỗi: "Chưa có món ăn nào trong cơ sở dữ liệu!". Ô kết quả (món ăn ngẫu nhiên) không thay đổi. |  |

*Câu 7:*  *Duyệt thư mục*

***Trả lời:***

Ý tưởng chính: Xây dựng một ứng dụng Windows Forms cho phép người dùng duyệt toàn bộ cấu trúc thư mục và file trên máy tính theo dạng cây phân cấp (TreeView). Chương trình hiển thị tất cả các ổ đĩa (drives), thư mục (folders) và file, cho phép mở rộng thư mục khi nhấp đúp chuột, và tự động hiển thị nội dung file khi người dùng chọn file. Các loại file được hỗ trợ hiển thị bao gồm: hình ảnh (jpg, png, bmp) hiển thị bằng PictureBox, file văn bản (txt, cs, log) hiển thị bằng RichTextBox, và thông báo cho các loại file khác không được hỗ trợ.

Các bước thực hiện chính:

1. Nhận sự kiện: Chương trình xử lý các sự kiện chính sau:

* Sự kiện khởi tạo Form (Lab02\_Bai07()): Khi form được tạo, chương trình tự động gọi phương thức LoadDrives() để tải danh sách các ổ đĩa có trên máy tính.
* Sự kiện BeforeExpand (TreeView1\_BeforeExpand): Khi người dùng nhấp đúp chuột hoặc nhấn vào dấu "+" để mở rộng một node trong TreeView, sự kiện này được kích hoạt. Chương trình sẽ tải và hiển thị tất cả các thư mục con và file bên trong thư mục được chọn.
* Sự kiện AfterSelect (TreeView1\_AfterSelect): Khi người dùng nhấp chọn một node (file hoặc thư mục) trong TreeView, sự kiện này được kích hoạt. Chương trình sẽ kiểm tra loại node và hiển thị nội dung file tương ứng trong GroupBox nếu là file.

1. Đọc dữ liệu:

* Đọc danh sách ổ đĩa: Trong phương thức LoadDrives(): Sử dụng DriveInfo.GetDrives() để lấy danh sách tất cả các ổ đĩa (drive) có trên máy tính (C:, D:, E:, v.v.)
* Đọc cấu trúc thư mục và file: Trong phương thức TreeView1\_BeforeExpand(): Đọc thư mục con: Sử dụng Directory.GetDirectories(path) để lấy danh sách tất cả các thư mục con trong đường dẫn hiện tại. Đọc file: Sử dụng Directory.GetFiles(path) để lấy danh sách tất cả các file trong đường dẫn hiện tại. Lấy tên file/thư mục: Sử dụng Path.GetFileName(dir) hoặc Path.GetFileName(file) để trích xuất tên file/thư mục từ đường dẫn đầy đủ.
* Đọc nội dung file:
* Trong phương thức TreeView1\_AfterSelect(): Kiểm tra đường dẫn: Sử dụng File.Exists(path) để xác định node được chọn có phải là file hay không. Đọc file văn bản: Sử dụng File.ReadAllText(path) để đọc toàn bộ nội dung file text thành một chuỗi string duy nhất. Đọc file hình ảnh: Sử dụng Image.FromFile(path) để tải hình ảnh từ file và chuyển thành đối tượng Image để hiển thị.

1. Xác thực dữ liệu: Kiểm tra loại file: Trong phương thức TreeView1\_AfterSelect(): Kiểm tra file tồn tại: Sử dụng File.Exists(path) trước khi cố gắng đọc file. Xác định phần mở rộng: Sử dụng Path.GetExtension(path).ToLower() để lấy phần mở rộng của file và chuyển về chữ thường để so sánh không phân biệt hoa/thường.
2. Tính toán:

* Xây dựng cấu trúc cây: Tạo TreeNode cho mỗi ổ đĩa/thư mục/file -> Lưu đường dẫn đầy đủ vào thuộc tính Tag -> Thêm node giả "..." để hiển thị dấu "+" cho thư mục
* Lazy Loading: Chỉ tải nội dung thư mục khi người dùng mở rộng, tiết kiệm bộ nhớ và thời gian
* Xóa và tải lại: Khi mở rộng node, xóa node giả bằng node.Nodes.Clear() và thay bằng nội dung thực

1. Hiển thị kết quả:

* Hiển thị file hình ảnh: Tạo PictureBox với Dock = DockStyle.Fill và sử dụng SizeMode.Zoom để tự động scale giữ tỷ lệ
* Hiển thị file văn bản: Tạo RichTextBox với Dock = DockStyle.Fill và hiển thị toàn bộ nội dung file với khả năng cuộn
* Hiển thị file không hỗ trợ: Tạo Label thông báo "Không thể hiển thị loại file này."
* Xóa nội dung cũ: Luôn gọi groupBox1.Controls.Clear() trước khi thêm control mới
* ***Test case:***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Mã TC | Mô tả | Dữ liệu đầu vào | Kết quả mong đợi | Hình ảnh kết quả |
| TC7.1 | Mở form Lab02\_Bai07 |  | Kết quả:  C:\  D:\  E:\ |  |
| TC7.2 | Mở ổ đĩa D | D:\ | Danh sách cái thư mục trong ổ đĩa D |  |
| TC7.3 | Mở file txt | File input1.txt:  Võ Duy Hiếu  23520498  ATTT2023.1 | Kết quả:  Võ Duy Hiếu  23520498  ATTT2023.1 |  |
| TC7.4 | Mở file png | File CauB21\_6.png | Kết quả:  Hình ảnh |  |
| TC7.5 | (Validation) Mở file docx | 23520498\_Lab02\_Report.docx | Hiển thị t hông báo "Không thể hiển thị loại file này." trong file content |  |