

Đề tài dự án và phân tích bài toán cho phát triển ứng dụng

Hướng dẫn tự học lập trình C# toàn tập > Đề tài dự án và phân tích bài toán cho phát triển ứng dụng

Trong bài này, chúng ta sẽ đưa ra đề tài cho dự án và tiến hành phân tích nhằm làm rõ các yêu cầu về chức năng và dữ liệu của phần mềm. Các phân tích ở đây sẽ được sử dụng làm cơ sở cho việc phát triển ứng dụng trong suốt quá trình học.

NỘI DUNG CỦA BÀI [Ấn]

1. Các vấn đề thường gặp khi học lập trình C#
 - 1.1. Không biết cách vận dụng và kết hợp các kỹ thuật đã biết khi giải quyết bài toán
 - 1.2. Mắc lỗi trong quá trình xây dựng và sử dụng class
 - 1.3. Không biết cách tổ chức code của chương trình
 - 1.4. Không biết bắt đầu từ đâu
2. Đề tài dự án
3. Phân tích tổng quát
 - 3.1. Các tình huống sử dụng (use case)
 - 3.2. Thông tin cần quản lý
4. Phân tích chi tiết
5. Kết luận

Các vấn đề thường gặp khi học lập trình C#

Khi học lập trình cơ bản, rất nhiều người thường gặp những vấn đề có liên quan trực tiếp đến việc vận dụng kiến thức và kỹ năng đã được học để giải quyết một bài toán trọn vẹn. Qua kinh nghiệm giảng dạy thực tế, chúng tôi nhận thấy một số vấn đề mà người học thường gặp phải.

Không biết cách vận dụng và kết hợp các kỹ thuật đã biết khi giải quyết bài toán

Người học chủ yếu thường được giới thiệu các khái niệm và kỹ thuật lập trình thông qua các ví dụ nhỏ tách rời. Cách thức này có hiệu quả để người học hiểu riêng những vấn đề đang được trình bày.

Tuy nhiên, khi gặp các tình huống thực tế, người học thường không biết cách áp dụng các kỹ thuật mình đã biết, đặc biệt là cách phối hợp các kỹ thuật để giải quyết vấn đề đang gặp phải.

Với cách tiếp cận của bài giảng này, người học bên cạnh việc tiếp thu những khái niệm và kỹ thuật mới sẽ được chỉ dẫn cách vận dụng của chúng để giải quyết những vấn đề gặp phải trong một dự án nhỏ.

Mắc lỗi trong quá trình xây dựng và sử dụng class

Ví dụ, người học lập trình hướng đối tượng mặc dù hiểu được rõ ràng các khái niệm cơ bản như lớp, đối tượng, kế thừa, đa hình, nạp chồng, ghi đè, v.v. cũng như kỹ thuật thực hiện

chúng trong một ngôn ngữ lập trình cụ thể.

Tuy nhiên, đến lúc lập trình lại thường xuyên mắc những lỗi như trộn lẫn code xử lý giao diện với code xử lý logic hay dữ liệu, đưa quá nhiều code không liên quan vào một class, lạm dụng trường dữ liệu và phương thức tĩnh, v.v..

Những lỗi này xuất hiện là do người học thường không được giới thiệu về những nguyên lý cơ bản trong vận dụng lập trình hướng đối tượng, ví dụ bộ nguyên lý SOLID, các mẫu kiến trúc cho giao diện (MVC, MVP), các mẫu thiết kế cho dữ liệu (repository, unit of work), các mẫu thiết kế class (singleton, mediator, v.v.), các nguyên lý chung phổ biến (separation of concern, inversion of control, v.v.).

Nói một cách khác, đó là học lập trình hướng đối tượng nhưng chưa biết cách vận dụng của nó (học không đi đôi với hành).

Bài giảng này không hướng tới cung cấp cho người học những vấn đề chuyên sâu về cách vận dụng của các nguyên lý trên mà cố gắng vận dụng những nguyên lý trên ở những nơi phù hợp (cùng với giải thích nguyên lý) giúp người học lưu ý và bắt đầu ý thức được việc vận dụng chúng trong quá trình lập trình ứng dụng.

Không biết cách tổ chức code của chương trình

Với cách học từ những ví dụ nhỏ, code thường rất ít và được gộp trong một vài file mã nguồn. Từ đây dẫn đến những thói quen xấu trong việc tổ chức code và có hại về lâu dài. Ví dụ, người học không có thói quen tách code ra các file và thư mục (và khi muốn tách thì không biết nên tách những gì), không viết ghi chú (comment) cho các đoạn code, đặt tên file mã nguồn bất quy tắc, v.v..

Những thói quen này làm cho việc bảo trì code trở nên khó khăn hơn vì chỉ sau vài tháng thường rất khó nhớ được chi tiết những gì mình làm, khó tìm lại code theo nhu cầu, vất vả khi đọc lại code kể cả khi đọc code do chính mình viết ra.

Khi học qua một dự án với khối lượng code đủ lớn, tất cả những vấn đề nêu trên sẽ xuất hiện rất nhanh và người học bắt buộc phải vận dụng những cách thức tổ chức code phù hợp.

Không biết bắt đầu từ đâu

Khi được giao xử lý một vấn đề, người học thường không biết bắt đầu từ đâu. Thông thường, người học thường suy nghĩ sơ lược về chức năng (chương trình sẽ làm gì), tưởng tượng ngay ra giao diện (chương trình nhìn như thế nào) và bắt tay thẳng vào code. Trong quá trình code sẽ tiếp tục suy nghĩ chương trình sẽ làm gì tiếp theo, và liên tục sửa code theo suy nghĩ của mình.

Thói quen xấu này làm việc giải quyết bài toán mất nhiều công sức hơn, kết quả thu được thường khá lộn xộn và ít có khả năng cải tiến tiếp.

Phần thực hành làm project trong bài giảng này sẽ cố gắng giúp bạn biết cách bắt đầu giải quyết một bài toán mà không sa đà lạc lối ngay vào code.

Đề tài dự án

Trong thời gian học đại học, tôi thường có thói quen sưu tầm sách điện tử, chủ yếu là các file e-book ở dạng .pdf, .epub, .djvu, .chm, và thuộc nhiều thể loại khác nhau (sách học tiếng Anh, truyện, sách thuộc các lĩnh vực chuyên môn của cá nhân, v.v.).

Qua thời gian, số lượng sách sưu tầm được lên đến khoảng 30 Gb (với hàng nghìn file), và số lượng này vẫn tiếp tục tăng lên. Số sách này được lưu trữ trên ổ đĩa cứng và có một bản backup trên dịch vụ OneDrive.

Vì tôi luôn phải cập nhật kiến thức và công nghệ mới, số lượng sách và chủng loại sách vẫn ngày một tăng lên khiến việc tìm kiếm (để đọc lại hoặc cung cấp cho người khác) những cuốn mình cần trong kho sách ngày một khó hơn.

Cũng có nhiều cuốn sách về các chủ đề hoặc công nghệ đã cũ giờ không sử dụng đến nữa cần loại bỏ cho nhẹ bớt kho (và tiết kiệm tài nguyên của OneDrive).

Ngoài ra, do sưu tầm sách từ nhiều nguồn khác nhau, cách đặt tên cho các cuốn sách cũng rất lộn xộn khiến việc tìm kiếm sách rất mất công (phải mở thử từng file một để kiểm tra). Vì vậy, tôi có nhu cầu phát triển một ứng dụng nhỏ giúp tôi quản lý được kho tài liệu trên dễ dàng hơn.

Khi bạn được giao thực hiện một bài toán phát triển ứng dụng nào đó, thông thường yêu cầu đưa ra thường rất chung chung, vì chính người đưa ra bài toán có khi vẫn chưa hiểu thấu đáo bài toán của mình.

Vì vậy, hãy tự mình suy nghĩ (theo kinh nghiệm của bản thân)/tìm kiếm trên Internet/trao đổi với chuyên gia về các vấn đề có liên quan để hiểu rõ về vấn đề (cái này người ta gọi là tìm hiểu nghiệp vụ).

Khi đã hiểu được nghiệp vụ, hãy quay lại trao đổi với người đưa ra bài toán để cả hai cùng làm rõ và thống nhất từng vấn đề.

Trong bài toán này, nghiệp vụ rất đơn giản, các yêu cầu cũng đã tương đối rõ ràng nên mọi chuyện đơn giản hơn. Nếu chưa hiểu gì về bài toán thì đừng vội nghĩ về code hay giao diện.

Phân tích tổng quát

Tuy rằng đây là một bài toán về phát triển ứng dụng nhỏ và đơn giản, hàng loạt vấn đề vẫn cần được làm rõ. Các vấn đề này có thể chia làm ba loại: các tình huống (ca) sử dụng, thông tin cần lưu trữ/xử lý, và quy trình làm việc trong từng tình huống sử dụng. Tất cả các vấn đề này liên quan đến quy trình nghiệp vụ, chưa phải là lập trình và cần thực hiện kỹ càng.

Các tình huống sử dụng (use case)

Về tình huống sử dụng, chúng ta cần xác định được những ai sẽ sử dụng phần mềm và từng nhóm người dùng đó có thể làm được những gì trên phần mềm. Đây thông thường cũng là bước đầu tiên trong phát triển ứng dụng. Việc phân tích này cho chúng ta cái nhìn tổng thể về phần mềm tương lai và chi phối các bước phân tích tiếp theo.

Cụ thể, phần mềm này sẽ chỉ dành một người sử dụng và có thể:

1. Tìm kiếm và liệt kê sách theo tiêu chí
2. Xem thông tin chi tiết
3. Bổ sung thêm sách mới
4. Cập nhật thông tin sách
5. Loại bỏ sách khỏi kho
6. Tìm kiếm và sắp xếp sách theo tiêu chí
7. Thống kê sách theo tiêu chí
8. Xuất thông tin (sau khi tìm kiếm/thống kê) ra file
9. Mở đọc file sách
10. Tự động tìm sách trong thư mục
11. Đồng bộ hóa thông tin quản lý với các file sách thực có trong thư mục
12. Sao chép một/một số file sách sang vị trí khác
13. Đổi tên file theo quy tắc
14. Mở thư mục chứa file sách

Đừng lo lắng nếu bạn chưa hiểu hết các tình huống sử dụng trên, vì chúng ta còn một bước phân tích chi tiết nữa.

Thông tin cần quản lý

Về thông tin, câu hỏi đặt ra là để quản lý một kho sách điện tử, chúng ta cần những thông tin gì. Một cuốn sách (dù là sách in hay sách điện tử) đều có thông tin sau:

- tác giả (hoặc nhóm tác giả);
- tựa đề;
- nhà xuất bản;
- năm xuất bản;
- lần tái bản;
- mã ISBN (mã số tiêu chuẩn quốc tế cho sách).

Ngoài ra, đối với sách điện tử có thể có thêm những thông tin như:

- mô tả tóm tắt nội dung (rất tiện lợi cho việc tìm kiếm);
- từ khóa mô tả nội dung/thể loại, đánh giá của cá nhân (rating, sau này có thể dùng trong sắp xếp);
- đánh dấu cuốn sách nào hiện đang đọc (để sau dễ dàng tìm đến những cuốn được đánh dấu).

Vì số lượng sách lớn và thuộc nhiều thể loại, cần có thêm một thông tin nữa, tạm gọi là “giá sách” để giúp phân loại sách, tương tự như việc mỗi file sách nằm trong một thư mục nào đó. Mỗi giá sách chứa nhiều cuốn sách; mỗi cuốn sách chỉ nằm trong một giá nào đó. Giá sách giúp gom các cuốn sách cùng loại lại để dễ dàng quản lý, tương tự như việc dùng thư mục để quản lý file.

Để có thể thực hiện quản lý, mỗi thông tin sách (dữ liệu) sẽ phải tương ứng với một file sách trong thư mục kho sách. Vì vậy, cần thêm một thông tin về đường dẫn tới file sách tương ứng. Vì mỗi dữ liệu sách tương ứng với một file sách, mỗi giá sách tương ứng với một thư mục chứa file sách, toàn bộ dữ liệu sách sẽ tương ứng và đồng bộ với thư mục kho sách.

Về mặt kỹ thuật, để có thể nhanh chóng xác định đúng cuốn sách đang cần tìm, chúng ta thêm một thông tin định danh duy nhất cho mỗi cuốn sách, gọi là Id.

Tổng kết lại, dữ liệu về mỗi cuốn sách bao gồm những trường sau:

STT	Tên	Mô tả	Kiểu	Ghi chú
1	Id	Số định danh duy nhất	Số nguyên	Bắt buộc
2	Nhóm tác giả (authors)	Danh sách tên tác giả, phân tách bởi dấu phẩy	Văn bản	Bắt buộc
3	Tiêu đề (title)	Tiêu đề sách	Văn bản	Bắt buộc
4	Nhà xuất bản (publisher)	Tên nhà xuất bản	Văn bản	Bắt buộc
5	Năm xuất bản (year)	Năm xuất bản sách	Số nguyên	Bắt buộc, phải lớn hơn 1950, nhỏ hơn năm hiện tại
6	Lần tái bản (edition)	Lần tái bản của sách	Số nguyên	Bắt buộc, phải lớn hơn hoặc bằng 1, mặc định là 1
7	Mã xuất bản (Isbn)	Mã số tiêu chuẩn quốc tế	Văn bản	Không bắt buộc
8	Từ khóa (tags)	Danh sách các từ khóa mô tả nội dung, thể loại	Văn bản	Không bắt buộc
9	Mô tả (description)	Mô tả tóm tắt nội dung	Văn bản	Không bắt buộc

10	Đường dẫn (file)	Đường dẫn (đầy đủ) tới file pdf	Văn bản	Bắt buộc, phải là một đường dẫn đúng
11	Đánh dấu đọc (reading)	Dùng để đánh dấu một cuốn sách đang đọc	Logic	Không bắt buộc, mặc định là false
12	Đánh giá (rating)	Đánh giá chất lượng cuốn sách	Số nguyên	Không bắt buộc, có giá trị từ 1 (dở nhất) đến 5 (tốt nhất), mặc định là 1

Việc phân tích về mặt dữ liệu giúp chúng ta sau này xây dựng lớp thực thể.

Phân tích chi tiết

Như ở phần phân tích về tình huống sử dụng chúng ta thấy, nếu chỉ mô tả sơ lược như vậy thì rất khó hình dung về hoạt động cụ thể của chúng. Do đó, mỗi tình huống sử dụng thường được mô tả thêm một cách kỹ lưỡng (gọi là đặc tả) bằng lời hoặc các sơ đồ trực quan để diễn tả đầy đủ quy trình, thuật toán của từng tình huống sử dụng.

Ở đây chúng ta chỉ mô tả bằng lời vì các tình huống sử dụng trên không phức tạp. Mỗi tình huống sử dụng sẽ được phân tích kỹ hơn ở phần dưới đây.

#	Tình huống sử dụng	Mô tả chi tiết
1	Tìm kiếm và liệt kê sách theo tiêu chí	Người dùng có thể liệt kê tất cả các cuốn sách đang được quản lý theo một tiêu chí cụ thể. Nếu nhập một từ khóa, chương trình phải tìm và liệt kê ra tất cả các cuốn sách mà tên tác giả, nhà xuất bản, mô tả, tag, tiêu đề chứa từ khóa đó. Nếu không nhập từ khóa nào thì sẽ liệt kê tất cả các cuốn sách. Nếu không tìm thấy cuốn sách nào phù hợp tiêu chí hoặc kho sách đang trống sẽ báo lại cho người dùng biết.
2	Xem thông tin chi tiết	Người dùng cung cấp Id của cuốn sách, chương trình sẽ hiển thị toàn bộ thông tin chi tiết của cuốn sách đó. Nếu không tìm thấy cuốn sách có Id tương ứng thì báo lỗi.
3	Bổ sung thêm sách mới	Khi cần thêm một cuốn sách mới vào kho, người dùng copy file vào nơi cần thiết và cung cấp đầy đủ thông tin về cuốn sách. Chương trình sẽ bổ sung dữ liệu về cuốn sách mới và lưu vào file dữ liệu.
4	Cập nhật thông tin sách	Người dùng cung cấp Id của cuốn sách, chương trình sẽ lần lượt hiển thị từng trường dữ liệu (cũ) và yêu cầu nhập giá trị mới. Nếu trường nào không cần cập nhật thì bỏ qua. Nếu không tìm thấy cuốn sách thì báo lỗi.
5	Loại bỏ sách khỏi kho	Người dùng cung cấp Id của cuốn sách. Chương trình tìm cuốn sách có Id tương ứng. Nếu tìm thấy sẽ xóa xóa bỏ dữ liệu của cuốn sách nhưng không xóa bỏ file tương ứng. Nếu không tìm thấy sẽ báo lỗi. Người dùng cũng có thể yêu cầu xóa bỏ toàn bộ dữ liệu sách (nhưng không xóa các file tương ứng). Trong cả hai tình huống đều yêu cầu người dùng phải xác nhận hành động trước khi thực hiện.

#	Tình huống sử dụng	Mô tả chi tiết
6	Tìm kiếm và sắp xếp sách theo tiêu chí	Tương tự như tình huống số 1 nhưng có thêm khả năng sắp xếp theo tiêu chí do người dùng cung cấp, bao gồm sắp xếp theo tên tác giả, tiêu đề, nhà xuất bản, năm xuất bản.
7	Thống kê sách theo tiêu chí	Người dùng cung cấp một tiêu chí để thống kê. Chương trình hiển thị được danh sách toàn bộ các cuốn sách theo nhóm. Các tiêu chí để nhóm bao gồm: thư mục, tác giả, nhà xuất bản, năm xuất bản.
8	Xuất thông tin (sau khi tìm kiếm/thống kê) ra file	Tình huống 1, 2, 6, 7 đều xuất thông tin ra màn hình. Để người dùng có thể dễ dàng sử dụng thông tin đó trong các ứng dụng khác, ví dụ nhập vào excel, các thông tin này có thể đồng thời phải xuất ra một file dữ liệu riêng theo các định dạng phổ biến như xml, json. Đây là chức năng bổ sung cho 1,2,6,7. Người dùng có thể lựa chọn xuất thông tin ra màn hình hoặc xuất vào file.
9	Mở đọc file sách	Chức năng này cho phép người dùng cung cấp thông tin để tìm ra một cuốn sách cụ thể và mở file bằng chương trình đọc mặc định của hệ thống. Chức năng này hoạt động theo quy trình như tình huống 2 nhưng bổ sung thêm khả năng mở file để đọc.
10	Tự động tìm sách trong thư mục	Người dùng cung cấp đường dẫn tới một thư mục, chương trình tự dò tìm các file sách có phần mở rộng pdf trong thư mục đó (và thư mục con của nó). Ứng với mỗi file, chương trình tạo ra một dữ liệu sách. Chức năng này được sử dụng để tự động tạo dữ liệu cho chương trình, hạn chế phải tạo dữ liệu thủ công. Người dùng về sau chỉ cần dùng chức năng cập nhật để điều chỉnh thông tin của sách. Nếu đường dẫn tới thư mục nguồn sai sẽ báo lỗi. Nếu không có file sách nào tìm thấy sẽ thông báo lại. Nếu chức năng chạy tốt sẽ thông báo số file tìm được.
11	Đồng bộ hóa thông tin quản lý với các file sách thực có trong thư mục	Khi tải được cuốn sách mới, người dùng có thể không có thời gian để bổ sung ngay thông tin cho chương trình quản lý mà chỉ trực tiếp copy file sách vào thư mục. Người dùng cũng có thể tự xóa bớt các file sách cũ. Chức năng này giúp tự động phát hiện những cuốn sách mới trong các thư mục. Nếu có file mới, chương trình tự động tạo ra dữ liệu tương ứng. Nếu file sách đã bị xóa, dữ liệu tương ứng của nó trong chương trình sẽ bị xóa bỏ.
12	Sao chép một/một số file sách sang vị trí khác	Khi người dùng muốn copy sách cho người khác, thay vì phải mở thư mục và copy từng file, chương trình giúp copy tất cả các file được lựa chọn tới một thư mục đích. Có hai tình huống bên trong: – Người dùng cung cấp Id và thư mục đích. Nếu tìm thấy cuốn sách và thư mục đích chính xác, chương trình sẽ copy file sách tương ứng vào đó. Nếu không tìm thấy file sách hoặc thư mục đích sẽ báo lỗi. – Người dùng cung cấp một tiêu chí tìm kiếm (tương tự tình huống 2) và thư mục đích. Chương trình tìm tất cả sách đáp ứng tiêu chí và copy vào thư mục đích. Nếu không tìm file sách nào phù hợp hoặc không thấy thư mục đích sẽ báo lỗi.

#	Tình huống sử dụng	Mô tả chi tiết
13	Đổi tên file theo quy tắc	Người dùng lưu trữ quy tắc đặt tên file trong thông tin cấu hình của chương trình. Chương trình sử dụng quy tắc này để đặt lại tên cho toàn bộ sách có trong kho dựa trên dữ liệu sách đang có. Quy tắc này không được gán cố định mà phải do người dùng cuối thiết lập. Ví dụ quy tắc đặt tên: [<tên tác giả>] <tên sách>, <nhà xuất bản>, - <năm xuất bản, <lần tái bản>.pdf
14	Mở thư mục chứa file sách	Khi cuốn sách có các file đi kèm (như mã nguồn, video), chức năng này cho phép nhanh chóng mở thư mục chứa file sách. Người dùng cung cấp Id của sách. Nếu tìm thấy dữ liệu sách và đường dẫn tới file chính xác, chương trình sẽ mở ra thư mục chứa file. Nếu không tìm thấy dữ liệu hoặc đường dẫn sai sẽ báo lỗi.

Mặc dù ở trên chúng ta phân tích bài toán tương đối chi tiết và đưa ra nhiều tình huống sử dụng khác nhau, thực tế không phải lúc nào cũng có thể thực hiện ngay được tất cả các yêu cầu đó khi phát triển ứng dụng.

Có thể ở những phiên bản đầu chỉ thực hiện được những yêu cầu chính, sau đó mới lần lượt bổ sung. Vì vậy, khi phát triển ứng dụng cần vận dụng thiết kế phù hợp để có thể liên tục thực hiện bổ sung các yêu cầu mới một cách dễ dàng mà không ảnh hưởng đến code đã có.

Ngoài ra, nếu làm việc theo nhóm cũng đặt ra những yêu cầu đối với thiết kế để có thể phát triển song song các thành phần của ứng dụng.

Trong quá trình phát triển ứng dụng trong tập bài giảng này sẽ đề cập đến tất cả các vấn đề trên. Chúng ta cũng không thực hiện tất cả các yêu cầu đưa ra trong phần phân tích mà để dành lại một phần cho người học tự hoàn thiện.

Kết luận

Trong bài này chúng ta đã đưa ra đề bài và phân tích bài toán một cách chi tiết, tập trung vào hai khía cạnh: các ca sử dụng và dữ liệu. Các phân tích này sẽ được sử dụng làm cơ sở cho việc phát triển ứng dụng trong dự án.

Trong phần tiếp theo, chúng ta sẽ tìm hiểu kiến trúc .NET framework và ngôn ngữ lập trình C#, trước khi chính thức bắt tay vào thực hiện dự án.