

ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

TÀI LIỆU HƯỚNG DẪN THỰC HÀNH

Thiết kế và xây dựng phần mềm - IT4490

(LƯU HÀNH NỘI BỘ)

© 2020 – HUST – SOICT

MỤC LỤC

1. GIỚI THIỆU.....	2
1.1. MỤC ĐÍCH VÀ PHẠM VI CỦA TÀI LIỆU.....	2
1.2. MỤC TIÊU THỰC HÀNH	2
1.3. THÔNG TIN CHUNG	2
2. CÁC QUY ĐỊNH ĐỐI VỚI SINH VIÊN	2
3. BÀI THỰC HÀNH SỐ HAI	2
3.1. MỤC ĐÍCH VÀ NỘI DUNG.....	2
3.2. CHUẨN BỊ	3
3.3. HƯỚNG DẪN CÁC THAO TÁC CƠ BẢN.....	3
3.4. BÀI TẬP THỰC HÀNH.....	4
3.5 HƯỚNG DẪN THIẾT KẾ CHO USE CASE “XEM DANH SÁCH BOOK”	4
3.5.1 Bước lập 1	4
3.5.2 Bước lập 2	6

1. GIỚI THIỆU

1.1. MỤC ĐÍCH VÀ PHẠM VI CỦA TÀI LIỆU

Tài liệu này được cung cấp cho sinh viên đăng ký học phần Thiết kế và xây dựng phần mềm – IT4490. Sinh viên có 5 buổi thực hành; mỗi buổi sinh viên sẽ được cung cấp các tài liệu độc lập cùng với các tài nguyên cần thiết khác. Sinh viên cần đọc kỹ hướng dẫn trong tài liệu và hoàn thành các bài tập được yêu cầu trong tài liệu này.

1.2. MỤC TIÊU THỰC HÀNH

Các giờ THỰC HÀNH thuộc học phần IT4490 có mục tiêu trang bị các kỹ năng thực hành và làm rõ các kiến thức sinh viên đã học ở giờ lý thuyết trên lớp. Thông qua quá trình thực hành, sinh viên hiểu và vận dụng được nhuần nhuyễn các kiến thức đã học trong quá trình thiết kế và xây dựng phần mềm.

1.3. THÔNG TIN CHUNG

Tổng thời lượng thực hành là 15 tiết, phân bố vào 5 buổi thực hành; mỗi buổi thực hành có thời lượng là 3 tiết

2. CÁC QUY ĐỊNH ĐỐI VỚI SINH VIÊN

Bạn phải có trách nhiệm tham gia đầy đủ các buổi thực hành. Trong quá trình thực hành, bạn cần có thái độ nghiêm túc, không cười đùa, nói chuyện riêng, và cần tích cực hoàn thành các bài tập được giao. Trừ khi phục vụ trực tiếp cho việc thực hành, nghiêm cấm các hoạt động chat, lướt web, chơi game. Bạn cần tự hoàn thành các bài thực hành. Tuyệt đối nghiêm cấm hành vi sao chép bài của người khác. Nếu bị phát hiện, bạn sẽ bị đánh giá trượt môn học.

Khi tham gia buổi thực hành, bạn cần tuân thủ quy định nội quy phòng thực hành, và chấp hành các hiệu lệnh của giáo viên hướng dẫn thực hành.

3. BÀI THỰC HÀNH SỐ HAI

3.1. MỤC ĐÍCH VÀ NỘI DUNG

Bài thực hành này hướng dẫn bạn thực hiện bước thiết kế (design) trong pha phân tích và thiết kế phần mềm.

3.2. CHUẨN BỊ

Trong bài Lab 02, bạn sẽ tiếp tục thực hành với công cụ Astah UML. Bạn cũng cần sử dụng lại các kết quả mình đã làm trong bài thực hành Lab 01.

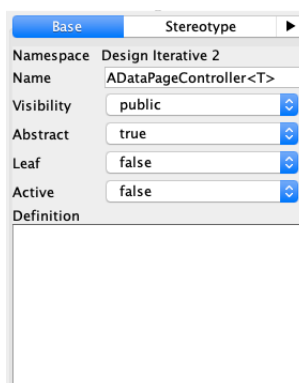
Tài liệu tham khảo hướng dẫn sử dụng Astah UML:

- Web-<https://astah.net/support/astah-pro/user-guide/>
- PDF-https://astah.net/wp-content/uploads/2019/12/ReferenceManual-astah-UML_professional.pdf

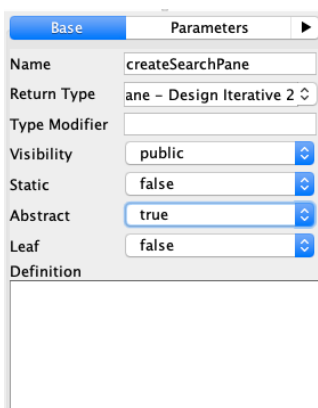
3.3. HƯỚNG DẪN CÁC THAO TÁC CƠ BẢN

Lưu ý: Bạn cần đọc lại các thao tác cơ bản đã hướng dẫn trong bài Lab 01.

Tạo lớp trừu tượng: để tạo lớp trừu tượng, trước tiên bạn tạo lớp như cách thông thường. Bấm chọn lớp vừa tạo để mở view đặc tính của lớp ở góc dưới trái màn hình rồi chọn giá trị thuộc tính Abstract là true như sau



Tạo phương thức trừu tượng: để tạo phương thức trừu tượng, trước tiên bạn tạo phương thức như cách thông thường. Nhấn chọn phương thức vừa tạo để mở view đặc tính của phương thức ở góc dưới trái màn hình rồi chọn giá trị thuộc tính Abstract là true như sau



Tạo giao diện (Interface): để tạo một Interface, trước tiên bạn tạo lớp như cách thông thường, sao đó nhấn chuột phải vào lớp vừa tạo, và chọn menu “Convert To Interface”.

3.4. BÀI TẬP THỰC HÀNH

Bài tập 2.1: Tham khảo ví dụ mẫu với biểu đồ trình tự và biểu đồ lớp thiết kế cho use case “Xem danh sách Book” của tác nhân User được cung cấp trong bài lab. Đọc hiểu ví dụ này. Xem thêm hướng dẫn ở phần 3.5

Bài tập 2.2: Vẽ biểu đồ trình tự và biểu đồ lớp thiết kế cho use case “Xem danh sách Book quản lý” của tác nhân Admin. Trong các bài lab, sinh viên chỉ cần đưa ra thiết kế cuối cùng.

Bài tập 2.3: Vẽ biểu đồ trình tự và biểu đồ lớp thiết kế cho use case “Thêm Book vào giỏ hàng” của tác nhân User.

Bài tập 2.4: Vẽ biểu đồ trình tự và biểu đồ lớp thiết kế cho use case “Chỉnh sửa Book” của tác nhân Admin.

3.5 HƯỚNG DẪN THIẾT KẾ CHO USE CASE “XEM DANH SÁCH BOOK”

Thiết kế là bước sau của bước phân tích. Khi phân tích use case, ta chưa đi vào chi tiết, chưa quan tâm tới công nghệ, hiệu năng, tính tái sử dụng, ... Nhưng các vấn đề này cần phải được giải quyết trong bước thiết kế. Do vậy, thiết kế là một công việc khó, đòi hỏi nhiều kỹ năng tổng hợp. Thiết kế là một quá trình lặp. Sau khi đã thiết kế các lớp cần thiết cho một use case, cần phải xem xét thêm các use case khác để điều chỉnh lại thiết kế cho tối ưu hơn. Ví dụ mẫu với use case “Xem danh sách Book” trong bài lab này đưa ra 2 bước lặp như vậy. Trên thực tế có thể có 1, 2, 3, hoặc nhiều bước lặp, tùy từng tình huống cụ thể.

3.5.1 Bước lặp 1

Trước tiên xét bước lặp 1. Bước này sẽ sử dụng kết quả của bước phân tích. Trong bước phân tích, toàn bộ xử lý hiển thị giao diện tìm kiếm và danh sách các Book trả về đều được đưa vào lớp UseBookPage. Cách làm này không tốt, vì khi đó UseBookPage đảm nhiệm nhiều nhiệm vụ, vừa phức tạp, vừa không có tính tái sử dụng cao. UserBookPage và AdminBookPage sẽ có nhiều phần code rất giống nhau, dẫn đến tốn chi phí bảo trì.

Để khắc phục, chúng ta sẽ tách giao diện UserBookPage thành BookSearchPane và UserBookListPane. UserBookListPane sẽ hiển thị danh sách các book trả về. mỗi book trả về sẽ được hiển thị trong một BookSinglePane. Để thống nhất cách đặt tên, ta cũng đổi UserBookPage trong bước phân tích thành UserBookPagePane.

Lưu ý: BookSearchPane sẽ dùng chung cho cả admin và user do giao diện và xử lý giống nhau. Nhưng UserBookListPane dù có nhiều nét tương tự, vẫn có giao diện và xử lý khác với AdminBookListPane. Trong bước lập sau ta sẽ xử lý tái sử dụng mã nguồn triệt để hơn.

BookSearchPane sẽ đồng thời được sử dụng trong use case “Xem danh sách Book quản lý” của Admin. Do vậy, để đảm bảo tính phụ thuộc thấp (low coupling), controller của BookSearchPane được thiết kế là một giao diện IBookSearchController.

Trong biểu đồ trình tự ở bước lập 1, message số 1 (getUserBookPagePane) có nhiệm vụ khởi tạo các đối tượng, và kết nối các đối tượng lại với nhau. Lớp thực thể Book thao tác với CSDL trước đây sẽ được thay bằng SubSystem MediaApi. Lớp UserBookPageController đóng vai trò trung tâm điều khiển, kết nối các thành phần giao diện BookSearchPane, BookSinglePane, và UserBookListPane.

Khi UserBookPageController khởi tạo BookSearchPane, sẽ đồng thời khởi tạo một đối tượng IBookSearchController với kỹ thuật lớp vô danh Anonymous Class như sau:

```
public class UserBookPageController {
    public UserBookPageController() {
        // ...
        BookSearchPane searchPane = new BookSearchPane();
        IBookSearchController controller = new IBookSearchController() {
            @Override
            public void search(Map<String, String> searchParams) {
                UserBookPageController.this.search(searchParams);
            }
        };
        searchPane.setController(controller);
        //...
    }
    public void search(Map<String, String> searchParams) {
        List<Book> list = mediApi.getBooks(searchParams);
        listPane.updateData(list);
    }
}
```

Xử lý khi người dùng nhấn vào nút tìm kiếm trên BookSearchPane:

```
searchButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        controller.search(getQueryParams());
    }
});
```

3.5.2 Bước lập 2

Thiết kế ở trên có nhược điểm là UserBookListPane sẽ có code rất giống với AdminBookListPane. Hơn nữa, các loại dữ liệu CD, DVD có nhiều thuộc tính giống với Book, cũng sẽ dẫn đến lặp lại code. Bước lập 1 cũng chưa làm rõ công nghệ sử dụng (Java Swing)

Để khắc phục nhược điểm này và tổng quát hóa các loại mặt hàng, thiết kế ở bước lập 1 cần được điều chỉnh và làm rõ thêm.

Ở bước lập 2 này, ta phải kết hợp vừa thiết kế lớp, vừa vẽ biểu đồ trình tự. Trong quá trình thiết kế, vừa kiểm tra và tinh chỉnh liên tục. Đồng thời, trong quá trình thiết kế, ta cần sử dụng các kỹ thuật trong lập trình hướng đối tượng như kế thừa, đa hình, giao diện, lớp trừu tượng, ... Sinh viên cần tham khảo biểu đồ lớp trước khi đọc tiếp hướng dẫn này.

Như trong biểu đồ lớp trình bày, ta có giao diện tổng quát ADataPagePane<T>, bao gồm 2 phần là phần tìm kiếm - ADataSearchPane và phần hiển thị danh sách dữ liệu - ADataListPane<T>. Một ADataListPane<T> lại bao gồm nhiều ADataSinglePane<T>, để hiển thị mỗi phần tử dữ liệu trong danh sách. Lớp GUI ADataPagePane này áp dụng cho loại dữ liệu T bất kỳ (hóa đơn, sách, CD, DVD, hoặc bất kỳ kiểu dữ liệu mở rộng về sau). ADataSearchPane không cần lập trình template, do các tham số tìm kiếm được xuất ra dưới dạng Map<String, String>.

ADataPagePane<T> sẽ có lớp điều khiển là ADataPageController<T> và ADataSearchPane có controller là IDataSearchController

Dựa trên các lớp cơ sở này, chúng ta xây dựng các cây phân cấp kế thừa như sau

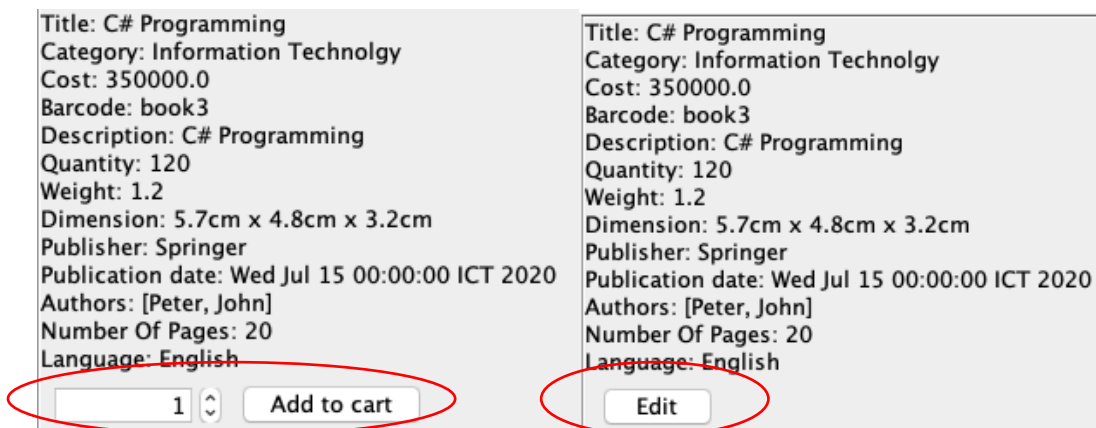
Các lớp hiển thị dữ liệu đơn: ADataSinglePane, MediaSinglePane, PhysicalMediaSinglePane, và BookSinglePane (lớp đứng trước là cha của lớp đứng sau):

Lớp `UserMediaListPane` là con của lớp `ADataListPane`. Lớp `UserMediaListPane` này dùng chung được cho tất cả các loại Media (không cần tách rõ `UserBookListPane`, `UserCDListPane`, ...), do lớp này chỉ chứa danh sách các `ADataSinglePane`.

Lớp `ADataPageController`, `UserMediaPageController`, và `UserBookPageController`. Sau này ta sẽ có thêm `UserCDPageController`, `UserDVDPageController`, do các controller này xử lý dữ liệu khác nhau. Với chức năng cho Admin, ta sẽ có `AdminMediaPageController`. `UserMediaPageController` sẽ có `CartController` và `AdminMediaPageController` có thêm phương thức để quản lý dữ liệu (CRUD).

Một số lưu ý khác trong thiết kế:

- `ADataListPane` có phương thức trừu tượng `decorateSinglePane`. User, Admin có giao diện xem danh sách mặt hàng tương tự như nhau. Khác biệt là User sẽ thêm mặt hàng vào giỏ, Admin thì edit mặt hàng. `UserMediaListPane` sẽ cài đặt phương thức `decorateSinglePane` này để hiển thị phần giao diện chọn mặt hàng và nút thêm hàng vào giỏ. `AdminMediaListPane` thì hiển thị phần giao diện Edit



- `ADataPageController` có 4 phương thức trừu tượng: `createSearchPane`, `search`, `createSinglePane`, `createListPane`. Việc cài đặt giao diện tìm kiếm, gọi API tìm kiếm, giao diện hiển thị sẽ do các lớp con phụ trách, tùy loại dữ liệu mong muốn. Ví dụ với Book, ta có lớp `UserBookPageController` xử lý các công việc này.
- `ADataSinglePane` cần cung cấp phương thức `getData`, trả về dữ liệu tương ứng mà nó hiển thị, dùng để xử lý khi User/Admin nhấn vào nút thêm hàng vào giỏ/Edit
- `ADataSinglePane` có phương thức trừu tượng `displayData`, để các lớp con tự trình bày dữ liệu