

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 NĂM	2
3.1. MỤC ĐÍCH VÀ NỘI DUNG.....	2
3.2. CHUẨN BỊ	3
3.3. BÀI TẬP THỰC HÀNH.....	3
3.4. BÀI TẬP NÂNG CAO	5

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 bổ 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 NĂM

3.1. MỤC ĐÍCH VÀ NỘI DUNG

Bài thực hành này hướng dẫn bạn tối ưu mã nguồn, sử dụng các nguyên lý thiết kế và các mẫu thiết kế

3.2. CHUẨN BỊ

Trong bài Lab 04, bạn sẽ tiếp tục thực hành với công cụ IDE Eclipse. Bạn cần import project Lab05-OMSApp và project Lab05-OMSVirtualServer vào workspace và tìm hiểu nhanh mã nguồn được cung cấp trong 2 project này

3.3. BÀI TẬP THỰC HÀNH

Bài tập 5.1: Nguyên lý Dependency Injection

Lớp `com.oms.components.media.book.controller.AdminBookPageController` ở tầng trên bị phụ thuộc vào lớp `com.oms.serverapi.MediaApi` ở tầng dưới. Sinh viên áp dụng nguyên lý Dependency Injection, tạo interface `IMediaApi` để khắc phục hạn chế này

Bài tập 5.2: Nguyên lý Interface Segregation và nguyên lý Single Responsibility:

- a) Lớp `com.oms.serverapi.MediaApi` sau khi cải tiến trong phần áp dụng nguyên lý Dependency Injection của bạn có nhiều phương thức liên quan tới các đối tượng dữ liệu như: phương thức `getAllMedias`, `getBooks`, `updateBook`, `getCds`, `getDvds`. Số lượng các phương thức này sẽ tăng dần lên khi bạn cài đặt hết các tính năng của hệ thống thương mại điện tử. Lớp `MediaApi` do vậy có kích thước ngày một lớn, dẫn đến khó theo dõi, khó sửa đổi và bảo trì. Bạn cần áp dụng nguyên lý Single Responsibility để khắc phục nhược điểm này. Lưu ý, để tối ưu, bạn có thể cần phải kết hợp với nguyên lý Interface Segregation khi cài đặt.
- b) Giao diện `com.oms.components.abstractdata.controller.IDataManageController` định nghĩa 4 phương thức quản lý dữ liệu: tạo, đọc, xóa, sửa. Lớp `com.oms.components.media.gui.AdminMediaListViewPane` tạo lớp vô danh (anonymous class) thực thi giao diện này và chỉ cần phương thức `update`, nhưng bắt buộc phải tạo cài đặt rỗng cho 3 phương thức `create`, `read`, `delete`. Thực tế, lớp `BookEditDialog` cũng chỉ dùng tới phương thức `edit`. Bạn cần áp dụng nguyên lý Interface Segregation để cài đặt giải pháp khắc phục. Gợi ý: `IDataManageController` chỉ một phương thức chung là `onAct`, dùng chung cho các thao tác CRUD.

```
public interface IDataManageController<T> {
    public void onAct(T t);
}
```

- c) Lớp `com.oms.components.media.gui.AdminMediaListPane` chưa đảm bảo tuân thủ mẫu thiết kế MVC và thực chất là chưa tuân thủ nguyên lý Single Responsibility. Phương thức `decorateSinglePane` chỉ nên xử lý về mặt giao diện, không nên khởi tạo đối tượng `IDataManageController` và kết nối đối

tượng này với đối tượng lớp BookEditDialog để cài đặt tính năng chỉnh sửa sách cho admin. Phần nhiệm vụ này nên chuyển cho lớp AdminMediaPageController. Sinh viên cần chỉnh sửa mã nguồn cho tối ưu hơn. Gợi ý: thêm phương thức onEdit cho lớp AdminMediaPageController như sau

```
public void onEdit(ADataSinglePane<Media> singlePane) {
    ...
}
```

Bài tập 5.3: Nguyên lý Open/Closed

Tìm trong source code và chỉ ra MỘT ví dụ áp dụng nguyên lý Open/Closed. Việc áp dụng như vậy đem lại lợi ích gì?

Bài tập 5.4: Nguyên lý Liskov Substitution

Kiểm tra lại toàn bộ mã nguồn dự án, có phần cài đặt nào bị vi phạm nguyên lý thay thế Liskov ở đâu không? Nếu có, giải thích tại sao vi phạm, và chỉ ra cách khắc phục

Bài tập 5.5: Mẫu thiết kế Singleton

Lớp com.oms.serverapi.MediaApi hiện có thể được khởi tạo nhiều lần. (Xem lớp com.oms.components.media.book.controller.AdminBookPageController và lớp com.oms.components.media.book.controller.UserBookPageController). Bạn được yêu cầu sử dụng mẫu thiết kế Singleton với MediaAPI để tối ưu hơn

Bài tập 5.6: Mẫu thiết kế Factory

Để tạo tab quản lý Book, lớp com.oms.app.admin.OMSAdminController tạo ra đối tượng AdminBookPage. Mỗi khi thêm một sản phẩm mới, tương tự, chúng ta có thể tạo ra đối tượng các lớp AdminCdPage, AdminDvdPage. Tuy nhiên, nhược điểm là mỗi lần thêm sản phẩm mới như vậy, chúng ta phải sửa code ở cả lớp GUI (OMSAdmin và OMSUser) và lớp Controller (OMSAdminController và OMSUserController), vi phạm nguyên lý đóng mở. Bạn cần tạo lớp com.oms.factory.AdminPageFactory và lớp com.oms.factory.UserPageFactory để áp dụng mẫu thiết kế Factory khắc phục hạn chế này. Mỗi lớp có phương thức createPage(type: String), tạo ra Page để thêm vào tab mặt hàng tương ứng. Áp dụng tiếp mẫu Singleton cho các lớp Factory vừa tạo.

3.4. BÀI TẬP NÂNG CAO

Sau khi hoàn thành các bài lab trên, bạn chuyển sang các bài lab bổ sung sau đây. Nếu bạn chưa kịp hoàn thành, bạn có thể thực hiện các bài lab bổ sung ở nhà.

Bài tập 5.7: Bạn tạo thêm các chức năng cho mặt hàng CD, tương tự như mặt hàng Book, cho cả ứng dụng OMSAdmin và OMSUser. Việc tạo thêm mặt hàng mới với bạn có dễ dàng không. Nếu dễ dàng, giải thích vì sao mã nguồn hiện tại giúp bạn thêm mới dễ dàng các mặt hàng? Nếu khó thêm mặt hàng, bạn cần trình bày các khó khăn mình gặp phải.

Bài tập 5.8: Sản phẩm mã nguồn cuối cùng còn có nhược điểm gì không? Mô tả và trình bày cách khắc phục các nhược điểm này.