**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

======\*\*\*======

BÁO CÁO THỰC NGHIỆM  
 LẬP TRÌNH JAVA

**XÂY DỰNG ỨNG DỤNG HỖ TRỢ ĐĂNG KÝ HỌC PHẦN CHO SINH VIÊN HAUI**

|  |  |
| --- | --- |
| **GVHD:** | ***Ts. Nguyễn Ngọc Quang*** |
| **Sinh viên:** | Nguyễn Thế Đạt 2021602720  Nguyễn Hoàng Hải 2021602114  Vũ Thị Hường 2021601944  Nguyễn Minh Đức 2021607287  Trần Đại Phú 2019605940 |
| **Nhóm:** | 8 |
| **Lớp:** | 20233IT6019001 **Khóa:** 16 |

Hà Nội – Năm 2024

**MỤC LỤC**

[PHẦN 1: MỞ ĐẦU 1](#_Toc159683925)

[1.1. Lý do chọn đề tài 1](#_Toc159683926)

[1.2. Xác định nội dung học tập 1](#_Toc159683927)

[1.2.1. Các kiến thức cơ bản cần có 1](#_Toc159683928)

[1.2.2. Các kỹ năng đã có để thực hiện chủ đề nghiên cứu 2](#_Toc159683929)

[PHẦN 2: KẾT QUẢ NGHIÊN CỨU 3](#_Toc159683930)

[2.1. Giới thiệu 3](#_Toc159683931)

[2.1.1. Những nhiệm vụ chính 3](#_Toc159683932)

[2.1.2. Kết quả đạt được. 3](#_Toc159683933)

[2.1.3. Công cụ sử dụng 3](#_Toc159683934)

[2.2. Khảo sát hệ thống 5](#_Toc159683935)

[2.2.1. Khảo sát sơ bộ 5](#_Toc159683936)

[2.2.2. Tài liệu đặc tả yêu cầu 5](#_Toc159683937)

[2.3. Phân tích hệ thống 6](#_Toc159683938)

[2.3.1. Mô hình hóa chức năng hệ thống 6](#_Toc159683939)

[2.3.2. Mô hình hóa dữ liệu và giao diện hệ thống 11](#_Toc159683940)

[2.4. Thực hiện bài toán 20](#_Toc159683941)

[2.4.1. Vũ Thị Hường – Chức năng đăng nhập 20](#_Toc159683942)

[2.4.2. Vũ Thị Hường – Chức năng đổi mật khẩu 23](#_Toc159683943)

[2.4.3. Nguyễn Minh Đức - Chức năng gửi đơn đăng ký cá nhân 26](#_Toc159683944)

[2.4.4. Nguyễn Minh Đức - Chức năng gửi đơn đăng ký theo tập thể 30](#_Toc159683945)

[2.4.5. Trần Đại Phú - Chức năng gửi đơn đề xuất 34](#_Toc159683946)

[2.4.6. Trần Đại Phú - Chức năng xem danh sách đơn đăng ký 36](#_Toc159683947)

[2.4.7. Nguyễn Thế Đạt - Chức năng Quản lý thông tin khoa 38](#_Toc159683948)

[2.4.8. Nguyễn Thế Đạt - Chức năng Quản lý danh sách ngành theo khoa 41](#_Toc159683949)

[2.4.9. Nguyễn Thế Đạt - Chức năng Quản lý danh sách lớp học phần theo ngành 43](#_Toc159683950)

[2.4.10. Nguyễn Minh Đức- Chức năng thống kê danh sách học phần có đơn đăng ký theo khoa 45](#_Toc159683951)

[2.4.11. Nguyễn Hoàng Hải - Chức năng thống kê danh sách đơn đăng ký theo lớp học phần 48](#_Toc159683952)

[2.4.12. Nguyễn Hoàng Hải - Chức năng thống kê đơn đề xuất 51](#_Toc159683953)

[2.4.13. Lớp FakeData để lấy dữ liệu 53](#_Toc159683954)

[PHẦN 3: KẾT LUẬN VÀ BÀI HỌC KINH NGHIỆM 56](#_Toc159683955)

[3.1. Nội dung đã thực hiện 56](#_Toc159683956)

[3.1.1. Kiến thức và kỹ năng đã đạt được 56](#_Toc159683957)

[3.1.2. Các chuẩn đầu ra đã đạt được 57](#_Toc159683958)

[3.1.3. Các bài học kinh nghiệm 57](#_Toc159683959)

[3.2. Hướng phát triển 58](#_Toc159683960)

[TÀI LIỆU THAM KHẢO 60](#_Toc159683961)

# MỞ ĐẦU

## Lý do chọn đề tài

Tên chủ đề nghiên cứu: Xây dựng ứng dụng hỗ trợ đăng ký học phần cho sinh viên HaUI.

Trong thời đại công nghệ thông tin phát triển mạnh mẽ, việc tin học hóa các công việc trong các trường đại học là đúng với xu thế hiện tại và rất cần thiết. Việc ứng dụng công nghệ thông tin vào các công việc sẽ tiết kiệm thời gian, chi phí cho nhà trường, giảng viên cũng như sinh viên và mang lại hiệu quả rất cao. Việc chọn đề tài "Xây dựng ứng dụng hỗ trợ đăng ký học phần cho sinh viên HaUI" được đưa ra với một số lý do quan trọng. Trước hết, chương trình giải quyết một vấn đề thực tế đang tồn tại trong quá trình quản lý học phần tại trường Đại học Công nghiệp Hà Nội (HaUI). Sinh viên thường gặp khó khăn khi đăng ký học phần, và ứng dụng này có thể cung cấp giải pháp hiệu quả. Bên cạnh đó, đề tài đồng thời hướng đến việc tối ưu hóa quá trình quản lý học phần, giúp nhà trường tiết kiệm thời gian và nguồn lực trong công tác quản lý học vụ. Điều này sẽ tăng cường sự linh hoạt và hiệu suất của hệ thống đăng ký, giúp sinh viên dễ dàng chọn lựa và theo dõi học phần một cách thuận lợi. Thứ ba, việc xây dựng ứng dụng này đồng thời đáp ứng xu hướng công nghệ hiện đại, tạo điều kiện thuận lợi cho sinh viên sử dụng trên nhiều nền tảng, từ máy tính đến thiết bị di động. Điều này giúp cải thiện trải nghiệm người dùng và đồng thời đưa ra một giải pháp tiện ích trong thời đại số ngày nay.

Tóm lại, việc chọn đề tài này không chỉ phản ánh nhu cầu thực tế mà còn mang lại nhiều lợi ích trong quản lý học vụ và cải thiện trải nghiệm học tập cho sinh viên tại HaUI.

## Xác định nội dung học tập

### Các kiến thức cơ bản cần có

Tổng hợp các kiến thức cơ bản cần có để thực hiện xây dựng hệ thống:

* Các kiến thức về lập trình hướng đối tượng để thiết kế các đối tượng cần thiết và sử dụng ngôn ngữ lập trình java để trình bày.
* Các kiến thức cơ bản về ngôn ngữ lập trình java: lớp, đối tượng, bẫy lỗi, gom rác, xử lý giao diện (form), thao tác với file.
* Các phương thức xử lý thông tin, kết nối với cơ sở dữ liệu trong java.

### Các kỹ năng đã có để thực hiện chủ đề nghiên cứu

Nhóm em đã thực hiện mô tả và xây dựng ứng dụng hỗ trợ đăng ký học phần cho sinh viên Haui cùng với những kiến thức và kỹ năng dưới đây:

* Về kiến thức:
* Kiến thức cơ bản về lập trình java gồm: lớp, đối tượng, bẫy lỗi, gom rác, xử lý giao diện(form), thao tác với file.
* Thành thạo sử dụng các công cụ hỗ trợ viết code: Apache Netbeans, công cụ hỗ trợ viết báo cáo thực nghiệm: Microsoft Word và công cụ hỗ trợ phân tích và thiết kế phần mềm: Rational Rose, Case Studio.
* Kiến thức về các thao tác cơ bản trong việc xử lý kết nối với cơ sở dữ liệu từ SQL Server.
* Ngoài việc vận dụng các kiến thức ở môn “Lập trình java” nhóm chúng em còn vận dụng kiến thức môn “Phân tích và đặc tả yêu cầu phần mềm” để phân tích, phát hiện yêu cầu bài thực nghiệm
* Về kỹ năng: Các kỹ năng làm việc nhóm, thuyết trình, tổng hợp kiến thức.
* Với các kiến thức và kỹ năng có được ở trên và là triển khai của mô hình xây dựng bản mẫu. Xây dựng theo mô hình bản mẫu là cách tiếp cận thực tế nhất, phù hợp với hệ thống vừa và nhỏ. Nhóm em mong rằng với cách chọn mô hình và xây dựng chương trình sẽ đạt hiệu quả tốt nhất đem lại kết quả cao khi thực hiện bài tập thực nghiệm lần này cũng như cho chúng em thêm kinh nghiệm, trải nghiệm học tập về ngôn ngữ lập trình java.

# KẾT QUẢ NGHIÊN CỨU

## Giới thiệu

### Những nhiệm vụ chính

* Khảo sát hệ thống.
* Phân tích hệ thống từ đó xác định các thực thể, các use case cũng như quan hệ giữa các thực thể, các use case đó.
* Đặc tả, vẽ biểu đồ cho các use case kể trên, Xác định kiểu dữ liệu cũng như thuộc tính các thông tin trong thực thể.
* Xây dựng chương trình bài toán theo ngôn ngữ lập trình Java, mô tả nội dung mã lệnh mình đã thực hiện.

### Kết quả đạt được.

* Bài báo cáo tổng hợp nội dung các thành viên trong nhóm đã tìm hiểu và tổng hợp được theo đề tài đã nhận.
* Chương trình được viết bằng ngôn ngữ lập trình Java sử dụng bộ công cụ GUI Java Swing.

### Công cụ sử dụng

#### Apache Netbeans

* *Apache NetBeans* là một môi trường phát triển tích hợp (IDE) mã nguồn mở được phát triển bởi Apache Software Foundation. Phần mềm cung cấp một loạt các tính năng hỗ trợ phát triển ứng dụng Java
* Ưu điểm:
* **Miễn phí:** Đây là một phần miễn phí hữu ích cho lập trình viên, chúng ta có thể dễ dàng tải nó trên internet.
* **Thiết kế GUI:** NetBeans cung cấp một công cụ thiết kế GUI giúp tạo ra các giao diện người dùng một cách dễ dàng và nhanh chóng.
* **Dễ sử dụng:** Apache NetBeans có giao diện người dùng thân thiện và dễ sử dụng, đặc biệt là đối với những người mới bắt đầu trong lập trình.
* Nhược điểm:
* **Tài nguyên hệ thống:** So với một số IDE khác, NetBeans có thể sử dụng nhiều tài nguyên hệ thống hơn, đặc biệt là khi làm việc với các dự án lớn.
* **Cộng đồng và hỗ trợ:** So với một số IDE khác, cộng đồng sử dụng và hỗ trợ cho NetBeans ít phát triển hơn.

#### Case Studio 2

* *Case Studio 2* là một công cụ thiết kế cơ sở dữ liệu chuyên nghiệp, cho phép chúng ta tạo các sơ đồ quan hệ thực thể (ERD) cho nhiều hệ thống cơ sở dữ liệu khác nhau như MS SQL Server 2005, Oracle, DB2, Access, Sybase, Firebird, MaxDB, MySQL, PostgreSQL... Với Case Studio 2, chúng ta có thể tự tạo và xây dựng nên một sơ đồ ERD nhanh chóng và chính xác nhất.
* Ưu điểm:
  + - Giao diện đơn giản dễ sử dụng.
    - Dễ dàng tạo ra các entity và các thuộc tính của chúng.
    - Dễ dàng tạo liên kết giữa các bảng.
* Nhược điểm: Đây là một phần mềm trả phí.

#### Rational Rose

* IBM Rational Rose là một công cụ mô hình hóa chuyên nghiệp được sử dụng trong quá trình phát triển phần mềm.
* Ưu điểm:
  + - Mô hình hóa đa dạng: Hỗ trợ mô hình hóa ở nhiều cấp độ, từ mô hình khái niệm đến mô hình logic và mô hình cơ sở dữ liệu.
    - UML (Unified Modeling Language): Hỗ trợ toàn diện cho UML, giúp xây dựng các biểu đồ UML như biểu đồ lớp, biểu đồ tuần tự, biểu đồ hoạt động, và nhiều loại khác.
* Nhược điểm: Đây là một phần mềm trả phí

## Khảo sát hệ thống

### Khảo sát sơ bộ

Đơn vị đặt hàng là trường Đại học Công Nghiệp Hà Nội. Đây là đơn vị có nhiều khoa và chương trình đào tạo khác nhau. Trường quản lý nhiều khoa, và mỗi khoa sẽ tổ chức các học phần cho sinh viên đăng ký trong mỗi kỳ học.

Tổng quan các vấn đề bài toán giải quyết: Ứng dụng “Hỗ trợ đăng ký học phần cho sinh viên HaUI” giải quyết, hỗ trợ sinh viên gửi đơn đăng ký học phần giảm thiểu các khó khăn thường gặp phải khi mở chương trình đào tạo có sẵn cố định các môn. Ứng dụng giúp tiết kiệm thời gian, tăng tính hiệu quả, độ chính xác trong việc gửi đơn đăng ký học phần.

Thu thập các biểu mẫu: Tổ chức đặt hàng cung cấp các biểu mẫu đăng ký học phần, các thông tin liên quan về các học phần, và các yêu cầu của ứng dụng. Đội ngũ phát triển sẽ sử dụng các biểu mẫu này để thiết kế giao diện và tính năng của ứng dụng.

### Tài liệu đặc tả yêu cầu

#### Mục đích

Ứng dụng hỗ trợ đăng ký học phần cho sinh viên trường Đại học Công nghiệp Hà Nội nhằm hỗ trợ sinh viên đăng ký học phần một cách dễ dàng, thuận tiện và nâng cao hiệu quả.

#### Yêu cầu chức năng

* Đăng nhập hệ thống và phân quyền truy cập cho quản trị viên và sinh viên.
* Quản lý thông tin về khoa, ngành, lớp học phần có thể xem, thêm, sửa, xóa thông tin liên quan.
* Quản lý o từng khoa.
* Quản lý thông tin đơn đăng ký, xem xét đơn được duyệt hay không, gửi thông báo cho sinh viên.
* Sinh viên có thể tạo đơn đăng ký cá nhân, tập thể.
* Xem thông tin cá nhân, danh sách đơn đăng ký.
* Đề xuất cải tiến.

#### Yêu cầu phi chức năng

* Giao diện đơn giản, dễ sử dụng và thân thiện với người dùng.
* Ổn định, xử lý nhanh, hiển thị thông báo khi kết thúc một thao tác hay công việc nào đó.
* An toàn, bảo mật:Đảm bảo an toàn thông tin người dùng dữ liệu của nhà trường. Các thao tác nghiệp vụ của người quản trị chỉ thực hiện được khi có quyền.
* Trang web có thể tương tích với các trình duyệt khác nhau các thiết bị truy cập khác nhau.
* Hỗ trợ đa ngôn ngữ.

## Phân tích hệ thống

### Mô hình hóa chức năng hệ thống

|  |  |
| --- | --- |
| **Tác nhân** | **Hoạt động** |
| Sinh viên | * Đăng nhập * Đăng xuất * Đổi mật khẩu * Gửi đơn đăng ký theo cá nhân * Gửi đơn đăng ký theo tập thể * Gửi đơn đề xuất cải tiến * Xem danh sách đơn đăng ký |
| Quản trị viên | * Đăng nhập * Đăng xuất * Đổi mật khẩu * Quản lý danh sách các khoa * Quản lý danh sách ngành theo khoa * Quản lý danh sách học phần theo ngành * Thống kê đơn đăng ký theo học phần * Thống kê danh sách học phần có đơn đăng ký theo khoa * Thống kê đề xuất cải tiến * Duyệt đơn |

* Biểu đồ Use case

A diagram of a diagram

Description automatically generated

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Tên use case | Mô tả ngắn gọn | Chức năng | Ghi chú |
| DN01 | Đăng nhập | Chức năng đăng nhập | Cho phép người sử dụng đăng nhập vào hệ thống bằng tài khoản của mình | Quản trị viên, sinh viên |
| DX01 | Đăng xuất | Chức năng đăng xuất | Cho phép người sử dụng đăng xuất khỏi hệ thống | Quản trị viên, sinh viên |
| DMK01 | Đổi mật khẩu | Chức năng đổi mật khẩu | Cho phép người sử dụng thay đổi mật khẩu | Quản trị viên, sinh viên |
| QLDSHP01 | Quản lý danh sách lớp học phần theo ngành | Quản lý thông tin các lớp học phần của ngành trong hệ thống | Cho phép người dùng thêm, sửa, xóa thông tin các lớp học phần theo ngành. Hiển thị danh sách các lớp học phần trong khoa. | Quản trị viên |
| QLDSN1 | Quản lý danh sách ngành theo khoa | Quản lý thông tin các ngành trong hệ thống | Cho phép người dùng thêm, sửa xóa thông tin các ngành. Hiển thị danh sách các ngành trong hệ thống. | Quản trị viên [LĐ1] |
| QLK01 | Quản lý các khoa | Quản lý thông tin các lớp học trong hệ thống | Cho phép người dùng thêm, sửa, xóa thông tin các khoa. Hiển thị danh sách các khoa trong hệ thống. | Quản trị viên |
| TKDHP01 | Thống kê đơn đăng ký theo học phần | Thống kê số lượng đơn đăng ký theo học phần | Cho phép người dùng hệ thống xem chi tiết danh sách đơn đăng ký theo từng học phần có đơn đăng ký gửi đến | Quản trị viên |
| TKHPTK01 | Thống kê danh sách học phần có đơn đăng ký theo khoa | Thống kê danh sách học phần có đơn đăng ký và số lượng đơn | Cho phép người dùng xem danh sách các lớp học phần có đơn đăng ký và số lượng đơn đăng ký trong học phần đó theo khoa | Quản trị viên |
| TKDXCT01 | Thống kê đề xuất cải tiến | Thống kê danh sách các đơn đề xuất | Cho phép người quản trị xem thông tin chi tiết về đơn đề xuất cải tiến | Quản trị viên |
| DD01 | Duyệt đơn | Quản trị viên duyệt đơn | Người quản trị thực hiện chức năng duyệt đơn và thống kê số lượng lớp được mở của học phần đó và cập nhật trạng thái cho các đơn | Quản trị viên |
| GDCN01 | Gửi đơn đăng ký học phần theo cá nhân | Gửi đơn đăng ký | Cho phép cá nhân sinh viên gửi đơn đăng ký | Sinh Viên |
| GDTT01 | Gửi đơn đăng ký học phần theo tập thể | Gửi đơn đăng ký theo tập thể | Cho phép cá nhân sinh viên đại diện cho một tập thể sinh viên gửi đơn đăng ký học phần | Sinh viên |
| GDDXCT01 | Gửi đơn đề xuất cải tiến | Gửi đơn đề xuất cải tiến | Cho phép sinh viên gửi đơn đề xuất cải tiến về các lớp học phần của trường | Sinh viên |
| XDSD01 | Xem danh sách đơn đăng ký | Xem danh sách đơn đăng ký của sinh viên | Cho phép cá nhân sinh viên xem danh sách đơn đăng ký mà sinh viên đã gửi | Sinh viên |

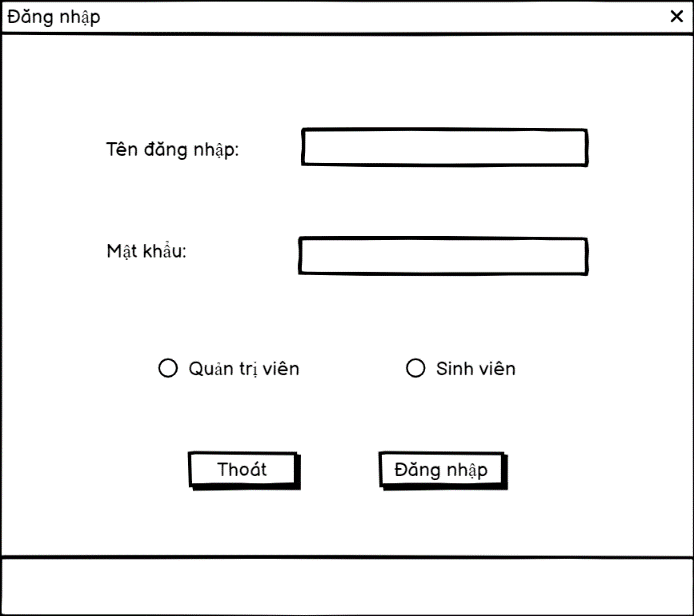
* Biểu đồ mô tả các thực thể và cấu trúc cơ sở dữ liệuA diagram of a network

  Description automatically generated

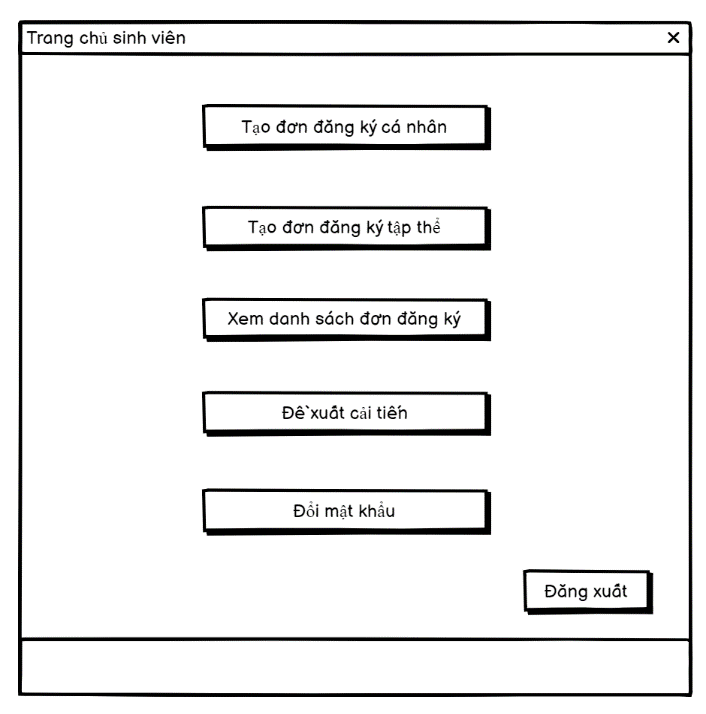
### Mô hình hóa dữ liệu và giao diện hệ thống

#### Chức năng đăng nhập (Vũ Thị Hường)

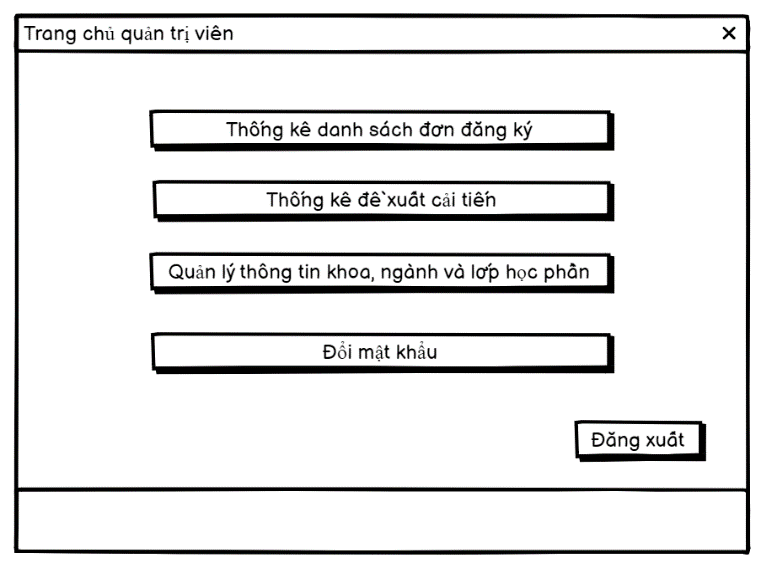
* Các thông tin đầu vào: mã tài khoản (mã tài khoản trùng với mã sinh viên với vai trò đăng nhập là sinh viên), mật khẩu.
* Form cho phép người dùng (sinh viên hoặc quản trị viên) nhập thông tin đăng nhập vào chương trình để sử dụng các chức năng.
* Đăng nhập thành công, chương trình hiển thị thông báo xác nhận và chuyển hướng đến giao diện trang chủ đối với mỗi vai trò đăng nhập.
* Phác thảo giao diện
* Giao diện đăng nhập:



* Giao diện trang chủ sinh viên



* Giao diện trang chủ quản trị viên

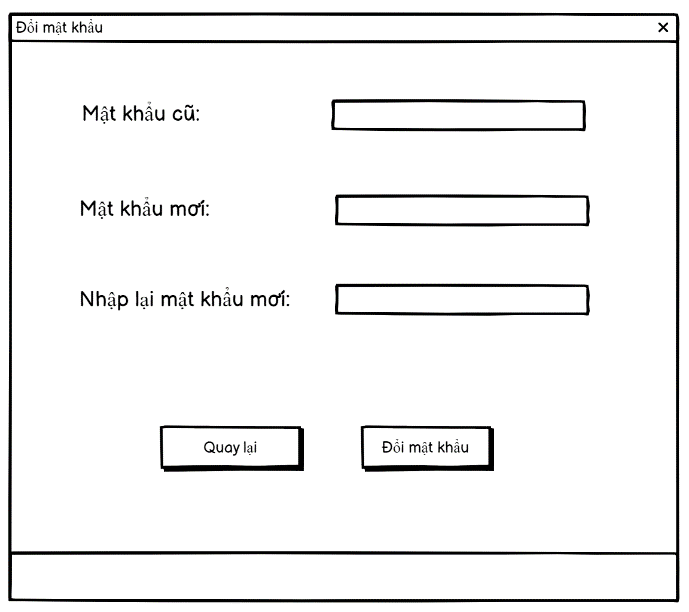


#### Chức năng đăng xuất (Vũ Thị Hường)

* Người dùng thực hiện kích chuột vào nút “Đăng xuất” trên giao diện trang chủ của mỗi vai trò đăng nhập.
* From cho phép người dùng thoát khỏi chương trình và quay lại màn hình đăng nhập.

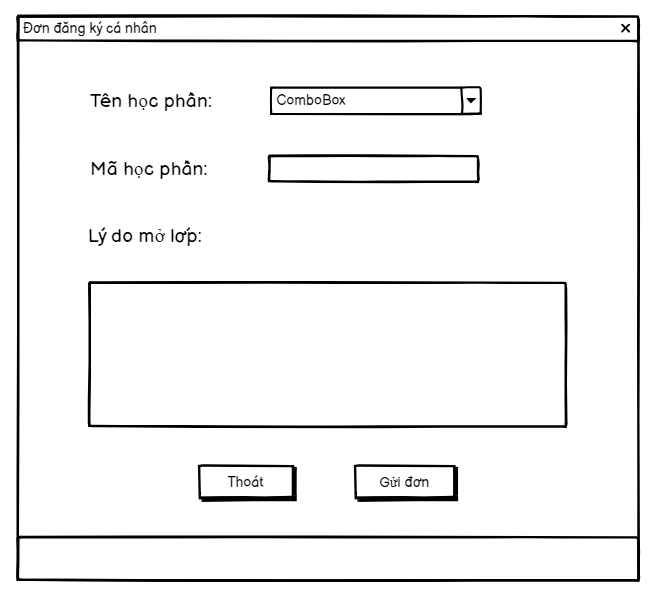
#### Chức năng đổi mật khẩu (Vũ Thị Hường)

* Các thông tin đầu vào: mã tài khoản.
* Chức năng này cho phép người dùng thay đổi mật khẩu của mình. Yêu cầu nhập đúng mật khẩu hiện tại và nhập mật khẩu mới.
* Đổi mật khẩu thành công, chương trình hiển thị thông báo xác nhận và làm mới các trường nhập dữ liệu.
* Phác thảo giao diện:



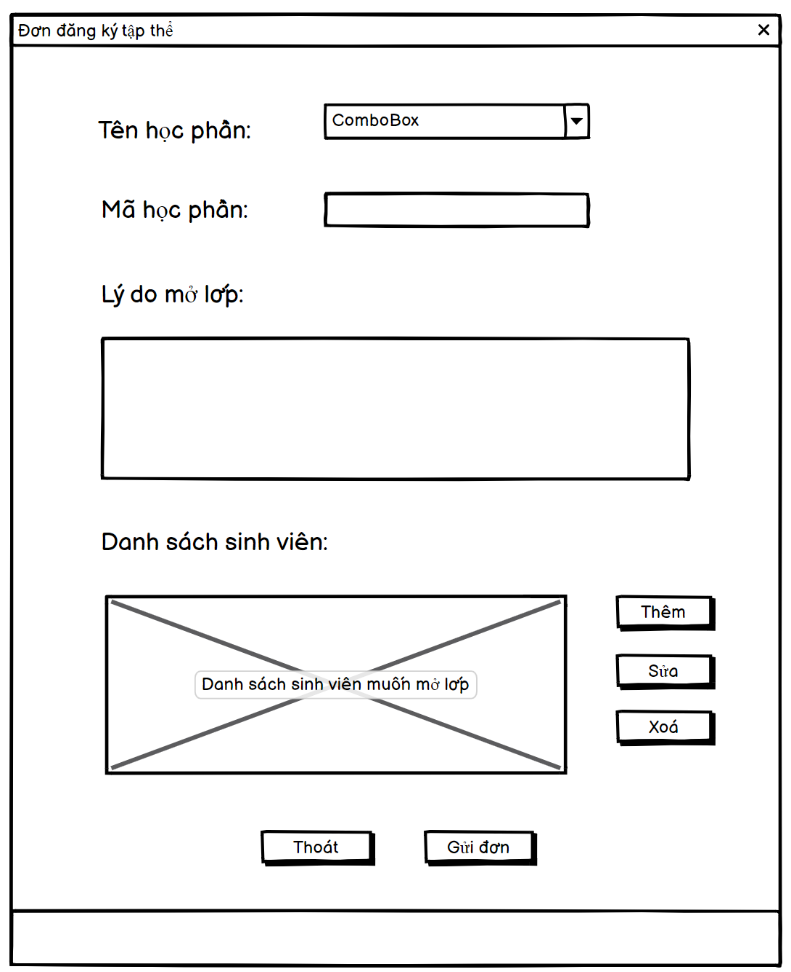
#### Chức năng gửi đơn đăng ký cá nhân (Nguyễn Minh Đức)

* Các thông tin đầu vào: tên học phần, mã học phần, lý do mở lớp.
* Form cho phép sinh viên nhập thông tin đầu vào là đơn đăng ký cá nhân.
* Khi sinh viên gửi đơn đăng ký thành công chương trình sẽ tự động tạo mã đơn đăng ký theo tập thể, lấy mã sinh viên đại diện(mã sinh viên đang đăng nhập) và những thông tin đầu vào để ghi dữ liệu vào file DSDonCaNhan.txt
* Phác thảo giao diện



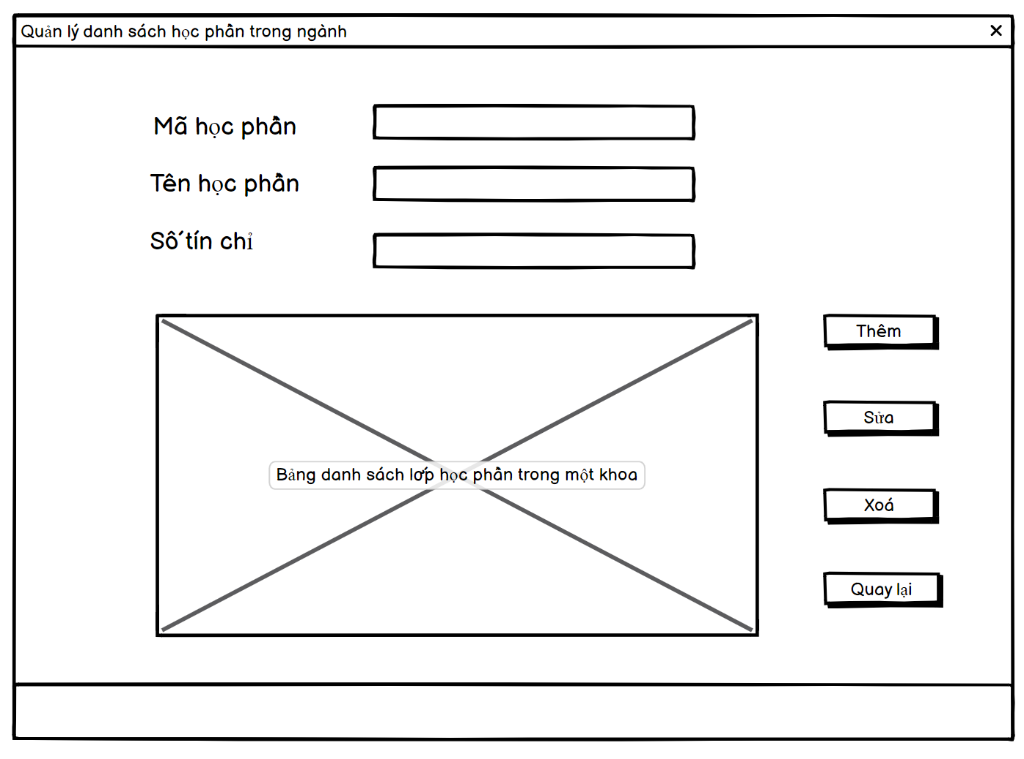
#### Chức năng gửi đơn đăng ký theo tập thể (Nguyễn Minh Đức)

* Các thông tin đầu vào: tên học phần, mã học phần, lý do mở lớp, danh sách sinh viên tập thể.
* Form cho phép sinh viên nhập thông tin đầu vào là đơn đăng ký tập thể.
* Khi sinh viên gửi đơn đăng ký thành công chương trình sẽ tự động tạo mã đơn đăng ký theo tập thể, lấy mã sinh viên đại diện (mã sinh viên đang đăng nhập) và những thông tin đầu vào để ghi dữ liệu vào file DSDonTapThe.txt
* Phác thảo giao diện



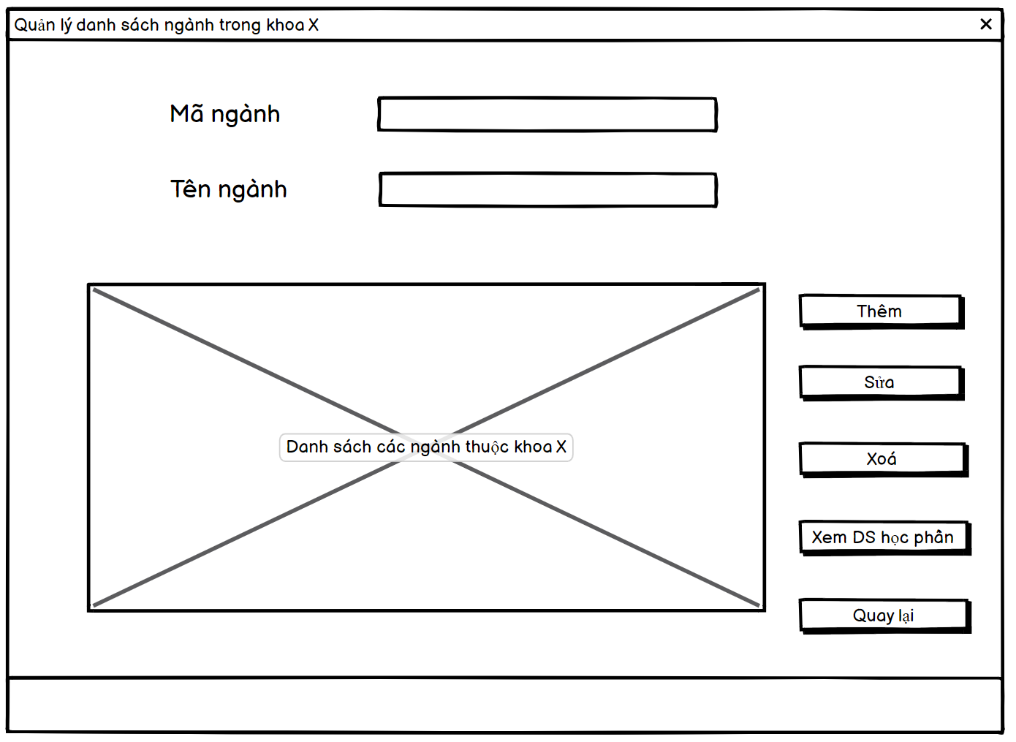
#### Chức năng Quản lý danh sách lớp học phần theo ngành (Nguyễn Thế Đạt)

* Các thông tin đầu vào: mã học phần, tên học phần, số tín chỉ
* Chức năng này cho phép quản trị viên nhập thông tin, thêm, sửa, xoá về học phần có mã khoa được chọn trước đó.
* Khi quản trị viên thêm, sửa, xóa thông tin trong bảng thành công, dữ liệu sẽ được cập nhật lại trong cơ sở dữ liệu.
* Phác thảo giao diện:



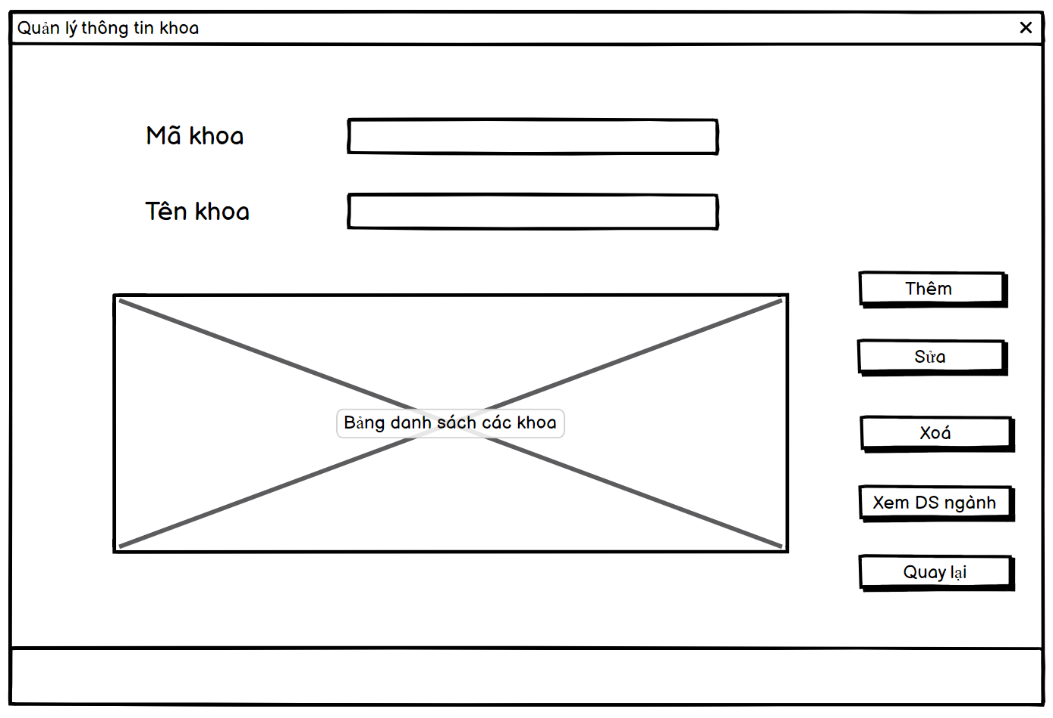
#### Chức năng Quản lý danh sách ngành theo khoa (Nguyễn Thế Đạt)

* Các thông tin đầu vào: mã ngành, tên ngành, mã khoa
* Form hiển thị thông tin đầu vào là mã khoa, thông tin về các ngành có trong khoa, quản trị viên có thể thêm sửa xóa thông tin ngành.
* Khi quản trị viên chọn 1 thông tin ngành và bấm xem danh sách học phần hiển thị thông tin các lớp học phần có trong ngành.
* Phác thảo giao diện:



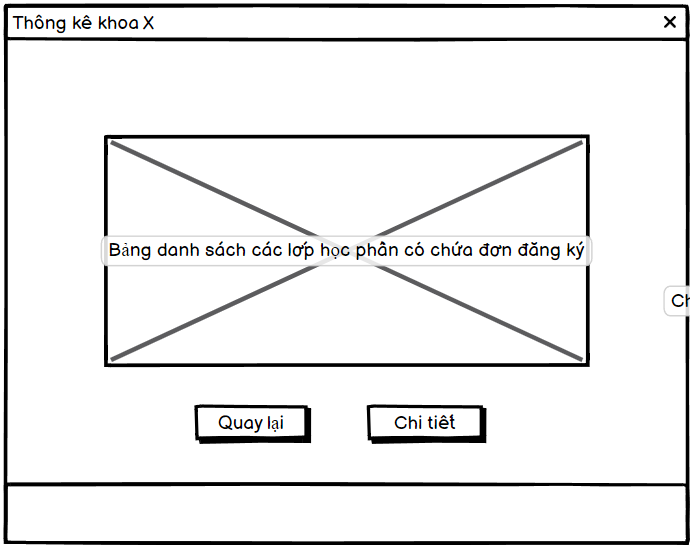
#### Chức năng Quản lý thông tin khoa (Nguyễn Thế Đạt)

* Các thông tin đầu vào: mã khoa, tên khoa
* Chức năng này cho phép quản trị viên nhập thông tin, thêm, sửa, xóa thông tin khoa.
* Khi quản trị viên chọn 1 thông tin khoa và bấm xem danh sách học phần hiển thị thông tin các lớp học phần có trong khoa.
* Phác thảo giao diện:



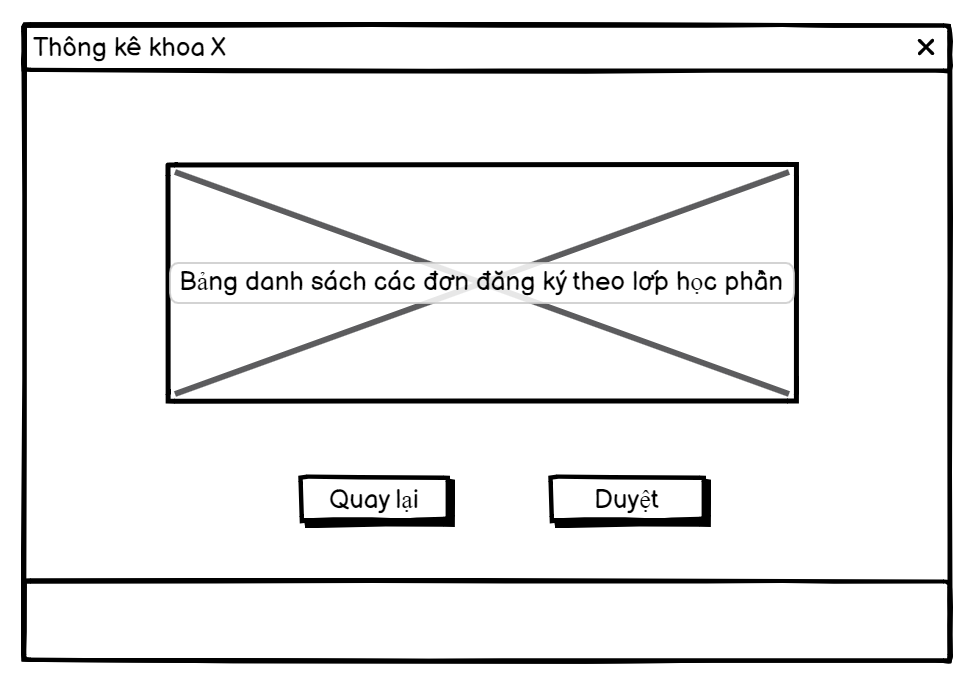
#### Chức năng thống kê danh sách học phần có đơn đăng ký theo khoa (Nguyễn Minh Đức)

* Các thông tin đầu vào: mã học phần, tên học phần, số đơn đăng ký. Những thông tin này sẽ lấy dữ liệu từ file DSDonCaNhan.txt và DSDonTapThe.txt
* Form hiển thị thông tin đầu vào là danh sách các học phần có đơn đăng ký trong khoa.
* Khi quản trị viên chọn một dòng ở trên table và bấm nút chi tiết thì sẽ hiển thị cửa sổ thống kê đơn đăng ký theo học phần.
* Phác thảo giao diện



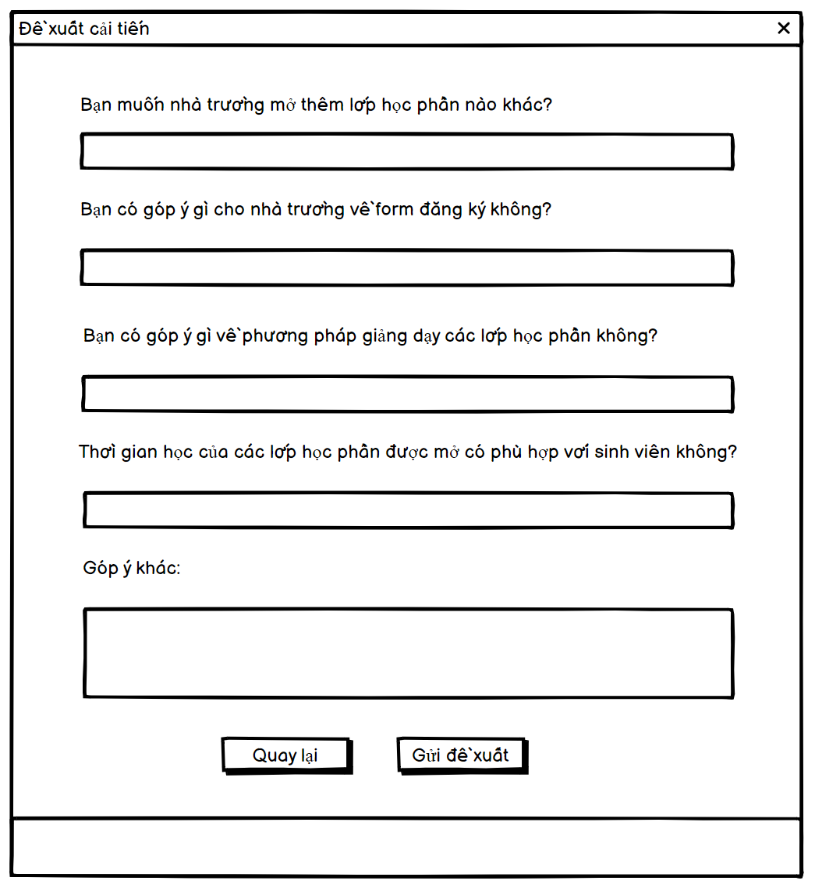
#### Chức năng thống kê chi tiết danh sách đơn đăng ký theo học phần (Nguyễn Hoàng Hải)

* Các thông tin đầu vào: mã đơn đăng ký, mã sinh viên, mã học phần, mã khoa, loại đơn, trạng thái, số lượng sinh viên.
* Chức năng này cho phép quản trị viên thống kê danh sách đơn đăng ký có trạng thái chưa duyệt trong một lớp học phần.
* Khi người dùng kích vào nút “Duyệt”, chương trình sẽ thống kê số lớp học phần được mở của học phần đó theo số lượng sinh viên đăng ký và cập nhật trạng thái cho các đơn trong cơ sở dữ liệu.
* Phác thảo giao diện:



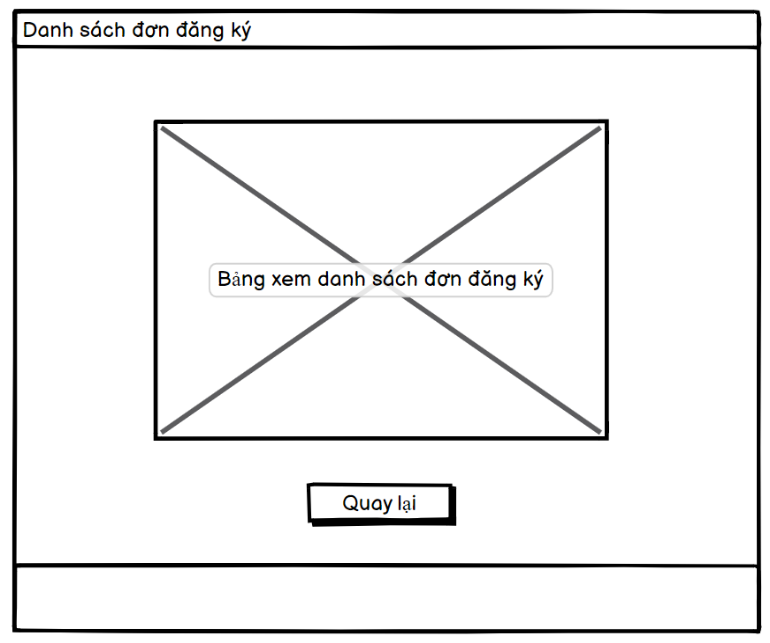
#### Chức năng gửi đơn đề xuất cải tiến (Trần Đại Phú)

* Các thông tin đầu vào: Mã sinh viên, mã đơn, nội dung các câu hỏi.
* Chức năng này cho phép sinh viên trả lời các câu hỏi đề xuất và gửi đề xuất.
* Khi sinh viên gửi đơn đề xuất thành công chương trình sẽ tự động tạo mã đơn đề xuất, lấy mã sinh viên đại diện (mã sinh viên đang đăng nhập) và những thông tin đầu vào để ghi dữ liệu vào file DonDeXuat.txt
* Phác thảo giao diện



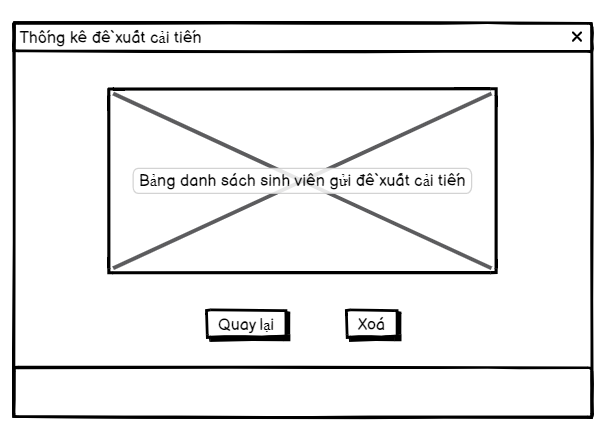
#### Chức năng xem danh sách đơn đăng ký (Trần Đại Phú)

* Các thông tin đầu vào: mã đơn đăng ký, mã học phần, tên học phần, loại đơn, trạng thái.
* Chức năng này cho phép sinh viên xem danh sách đơn đăng ký theo cá nhân và đơn đăng ký theo tập thể có trạng thái chưa duyệt.
* Phác thảo giao diện:



#### Chức năng thống kê đề xuất cải tiến (Nguyễn Hoàng Hải)

* Các thông tin đầu vào: Mã sinh viên, mã đơn, nội dung các câu hỏi.
* Chức năng này cho phép quản trị viên thống kê danh sách, xem chi tiết, xóa đơn đề xuất của sinh viên.
* Phác thảo giao diện

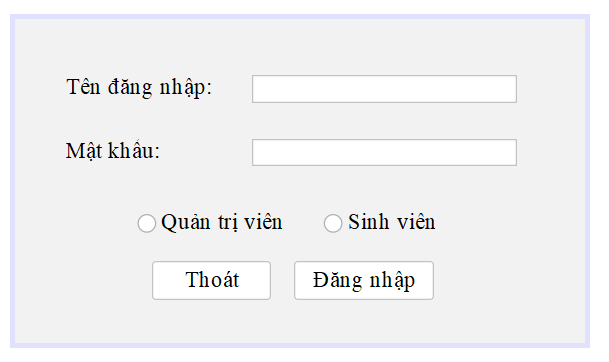


## Thực hiện bài toán

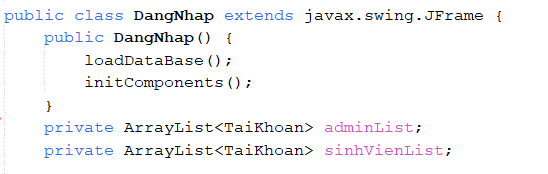
* **Về giao diện:**
* Các giao diện của chương trình được xây dựng bằng phương pháp kéo thả trên IDE Netbean. Việc xây dựng giao diện của hệ thống bằng phương pháp kéo thả trên IDE NetBeans sẽ giúp phát triển các ứng dụng và hệ thống có giao diện người dùng nhanh chóng hơn. Bên cạnh đó có thể tạo các ứng dụng với giao diện người dùng đơn giản hoặc phức tạp một cách thuận tiện mà không cần phải viết mã Java bằng tay.
* Thực hiện kéo thả các component JLabel, JTextField, JRadioButton, JButton, JTable…. từ Palette vào JFrame.
* Thực hiện kích chuột phải vào component, chọn Property để thay đổi các thuộc tính name, fontsize, text... của các components đó.
* Cơ chế kết hợp với Table: đặt tên cho table, tạo một DefaultTableModel mới sau đó getModel() của bảng đang làm việc với DefaultTableModel vừa tạo, thực hiện các thao tác thêm sửa xóa các dòng và hiển thị lên table.
* **Về thao tác đọc ghi file:** Do CSDL được lưu vào file txt nên nhóm sử dụng 2 lớp đối tượng Writer và Reader để đơn giản hoá cho việc thực hiện truy xuất và lưu trữ dữ liệu vào/ra file txt.

### Vũ Thị Hường – Chức năng đăng nhập

* Chức năng này cho phép người dùng sử dụng tài khoản của mình để truy cập vào chương trình và sử dụng các tính năng.
* Giao diện code trong chương trình:

****

* Mã lệnh
* **Hướng đối tượng:** Áp dụng tính đóng gói, tính kế thừa:



A group of black text

Description automatically generated

Lớp DangNhap kế thừa lớp javax.swing.JFame sử dụng các chức năng tương tác với với các thành phần trên giao diện.

Các thuộc tính danh sách tài khoản Quản trị viên (adminList) và Sinh viên (sinhVienList) được đóng gói, ngăn không cho bên ngoài truy cập trực tiếp vào. Tương tự với các thuộc tính maTK, matKhau của lớp TaiKhoan.

* **Bắt lỗi và gom rác:**
* Mã nguồn:

A computer screen shot of text

Description automatically generated

Hàm `checkLogin` thực hiện bẫy lỗi sử dụng `throws Exception` mang ý nghĩa phương thức trả ra một ngoại lệ, yêu cầu hàm gọi đến phương thức này phải thực hiện xử lý ngoại lệ đó.

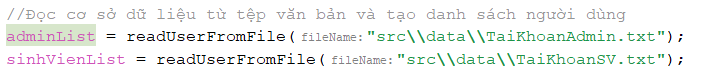
Đầu tiên, kiểm tra các trường có rỗng không, nếu cả hai đều rỗng, ném ra một ngoại lệ (Exception) với thông báo "Vui lòng điền đầy đủ thông tin!".

Sau đó, kiểm tra xem người dùng đã lựa chọn vai trò đăng nhập vào chương trình hay chưa (qua việc kiểm tra xem radio button "Quản trị viên" và "Sinh viên" có được chọn hay không). Nếu cả hai đều không được chọn, ném ra một ngoại lệ với thông báo "Vui lòng chọn vai trò đăng nhập".

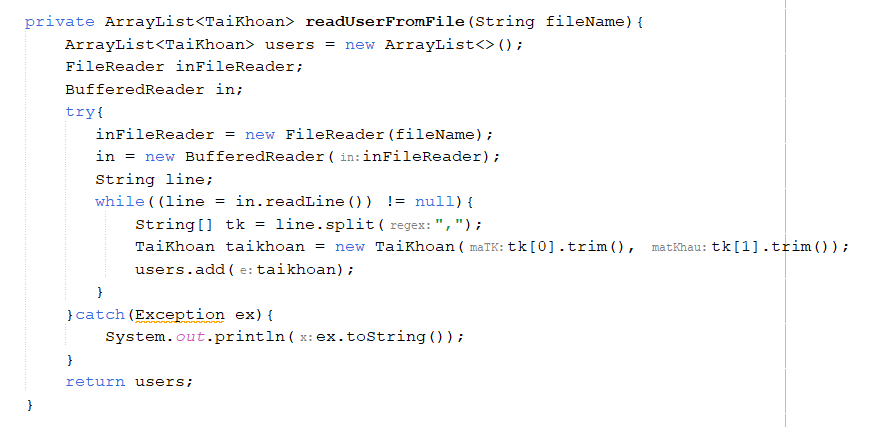
Cuối cùng, kiểm tra thông tin đăng nhập của Quản trị viên nếu radio button "Quản trị viên" được chọn, ngược lại, kiểm tra thông tin đăng nhập của Sinh viên. So sánh mã tài khoản và mật khẩu được nhập với thông tin của từng tài khoản trong danh sách. Nếu tìm thấy tài khoản khớp trong danh sách quản trị viên adminList thì trả về 0, tìm thấy trong danh sách tài khoản sinh viên sinhVienList thì trả về 1, không tìm thấy trong danh sách nào thì trả về 2.

* **Tập hợp:** Danh sách tài khoản sinh viên và tài khoản quản trị viên được lưu vào 2 danh sách sử dụng ArrayList giúp xét duyệt danh sách một cách dễ dàng, thuận tiện khi sử dụng.





* **Thao tác file**
* Đọc file



Hàm `readUserFromFile` trên có chức năng đọc thông tin tài khoản từ một tệp tin và trả về một danh sách (ArrayList) chứa các đối tượng TaiKhoan.

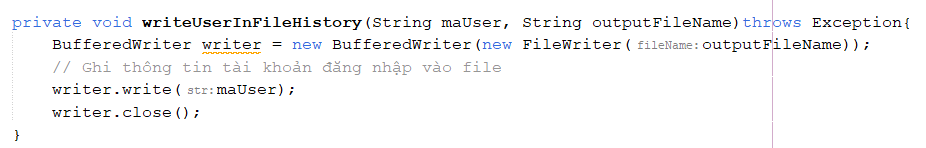
Trong khối try, sử dụng đối tượng FileReader và BufferedReader để đọc từng dòng dữ liệu trong tệp tin (mỗi dòng trong file là một tài khoản).

Nếu có bất kỳ lỗi nào xảy ra trong quá trình đọc tệp tin sẽ được xử lý trong khối catch, và thông báo lỗi được in ra màn hình.

Tách thông tin trong mỗi dòng thành hai phần, mã tài khoản và mật khẩu, sử dụng phương thức split(",").

Tạo đối tượng TaiKhoan mới với mã tài khoản và mật khẩu được tách trên. Sau đó thêm đối tượng này vào danh sách users thông qua phương thức add.

* Ghi file



Hàm `writeUserInFileHistory` được dùng để lưu dấu tài khoản mà người dùng đăng nhập vào file LichSuDangNhap.txt

### Vũ Thị Hường – Chức năng đổi mật khẩu

* Giao diện code trong chương trình



* Mã lệnh
* **Hướng đối tượng:** Áp dụng tính kế thừa



* **Bắt lỗi và gom rác**

A screenshot of a computer code

Description automatically generated

Hàm `checkChangePassword()` giúp kiểm tra sự hợp lệ của việc thay đổi mật khẩu của người dùng.

Đầu tiên, lấy ra các giá trị trong các ô textbox mà người dùng nhập vào sử dụng `getPassword()`.

Tiếp theo, kiểm tra các trường hợp ngoại lệ:

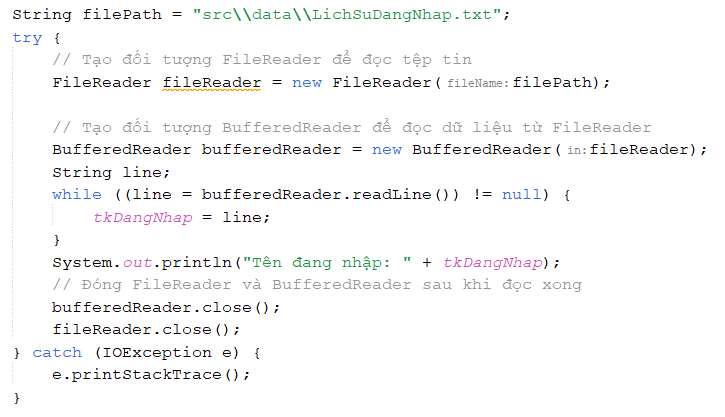
* Có ô nào bị bỏ trống hay không.
* Mật khẩu cũ nhập vào đã khớp với mật khẩu đăng nhập chưa
* Trường mật khẩu mới và nhập lại mật khẩu mới có khớp nhau hay không.

Nếu có bất kỳ ngoại lệ nào được ném (do việc kiểm tra không hợp lệ), hiển thị một hộp thoại thông báo lỗi (JOptionPane) với thông điệp từ ngoại lệ và trả về false. Nếu mọi thứ đều hợp lệ, trả về true.

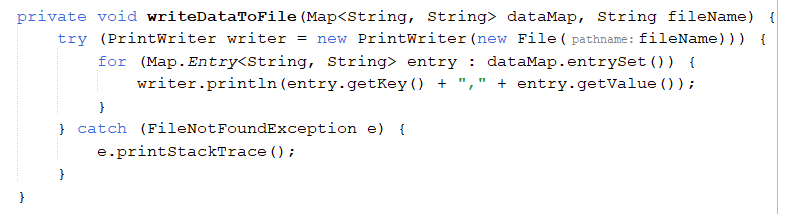
* **Tập hợp:** Sử dụng Map để lưu danh sách các tài khoản dưới dạng key-value, với key là mã tài khoản (tương ứng là tên đăng nhập) và value là mật khẩu tương ứng với tài khoản đó.

Lý do sử dụng Map:

* Map cung cấp cơ chế tra cứu hiệu quả, giúp kiểm soát tài khoản và truy xuất thông tin tài khoản một cách nhanh chóng.
* Tổ chức dữ liệu một cách đồng nhất và có tổ chức giúp thêm, sửa, xóa các tài khoản dễ dàng. Mỗi khóa định rõ một tài khoản cụ thể và giữ thông tin liên quan đến tài khoản đó.
* **Thao tác file**
  + Đọc file: đọc lấy mã tài khoản người dùng đang đăng nhập vào hệ thống từ file LichSuDangNhap.txt để kiểm tra mật khẩu của tài khoản này trong file tài khoản của người dùng đăng nhập tương ứng.

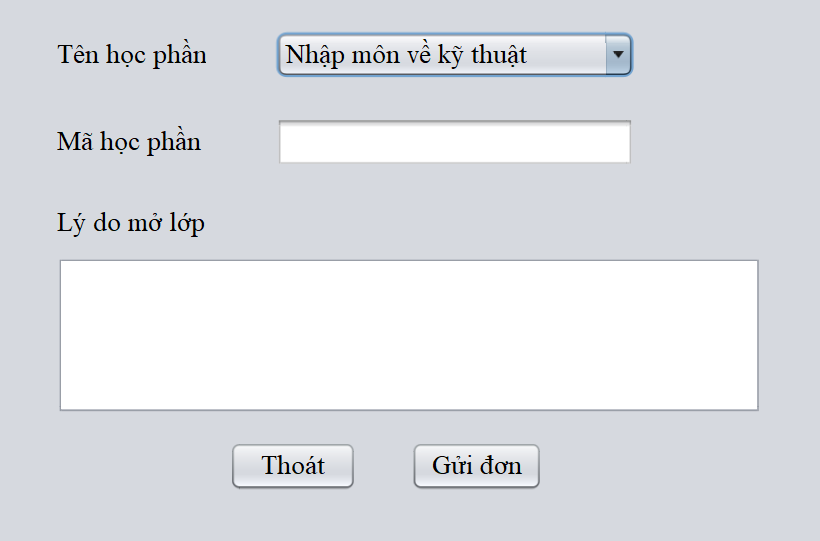


* Ghi file: Sau khi thay đổi mật khẩu thành công, tiến hành cập nhật lại mật khẩu mới của người dùng đó.



### Nguyễn Minh Đức - Chức năng gửi đơn đăng ký cá nhân

* Giao diện:



Ghi chú:

+ Nội dung nhận là một jTextArea dùng để nhập lý do gửi đơn

+ Nội dung chọn là một jComboBox dùng để chọn tên học phần muốn đăng kí.

* Mã lệnh
* **Hướng đối tượng**
  + - Áp dụng tính kế thừa và tính đóng gói để bảo vệ dữ liệu, ngăn chặn truy cập trực tiếp đến các thuộc tính của dữ liệu từ bên ngoài.

A screen shot of a computer code

Description automatically generated

A screenshot of a computer program

Description automatically generated

* Các nội dung thực hiện
* Tự động sinh mã đơn cá nhân khi người dùng gửi một đơn cá nhân mới.
* Tự động hiển thị ra mã học phần tương ứng khi người dùng chọn một học phần.
* Sử dụng cửa sổ cha, cửa sổ con bằng cách cho cửa sổ con kế thừa javax.swing.JDialog
* Kiểm tra nội dung các ô nhập không được để trống
* Kiểm tra tính hợp lệ của đơn: Nếu đơn đăng ký đã tồn tại trong file DSDonCaNhan.txt hoặc đã tồn tại trong file DSDonTapThe.txt tức mã sinh viên và mã học phần đã tồn tại trên 1 bản ghi.
* **Bẫy lỗi và gom rác**
* Mã nguồn

A computer code with text

Description automatically generated

A screenshot of a computer code

Description automatically generated

A screenshot of a computer

Description automatically generated

Ghi chú: Kiểm tra nội dung nhập không được để trống, đơn mới không được trùng với đơn cũ (không trùng mã sinh viên và mã học phần).

* **Tập hợp**
* Loại tập hợp sử dụng: List.



* **Thao tác với file**
* Đọc dữ liệu từ file

A computer screen shot of text

Description automatically generated

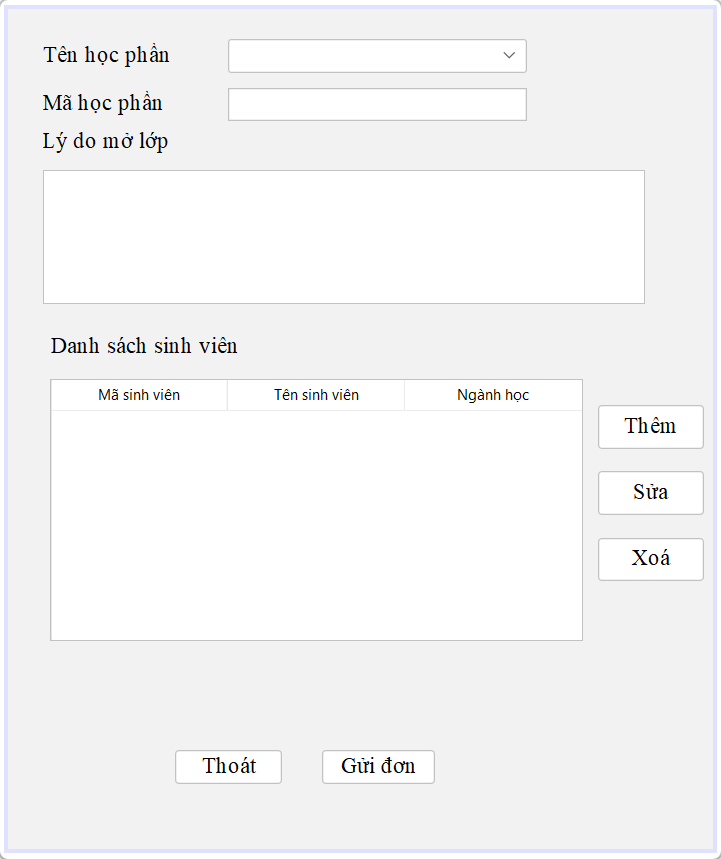
* Lưu dữ liệu vào file

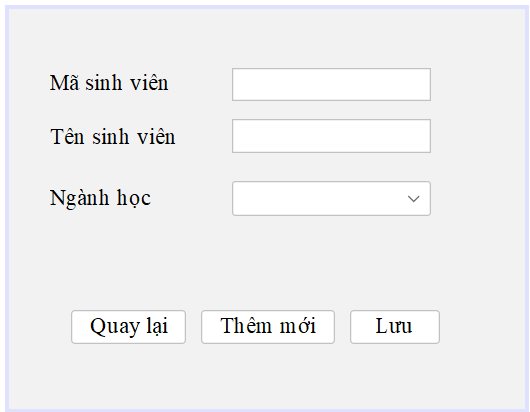
A close up of a text

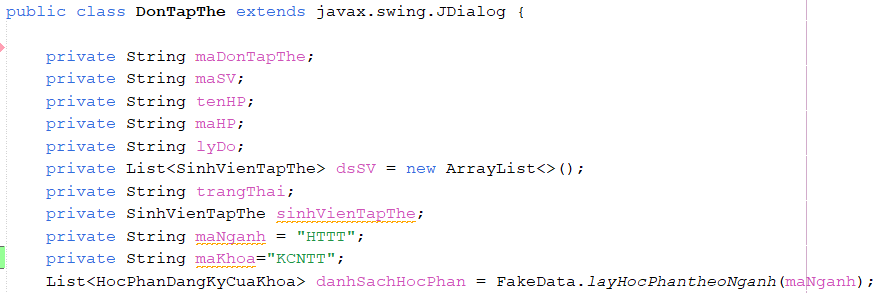
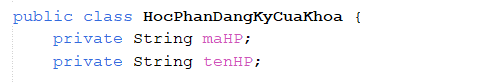
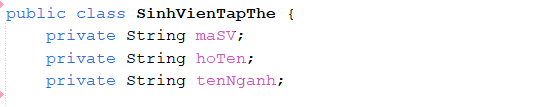
Description automatically generated

### Nguyễn Minh Đức - Chức năng gửi đơn đăng ký theo tập thể

* Giao diện này cho phép sinh viên nhập các thông tin tên học phần, mã học phần, lý do mở lớp, danh sách sinh viên tập thể. Khi người dùng bấm vào nút thêm, sửa sẽ hiện ra giao diện con để sinh viên có thể nhập hoặc sửa thông tin sinh viên trong danh sách sinh viên.



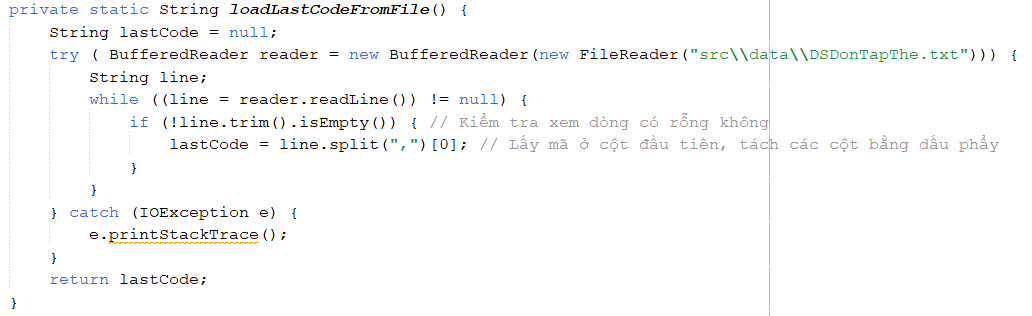


* Mã lệnh
* **Hướng đối tượng**
* Áp dụng được tính đóng gói, tính kế thừa
* Các nội dung thực hiện
* Tự động sinh mã đơn tập thể khi người dùng gửi một đơn tập thể mới.
* Tự động hiển thị ra mã học phần tương ứng khi người dùng chọn một học phần và ngược lại.
* Cho phép người dùng thêm, sửa, xóa thông tin sinh viên tập thể bao gồm: mã sinh viên, họ tên, ngành học
* Sử dụng cửa sổ cha, cửa sổ con bằng cách cho cửa sổ con kế thừa javax.swing.JDialog
* Kiểm tra nội dung các ô nhập không được để trống
* Kiểm tra tính hợp lệ của đơn: Nếu đơn đăng ký đã tồn tại trong file DSDonTapthe.txt tức Mã sinh viên và mã học phần đã tồn tại trên 1 bản ghi. Nếu trong danh sách sinh viên tập thể có sinh viên trùng mã sinh viên hoặc mã sinh viên đại diện tồn tại trong danh sách sinh viên tập thể thì đơn không hợp lệ.
* **Bẫy lỗi và gom rác**

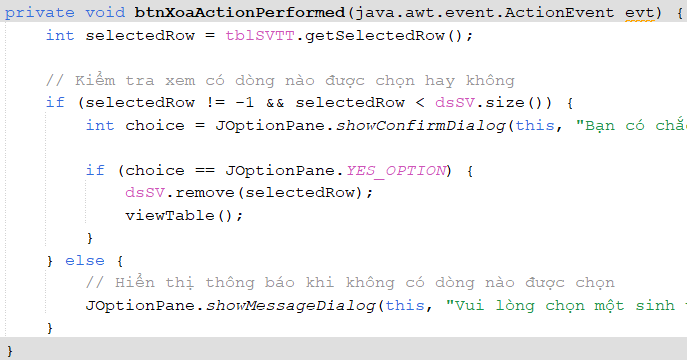
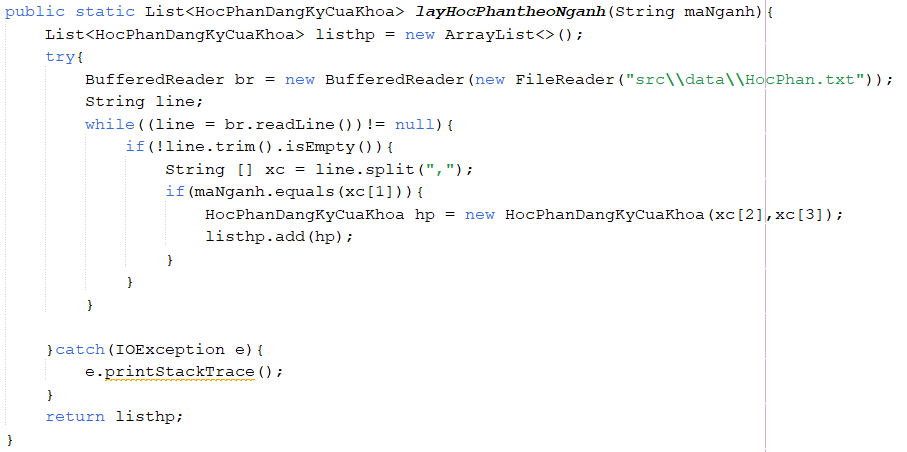
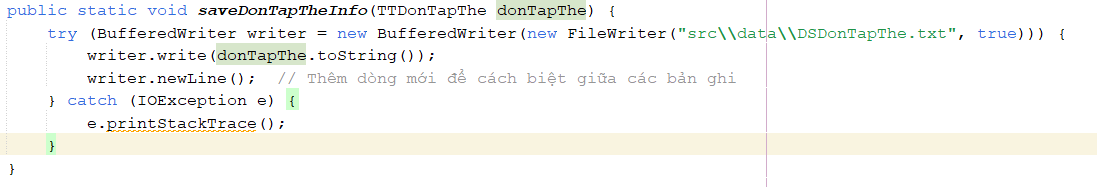
Ghi chú: Sử dụng cơ chế bẫy lỗi bằng try catch. Khi người dùng không chọn sinh viên nào để sửa thì khi người dùng bấm nút sửa chương trình sẽ thống báo lỗi lên màn hình.



Chú thích: bẫy lỗi kiểm tra tính hợp lệ của đơn đăng ký theo tập thể



Chú thích: bẫy lỗi đọc file từ tệp

* **Tập hợp**
* Loại tập hợp sử dụng: List, ArrayList. Việc sử dụng loại tập hợp này vì nó cung cấp các chức năng thêm, sửa, xóa các phần tử trong tập hợp. Nó cũng cho phép truy cập các phần tử thông qua chỉ số.
* Các phép toán sử dụng: add(), remove(), size(), get()
* **Thao tác với files**
* Lấy học phần theo ngành từ file txt
* Lưu dữ liệu vào file khi người dùng gửi thông tin đơn tập thể

### Trần Đại Phú - Chức năng gửi đơn đề xuất

* Giao diện này cho phép sinh viên nhập các thông tin tên học phần, mã học phần, lý do mở lớp, danh sách sinh viên tập thể. Khi người dùng bấm vào nút thêm, sửa sẽ hiện ra giao diện con để sinh viên có thể nhập hoặc sửa thông tin sinh viên trong danh sách sinh viên.
* Giao diện chương trình

A screenshot of a computer

Description automatically generated

* Mã lệnh
* **Hướng đối tượng**
* Áp dụng được tính đóng gói, tính kế thừa

A screenshot of a computer code

Description automatically generated

* Các nội dung thực hiện
* Tự động sinh mã đơn đề xuất khi người dùng gửi một đơn đề xuất mới.
* Sử dụng cửa sổ cha, cửa sổ con bằng cách cho cửa sổ con kế thừa javax.swing.JDialog
* Kiểm tra nội dung các ô nhập không được để trống toàn bộ.
* **Bẫy lỗi và gom rác**A screenshot of a computer

  Description automatically generated

Sử dụng cơ chế bẫy lỗi bằng try catch. Khi cố gắng mở tệp để ghi, có thể xảy ra các lỗi như tệp không tồn tại hay thư mục không có quyền ghi,… khi đó ngoại lệ `IOException` sẽ được kích hoạt.

* **Thao tác file**
* Lấy mã sinh viên từ file txt

A computer screen shot of a computer code

Description automatically generated

* Lưu vào file txt

A screen shot of a computer

Description automatically generated

### Trần Đại Phú - Chức năng xem danh sách đơn đăng ký

* Giao diện code chương trình

A screenshot of a computer

Description automatically generated

* Mã lệnh
  + **Hướng đối tượng**
* Áp dụng tính kế thừa và tính đóng gói để bảo vệ dữ liệu, ngăn chặn truy cập trực tiếp đến các thuộc tính của dữ liệu từ bên ngoài.

A close-up of a computer code

Description automatically generated

A screenshot of a computer code

Description automatically generated

* **Bắt lỗi và gom rác**

**A computer screen shot of a code

Description automatically generated**

`listDonCaNhan`, `listDonTapThe` được đọc từ file txt danh sách đơn đăng ký. Thực hiện duyệt hai list này trong try-catch, nếu có bất ký lỗi nào trong quá trình đọc file, chương trình sẽ bắt lỗi và hiển thị thông báo.

* **Tập hợp**
* Sử dụng Map để lưu danh sách các đơn đăng ký, với key là mã sinh viên và value là danh sách các đơn có cùng mã sinh viên đó. Giúp việc truy xuất và cập nhật trạng thái của các đơn thuộc cùng một sinh viên dễ dàng hơn, đảm bảo tính nhất quán, có cấu trúc tổ chức rõ ràng cho dữ liệu.

A screenshot of a computer code

Description automatically generated

* Sử dụng List, ArrayList để lấy ra danh sách đơn theo mã sinh viên để hiển thị danh sách đơn này khi sinh viên muốn xem danh sách học phần của mình đã đăng ký.

A screen shot of a computer code

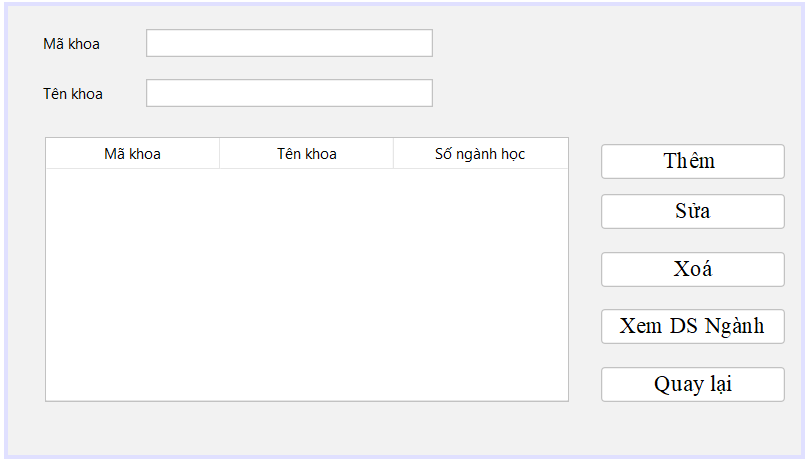
Description automatically generated

Hàm ‘dsDonTheoMaSV()’ giúp lấy ra danh sách các đơn theo mã sinh viên đang đăng nhập.

List giúp khả năng lặp qua danh sách một cách dễ dàng, không cần ràng buộc key-value một cách chặt chẽ.

### Nguyễn Thế Đạt - Chức năng Quản lý thông tin khoa

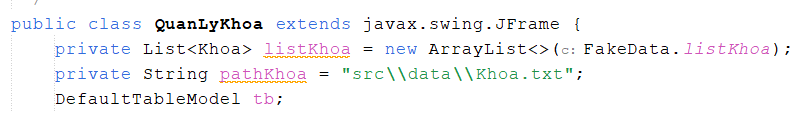
* Giao diện



* Mã lệnh
* **Hướng đối tượng**
* Áp dụng được tính đóng gói, tính kế thừa bảo vệ dữ liệu, bảo vệ tính toàn diện của dữ liệu, kiểm soát quyền truy cập dữ liệu từ ngoài đối tượng.

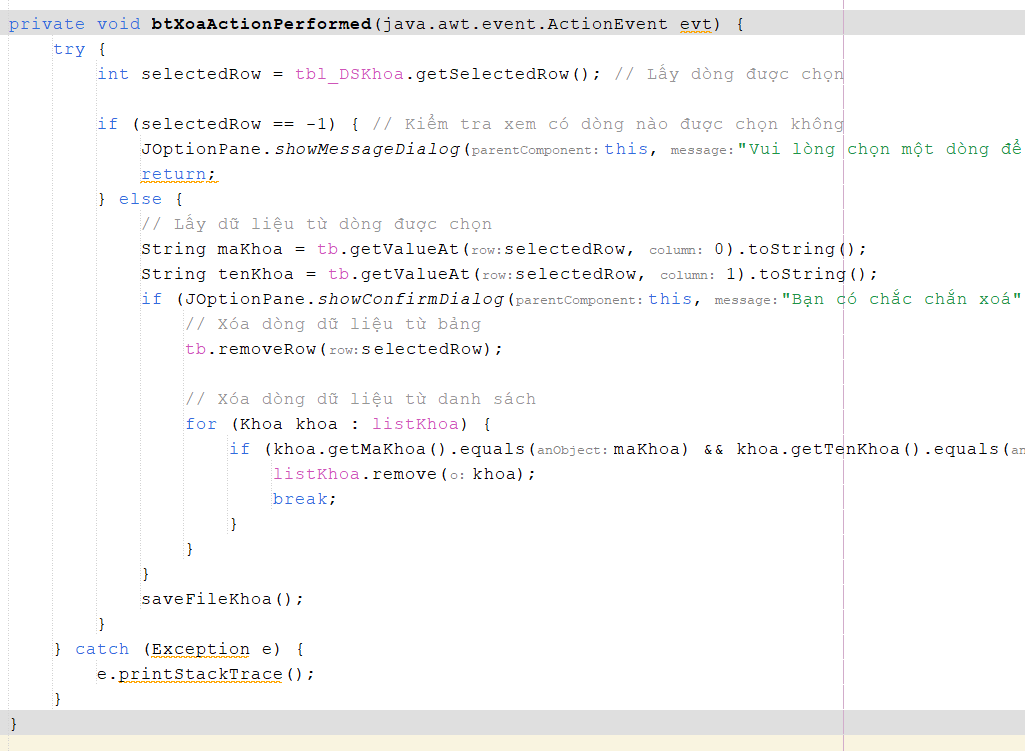
A close up of words

Description automatically generated



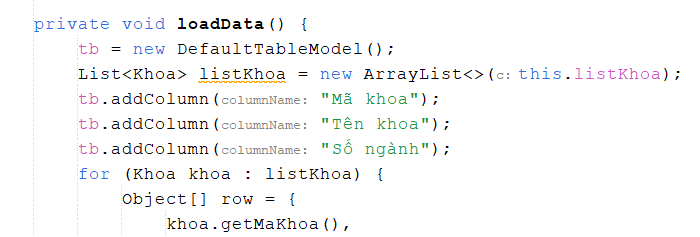
Nội dung thực hiện: Hiển thị thông tin các ngành có trong khoa dự trên thông tin mã khoa mà người dùng được chọn

* **Bắt lỗi và gom rác**



Chú thích: sử dụng bẫy lỗi try catch in thông tin lỗi nếu có lỗi xảy ra, giúp ngăn chặn chương trình dừng đột ngột khi gặp lỗi và xử lý lỗi một cách kiểm soát.

* **Tập hợp**

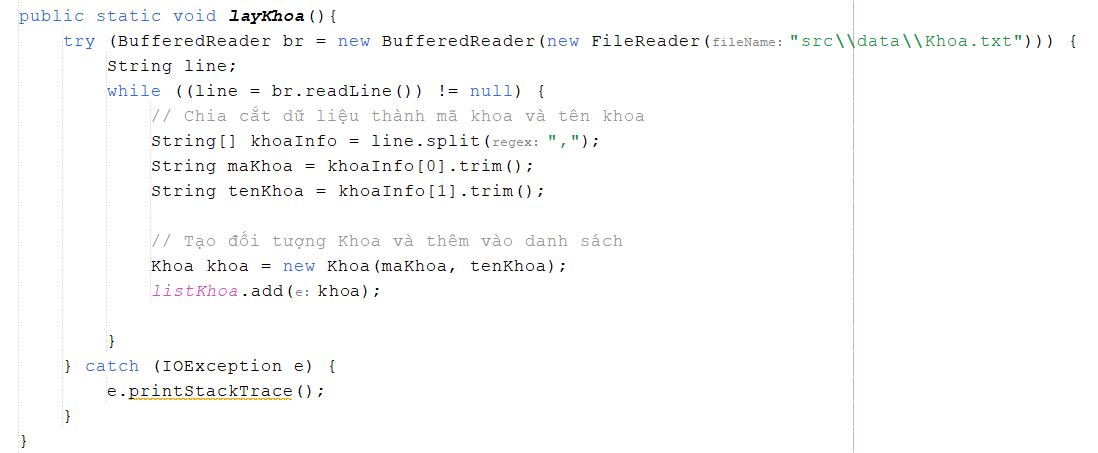


Sử dụng List, ArrayList lưu danh sách khoa để hiển thị lên table.

Tạo danh sách các khoa, số ngành có trong khoa hiển thị lên table.

Thêm, sửa, xoá phần tử truy cập vào danh sách để cập nhật giao diện hiển thị tương ứng.

* **Thao tác với files:**
* Đọc file:

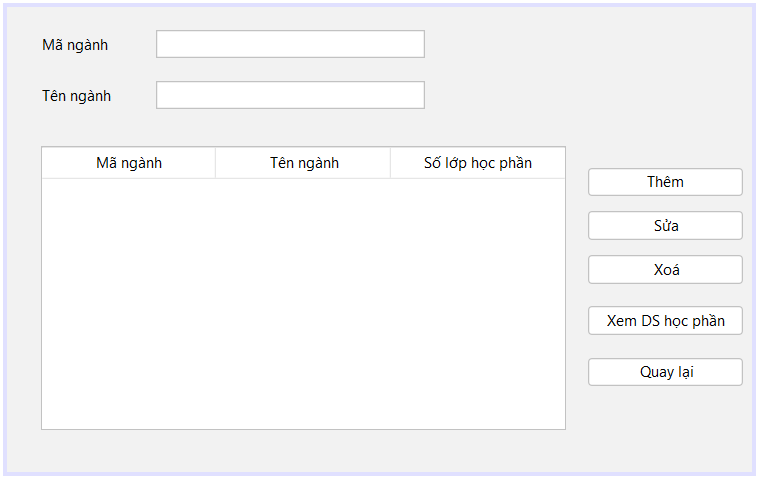


* Ghi file:

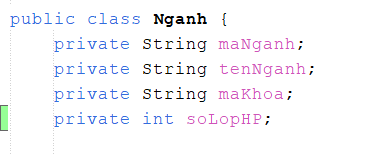


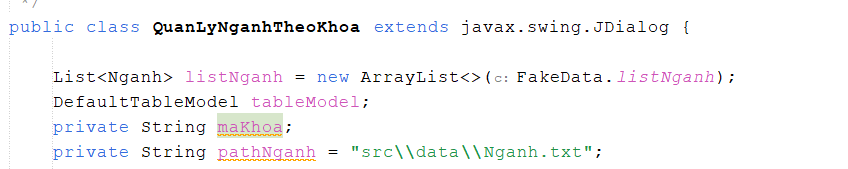
### Nguyễn Thế Đạt - Chức năng Quản lý danh sách ngành theo khoa

* Giao diện

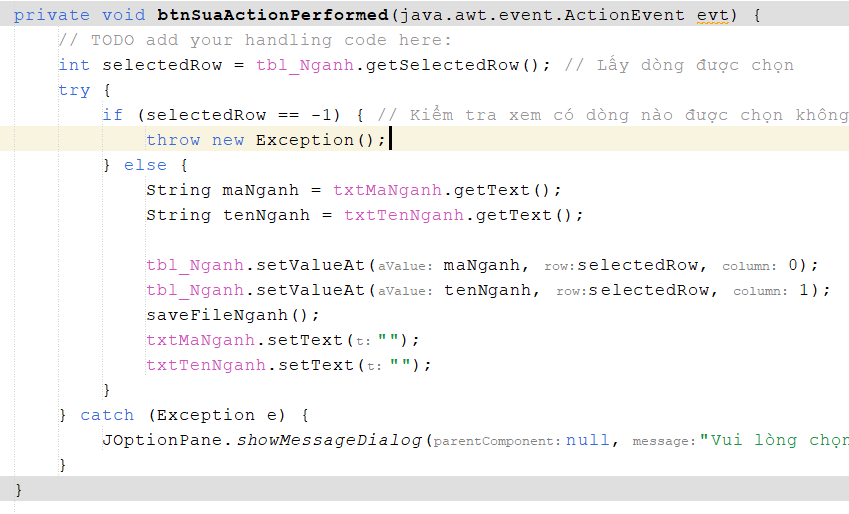


* Mã lệnh
* **Hướng đối tượng**
* Áp dụng được tính đóng gói, tính kế thừa



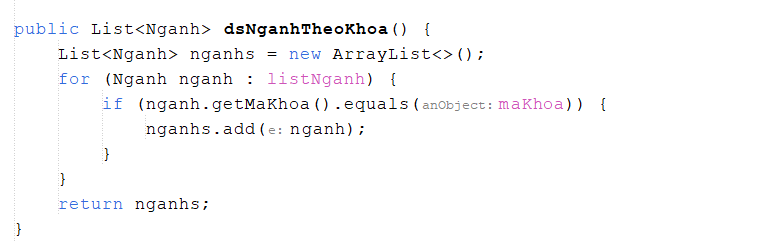


* Các nội dung thực hiện Cho phép người dùng thêm, sửa, xóa thông tin sinh viên tập thể bao gồm: mã sinh viên, họ tên, ngành học
* **Bắt lỗi và gom rác**



Chú thích: sử dụng bẫy lỗi try catch in thông tin lỗi chọn dòng nếu có lỗi xảy ra, nếu không có dòng nào được chọn, ném ra một ngoại lệ để không làm gián đoạn chương trình

* **Tập hợp**

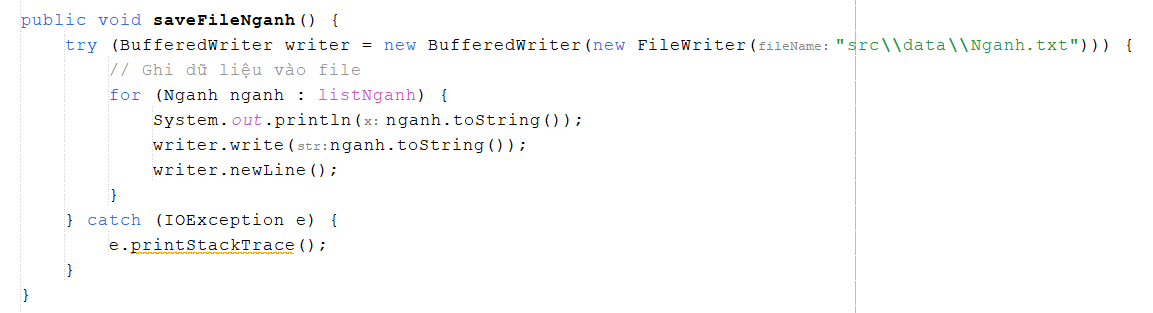


Sử dụng List, ArrayList để lấy ra danh sách ngành theo mã khoa, giúp lặp qua danh sách một cách dễ dàng.

* **Thao tác với files**
* Đọc file:

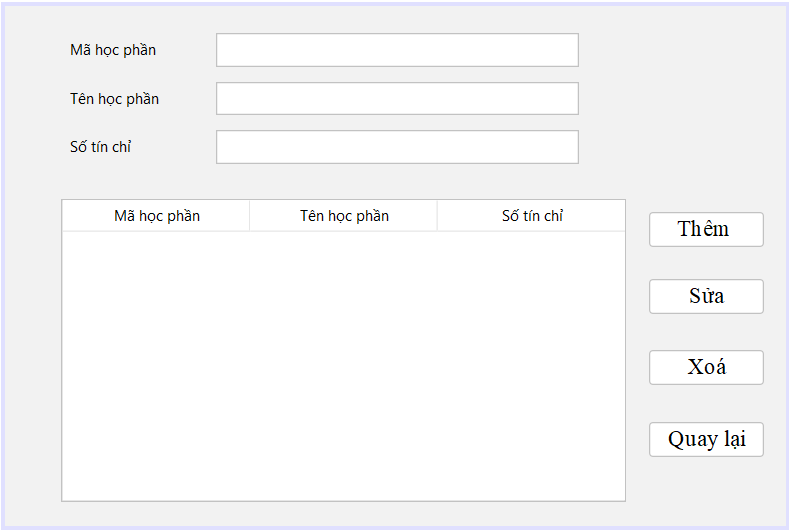


* Ghi file:



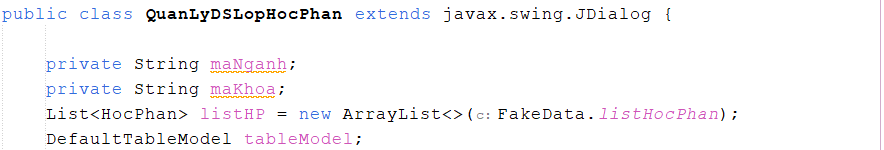
### Nguyễn Thế Đạt - Chức năng Quản lý danh sách lớp học phần theo ngành

* Giao diện



* Mã lệnh
* Hướng đối tượng
* Áp dụng được tính đóng gói, tính kế thừa





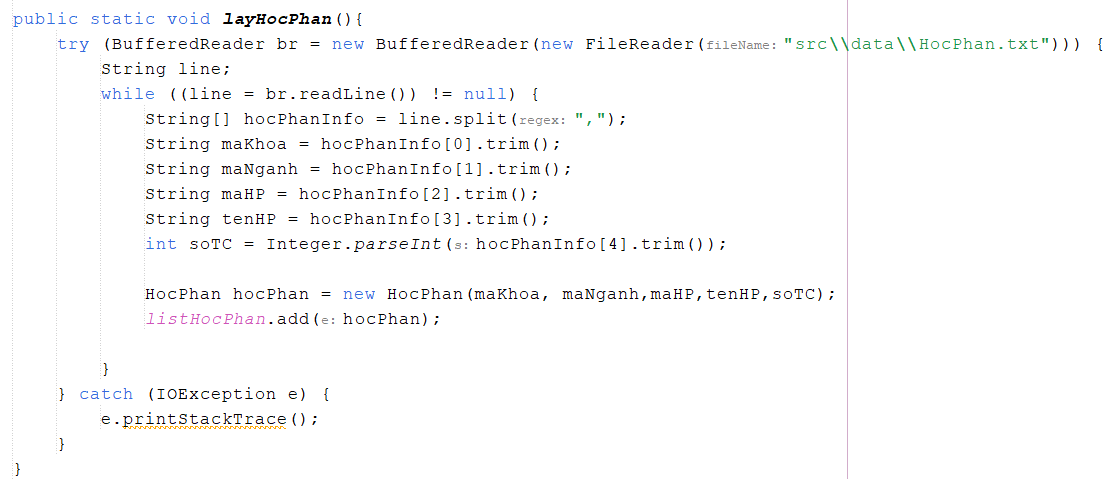
* Các nội dung thực hiện Cho phép người dùng thêm, sửa, xóa thông tin sinh viên tập thể bao gồm: mã sinh viên, họ tên, ngành học
* **Bắt lỗi và gom rác**

A close-up of a computer screen

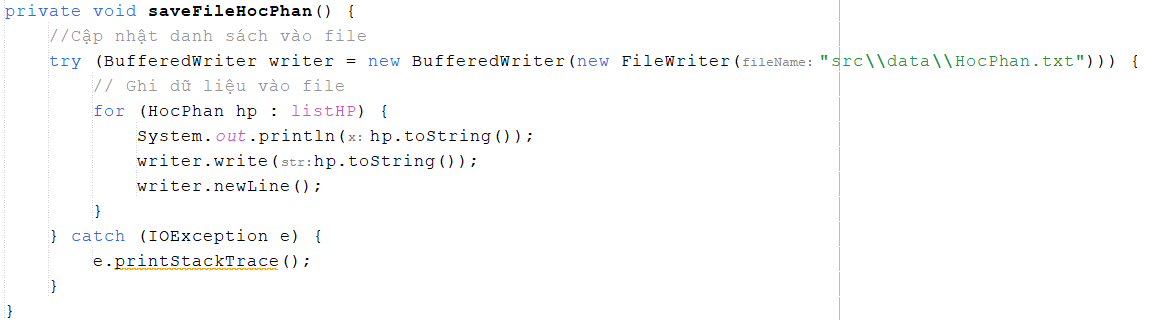
Description automatically generated

Chú thích: Khi người dùng nhập thông tin của số tín chỉ không phải là số nguyên thì thông báo lỗi cho người dùng, và trả về khỏi phương thức.

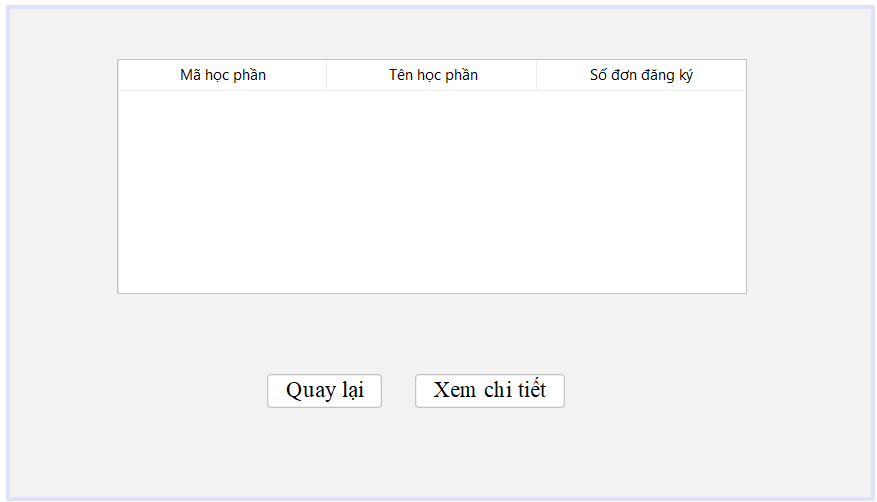
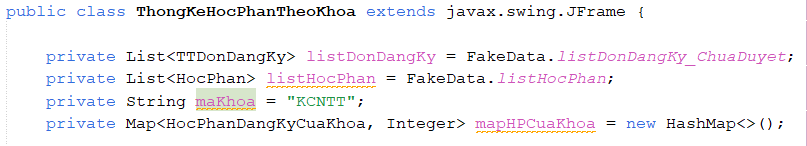
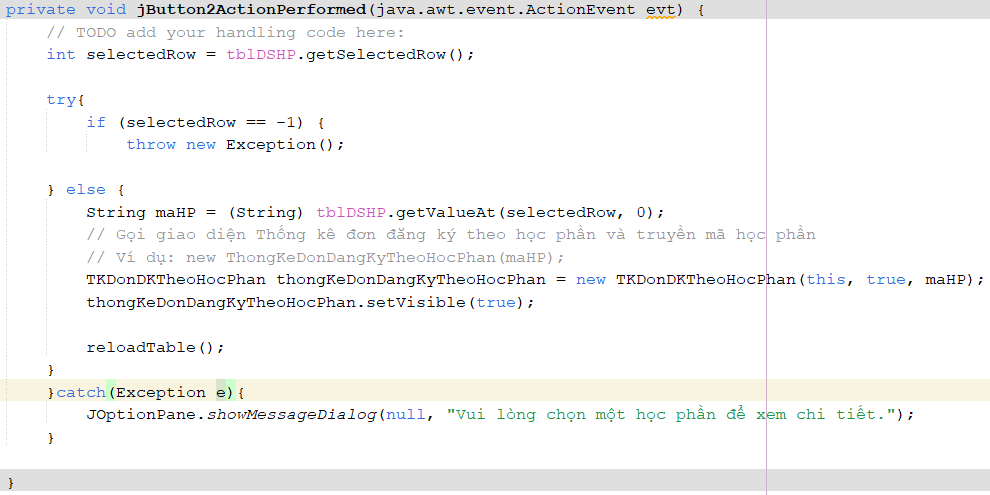
* **Tập hợp:** List giúp khả năng lặp qua danh sách một cách dễ dàng, không cần ràng buộc key-value một cách chặt chẽ.
* **Thao tác với files**
* Đọc file:



* Ghi file:

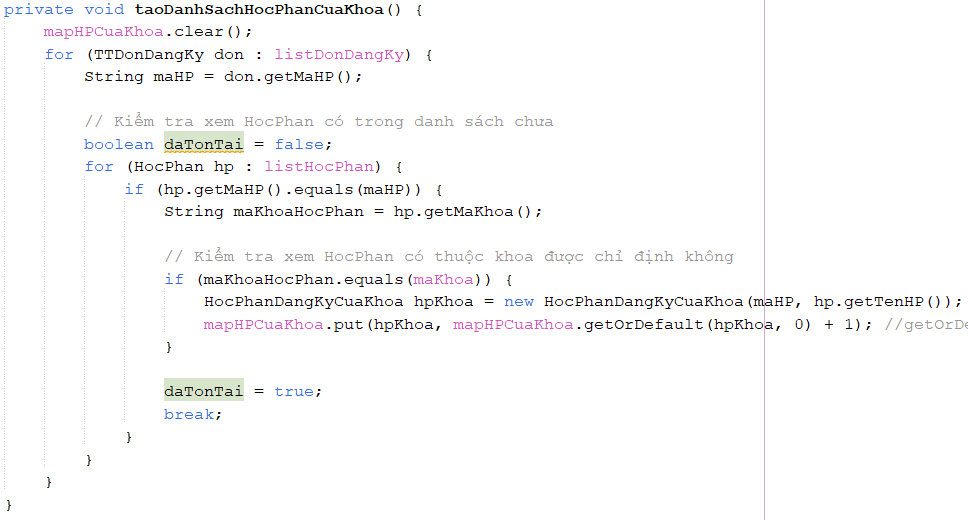


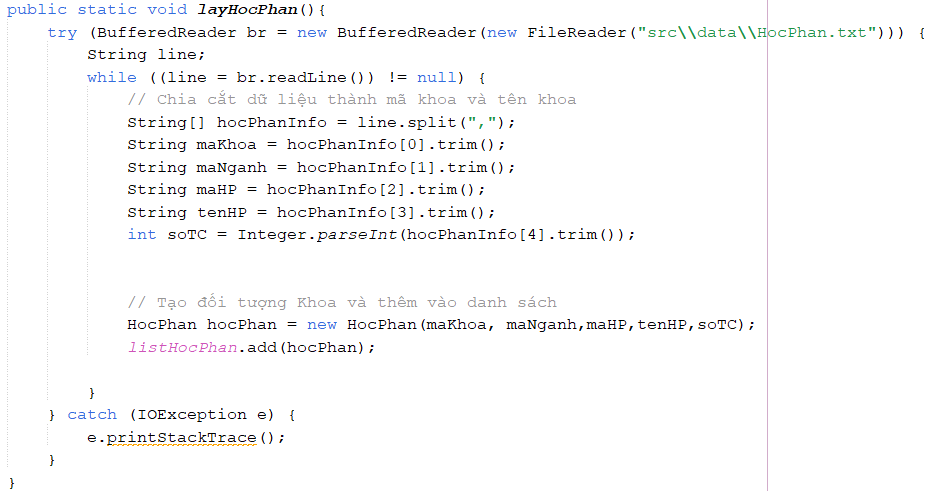
### Nguyễn Minh Đức- Chức năng thống kê danh sách học phần có đơn đăng ký theo khoa

* Giao diện này cho phép quản trị viên thống kê danh sách học phần có đơn đăng ký theo khoa
* 
* Mã lệnh
  + **Hướng đối tượng**
* Áp dụng được tính đóng gói, tính kế thừa
* Các nội dung thực hiện
* Tạo danh sách các học phần có đơn đăng ký theo khoa và số lượng đơn đăng ký.
* Hiển thị danh sách học phần có đơn đăng ký của khoa lên table và reload lại khi quản trị viên bấm duyệt đơn.
  + **Bẫy lỗi và gom rác** 

Ghi chú: Khi người dùng bấm nút xem chi tiết thì yêu cầu phải chọn một học phần trong table nếu không sẽ hiển thị thông báo lỗi.

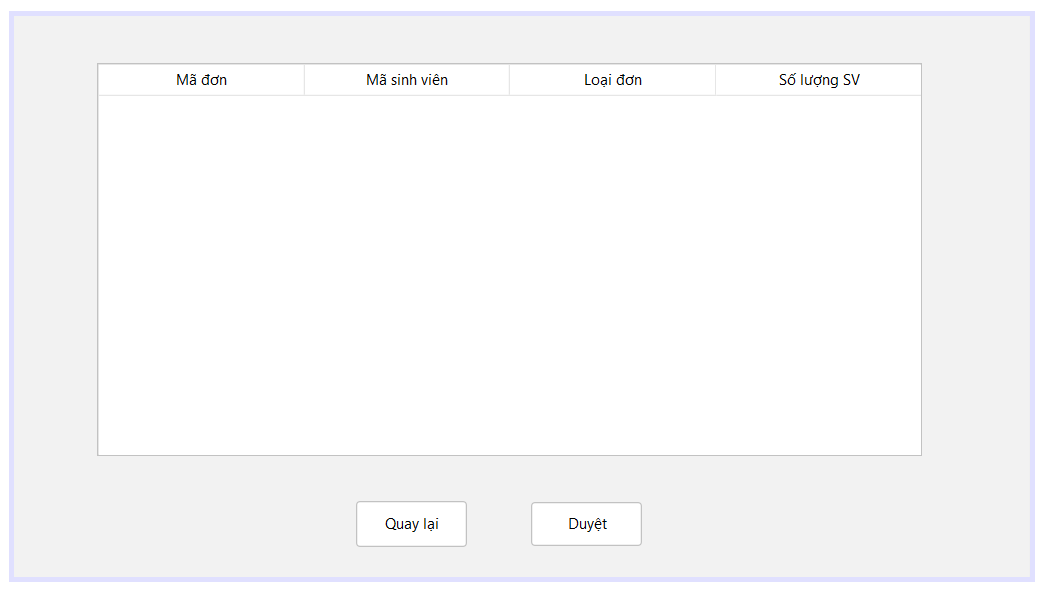
* **Tập hợp**
* Loại tập hợp sử dụng: List, ArrayList, Map, HashMap. Việc sử dụng loại tập hợp List, ArrayList vì nó cung cấp các chức năng thêm, sửa, xóa các phần tử trong tập hợp. Nó cũng cho phép truy cập các phần tử thông qua chỉ số. Việc sử dụng Map, HashMap vì loại tập hợp này cho phép lưu trữ phần tử dưới dạng key và value vậy nên key sẽ lưu các học phần để không bị trùng, còn value sẽ lưu số lượng đơn đăng ký có trong học phần đó.
* Các phép toán sử dụng: add(), remove(), size(), get(), clear(), put(), entrySet(), getOrDefault()



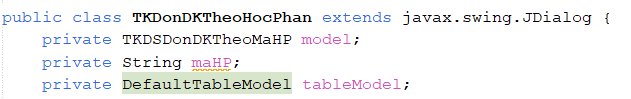
* **Thao tác với file**
* Lấy học phần từ file txt

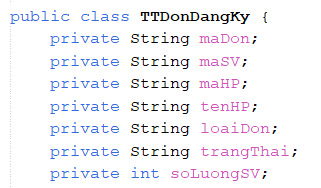
### Nguyễn Hoàng Hải - Chức năng thống kê danh sách đơn đăng ký theo lớp học phần

* Giao diện code chương trình

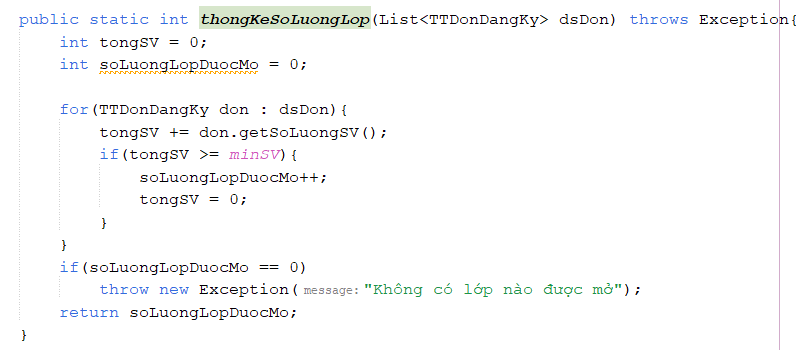


* Mã lệnh
  + **Hướng đối tượng**
* Áp dụng tính kế thừa và tính đóng gói để bảo vệ dữ liệu, ngăn chặn truy cập trực tiếp đến các thuộc tính của dữ liệu từ bên ngoài.





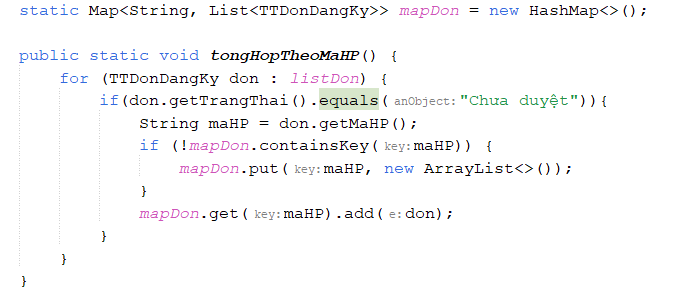
* **Bắt lỗi và gom rác**

****

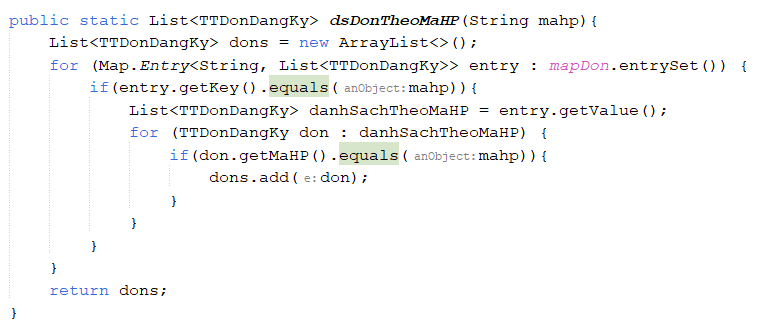
Hàm `thongKeSoLuongLop` trả về số lượng lớp học phần được mở dựa trên tổng số sinh viên trong tổng số các đơn thống kế được trong học phần đó

Khi người dùng ấn nút `Duyệt`, hàm này sẽ được gọi đến. Nếu soLuongLopDuocMo = 0 thì ném ra một ngoại lệ, thông báo cho người dùng và cập trạng thái của các đơn trong đó là `Không có lớp nào được mở`.

* **Tập hợp**
* Sử dụng Map để lưu danh sách các đơn đăng ký, với key là mã học phần và value là danh sách các đơn có cùng mã học phần đó. Giúp việc truy xuất và cập nhật trạng thái của các đơn thuộc cùng một học phần dễ dàng hơn, đảm bảo tính nhất quán, có cấu trúc tổ chức rõ ràng cho dữ liệu.



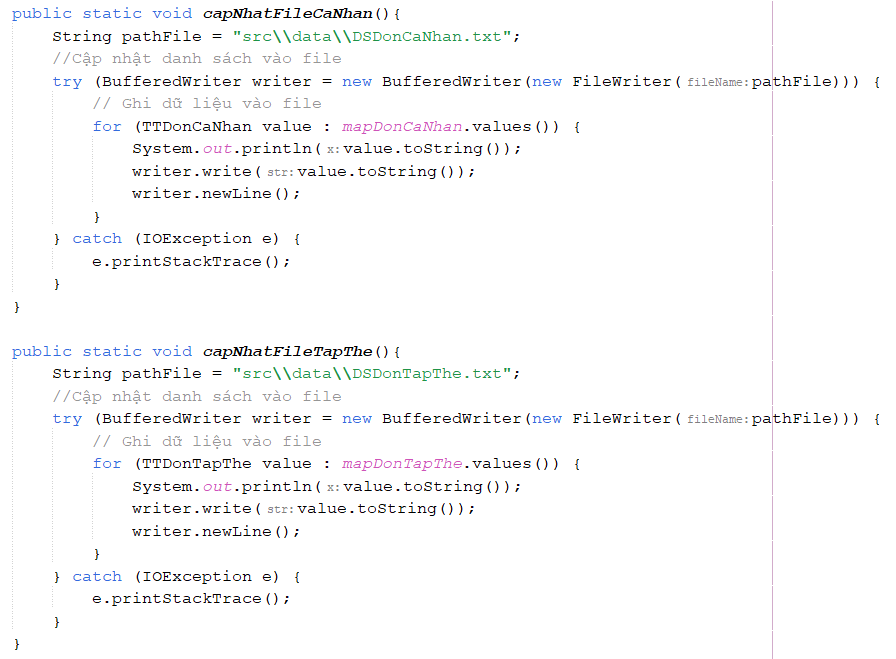
* Sử dụng List, ArrayList để lấy ra danh sách đơn theo mã học phần để hiển thị danh sách đơn này khi kích chuột vào một học phần nào đó.



Hàm `dsDonTheoMaHP()` giúp lấy ra danh sách các đơn theo mã học phần khi quản trị viên kích chuột vào một học phần nào đó trong bảng danh sách lớp học phần có đơn đăng ký.

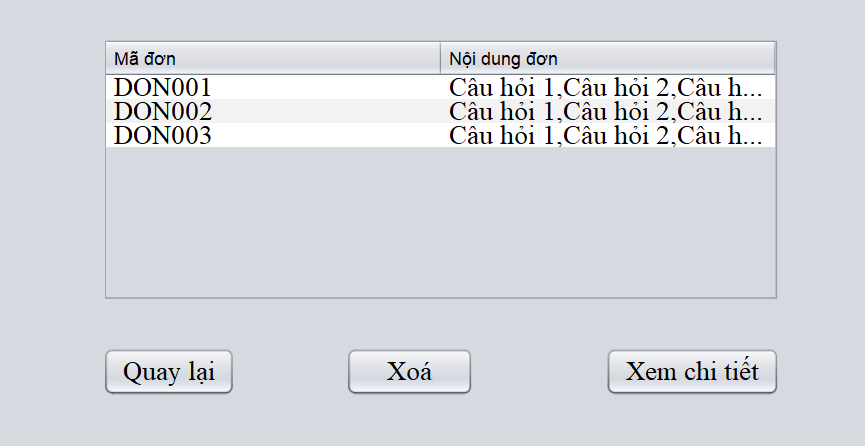
List giúp khả năng lặp qua danh sách một cách dễ dàng, không cần ràng buộc key-value một cách chặt chẽ.

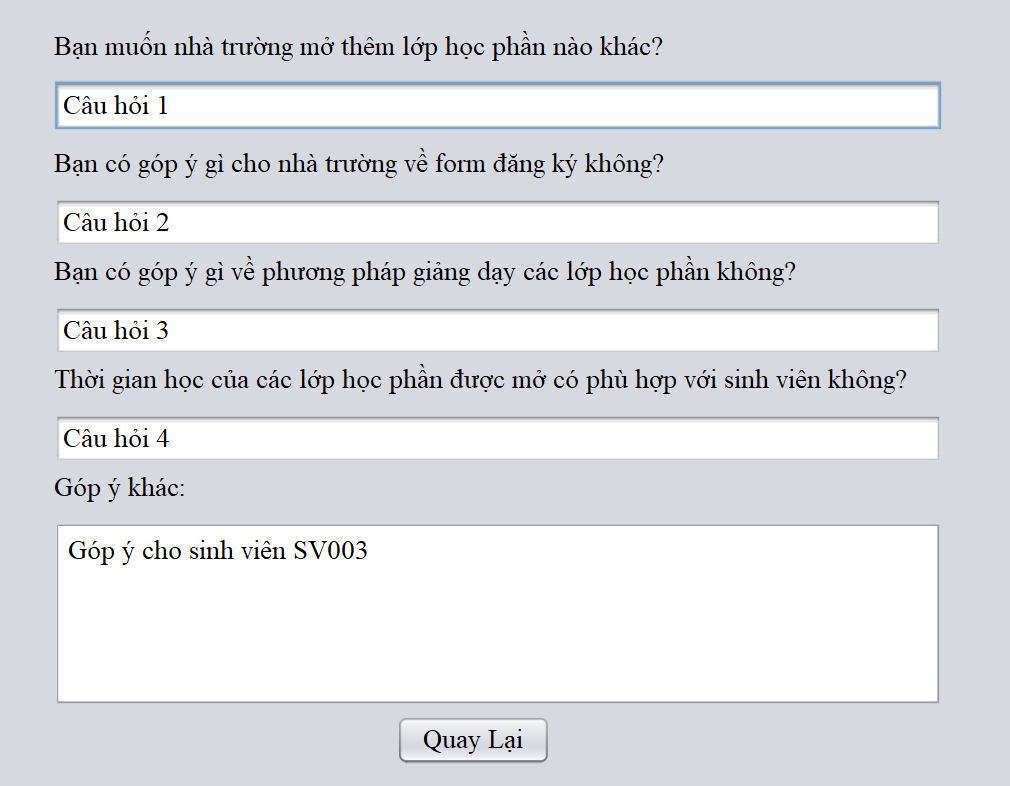
* **Thao tác file**
* Ghi file: Sau khi duyệt đơn, trạng thái của các đơn sẽ được cập nhật trong file txt lưu trữ danh sách các đơn.



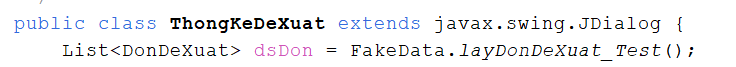
### Nguyễn Hoàng Hải - Chức năng thống kê đơn đề xuất

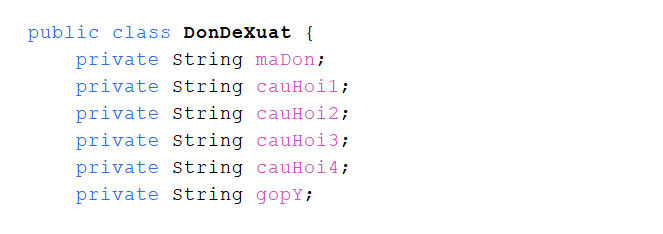
* Giao diện code chương trình





* Mã lệnh
  + **Hướng đối tượng**
* Áp dụng tính kế thừa, đóng gói.

****



* Các nội dung thực hiện
* Thống kê đơn đề xuất từ file hiển thị lên jTable.
* Nút ‘Xem chi tiết’ hiển thị chi tiết đơn được chọn.
* Sử dụng cửa sổ cha, cửa sổ con bằng cách cho cửa sổ con kế thừa javax.swing.JDialog
* Nút ‘Xóa’ sẽ xóa đơn được chọn trên bảng đồng thời cập nhật lại dữ liệu trong file.
* **Bẫy lỗi và gom rác**

A computer screen shot of a code

Description automatically generated

Ghi chú: Sử dụng cơ chế bẫy lỗi bằng try catch. Khi người dùng không bấm chọn dòng nào trên jTable thì chương trình sẽ thống báo lỗi lên màn hình.

* **Tập hợp**
* Loại tập hợp sử dụng: List. Việc sử dụng loại tập hợp này vì nó cung cấp các chức năng thêm, sửa, xóa các phần tử trong tập hợp. Nó cũng cho phép truy cập các phần tử thông qua chỉ số.



Phép toán : remove(), get()

* **Thao tác với file**
* Đọc dữ liệu từ file

A computer screen shot of text

Description automatically generated

* Lưu dữ liệu vào file

A computer screen shot of a computer code

Description automatically generated

### Lớp FakeData để lấy dữ liệu

Để thực hiện lấy dữ liệu từ file txt, Nhóm chúng em đã tiến hành xây dựng một lớp FakeData chung. Việc làm này giúp nhóm có thể thống nhất định dạng dữ liệu đầu ra cũng như dữ liệu đầu vào, tránh bị xung đột. Các phương thức được xây dựng trong lớp này được để static để khi sử dụng chỉ cần gọi phương thức qua tên lớp mà không phải khởi tạo đối tượng mới gây lãng phí bộ nhớ.

Các phương thức cụ thể trong lớp FakeData bao gồm:

* Lấy mã sinh viên đang đăng nhập
* Lấy danh sách khoa
* Lấy danh sách ngành
* Lấy danh sách học phần
* Lấy danh sách đơn đề xuất
* Lấy danh sách đơn cá nhân
* Lấy danh sách đơn tập thể
* Tạo danh sách đơn đăng ký
* Tạo danh sách đơn đăng ký chưa duyệt
* Lấy danh sách học phần theo ngành

Một số mã lệnh:

* Phương thức lấy danh sách khoa

A screenshot of a computer code

Description automatically generated

* Phương thức lấy danh sách đơn cá nhân

A computer code on a white background

Description automatically generated

# KẾT LUẬN VÀ BÀI HỌC KINH NGHIỆM

## Nội dung đã thực hiện

### Kiến thức và kỹ năng đã đạt được

Trong quá trình thực hiện bài tập lớn môn Java, chúng em đã học được rất nhiều kiến thức và kỹ năng mới liên quan đến lập trình Java và phát triển phần mềm:

* Kiến thức về lập trình java: Qua bài tập lớn, chúng em đã học được cách thiết kế và triển khai các lớp đối tượng, sử dụng đa hình, kế thừa và đóng gói trong Java.
* Kỹ năng về phát triển phần mềm: Bài tập lớn đã giúp chúng em học được các kỹ năng cần thiết để phát triển phần mềm. Chúng em đã học được cách thiết kế và triển khai các mô hình kiến trúc giao diện phần mềm. Bên cạnh đó là kỹ năng sử dụng các công cụ phát triển phần mềm như Apache NetBeans, Balsamip Mockup, Rational Rose, CASE studio …
* Kỹ năng phân tích bài toán: Bài tập lớn đã giúp chúng em học được cách đặt ra bài toán và phân tích các yêu cầu bài toán, vẽ sơ đồ usecase phân tích khái quát bài toán để phục vụ cho việc phát triển một phần mềm.
* Kỹ năng làm việc nhóm: Bài tập lớn đã đòi hỏi chúng em phải làm việc nhóm để hoàn thành các nhiệm vụ và giải quyết các vấn đề phát sinh. Chúng em đã học được cách phân chia công việc, trao đổi và làm việc với các thành viên khác trong nhóm để đạt được mục tiêu chung.
* Kỹ năng giải quyết vấn đề: Bài tập lớn đã giúp chúng em học được cách giải quyết các vấn đề phát sinh trong quá trình phát triển phần mềm. Chúng em đã học được cách phân tích các vấn đề, đưa ra các giải pháp và thực hiện chúng để đạt được kết quả tốt nhất.

Tổng quan lại, bài tập lớn môn Java đã giúp chúng em học được rất nhiều kiến thức và kỹ năng mới liên quan đến lập trình Java và phát triển phần mềm. Những kiến thức và kỹ năng này sẽ rất hữu ích cho chúng em trong việc phát triển các ứng dụng Java trong tương lai.

### Các chuẩn đầu ra đã đạt được

Các chuẩn đầu ra môn học mà chúng em đã đạt được:

* Chuẩn L1:
* Hiểu được cấu trúc cơ bản của ngôn ngữ lập trình Java và các khái niệm cơ bản như biến, kiểu dữ liệu, toán tử, điều kiện, vòng lặp và hàm.
* Cài đặt được các kỹ thuật lập trình căn bản của ngôn ngữ Java
* Hiểu được cách sử dụng các công cụ phát triển Java như Eclipse hoặc NetBeans để phát triển và chạy các chương trình Java.
* Chuẩn L2:
* Vận dụng được các kỹ thuật lập trình hướng đối tượng của Java vào giải quyết bài toán thực tế.
* Chuẩn L3:
* Chủ động tham gia cũng như có khả năng thành lập nhóm phù hợp với công việc
* Tổ chức hoạt động nhóm và có khả năng hợp tác kỹ thuật
* Khả năng làm việc nhóm với vài trò thành viên hoặc trưởng nhóm
* Chủ động tham gia cũng như có khả năng thành lập nhóm phù hợp với công việc
* Chuẩn L4:
* tổng hợp vào giải quyết bài toán thực tế.

### Các bài học kinh nghiệm

Sau khi hoàn thành bài tập lớn chúng em rút ra được một số kinh nghiệm như sau:

* Thiết kế kiến trúc hợp lý: Trước khi bắt đầu làm bài tập lớn, hãy đầu tư thời gian để thiết kế kiến trúc của phần mềm sao cho hợp lý nhất. Điều đó sẽ làm giảm đáng kể thời gian và công sức để phát triển phần mềm cũng như hạn chế được tối đa lỗi có thể xảy ra trong quá trình phát triển phần mềm.
* Sử dụng các công cụ phát triển phần mềm: Sử dụng công cụ phát triển phần mềm hợp lý sẽ khiến việc phát triển phần mềm trở nên đơn giản hơn, giúp tiết kiệm thời gian và hoàn thiện sản phẩm tốt hơn.
* Làm việc nhóm hiệu quả: Nếu bạn làm bài tập lớn Java trong nhóm, hãy đảm bảo rằng các thành viên trong nhóm đều hiểu rõ các yêu cầu của dự án và đóng góp vào quá trình phát triển. Hãy sử dụng các công cụ như Google Sheet, Google Drive để quản lý công việc và đảm bảo rằng mọi người đều đang làm việc trên cùng một trang.

## Hướng phát triển

Việc tạo một chương trình hỗ trợ đăng ký học phần cho sinh viên HaUI là hoàn toàn khả thi và vừa sức dựa trên các kiến thức chúng em đã được học tuy nhiên việc quản lý các lớp học phần trong khoa, đặc biệt là thống kê các đơn đăng ký mà sinh viên gửi đến đối với mỗi lớp học phần gồm nhiều yếu tố khác nhau như cần phải quản lý các học phần trong ngành cũng như các ngành trong khoa và các khoa trong nhà trường, từ đó hỗ trợ cho việc thống kê các đơn đăng ký và duyệt đơn theo từng học phần tương ứng. Phần mềm hỗ trợ sinh viên đăng ký học phần phải đáp ứng được tất cả các yếu tố này, điều này đòi hỏi phải có kiến thức về quản lý các lớp học phần trong từng ngành, từng khoa; kiến thức thống kê các đơn đăng ký theo từng lớp học phần đó và kỹ năng lập trình phần mềm.

Trong quá trình phát triển phần mềm chúng em đã gặp một số thuận lợi và khó khăn như sau:

* Thuận lợi:

Là một sinh viên HaUI, chúng em dựa trên những mong muốn của sinh viên để thiết kế các nghiệp vụ chức năng của chương trình một các dễ ràng, tương đối ổn định.

* Khó khăn:
* Việc quản lý các khoa, ngành và các lớp học phần của toàn trường đòi hỏi phải có đầy đủ thông tin về các chuyên ngành trong các khoa và các lớp học phần thuộc cơ chế đào tạo của từng chuyên ngành đó.
* Hiện tại, điểm hạn chế của chúng em là mới thống kê được các lớp học phần và các ngành thuộc khoa Công nghệ thông tin mà mình đang theo học.
* Lượng kiến thức chuyên môn còn hạn hẹp nên chưa thể tạo ra một chương trình hoàn hảo.

# TÀI LIỆU THAM KHẢO

1. Balsamiq, Bộ công cụ phác thảo giao diện người dùng, [muakey.vn/balsamiq-thiet-ke-website](https://muakey.vn/balsamiq-thiet-ke-website).
2. Giáo trình Lập trình HĐT với Java, Nguyễn Bá Nghiễn, Ngô Văn Bình, Vương Quốc Dũng, Đỗ Sinh Trường; NXB Thống kê, 2020.
3. The Java Programming Language; Author: K. Arnold, J. Gosling; Published: Addison-Wesley, 1996, ISBN 0-201-63455-4