TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

\_\_\_\_\_\_\_\_\_\_\_\_\*\*\*\_\_\_\_\_\_\_\_\_\_\_\_

**BÀI TẬP LỚN**

MÔN: LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Tên đề tài: Xây dựng hệ thống quản lý học phí sinh viên

Giảng viên hướng dẫn: TS. Đỗ Thị Ngọc Diệp

Nhóm sinh viên: 5 (làm bài tập lớn số 8)

|  |  |
| --- | --- |
| **Họ tên sinh viên** | **Mã số sinh viên** |
| Tô Đức Hiệp | 20194278 |
| Nguyễn Mạnh Duy | 20194262 |
| Nguyễn Phương Nam | 20194336 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Họ và tên | Mã sinh viên | Email | Công việc | Mức độ hoàn thành |
| 01 | Tô Đức Hiệp | 20194278 | hiep.td194278@sis.hust.edu.vn | -Restore database  -Xây dựng các lớp cơ bản Student, lớp kết thừa: StudentTinChi, StudentCtrMau  -Xây dựng lớp controller và giao diện đăng nhập, đăng kí tài khoản.  -Tìm hiểu các thư viện để làm việc với project, tổng hợp lại, viết báo cáo hoàn chỉnh | 100% |
| 02 | Nguyễn Mạnh Duy | 20194262 | duy.nm194262@sis.hust.edu.vn | -Restore databade  -Tạo sơ đồ lớp cơ bản.  -Xây dựng các lớp controller và giao diện phần sinh viên chương trình mẫu.  -Test các chức năng, hoạt động của project, phân tích ưu điểm nhược điểm.  -Làm báo cáo thiết kế chương trình, biểu đồ lớp UML. | 100% |
| 03 | Nguyễn Phương Nam | 20194336 | nam.np194336@sis.hust.edu.vn | -Tạo database chung, backup, tạo lớp kết nối project.  -Xây dựng các lớp controller và giao diện phần sinh viên tính chỉ.  -Tổng hợp,tạo project đồng nhất sửa lại code,thêm các chức năng thông báo, kiểm tra.  -Làm báo cáo hướng dẫn cài thư viện với Intellij IDE, mô tả hoạt động của các chương trình. | 100% |

***Ghi chú***: Trong quá trình làm bài, nhóm em thường xuyên làm việc chung, lên những ý tưởng, góp ý chung.Do quá trình học tập online, chúng em hoạt động thảo luận bằng nhóm trên Teams nên việc thảo luận cũng gặp một vài khó khăn. Vì vậy việc phân chia công việc cũng chỉ dựa trên phương diện khách quan.

**MỤC LỤC**

[1. Giới thiệu bài toán 3](#_heading=h.gjdgxs)

[2. Các thư viện sử dụng để thực thi bài toán 4](#_heading=h.30j0zll)

[2.1. Runtime Environment System Library 4](#_heading=h.1fob9te)

[2.2. JavaFX Software Development Kit (JavaFX SDK) 4](#_heading=h.3znysh7)

[2.3. Scene Builder 5](#_heading=h.2et92p0)

[2.4. Microsoft  JDBC Driver (JDBC) 5](#_heading=h.tyjcwt)

[3. Hướng dẫn cài đặt các thư viện 5](#_heading=h.3dy6vkm)

[3.1. Giới thiệu chung 5](#_heading=h.1t3h5sf)

[3.2. Hướng dẫn cách kết nối thư viện 6](#_heading=h.4d34og8)

[4. Thiết kế chương trình, biểu đồ lớp 12](#_heading=h.2s8eyo1)

[4.1. Thiết kế chương trình 12](#_heading=h.17dp8vu)

[4.2. Biểu đồ lớp hoàn chỉnh 18](#_heading=h.3rdcrjn)

[5. Mô tả hoạt động của các chương trình, lớp sử dụng 18](#_heading=h.26in1rg)

[5.1. Hoạt động của chương trình 18](#_heading=h.lnxbz9)

[5.2. Các lớp mẫu 27](#_heading=h.35nkun2)

[5.3. Các lớp điều khiển 30](#_heading=h.1ksv4uv)

[5.4. Lớp thông báo 41](#_heading=h.44sinio)

[5.5. Lớp kiểm tra 43](#_heading=h.2jxsxqh)

[5.6. Lớp kết nối 43](#_heading=h.z337ya)

[5.7. Các lớp giao diện fxml 44](#_heading=h.3j2qqm3)

[6. Tự đánh giá kết quả, các điểm hạn chế 45](#_heading=h.1y810tw)

[6.1. Ưu điểm 45](#_heading=h.4i7ojhp)

[6.2. Nhược điểm 46](#_heading=h.2xcytpi)

NỘI DUNG BÁO CÁO

# Giới thiệu bài toán

Bài toán quản lý học phí sinh viên chia danh sách sinh viên ra làm hai nhóm: nhóm học theo tín chỉ và nhóm học theo chương trình mẫu. Tùy theo hình thức đào tạo, cách thức tính học phí mỗi hình thức là khác nhau:

* Hình thức học theo tín chỉ: sinh viên đóng học phí hằng năm tính theo số tín chỉ một năm \* 250.000đ.
* Hình thức học theo chương trình mẫu: sinh viên đóng học phí hằng năm bằng số học phần của năm \* 2.5 \* 250.000đ + 1.000.000đ phí quản lý.

Sinh viên xây dựng hệ thống quản lý sinh viên gồm các chức

năng như thêm, bớt, sửa, xóa thông tin các sinh viên; tìm kiếm sinh viên; tính học phí theo hai hình thức; thống kê sinh viên.

# Các thư viện sử dụng để thực thi bài toán

## Runtime Environment System Library

* Thư viện này được tự động thêm vào ngay sau khi một project được tạo ra. JRE System Library áp dụng giao diện lập trình ứng dụng Java (Java API) trong một môi trường phát triển tích hợp (IDE). JRE lấy đoạn code Java, kết hợp nó với các thư viện cần thiết, khởi động máy ảo Java (JVM) để thực hiện nó.
* JRE là một tập các công cụ phần mềm cho việc phát triển các ứng dụng Java, nó kết hợp JVM, các lớp nền tảng và các thư viện hỗ trợ. JRE là một phần của Java Development Kit (JDK), nhưng có thể được tải riêng biệt. JRE ban đầu được phát triển bởi Sun Microsystems Inc., một công ty con của Tập đoàn Oracle. Còn được gọi là thời gian chạy Java. JRE được sử dụng với Eclipse IDE, còn JDK được tải về sử dụng với Intellij IDE. Khi tiến hành cài đặt Intellij, ta nên cài đặt JDK trước khi cài đặt Intellij để tránh lỗi không mong muốn.

## JavaFX Software Development Kit (JavaFX SDK)

* JavaFx được tạo ra nhằm thay thế Swing làm thư viện giao diện đồ họa người dùng tiêu chuẩn của Java bản tiêu chuẩn (Java SE), được dùng cho việc tạo và chuyển ứng dụng desktop, có hỗ trợ cho máy tính desktop và các trình duyệt web trên các hệ điều hành Windows, Linux and macOS, cũng như iOS và Android trên điện thoại.
* Những ưu điểm nổi bật của ngôn ngữ lập trình JavaFX:
* JavaFX là một thư viện Java bao gồm các lớp và các giao diện được viết bằng mã Java nguyên gốc.
* FXML là một ngôn ngữ khai báo dựa trên XML để xây dựng một giao diện người dùng trong ứng dụng JavaFX. Lập trình viên có thể sử dụng JavaFX Scene Builder để thiết kế giao diện đồ họa (GUI).
* JavaFX có thể được tùy biến giao diện thêm sinh động bằng cách sử dụng CSS.
* JavaFX hỗ trợ đồ họa 2D và 3D cũng như hỗ trợ âm thanh và video.
* JavaFX còn có WebView dựa trên trình duyệt WebKit, vì vậy bạn có thể nhúng các trang web hoặc các ứng dụng web bên trong JavaFX.

## Scene Builder

* Công cụ thiết kế giao diện đồ họa người dùng có tính tương tác, được tạo ra bởi Oracle. Công ty Gluon thông báo sẽ hỗ trợ cho việc phát triển công cụ này và ra mắt phiên bản đầu tiên của software3 vào tháng 3 năm 2015. Phần mềm được sử dụng bằng kỹ thuật kéo/thả, các giao diện được lưu ở các file dạng FXML, được đọc và tải vào bộ nhớ nhờ chương trình, sau đó được hiện trên màn hình.

## Microsoft  JDBC Driver (JDBC)

* + 1. Khái niệm
* JDBC (Java Database Connectivity) là một chuẩn API (Application Program Interface) cho phép kết nối các chương trình viết bởi Java với các hệ quản trị cơ sở dữ liệu (MySQL, MS SQL, Postgre SQL, Oracle, DB2…).
* JDBC chỉ là một tập các interface, các định nghĩa, thông báo lỗi, đặc tả chứ không phải là thư viện. Với mỗi hệ quản trị cơ sở dữ liệu ta sẽ có một cài đặt JDBC riêng cho nó, ví dụ JDBC cho MySQL, JDBC cho MS SQL…
  + 1. Các thành phần của JDBC
* DriverManager: Dùng để quản lý danh sách các Driver (database drivers).
* Driver: Dùng để liên kết các kết nối tới cơ sở dữ liệu, điều khiển các liên kết.
* Connection: Biểu thị kết nối tới cơ sở dữ liệu, dùng để tạo ra Statement, PreparedStatement và CallableStatement.
* Statement, PreparedStatement, CallableStatement: Chứa lệnh SQL gửi tới cơ sở dữ liệu để thực thi.
* ResultSet – biểu diễn một tập kết quả trong cơ sở dữ liệu tạo ra bởi việc sử dụng một câu lệnh SQL là SELECT.
* SQLException – một lớp xử lý lỗi ngoại lệ chứa các lỗi truy cập cơ sở dữ liệu.

# Hướng dẫn cài đặt các thư viện

## Giới thiệu chung

* Ở phần này sẽ hướng dẫn các bạn cách cài đặt và import các thư viện vào Intellij IDEA  để thiết kế 1 project giao diện đồ họa bằng javafx. Dưới đây là các đường link tải thư viện đối với hệ điều hành Windows:
* Tải JDK trên Windows 10:

<https://www.oracle.com/java/technologies/javase-downloads.html>

* Tải SDK javafx trên Windows 10:

<https://gluonhq.com/products/javafx/>

* Tải Java Scene Buider cho Windhow 10:

<https://gluonhq.com/products/scene-builder/>

* Tải JDBC kết nối SQL server và Intellij cho Windows 10:

<https://docs.microsoft.com/vi-vn/sql/connect/jdbc/download-microsoft-jdbc-driver-for-sql-server?view=sql-server-ver15>

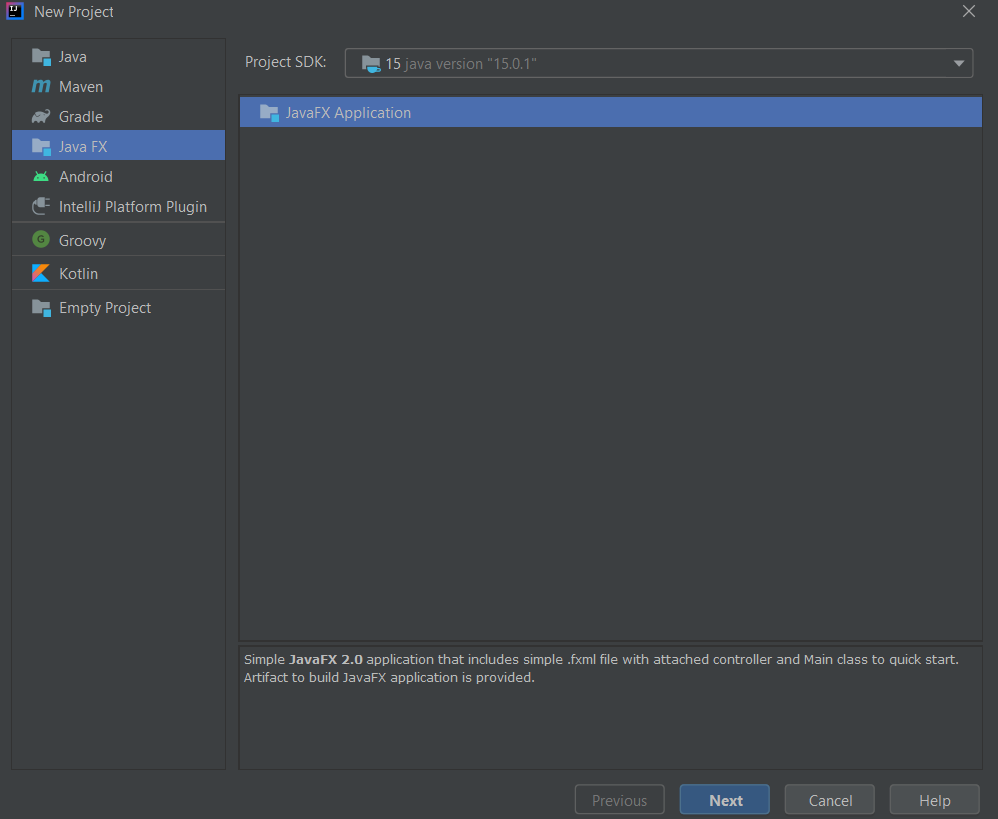
* Tham khảo cách liên kết thư viện tại đây:

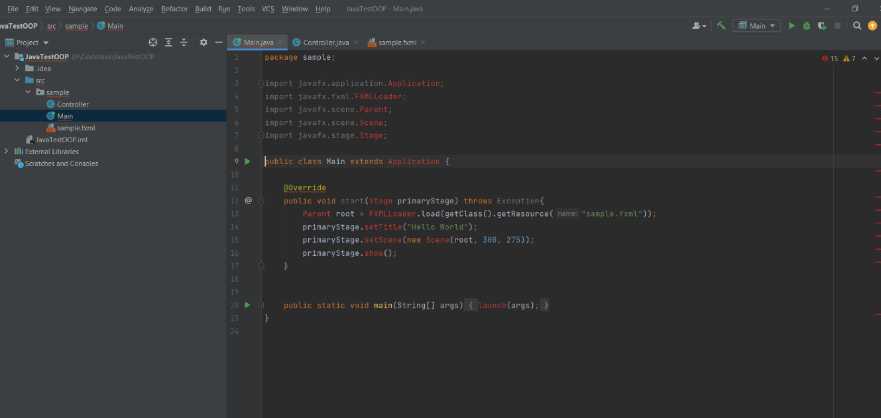
<https://www.youtube.com/watch?v=dlAe0vSPL8Q>

<https://www.youtube.com/watch?v=RF-_vchtV58>

## Hướng dẫn cách kết nối thư viện

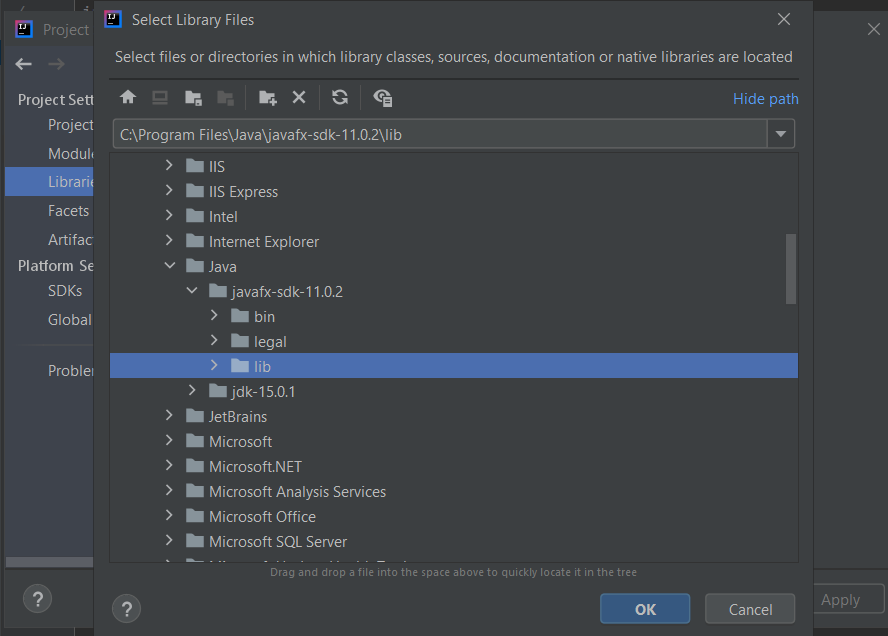
* Bước 1: Tạo 1 new Project Javafx dùng JDK phiên bản mới nhất đã tải về máy của bạn:



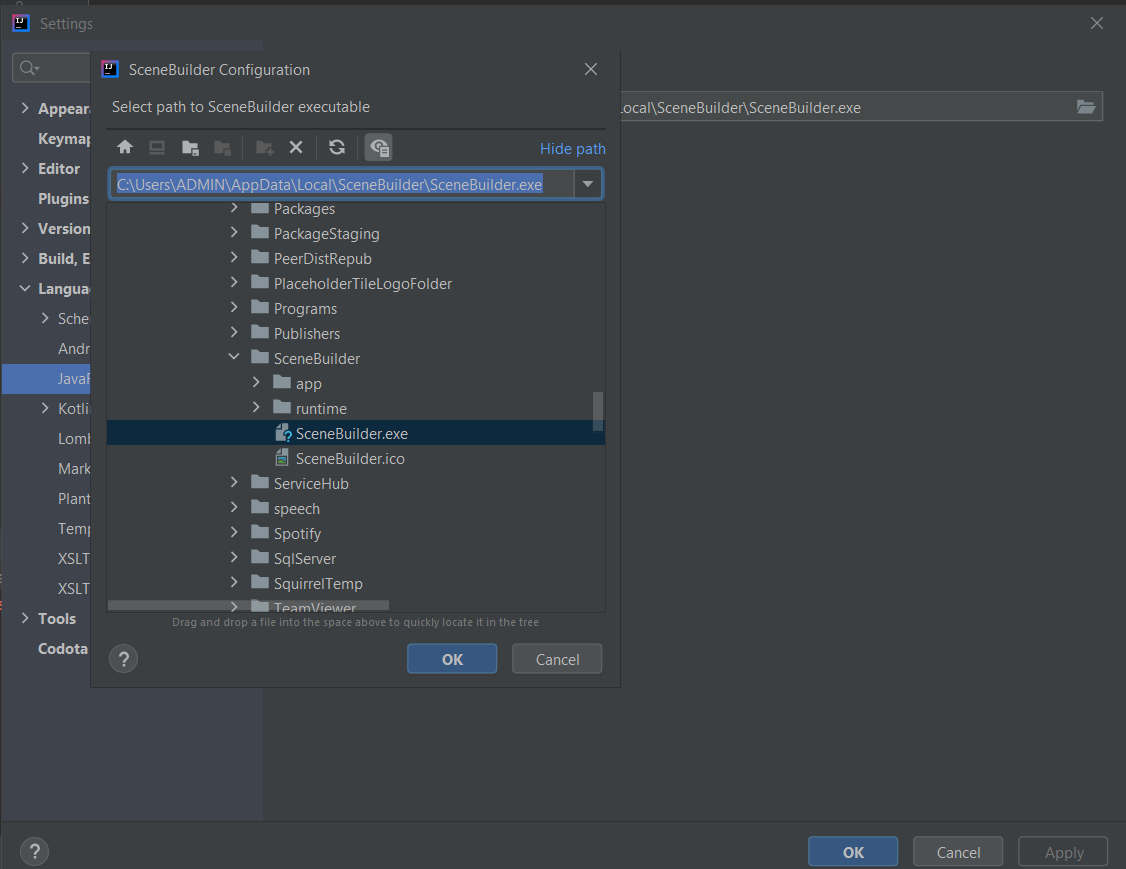


**1 package sample mặc định ban đầu được tạo ra và bị lỗi (màu đỏ) do chưa có thư viện Javafx:**

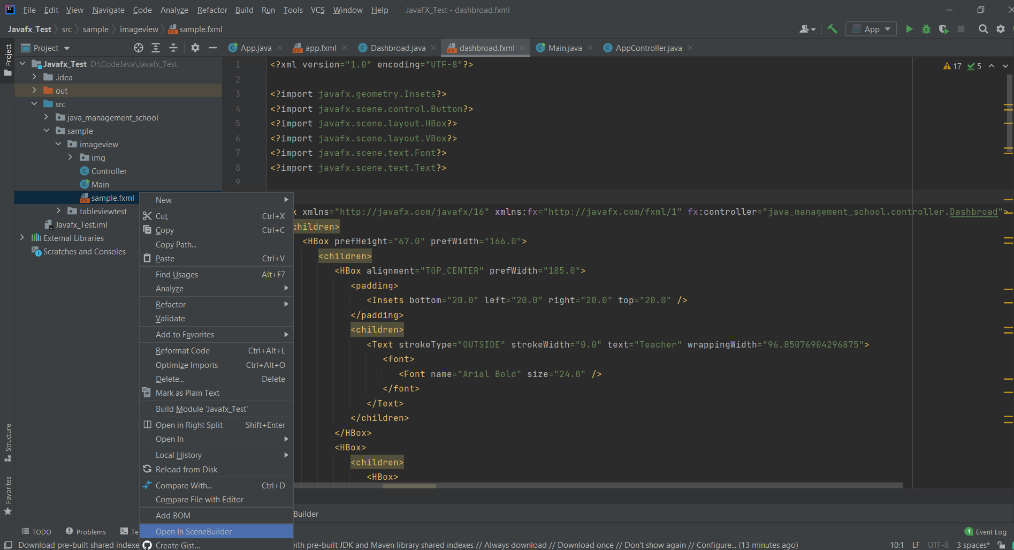
* Bước 2: Vào File -> Project Structure -> Library -> Java,sau đó bạn tìm đường dẫn thư viện lib của SDK Javafx đã tải về:

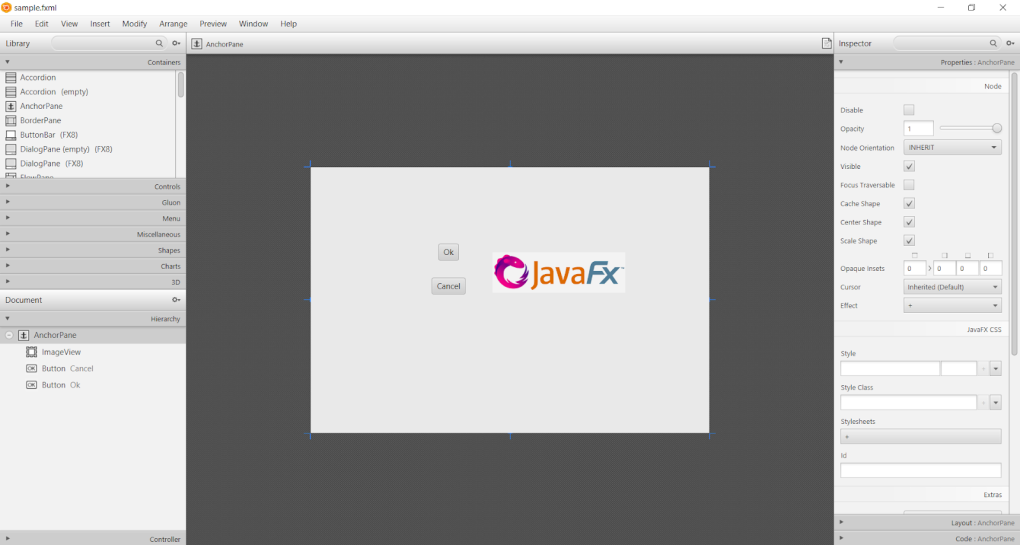


* Bước 3: Vào File -> Setting -> Languages & Frameworks -> Javafx. Tại mục Path to SceneBuider, bạn tìm đường dẫn đến thư viện SceneBuider.exe đã tải về:



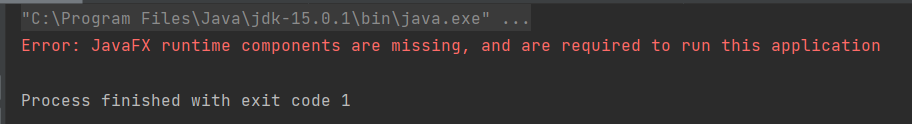
Sau đó tại sample.fxml -> click chuột phải -> Open SceneBuider:





SceneBuider được bật để thiết kế kéo thả dễ dàng

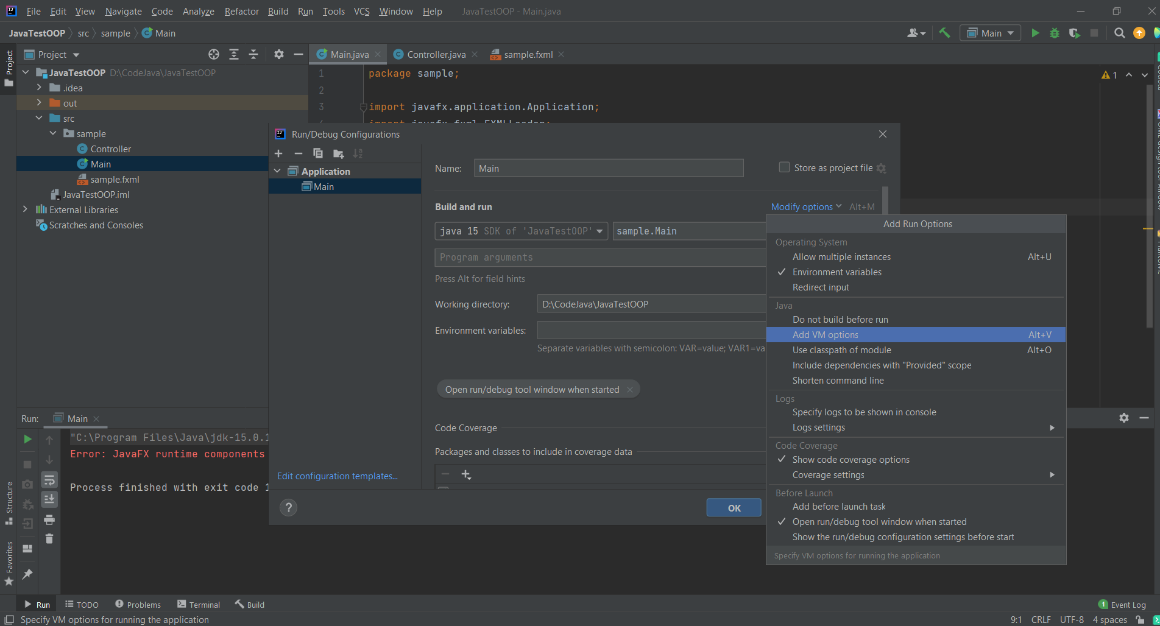
* Bước 4: Khi Run chương trình sẽ xuất hiện lỗi:

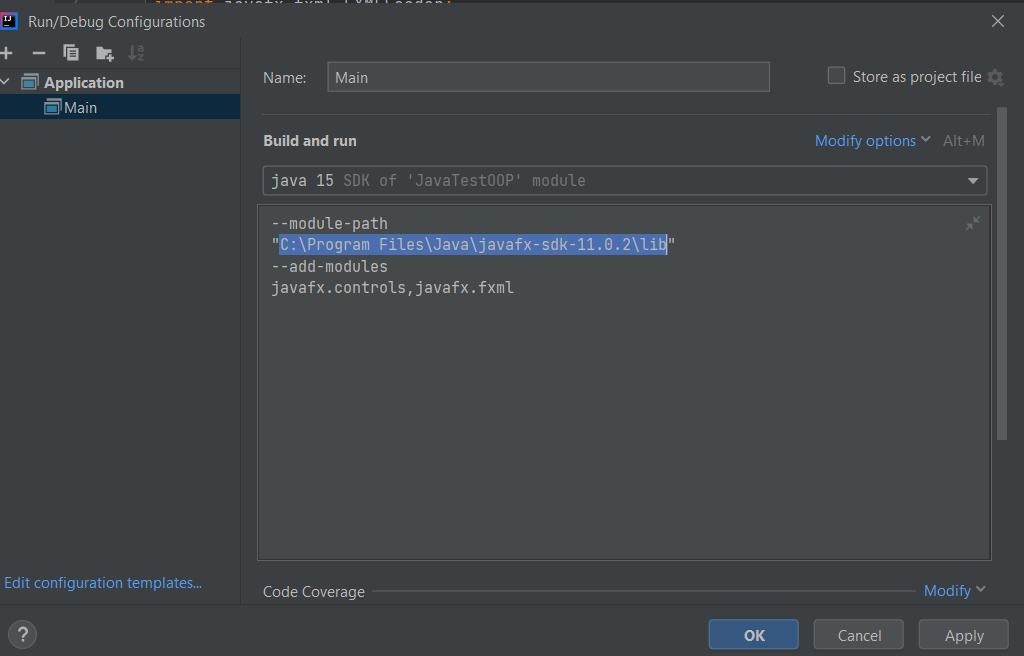


   Các bạn tiến hành: Run -> Edit Configuration -> Modify Options -> Add VM Options.

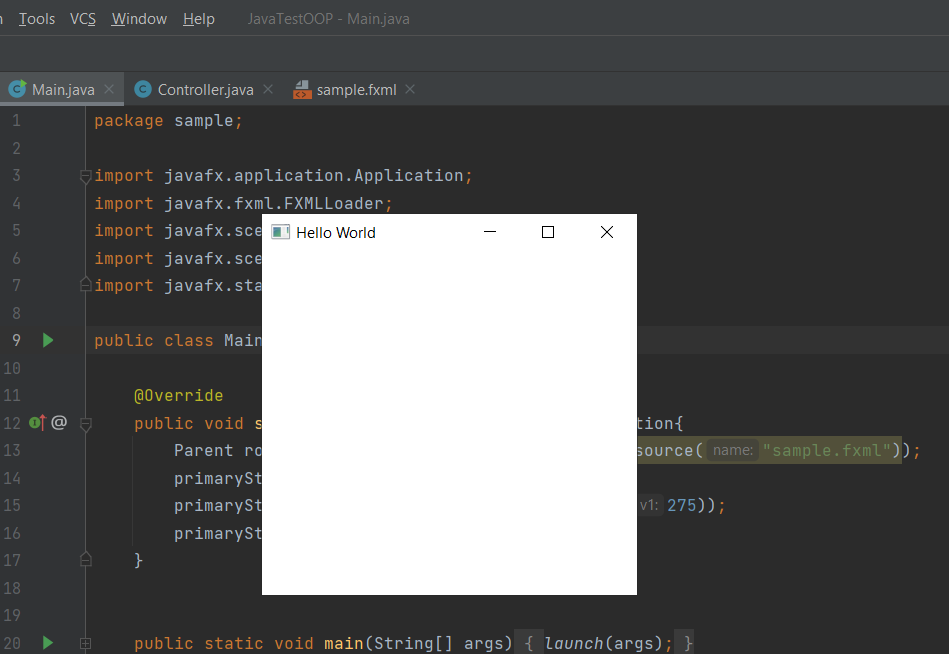
    Tại đây bạn nhập lệnh:  “ --module-path "C:\Program Files\Java\javafx-sdk-11.0.2\lib --add-modules javafx.controls,javafx.fxml“

    Thay “C:\Program Files\Java\javafx-sdk-11.0.2\lib” bằng địa chỉ thư viện sdk của bạn.



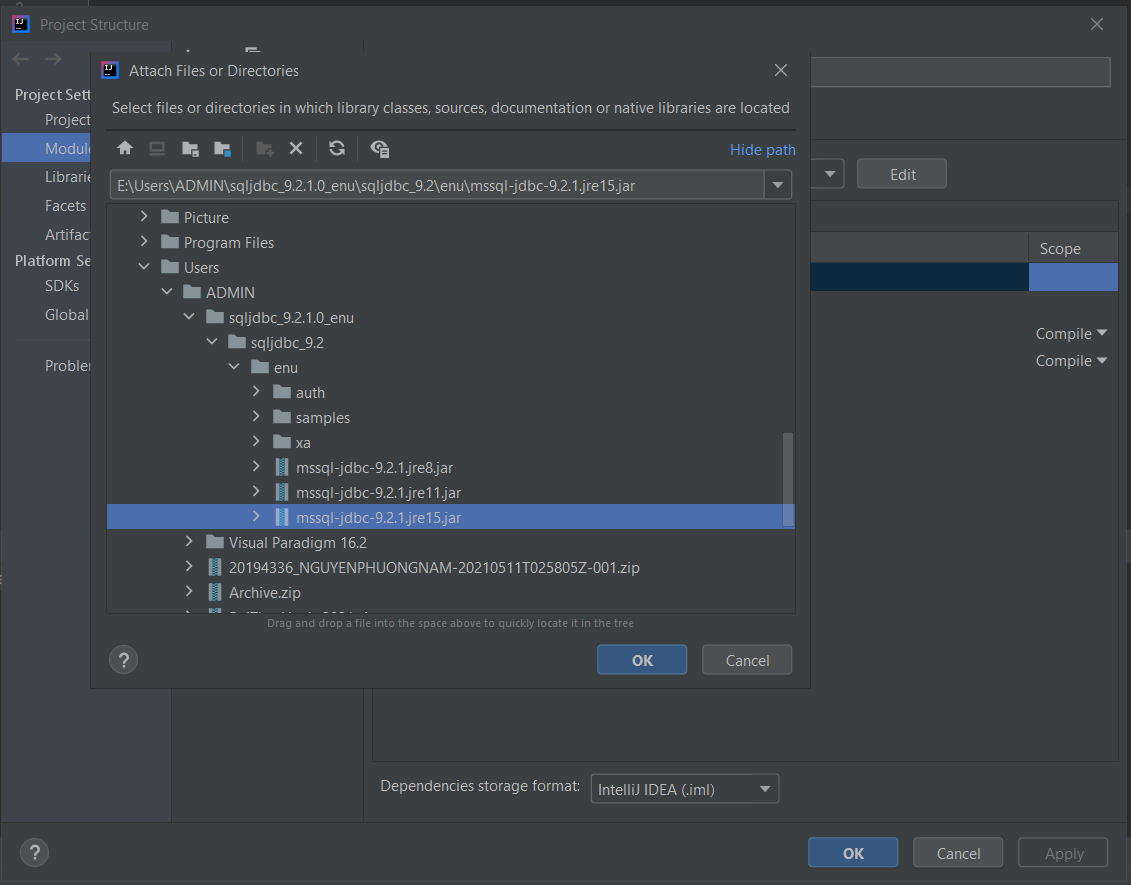


Sau khi chạy chương trình, 1 Stage mặc định được tạo ra:



* Bước 5: Liên kết SQL server với Intellij IDE bằng thư viện JDBC

File -> Project Struture -> Modules -> Dấu + bên dưới Module SDK. Sau đó tìm link dẫn đến thư viện jdbc đã tải về



Sử dụng phiên bản jdbc phù hợp với phiên bản IDE của bạn

 Liên kết Database trong SQL server bằng code:

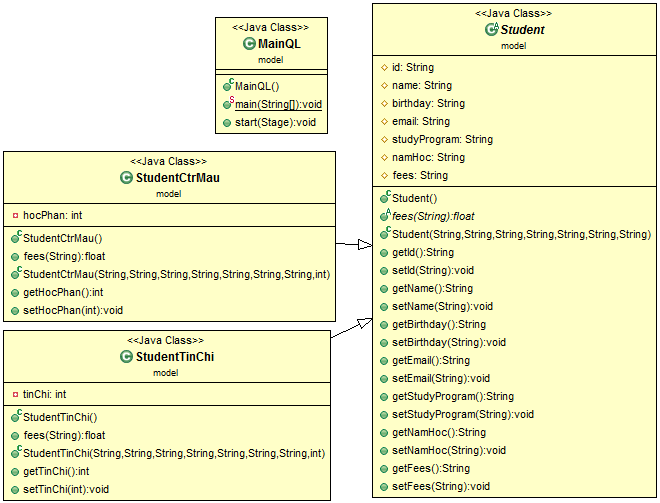


Như vậy là bạn đã hoàn thành xong việc liên kết tất cả các thư viên để tạo dựng 1 chương trình quản lí giao diện đồ họa.

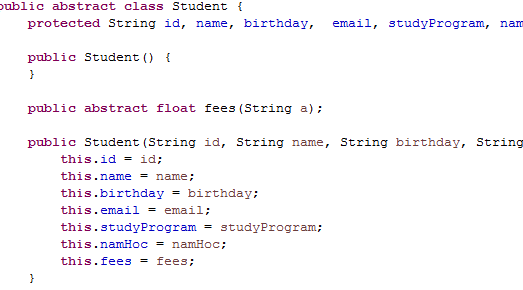
# Thiết kế chương trình, biểu đồ lớp

## Thiết kế chương trình

* Đầu tiên ta thiết kế các lớp, có thể chia ra làm các loại:
* Các lớp mẫu (Model):

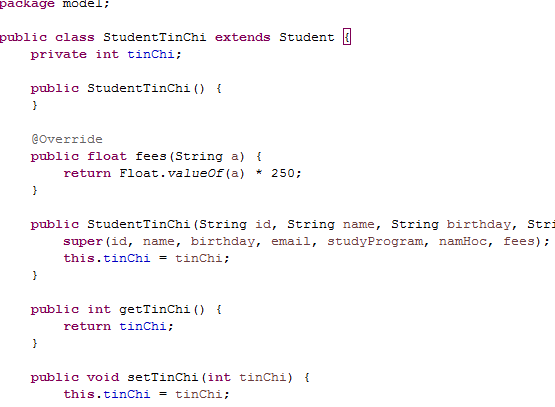


* MainQL: dùng để chạy chương trình
* Student: lớp trừu tượng phương thức trừu tượng là fees() (học phí), thuộc tính id, name, birthday, email, namHoc, fees và các phương thức get(), set().

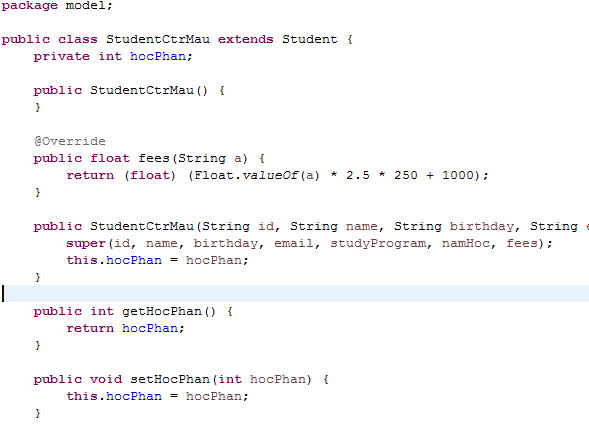


Đa hình: các lớp con thừa kế lớp cha Student có các phương thức ghi đè phương thức trừu tượng fees()

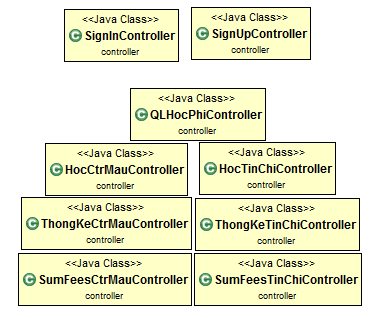
* StudentTinChi:



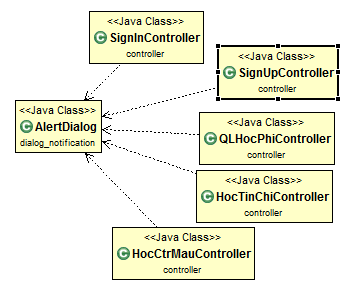
* StudentCtrMau:



* Các lớp điều khiển (controller): dùng để điều khiển các giao diện: đăng nhập, đăng ký, chọn hình thức tra cứu, tra cứu, thêm, cập nhật, xóa sinh viên, tính tổng học phí, thống kê sinh viên theo năm học theo từng hình thức học

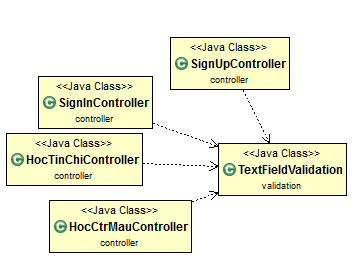


* Lớp thông báo (Alert): AlertDialog đưa ra thông báo khi một thao tác được thực hiện thành công, hoặc khi nhập sai (thêm sinh viên có ID trùng với sinh viên khác, để Text Field trống,…).



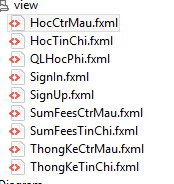
* Lớp kiểm tra (validation): kiểm tra xem Text Field có

trống hay không, có thì sẽ đưa ra thông báo không hợp lệ.

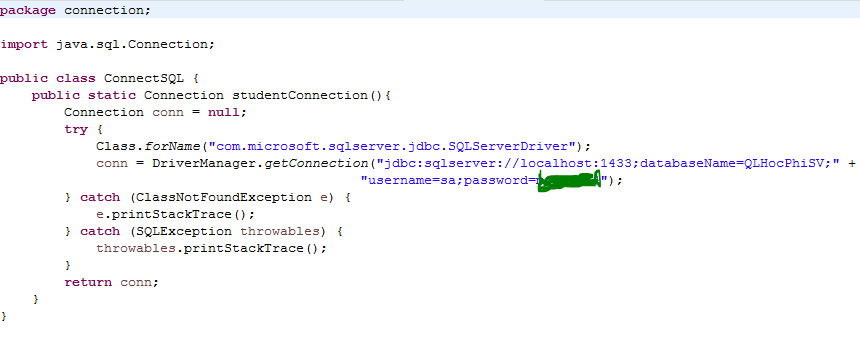


**Được gọi tới bởi các lớp điều khiển, phát hiện các Text Field bị bỏ trống khi đăng nhập, đăng ký hoặc tra cứu, thêm sinh viên**

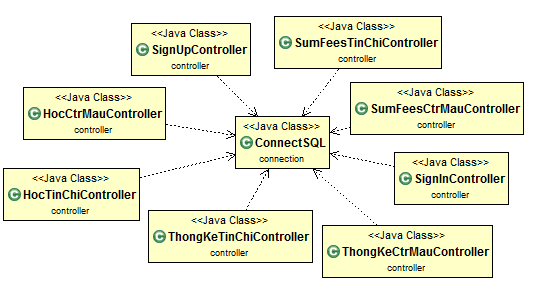
* Các lớp giao diện fxml: mở với Scene Buider để tạo giao diện bằng cách kéo/thả dễ dàng, có chín lớp điều khiển nên có chín lớp giao diện fxml.



* Hệ thống quản lý học phí sinh viên phải kết nối đến một hệ quản trị cơ sở dữ liệu để có thể tra cứu quản lý thông tin sinh viên. Do đó, việc thiết kế cơ sở dữ liệu và một lớp kết nối với cơ sở dữ liệu là rất quan trọng
* Lớp kết nối: ConnectSQL là lớp dùng để kết nối với SQL Server

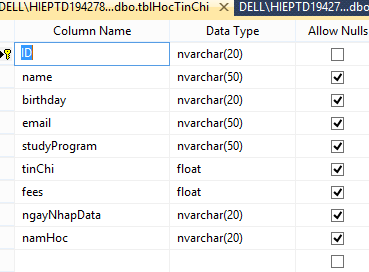
****

**Code của lớp ConnectSQL**

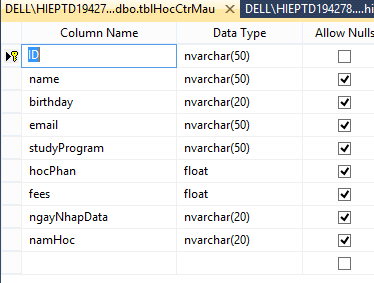
****

**Các lớp điều khiển dùng ConnectSQL để kết nối với hệ quản trị cơ sở dữ liệu**

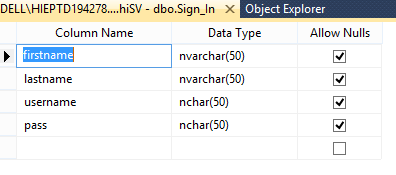
* Tạo các bảng trong cơ sở dữ liệu: ta tạo ba bảng



**HocTinChiController kết nối đến bảng tblHocTinChi để tra cứu thông tin về các sinh viên học tín chỉ**

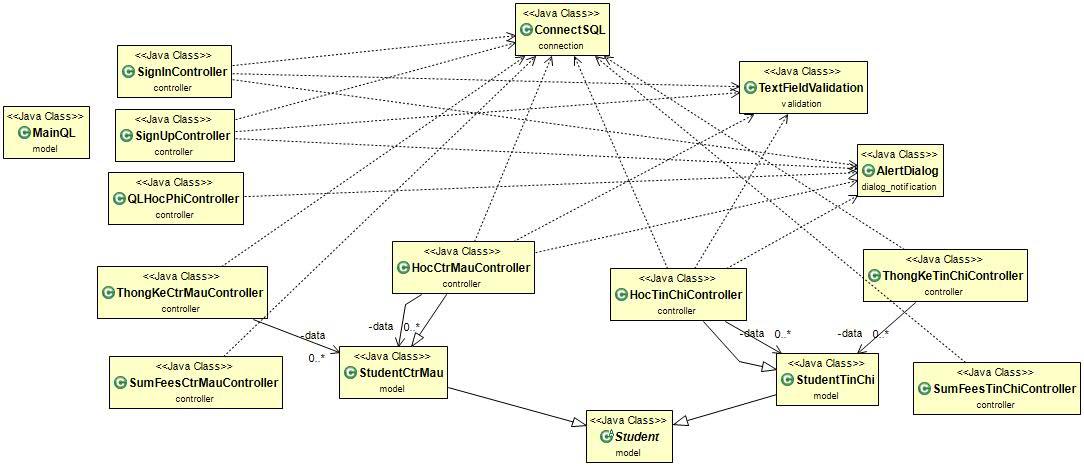
****

**HocCtrMauController kết nối đến bảng tblHocCtrMau để tra cứu thông tin về các sinh viên học chương trình mẫu**



**SignUpController kết nối đến bảng Sign\_In để xem tài khoản mới đăng ký có hợp lệ hay không (không bị trùng username)**

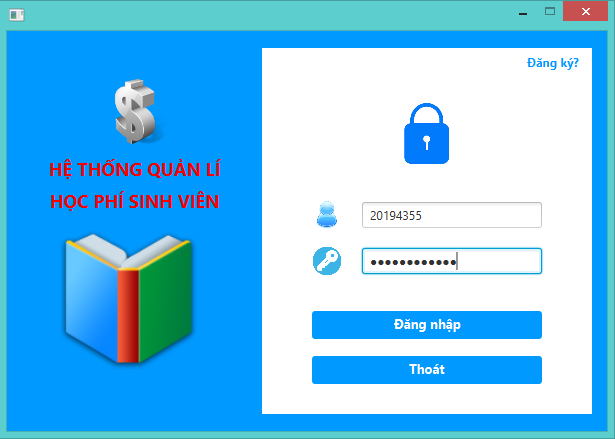
## Biểu đồ lớp hoàn chỉnh



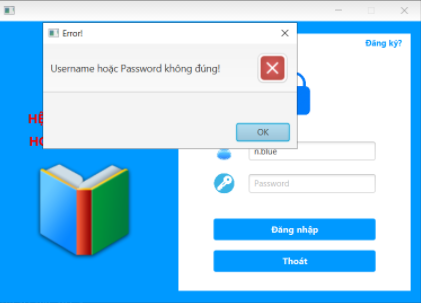
# Mô tả hoạt động của các chương trình, lớp sử dụng

## Hoạt động của chương trình

* Sau khi chạy lớp chính MainQL, giao diện đăng nhập sẽ được hiển thị trên màn hình, người dùng cần nhập đúng tài khoản và mật khẩu đã đăng ký để kết nối với hệ quản trị cơ sở dữ liệu và vào giao diện quản lý sinh viên:



**Nhập đúng tài khoản và mật khẩu để đăng nhập**

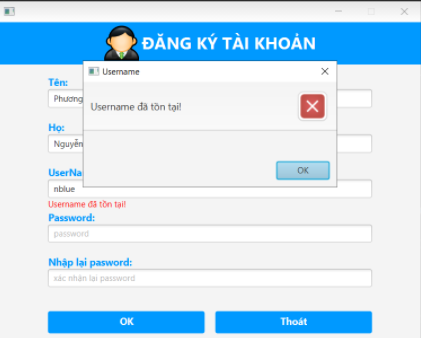


**Nhập sai thông tin hoặc để trống Text Field, hệ thống đưa ra thông báo lỗi**

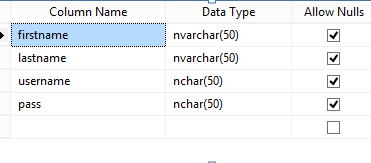
* Nếu chưa có tài khoản đăng nhập, người dùng cần phải đăng ký, bằng cách click chuột trái vào chữ “Đăng ký?” phía trên góc phải màn hình để đăng ký tài khoản

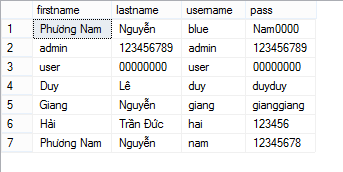


**Đăng ký thành công (username không bị trùng với một username đã được đăng ký và không để Text Field nào trống) sẽ có thông báo hiện lên**

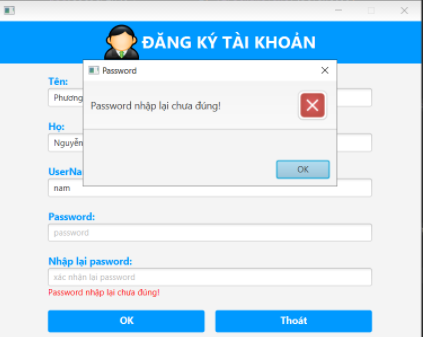


**Hệ thống sẽ kết nối với Microsoft SQL Server, truy vấn bảng Sign\_In để tìm xem username đã tồn tại hay chưa, nếu rồi sẽ hiển thị thông báo như trên**



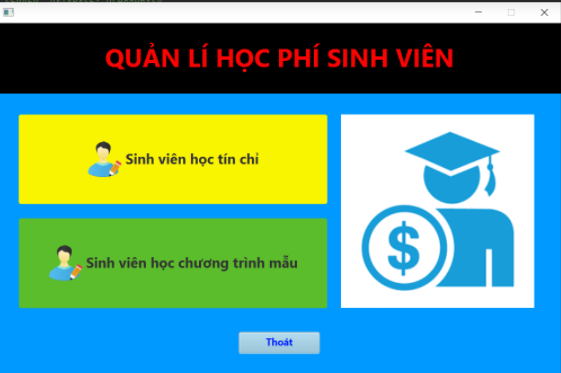


**Thiết kế và dữ liệu đã lưu trữ của bảng Sign\_In trong cơ sở dữ liệu**



**Password nhập lại không khớp cũng sẽ có thông báo lỗi**

* Sau khi đăng nhập thành công (nhập đúng tài khoản, mật khẩu đã đăng ký), click chuột vào nút “Đăng Nhập”, giao diện quản lý sinh viên sẽ xuất hiện:

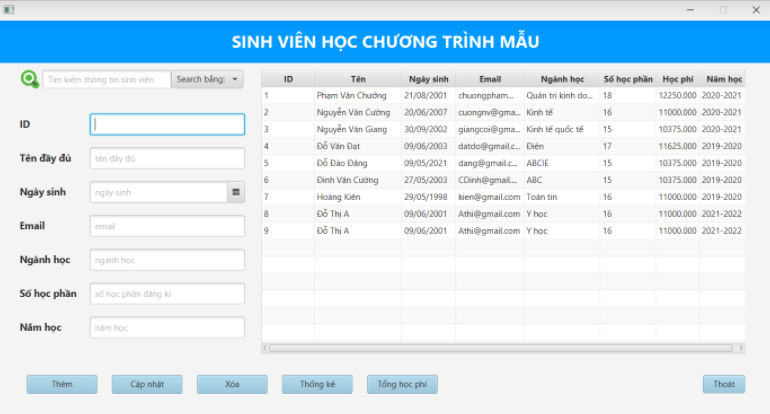


**Chọn tra cứu thông tin theo sinh viện học theo chương trình mẫu hoặc tín chỉ**

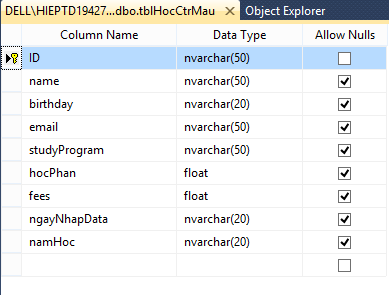


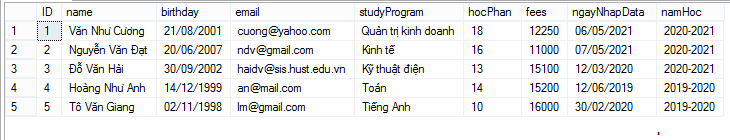
**Click chuột vào nút Thoát để ngắt kết nối SQL Server và thoát ra ngoài**

* Ta chọn một trong hai hình thức học để tra cứu, sửa đổi, thêm thông tin sinh viên, dưới ta sẽ chọn theo chương trình mẫu, với tín chỉ ta làm tương tự:



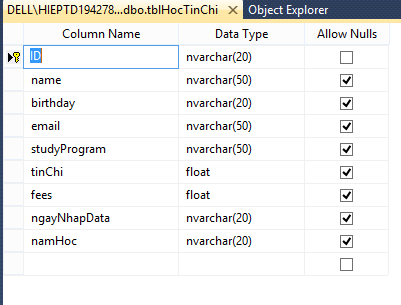
**Tương tác với giao diện quản lí sinh viện học chương trình mẫu, hệ thống đã được kết nối với bảng tblHocCtrMau của cơ sở dữ liệu**

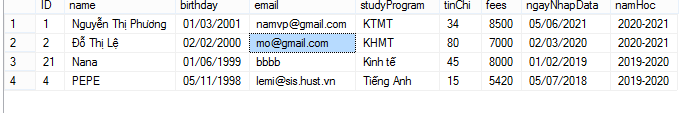




**Thiết kế và dữ liệu đã lưu trữ của bảng tblHocCtrMau trong cơ sở dữ liệu, chứa thông tin của các sinh viên học chương trình mẫu**

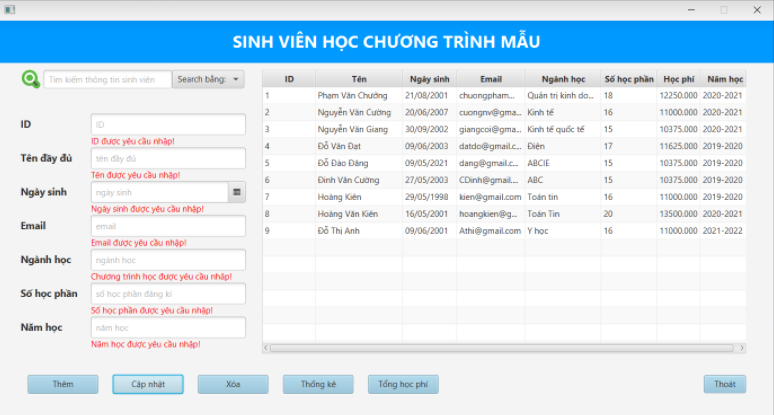
* Đối với các sinh viên học tín chỉ: giao diện tương tác tương tự chương trình mẫu, bảng dữ liệu có chút khác biệt



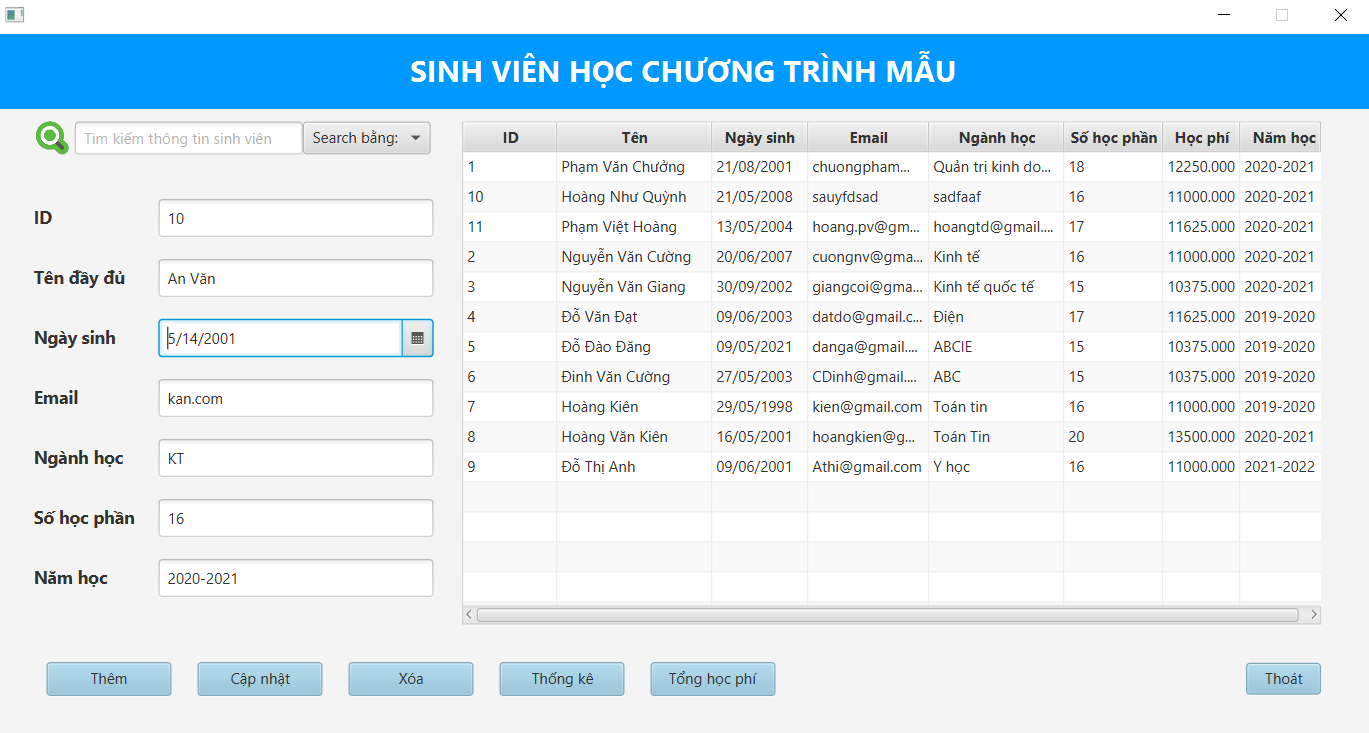


**Bảng tblHocTinChi, chứa thông tin các sinh viên học tín chỉ**

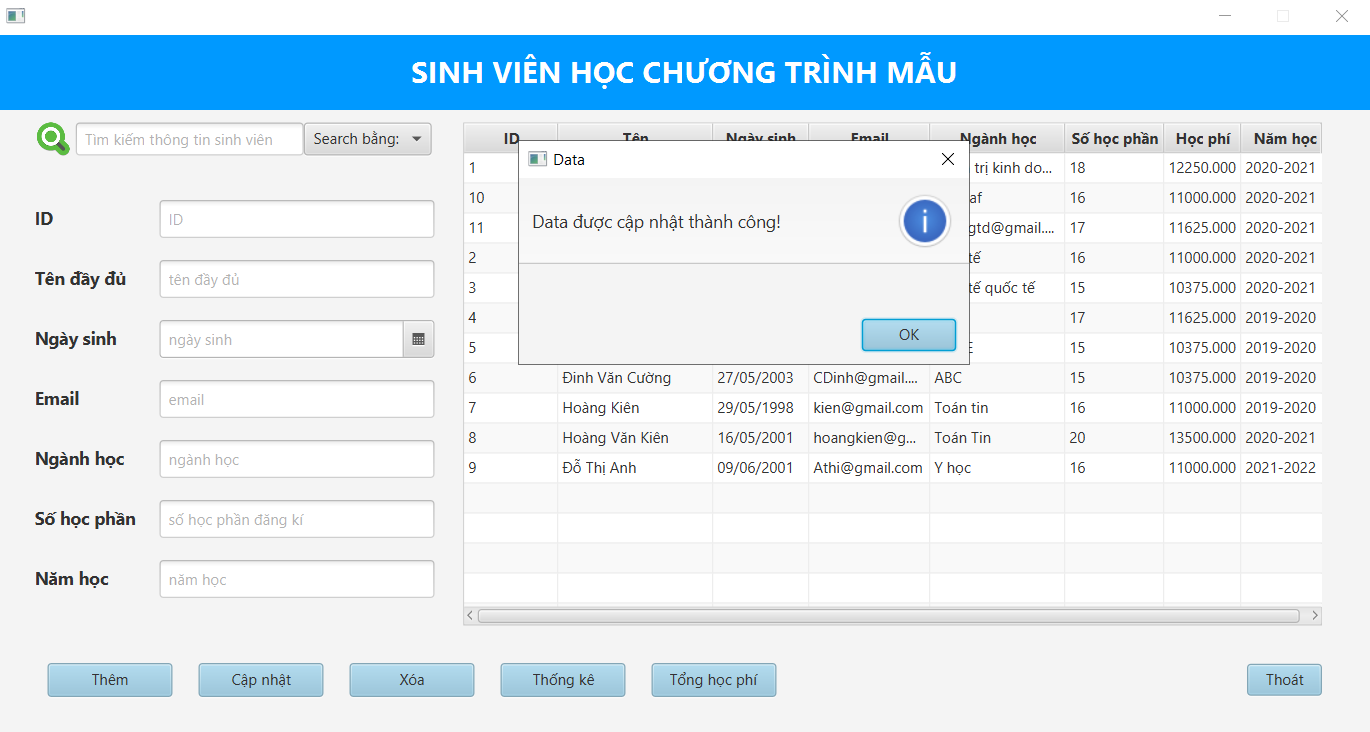
* Đến đây, ta có thể làm rất nhiều thứ: thêm, cập nhật, xóa thông tin sinh viên, thống kê các sinh viên theo năm học, tình tổng học phí theo các năm học được nhập vào, sẽ đưa ra thông báo lỗi nếu nhập sai quy định (để Text Field trống).



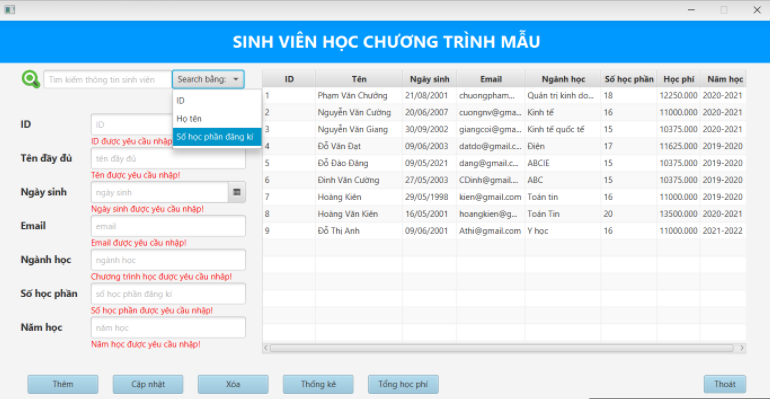
**Lỗi để Text Field trống**



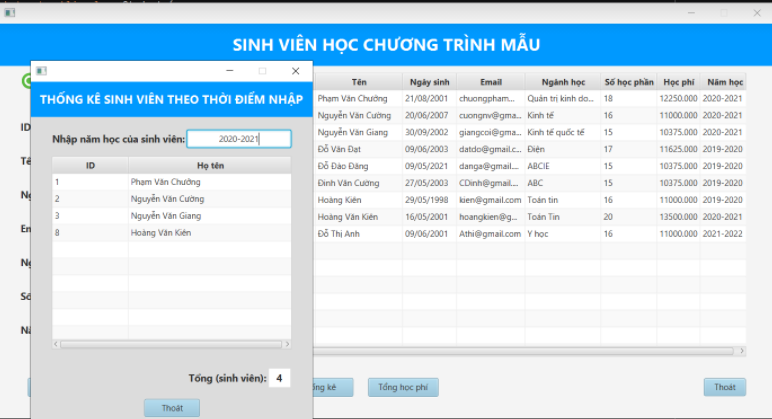


****

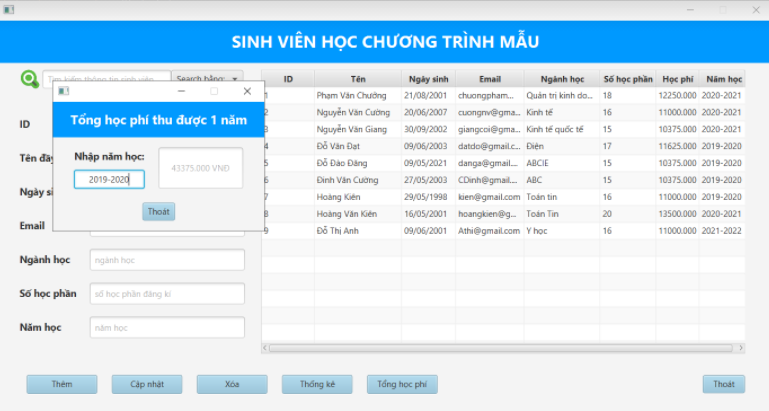
**Thêm, xóa hoặc cập nhật data thành công sẽ có thông báo thành công**



**Có thể chọn tìm kiếm theo ba phương án: theo ID, Họ tên hoặc Số học phần đăng ký**



**Click vào ô “Thống kê” để hiển thị danh sách sinh viên theo năm học**

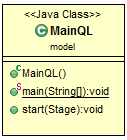


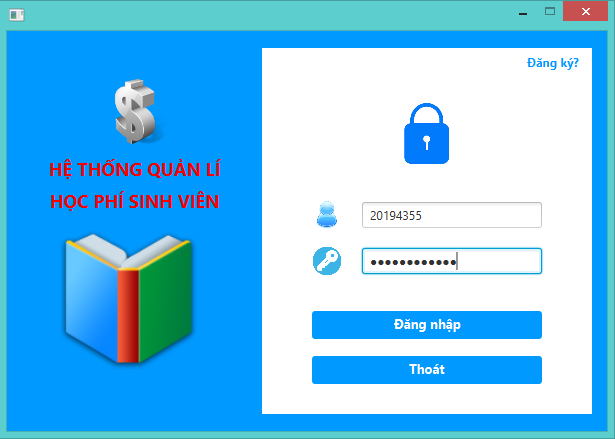
**Click vào ô “Tổng học phí” để xem tổng học phí các sinh viên theo năm nhập vào**

* Sau khi click chuột vào nút Thoát, hệ thống sẽ ngắt kết nối với bảng tblHocCtrMau và quay về giao diện quản lý học phí với hai hình thức học để chọn.

## Các lớp mẫu

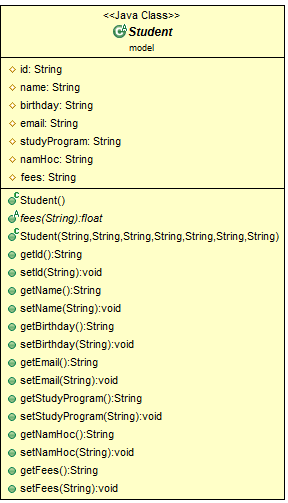
* MainQL: dùng để chạy hệ thống quản lý học phí sinh viên, xây dựng kích cỡ, cấu hình giao diện hiển thị.



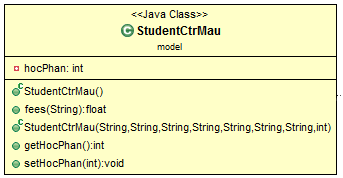


**Giao diện hệ thống quản lí học sinh xuất hiện sau khi chạy class MainQL**

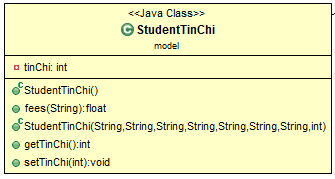
* Student: lớp với đối tượng là các học sinh, gồm các thuộc tính để lưu tên, chương trình học, id, email, năm học, học phí và các phương thức khởi tạo, get(), set() cơ bản, gồm các lớp mẫu con:



* StudentCtrMau: lớp với đối tượng là các học sinh học chương trình mẫu, có thêm thuộc tính hocPhan dùng để lưu số học phần, cũng như các phương thức khởi tạo, get(), set() mới để lưu và lấy số học phần.



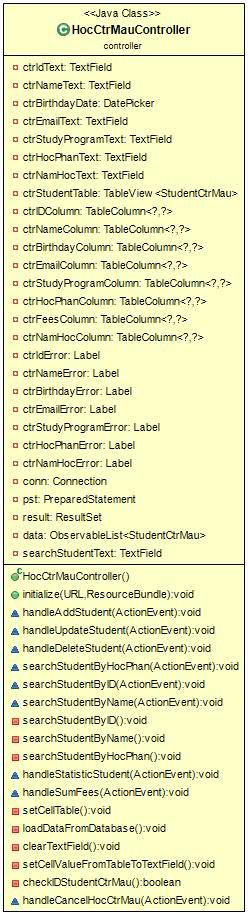
* StudentTinChi: lớp với đối tượng là các học sinh học theo tín chỉ, có thêm thuộc tính tinChi dùng để lưu số tín chỉ, cũng như các phương thức khởi tạo, get(), set() mới để lưu và lấy số tín chỉ.

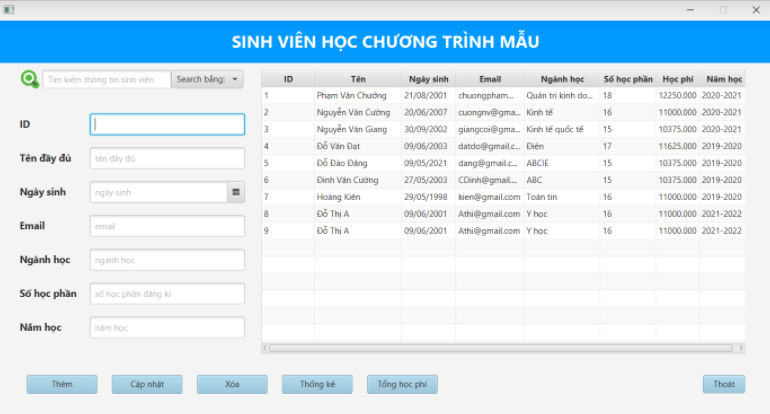
****



## Các lớp điều khiển

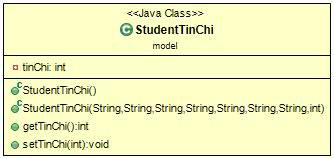
* HocCtrMauController: lớp dùng để điều khiển các thao tác đối với cơ sở dữ liệu có các học sinh học chương trình mẫu, giúp người sử dụng hệ thống tìm kiếm, tra cứu thông tin sinh viên theo một trong nhiều cách (theo id, tên, email,…). Ngoài ra còn có các chức năng thêm, xóa sinh viên (đưa ra lỗi khi ID sinh viên bị trùng với ID của một sinh viên đã được nhập vào), cập nhật thông tin sinh viên, thống kê sinh viên theo năm học được nhập từ bàn phím, tính tổng học phí của các sinh viên theo từng năm học được nhập vào. Người dùng cũng có thể trở về màn hình chính bằng cách click chuột trái vào nút Thoát ở phía dưới góc phải màn hình.

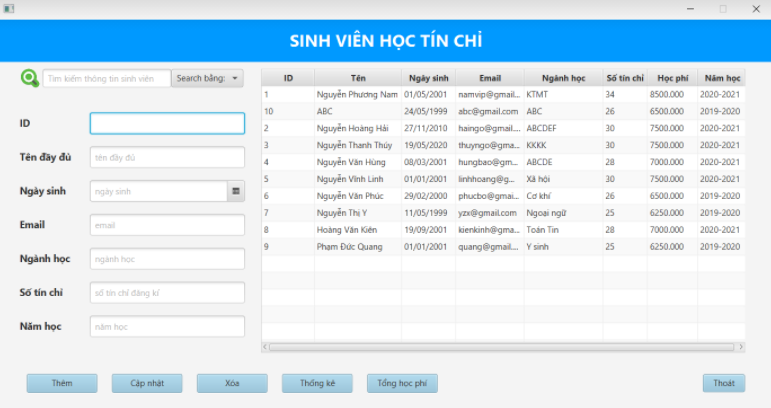




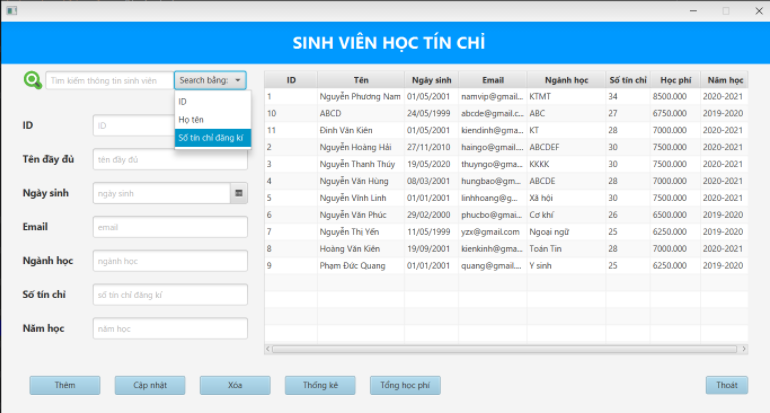
**Lớp HocCtrMauController điều khiển các thao tác với cơ sở dữ liệu của các sinh viên học theo chương trình mẫu**

* HocTinChiController: tương tự HocCtrMauController, nhưng áp dụng cho các sinh viên học theo hình thức tín chỉ.



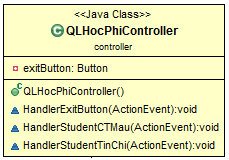


**Lớp HocTinChiController điều khiển các thao tác với cơ sở dữ liệu của các sinh viên học theo tín chỉ**



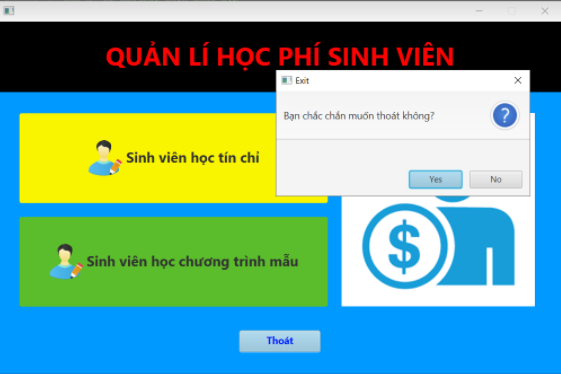
**Có thể tra cứu theo nhiều cách khác nhau**

* QLHocPhiController: lớp điều khiển giao diện quản lí học phí với 2 lựa chọn là danh sách chứa thông tin học sinh học chương trình mẫu và tín chỉ, với 1 nút Thoát để thoát ra giao diện quản lí học phí.



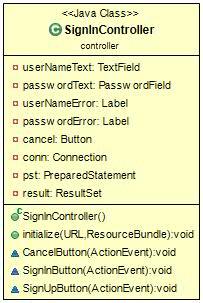


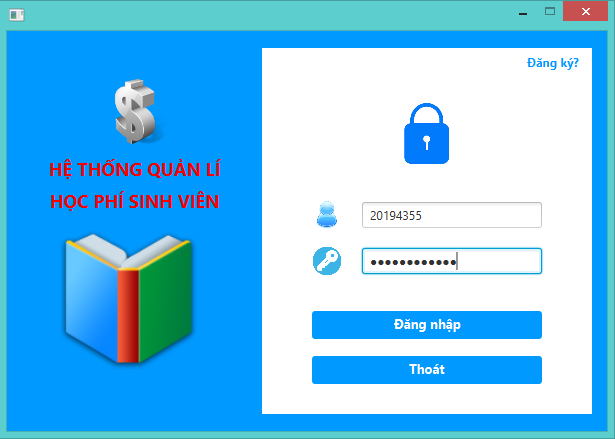
**Lớp QLHocPhiController điều khiển giao diện giúp người sử dụng có thể chọn một trong hai hình thức học của sinh viên để tra cứu danh sách sinh viên theo từng hình thức**



**Click vào nút Thoát để thoát khỏi hệ thống**

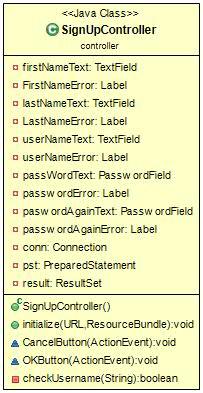
* SignInController: lớp điều khiển giúp người sử dụng có thể đăng nhập vào hệ thống, kiểm tra thông tin đăng nhập, hiển thị thông báo nhập không hợp lệ khi nhập sai thông tin hoặc tài khoản không tồn tại (trường hợp này ta nhấp chuột chữ “Đăng ký?” ở góc trên bên phải giao diện để đăng ký một tài khoản mới). Sau khi nhập tài khoản và mật khẩu hợp lệ, người sử dụng nhấp chuôt vào nút “Đăng nhập” để vào hệ thống quản lí học phí sinh viên. Nhấp vào nút “Thoát” để thoát khỏi chương trình.

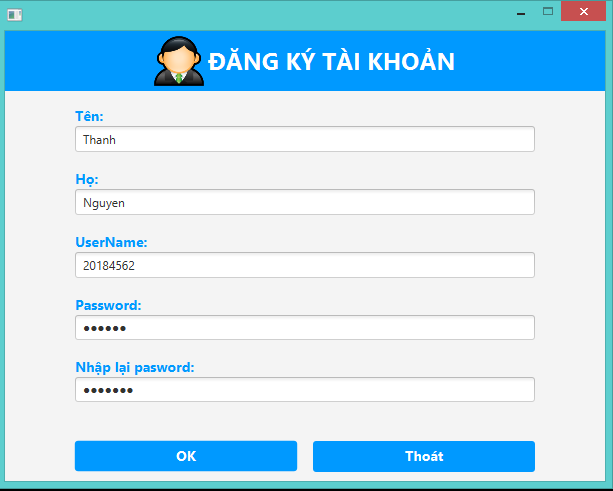




**Lớp SignInController điều khiển giao diện đăng nhập**

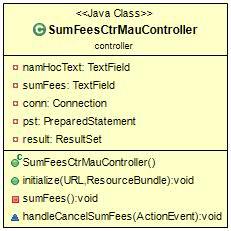
* SignUpController: lớp điều khiển giúp người sử dụng đăng ký tài khoản đăng nhập, nếu nhập hai mật khẩu khác nhau hệ thống sẽ báo lỗi, cần nhập đúng mật khẩu. Nếu nhập bất kì ô nào trống hệ thống sẽ báo lỗi.

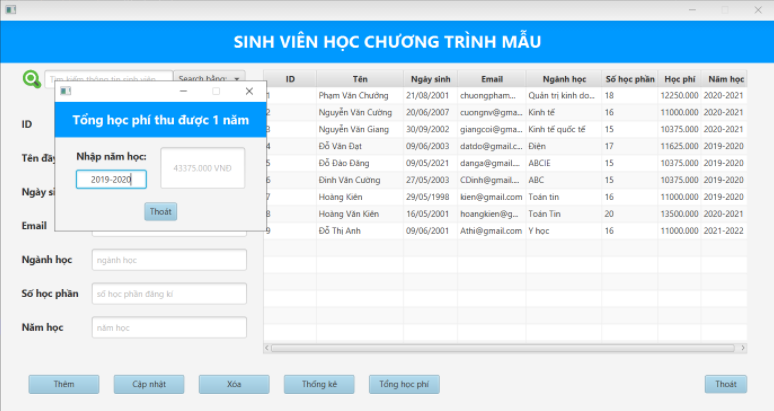




**Lớp SignUpController giúp người sử dụng đăng ký tài khoản**

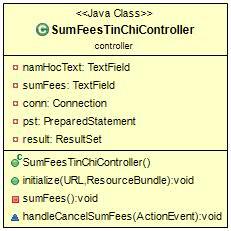
* SumFeesCtrMauController: lớp điểu khiển cho phép người dùng tra cứu tổng học phí của các sinh viên theo từng năm. Ví dụ, nhập “2018-2019” vào ô năm học để xem tổng học phí của các sinh viên học vào năm đó.

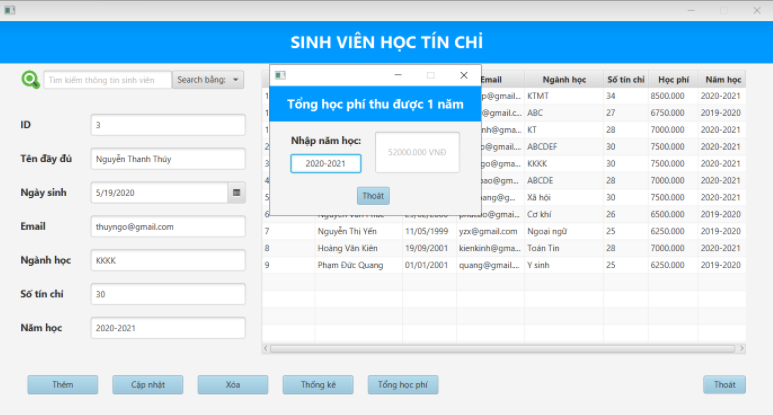




**SumFeesCtrMauController điều khiển giao diện xem tổng học phí của các sinh viên học theo chương trình mẫu theo từng năm học**

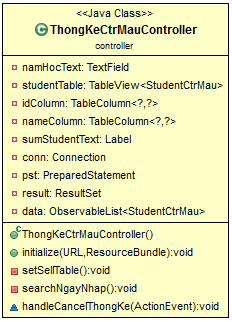
* SumFeesTinChiController: có chức năng tương tự như SumFeesCtrMauController, nhưng áp dụng cho các sinh viên học tín chỉ.

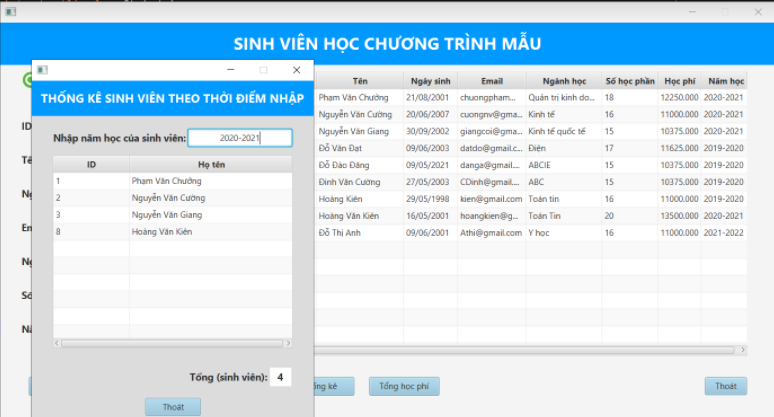




**SumFeesTinChiController điều khiển giao diện xem tổng học phí của các sinh viên học theo tín chỉ theo từng năm học**

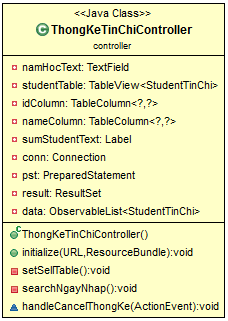
* ThongKeCtrMauController: đưa ra danh sách các sinh viên học theo chương trình mẫu, dữ liệu hiện ra sẽ là các sinh viên có năm học giống với năm học được nhập từ bàn phím. Có chức năng hiện tổng số sinh viên học chương trình mẫu năm học đó, click chuột vào nút Thoát để quay về giao diện quản lí sinh viên học chương trình mẫu.

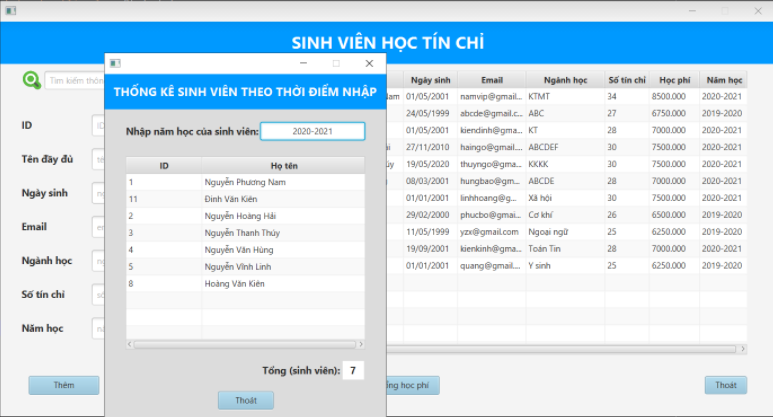




**ThongKeCtrMauController điều khiển giao diện thống kê sinh viên học theo chương trình mẫu theo năm học nhập vào từ bàn phím**

* ThongKeTinChiController: có chức năng tương tự như ThongKeCtrMauController nhưng áp dụng cho các sinh viên học theo tín chỉ. Nút Thoát dùng để quay về giao diện quản lí sinh viên học theo hình thức tín chỉ.

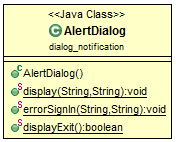


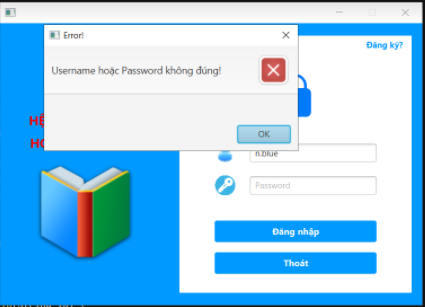


**ThongKeTinChiController điều khiển giao diện thống kê sinh viên học theo tín chỉ theo năm học nhập vào từ bàn phím**

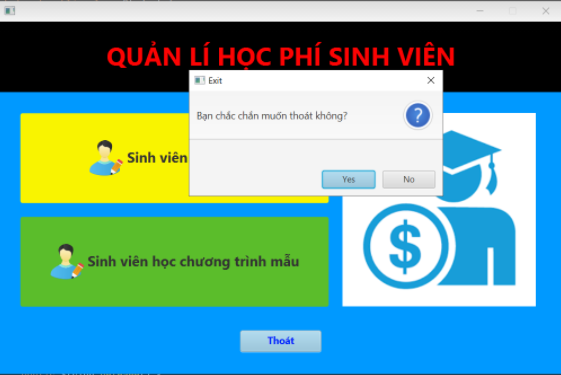
## Lớp thông báo

* AlertDialog: hiển thị thông báo thành công hoặc thông báo lỗi khi tra cứu hoặc đăng nhập, và hiện câu hỏi để hỏi lại xem người dùng có muốn thoát khỏi chương trình sau khi click chuột vào nút Thoát hay không.





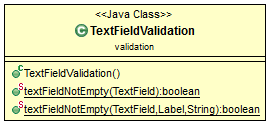
**Thông báo lỗi khi nhập sai thông tin đăng nhập**

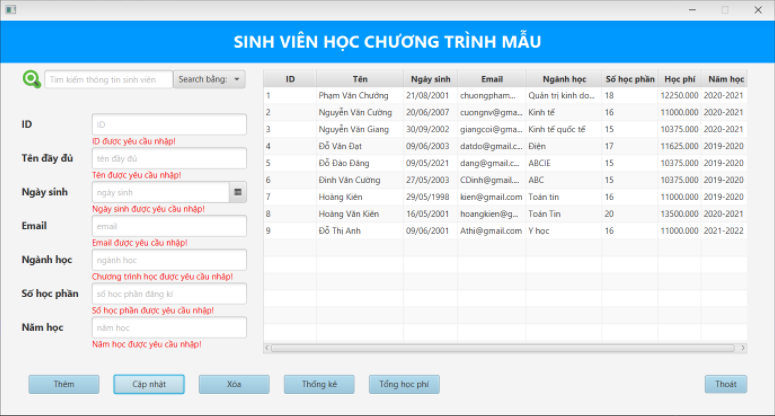


**Hỏi lại người dùng xem người dùng có muốn thoát khỏi hệ thống không**

## Lớp kiểm tra

* TextFieldValidation: kiểm tra xem Text Field có rỗng hay không, có thì hiển thị thông báo lỗi tùy theo từng Text Field.

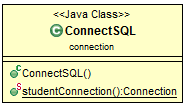


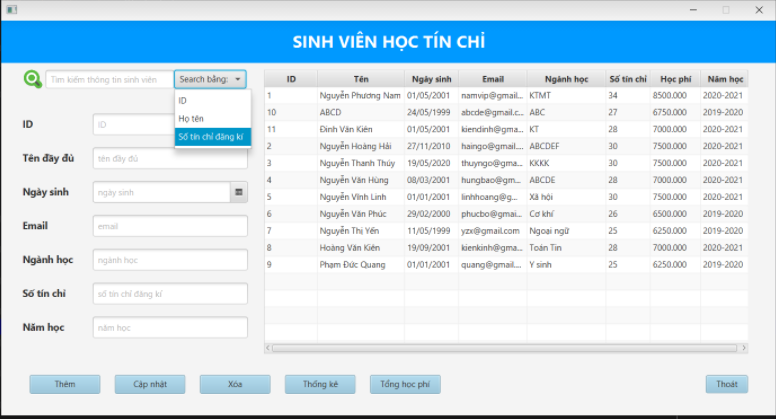


**Hiển thị thông báo lỗi khi người dùng nhập các Text Field trống**

## Lớp kết nối

* ConnectSQL: dùng để kết nối hệ thống quản lí với cơ sở dữ liệu chứa thông tin của các sinh viên, hệ thống được kết nối ngay sau khi người dùng đăng nhập thành công, ở đây ta kết nối với hệ quản trị cơ sở dữ liệu Microsoft SQL Server.

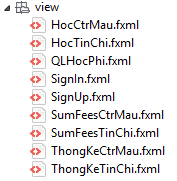


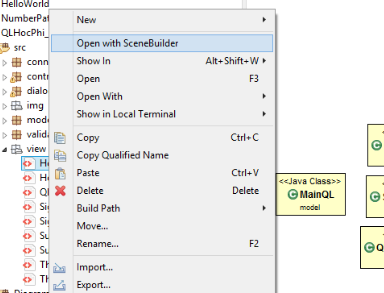


**Kết nối với cơ sở dữ liệu chưa thông tin các sinh viên, ở đây chỉ chứa các sinh viên học theo hình thức tín chỉ**

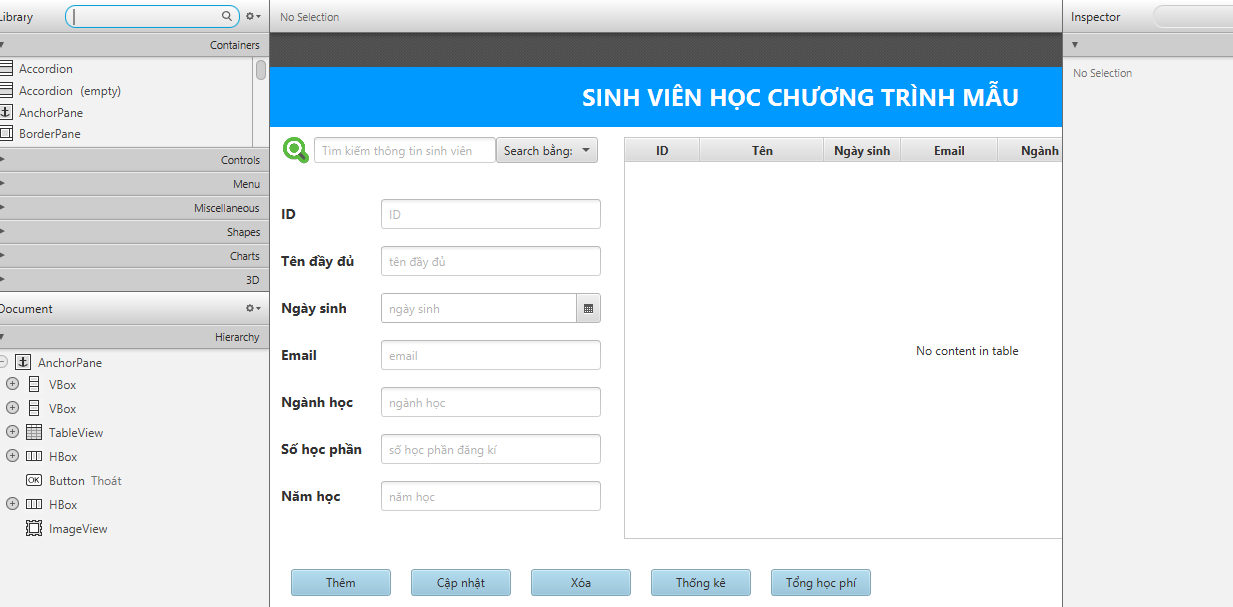
## Các lớp giao diện fxml

* Các lớp dùng để tạo giao diện, hình ảnh, các nút, được điều khiển bởi các lớp điều khiển, mỗi lớp điều khiển ứng với một lớp giao diện fxml:





**Click chuột phải và 1 lớp fxml, chọn Scene Buider để xem giao diện**



**Scene Buider hiện lên, bạn có thể thêm nút, box, pane, chèn hình ảnh ở đây**

# Tự đánh giá kết quả, các điểm hạn chế

## Ưu điểm

* Giao diện tối giản, vừa mắt.
* Bố cục rõ rang, trình bày hợp lý giúp người dùng dễ dàng nhận biết chức năng của các nút và dễ thao tác sử dụng.
* Chương trình có hệ thống đăng nhập giúp tăng cường bảo mật và hạn chế người truy cập.

## Nhược điểm

* Danh sách nhân viên chưa được đồng bộ mà hoạt động độc lập giữa hai loại sinh viên là theo hình thức tín chỉ và chương trình mẫu dẫn đến tính không toàn vẹn dữ liệu.
* Việc không đồng bộ dữ liệu dữ liệu khiến người dùng gặp khó khăn trong các thao tác thêm, xóa, sửa, đặc biệt là khi thiếu dữ liệu về sinh viên.