TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**Trần Mộc Cát Tường - 52100505**

**BÁO CÁO CUỐI KỲ**

**LẬP TRÌNH MẠNG CĂN BẢN**

Giáo viên hướng dẫn

**TS. Bùi Quy Anh**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2024**

LỜI CẢM ƠN

Em xin chân thành cảm ơn Khoa Công nghệ thông tin, Trường đại học Tôn Đức Thắng đã tạo điều kiện học tập thuận lợi để em có thể hoàn thành bài báo cáo này. Đặc biệt, chúng em xin bày tỏ lòng biết ơn sâu sắc đến thầy Bùi Quy Anh đã truyền đạt kiến thức, những kinh nghiệm quý giá và hướng dẫn chúng em trong quá trình làm bài.

Em đã cố gắng vận dụng những kiến thức đã học trong học kỳ vừa qua để hoàn thành bài báo cáo. Tuy nhiên, những kiến thức và kỹ năng về môn học này vẫn còn nhiều hạn chế do đó bài báo cáo của nhóm khó tránh khỏi những sai sót. Kính mong quý thầy xem xét và góp ý giúp bài báo cáo của chúng em được hoàn thiện hơn.

**BÁO CÁO ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam kết đây là chương trình do bản thân tự tìm hiểu và xây dựng. Các nội dung tìm hiểu nghiên cứu, kết quả trong đề tài này là trung thực. Những dữ liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung báo cáo của mình.** Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày tháng năm 2024*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Trần Mộc Cát Tường*

TÓM TẮT

Bài báo cáo chúng em tiến hành xây dựng các ứng dụng trao đổi thông tin giữa các máy tính sử dụng giao TCP và UDP gồm các chương trình chat giữa client và server, một số chương trình ứng dụng khác: Hệ thống quản lý sinh viên, máy tính.

MỤC LỤC

**[CHƯƠNG 1 – TỔNG QUAN BÁO CÁO](#_Toc14635)**

**[1.1 Tổng quan cơ sở lý thuyết. 6](#_Toc32207)**

**[1.2 Tổng quan chương trình. 6](#_Toc8904)**

*[1.2.1 Thiết kế tổng quát.](#_Toc24797)* [6](#_Toc24797)

*[1.2.2 Chương trình tổng quát.](#_Toc20114)* [7](#_Toc20114)

**[CHƯƠNG 2 – CƠ SỞ LÝ THUYẾT 8](#_Toc16584)**

**[2.1 Lý thuyết cơ bản về Java. 8](#_Toc24211)**

**[2.2 Lý thuyết về tra cứu địa chỉ Internet. 10](#_Toc29855)**

*[2.2.1 Lớp InetAddress.](#_Toc15905)* [10](#_Toc15905)

*[2.2.2 Lớp NetworkInterface.](#_Toc28887)* [12](#_Toc28887)

**[2.3 Lớp URL. 14](#_Toc21619)**

[2.4 URL Connection. 15](#_Toc31254)

**[2.5 Datagrams và Socket của UDP. 16](#_Toc3251)**

*[2.5.1 Tổng quan về UDP.](#_Toc11129)* [17](#_Toc11129)

*[2.5.2 Lớp DatagramPacket.](#_Toc26563)* [18](#_Toc26563)

**[2.6 Lý thuyết cho Socket của TCP. 19](#_Toc21445)**

*[2.6.1 Tổng quan về TCP socket.](#_Toc11538)* [19](#_Toc11538)

*[2.6.2 Lớp ServerSocket.](#_Toc2403)* [20](#_Toc2403)

**[2.7 JDBC cơ bản. 21](#_Toc15155)**

**[CHƯƠNG 3 – THIẾT KẾ. 26](#_Toc4902)**

**[3.1 Mô hình đề suất 26](#_Toc15895)**

[3.1.1 Mô hình hệ thống tổng quan 26](#_Toc29719)

[3.1.2 Xây dựng chương trình 34](#_Toc16)

**[CHƯƠNG 4: KẾT LUẬN 191](#_Toc6886)**

**[4.1 Những kết quả dạt được. 191](#_Toc29714)**

**[4.2 Những hạn chế. 191](#_Toc31158)**

**[4.3 Hướng phát triển. 191](#_Toc19142)**

**DANH MỤC CHỮ VIẾT TẮT**

**CÁC CHỮ VIẾT TẮT**

DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ

**DANH MỤC HÌNH**

[Hình 1 : Mô hình tổng quát chương trình. 6](#_Toc21618)

[Hình 2 : Các thành phần trong URL. 14](#_Toc13746)

[Hình 3 : Mô hình hoạt động của UDP socket. 17](#_Toc13597)

[Hình 4 : Quy trình hoạt động UDP socket. 18](#_Toc26541)

[Hình 5 : Quy trình hoạt động TCP socket. 20](#_Toc15261)

[Hình 6 : Kiến trúc JDBC. 22](#_Toc28042)

[Hình 7 : Tải xuống MySQL Connector/J. 23](#_Toc32649)

[Hình 8 : Giao diện sau khi thêm thành công MYSQL Connector/J. 24](#_Toc2609)

[Hình 9 : Ví dụ mã cho kết nối với cơ sở dữ liệu qlsv. 24](#_Toc16034)

[Hình 10 : Các thư viện danh cho JDBC. 25](#_Toc27151)

[Hình 11 : Biểu đồ tổng quát của đề tài. 26](#_Toc3673)

[Hình 12 : Biểu đồ chức năng đăng nhập. 27](#_Toc185)

[Hình 13 : Biểu đồ chức năng đổi mật khẩu. 27](#_Toc18065)

[Hình 14 : Biểu đồ chức năng đăng xuất. 28](#_Toc27090)

[Hình 15 : Biểu đồ chức năng xem điểm. 28](#_Toc2469)

[Hình 16 : Biểu đồ chức năng xem thông tin. 29](#_Toc20784)

[Hình 17 : Biểu đồ chức năng gửi tin nhắn. 29](#_Toc3501)

[Hình 18 : Biểu đồ chức năng sửa điểm. 30](#_Toc9786)

[Hình 19 : Biểu đồ chức năng sửa thông tin 30](#_Toc2813)

[Hình 20 : Biểu đồ chức năng xóa thông tin. 31](#_Toc7828)

[Hình 21 : Biểu đồ chức năng thêm học sinh. 31](#_Toc29679)

[Hình 22 : Biểu đồ chức năng tra cứu điểm theo ID Học Sinh. 32](#_Toc19455)

[Hình 23 : Biểu đồ chức năng tra cứu thông tin theo ID Học Sinh. 32](#_Toc23011)

[Hình 24 : Biểu đồ chức năng tính toán của máy tính. 33](#_Toc18810)

[Hình 25 : Biểu đồ chức năng lịch sử tính toán. 33](#_Toc9091)

[Hình 26 : Mô hình thực thể kiên kết. 34](#_Toc17325)

[Hình 27 : Khai báo thư viện Server\_UDP.java. 73](#_Toc30796)

[Hình 28 : Khai báo thư viện QuanLyDiem\_Server.java. 79](#_Toc24712)

[Hình 29 : Khai báo thư viện ChinhSuaDiem.java. 91](#_Toc1354)

[Hình 30 : Khai báo thư viện Server\_TCP\_MayTinh.java 129](#_Toc18984)

[Hình 31 : Khai báo thư viện Login.java 134](#_Toc32546)

[Hình 32 : Khai báo thư viện Client\_UDP.java 143](#_Toc24826)

[Hình 33 : Mở MySQL Workbench. 178](#_Toc1366)

[Hình 34 : Kết nối với MySQL server. 178](#_Toc22818)

[Hình 35 : Đăng nhập. 179](#_Toc19141)

[Hình 36 : Chọn Server > chọn DataImport. 179](#_Toc18206)

[Hình 37 : Import qlsv.sql. 180](#_Toc10097)

[Hình 38 : Import thành công. 180](#_Toc24486)

[Hình 39 : Import dự án java. 180](#_Toc30597)

[Hình 40 : Chọn Existing Projects into Workspace. 181](#_Toc28375)

[Hình 41 : Dẫn đường link và nhấn Finish. 181](#_Toc27543)

[Hình 42 : Dự án Final đã được thêm thành công. 182](#_Toc19089)

[Hình 43 : Cấu hình tài khoản kết nối. 182](#_Toc11818)

[Hình 44 : Mở MainMenu.jar 183](#_Toc17438)

[Hình 45 : Mở frame TCP Protocal. 183](#_Toc631)

[Hình 46 : Mở frame Chat Room. 184](#_Toc23828)

[Hình 47 : Mở Server và 2 Client. 184](#_Toc16493)

[Hình 48 : Thử giao tiếp với nhau. 184](#_Toc17276)

[Hình 49 : Bấm nút Máy tính để mở 2 frame MayTinh và Server. 185](#_Toc20982)

[Hình 50 : Thử tính toán. 185](#_Toc22835)

[Hình 51 : Chức năng xem lịch sử. 185](#_Toc5500)

[Hình 52 : UDP Protocal frame. 186](#_Toc24454)

[Hình 53 : Bấm vào nút Hệ thống quản lý học sinh. 186](#_Toc24286)

[Hình 54 : Đăng nhập với mật khẩu là "123". 186](#_Toc25904)

[Hình 55 : Xem thông tin ở Client. 187](#_Toc28755)

[Hình 56 : Chat giữa Client và Server 187](#_Toc16808)

[Hình 57 : Quản lý điểm. 187](#_Toc16341)

[Hình 58 : Chỉnh sửa điểm. 188](#_Toc5039)

[Hình 59 : Quản lý học sinh. 188](#_Toc16712)

[Hình 60 : Thêm học sinh. 189](#_Toc18027)

[Hình 61 : Xóa học sinh. 189](#_Toc21014)

[Hình 62 : Chỉnh sửa học sinh. 190](#_Toc1206)

**DANH MỤC BẢNG**

[Bảng 1 : Bảng các phương thức quan trọng trong lớp URL. 15](#_Toc27260)

[Bảng 2 : Bảng các phương thức quan trọng trong lớp URL Connection. 16](#_Toc5644)

[Bảng 3 : Bảng các phương thức được quan trọng trong lớp DatagramPacket. 19](#_Toc4261)

[Bảng 4 : Bảng các phương thức được quan trọng trong lớp ServerSocket. 21](#_Toc23946)

CHƯƠNG 1 – TỔNG QUAN BÁO CÁO

* 1. **Tổng quan cơ sở lý thuyết.**

- Lý thuyết cơ bản về Java.

- Lý thuyết về địa chỉ Internet.

- Lớp URL.

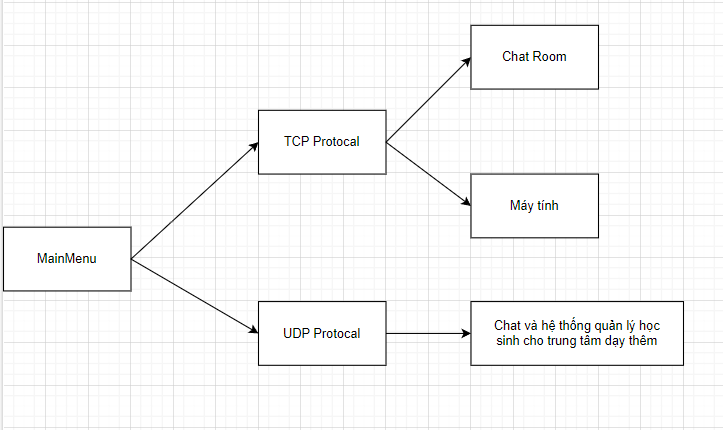
- URLConnection.

- Datagrams và Socket của UDP.

- Lý thuyết cho Socket của TCP.

- JDBC cơ bản.

* 1. **Tổng quan chương trình.**
     1. ***Thiết kế tổng quát.***



Hình 1: Mô hình tổng quát chương trình.

- Mô hình này được xây dựng dựa trên hai giao thức chính là UDP và TCP. Trong đó, ứng dụng Chat Room sử dụng TCP Protocol cho phép các Client giao tiếp với nhau thông qua một máy chủ (Server), trong khi ứng dụng Chat sử dụng UDP Protocol để giao tiếp với quản lý trung tâm. Ngoài ra, có một máy tính được sử dụng cho tính toán, dựa trên việc trao đổi dữ liệu qua giao thức TCP. Cuối cùng, hệ thống quản lý học sinh có đầy đủ các chức năng bao gồm đăng nhập, đổi mật khẩu, xem điểm, kiểm tra thông tin và chat ở phía Client, cũng như quản lý chỉnh sửa điểm kiểm tra, thêm/xóa/sửa thông tin học sinh và chat ở phía Server.

* + 1. ***Chương trình tổng quát.***

**-** TCP Chat Room.

- TCP Máy Tính.

- UDP Hệ Thống Quản Lý Học Sinh Cho Trung Tâm Dạy Thêm:

Client:

+ Đăng nhập, đăng xuất.

+ Đổi mật khẩu.

+ Chat.

+ Xem điểm, Xem thông tin.

Server:

+ Chat.

+ Chỉnh sửa điểm.

+ Xem điểm tổng.

+ Tìm kiếm bảng điểm dựa trên ID học sinh.

+ Thêm/ xóa/ sửa thông tin học sinh.

+ Xem thông tin tổng.

+ Tìm kiếm thông tin dựa trên ID học sinh.

CHƯƠNG 2 – CƠ SỞ LÝ THUYẾT

**2.1 Lý thuyết cơ bản về Java.**

***2.1.1 Biến.***

-Biến là một vùng nhớ dùngđể lưu trữ giá trị mà giá trị đó có thể thay đổi khi chương trình thực thi.

- Mỗi biến được gắn liền với một kiểu dữ liệu và một định danh duy nhất

gọi là tên biến.

Cú pháp: [Kiểu dữ liệu] [Tên biến].

***2.1.2 Hằng số.***

- Khác với biến, hằng sô là giá trị không thể thay đổi được (bất biến).

- Cú pháp: final [Kiểu dữ liệu] [Tên biến] = [Giá trị].

- Quy tắc đặt tên hằng số: Chữ in hoa + underscore.

VD: int YOUR\_BIRTHDAY = 2000;

***2.1.3 Kiểu dữ liệu.***

- Kiểu dữ liệu có 2 loại đó là kiểu dữ liệu gốc và kiểu dữ liệu đối tượng.

- Kiểu dữ liệu gốc: Boolean, Char, Byte, Short, Int, Long, Float, Double.

- Kiểu dữ liệu đối tượng: String, Array.

***2.1.4 Toán tử.***

- Toán tử số học+, -, \*, /, %

- Toán tử gán +=, -=, \*=, /=, %=

- Toán tử+ +, --

- Toán tử so sánh >, <, >=, <=, ==, !=

- Toán tử logic&&, ||,

***2.1.5 Điều kiện.***

Câu điều kiện IF ... ELSE

Câu điều kiện if

Câu điều kiện if ..... else

Câu điều kiện if ... else if .... else

Câu điều kiện SWITCH

***2.1.5 Vòng lặp.***

Các vòng lặp thường dùng trong Java:

- For

- While

- Do .... While

***2.1.5 Lập trình thủ tục.***

- Giải quết vấn đề từng bước đến khi đạt yêu cầu.

- Lập trình từ trên xuống.

- Lập trình theo hàm chỉ tạo ra hàm xử lý khi gặp vấn đề.

***2.1.5 Lập trình hướng đối tượng.***

- Lập trình hướng đối tượng (Object Oriented Programing – OOP) là một phương pháp để thiết kế một chương trình bởi sử dụng các lớp và các đối tượng.

- Java là một ngôn ngữ lập trình hướng đối tượng vì vậy nó cũng hỗ trợ các đặc tính của lập trình hướng đối tượng:

Đa hình (Polymorphism)

Thừa kế (Inheritance)

Đóng gói (Encapsulation)

Trừu tượng (Abstraction)

***2.1.5.1 Đối tượng (object).***

- Đối tượng là một thực thể có trạng thái và hành vi. Nó có thể mang tính vật lý hoặc logic.

- Nếu chúng ta xem xét thực tế chúng ta có thể tìm thấy nhiều đồ vật xung quanh chúng ta: cái bàn, con chó, con người, v.v… Tất cả các đối tượng này đều có thuộc tính và hành vi.

***2.1.5.1 Lớp (Class).***

- Chúng ta có thể xem lớp như một khuôn mẫu (template) của đối tượng (Object). Trong đó bao gồm dữ liệu của đối tượng (fields hay properties) và các phương thức(methods) tác động lên thành phần dữ liệu đó gọi là các phương thức của lớp.

**2.2 Lý thuyết về tra cứu địa chỉ Internet.**

***2.2.1 Lớp InetAddress.***

-**Lớp InetAddress** là một trong những lớp thuộc gói Java.net, xử lý các địa chỉ Internet cả dưới dạng tên máy chủ và địa chỉ IP.

***2.2.1.1 Tạo đối tượng InetAddress mới.***

- Cú pháp tạo đối tượng InetAddress gồm có ba phương thức:

1. public static InetAddress getByName(String hostName)

2. public static InetAddress[] getAllByName(String hostName)

3. public static InetAddress getLocalHost( )

Tất cả các phương thức này đều thực hiện kết nối tới server DNS cục bộ để biết được các thông tin trong đối tượng InetAddress.

***2.2.1.2 Nhận các trường thông tin của một đối tượng InetAddress.***

- Chỉ có các lớp trong gói java.net mới có quyền truy cập tới các trường của lớp InetAddress. Các lớp trong gói này có thể đọc các trường của một đối tượng InetAddress bằng cách gọi phương thức **getHostName()** và **getAddress()**.

- **public String getHostName()**: Phương thức này trả về một chuỗi biểu diễn hostname của một đối tượng InetAddress. Nếu máy không có hostname, thì nó sẽ trả về địa chỉ IP của máy này dưới dạng một chuỗi ký tự.

- **public byte[] getAddress()**: Nếu bạn muốn biết địa chỉ IP của một máy, phương thức getAddress() trả về một địa chỉ IP dưới dạng một mảng byte.

- Một số địa chỉ IP và một số mô hình địa chỉ có các ý nghĩa đặc biệt.

Ví dụ:

+ **127.0.0.1** là địa chỉ loopback.

+ Các địa chỉ IPv4 trong khoảng **224.0.0.0 => 239.255.255.255** là các địa chỉ multicast.

- Java 1.5 đã thêm vào hai phương thức cho lớp InetAddress cho phép các ứng dụng kiểm tra liệu một nút cụ thể có thể kết nối được hay không với nút xuất phát là nút hiện hành; nghĩa là kiểm tra xem một liên kết mạng đã được thiết lập hay chưa. Các liên kết có thể bị phong tỏa vì nhiều nguyên nhân như tường lửa, các server ủy quyền, các router hoạt động sai chức năng, dây cáp bị đứt, hoặc host ở xa không bật.

public boolean isReachable(int timeout)

throws IOException

public boolean isReachable(NetworkInterface netif, int ttl, int timeout)

throws IOException

- Các phương thức này cố gắng kết nối trên cổng echo của host ở xa để xác định xem nó có thể kết nối được hay không. Nếu host đáp ứng trong khoảng thời gian timeout tính bằng mili giây, các phương thức này sẽ trả về giá trị true nếu kết nối được, ngược lại nó sẽ trả về giá trị false.

***2.2.2 Lớp NetworkInterface.***

- Lớp NetworkInterface đại diện cho một địa chỉ IP cục bộ. Đây có thể là một giao diện vật lý như một card Ethernet bổ sung (thường thấy trên các tường lửa và bộ định tuyến), hoặc nó có thể là một giao diện ảo được gán cho cùng phần cứng vật lý với các địa chỉ IP khác của máy.

***2.2.2.1 Phương thức getByName().***

- Phương thức get Byname() của lớp InetAddress xác định địa chỉ IP của một máy chủ từ tên máy chủ được cung cấp.

- Nếu tên máy chủ là null, thì một đối tượng InetAddress đại diện cho địa chỉ của interface loopback sẽ được trả về.

Cú pháp:

public static InetAddress getByName(String host)

throws UnknownHostException

***2.2.2.2 Phương thức getByInetAddress().***

- Phương thức để tìm kiếm một interface mạng có địa chỉ IP được gán cho nó. Nếu địa chỉ IP được chỉ định được gán cho nhiều interface mạng, thì không xác định được interface mạng nào sẽ được trả về.

Cú pháp:

public static NetworkInterface getByInetAddress(InetAddress addr)

throws SocketException

***2.2.2.3 Phương thức getNetworkInterfaces().***

- Trả về tất cả các interface trên máy này. Đối tượng Enumeration chứa ít nhất một phần tử, có thể là một interface loopback chỉ hỗ trợ giao tiếp giữa các thực thể trên máy này.

Cú pháp:

public static Enumeration getNetworkInterfaces( )

throws SocketException

***2.2.2.4 Phương thức getInetAddresses().***

- Phương thức trả về một đối tượng Enumeration chứa tất cả hoặc một phần các InetAddress được gán cho interface mạng này.

- Nếu có một security manager, phương thức checkConnect của nó sẽ được gọi cho từng InetAddress. Chỉ các InetAddress mà checkConnect không ném ra SecurityException mới được trả về trong Enumeration. Tuy nhiên, nếu người gọi có quyền NetPermission("getNetworkInformation") thì tất cả các InetAddress sẽ được trả về.

Cú pháp:

public Enumeration<InetAddress> getInetAddresses()

***2.2.2.5 Phương thức getName().***

- Phương thức trả về một đối tượng Enumeration chứa tất cả hoặc một phần các InetAddress được gán cho interface mạng này.

- Nếu có một security manager, phương thức checkConnect của nó sẽ được gọi cho từng InetAddress. Chỉ các InetAddress mà checkConnect không ném ra SecurityException mới được trả về trong Enumeration. Tuy nhiên, nếu người gọi có quyền NetPermission("getNetworkInformation"), thì tất cả các InetAddress sẽ được trả về.

Cú pháp:

public String getName()

***2.2.2.6 Phương thức getDisplayName().***

- Lấy tên hiển thị của interface mạng này. Tên hiển thị là một chuỗi có thể đọc được bởi con người mô tả thiết bị mạng.

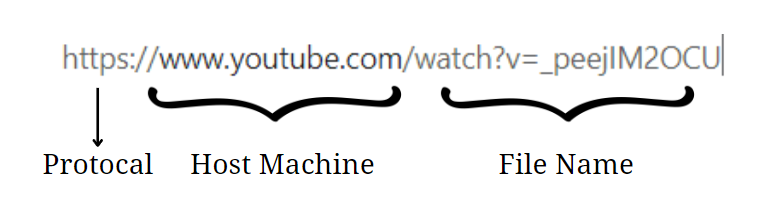
Cú pháp:

public String getDisplayName()

**2.3 Lớp URL.**

***2.3.1 Giới thiệu về URL.***

**-** URL là được biết đến như định vị tài nguyên thống nhất (Uniform Resource Locator), đơn giản là một chuỗi văn bản xác định tất cả các tài nguyên trên Internet, cho chúng ta biết địa chỉ của tài nguyên, cách giao tiếp với nó và lấy thông tin từ nó.



Hình 2: Các thành phần trong URL.

- Một URL có thể có nhiều dạng. Tuy nhiên, dạng tổng quát nhất thường tuân theo hệ thống ba thành phần như sau:

+ Giao thức (Protocol): Ví dụ: HTTP là giao thức ở đây.

+ Tên máy chủ (Hostname): Tên của máy chủ nơi tài nguyên tồn tại.

+ Tên tệp (File Name): Đường dẫn đến tệp trên máy chủ.

+ Số cổng (Port Number): Số cổng để kết nối (thường là tùy chọn).

***2.3.2 Lớp URL cơ bản.***

- Lớp URL là cổng thông tin tới bất kỳ tài nguyên nào có sẵn trên Internet. Một đối tượng URL đại diện cho một Định vị Tài nguyên Thống nhất, là một con trỏ tới một "tài nguyên" trên World Wide Web. Một tài nguyên có thể trỏ đến một tệp đơn giản hoặc thư mục, hoặc nó có thể tham chiếu đến một đối tượng phức tạp hơn, như một truy vấn đến một cơ sở dữ liệu hoặc đến một công cụ tìm kiếm.

***2.3.2.1 Các constructor của lớp URL.***

**URL(String address)** **throws MalformedURLException**: Tạo một đối tượng URL từ chuỗi được chỉ định.

**URL(String protocol, String host, String file)**: Tạo một đối tượng URL từ giao thức, máy chủ và tên tệp được chỉ định.

**URL(String protocol, String host, int port, String file)**: Tạo một đối tượng URL từ giao thức, máy chủ, cổng và tên tệp được chỉ định.

**URL(URL context, String spec)**: Tạo một đối tượng URL bằng cách phân tích chuỗi được chỉ định trong ngữ cảnh được cung cấp.

**URL(String protocol, String host, int port, String file, URLStreamHandler handler)**: Tạo một đối tượng URL từ giao thức, máy chủ, số cổng, tệp và bộ xử lý được chỉ định.

**URL(URL context, String spec, URLStreamHandler handler)**: Tạo một đối tượng URL bằng cách phân tích chuỗi được chỉ định với bộ xử lý được cung cấp trong ngữ cảnh được chỉ định.

***2.3.2.2 Các phương thức quan trọng được sử dụng trong lớp URL.***

|  |  |
| --- | --- |
| **Phương thức** | **Mô tả** |
| getAuthority() | Trả về phần quyền của URL hoặc null nếu nó trống. |
| getDefaultPort() | Trả về cổng mặc định được sử dụng. |
| getFile() | Trả về tên tệp. |
| getHost() | Trả về tên máy chủ của URL dưới dạng IPv6. |
| getPath() | Trả về đường dẫn của URL, hoặc null nếu nó trống. |
| getPort() | Trả về cổng liên kết với giao thức được chỉ định bởi URL. |
| getProtocol() | Trả về giao thức được sử dụng bởi URL. |
| getQuery() | Trả về phần truy vấn của URL. |
| getRef() | Trả về phần tham chiếu (anchor) của URL. |
| toString() | Trả về biểu diễn chuỗi của URL. |

Bảng 1: Bảng các phương thức quan trọng trong lớp URL.

**2.4 URL Connection.**

- **URLConnection** là một lớp trừu tượng đại diện cho một kết nối hoạt động tới một tài nguyên được chỉ định bởi một URL. Lớp URLConnection có hai mục đích khác nhau nhưng liên quan đến nhau.

***2.4.1. Constructor của lớp URL Connection.***

- URLConnection(URL url): Tạo một đối tượng URLConnection từ một đối tượng URL đã cho.

***2.4.1. Các phương thức quan trọng được sử dụng trong lớp URL Connection.***

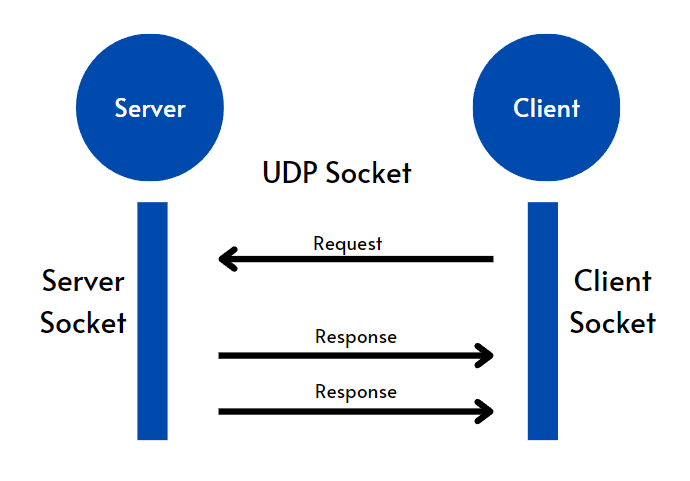
|  |  |
| --- | --- |
| **Phương thức** | **Mô tả** |
| getContent() | Nó lấy nội dung của URL Connection. |
| connect() | Nó mở một liên kết truyền thông tới tài nguyên được tham chiếu bởi URL này, nếu một kết nối như vậy chưa được thiết lập trước đó. |
| getInputStream() | Nó trả về một luồng đầu vào để đọc từ điều kiện mở. |
| getOutputStream() | Nó trả về một luồng đầu ra để ghi vào kết nối. |
| getContentType() | Nó trả về giá trị của trường tiêu đề ngày. |
| getContentLength() | Nó trả về giá trị của trường tiêu đề content-length. |
| getContentEncoding() | Nó trả về giá trị của trường tiêu đề content-encoding. |
| getDate() | Nó trả về giá trị của trường tiêu đề ngày. |
| getExpiration() | Nó trả về giá trị của các tệp tiêu đề hết hạn. |
| getLastModified() | Nó trả về giá trị của trường tiêu đề last-modified. |
| getHeaderField(String name) | Nó trả về giá trị của trường tiêu đề được đặt tên. |
| getHeaderFieldKey(int n) | Nó trả về khóa cho trường tiêu đề thứ n trong danh sách. |
| getHeaderFieldDate(String name, long Default) | Nó trả về giá trị của trường đã được đặt tên được phân tích dưới dạng một số. |
| setDoInput(boolean doinput) | Nó thiết lập giá trị của trường doInput cho URLConnection này thành giá trị được chỉ định. |
| getDoInput() | Nó trả về giá trị của cờ doInput của URLConnection. |
| setDoOutput(boolean dooutput) | Nó thiết lập giá trị của trường doOutput cho URLConnection thành giá trị được chỉ định. |
| getDoOutput() | Nó trả về giá trị của trường doOutput của URLConnection. |
| getIfModifiedSince() | Nó trả về giá trị của trường ifModifiedSince của đối tượng. |
| setIfModifiedSince(long ifModifiedSince) | Nó thiết lập giá trị của trường ifModifiedSince của đối tượng. |
| getUseCaches() | Nó trả về giá trị của trường useCaches của URLConnection. |
| setUseCaches() | Nó thiết lập giá trị của trường useCaches của URLConnection. |

Bảng 2: Bảng các phương thức quan trọng trong lớp URL Connection.

**2.5 Datagrams và Socket của UDP.**

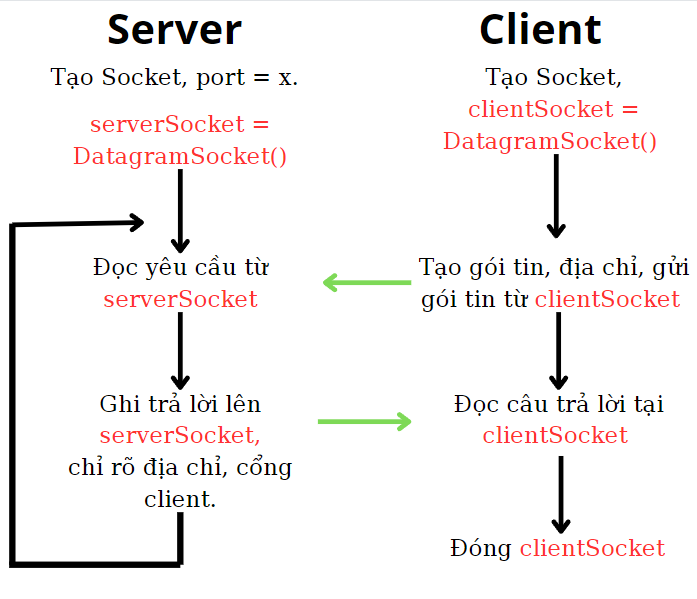
***2.5.1 Tổng quan về UDP.***

- **UDP** là viết tắt của User Datagram Protocol. UDP được sử dụng cho các ứng dụng yêu cầu tốc độ cao và thời gian phản hồi ngắn như phát video, trò chơi trực tuyến, hay dùng để truy vấn hệ thống tên miền (DNS). Giao thức này giúp tăng tốc độ truyền tải vì nó không yêu cầu thiết lập kết nối trước khi truyền thông điệp giữa máy gửi và máy nhận.



Hình 3: Mô hình hoạt động của UDP socket.

- Trước khi tìm hiểu lý thuyết và các phương thức của UDP socket, chúng ta cần phải biết quy trình hoạt động của mỗi bên client và server.



Hình 4: Quy trình hoạt động UDP socket.

***2.5.2 Lớp DatagramPacket.***

***2.5.2.1 Constructor của lớp DatagramPacket.***

- public DatagramPacket(byte[] b, int length, InetAddress dc, int port)

- public DatagramPacket(byte[] b, int offset, int length, InetAddress dc, int port)

- public DatagramPacket(byte[] b, int length, SocketAddress dc, int port)

- public DatagramPacket(byte[] b, int offset, int length, SocketAddress dc, int port)

- Mỗi constructor tạo ra một DatagramPacket mới để được gửi đi tới một host khác.Gói tin được điền đầy dữ liệu với chiều dài là length byte bắt đầu từ vị trí offset hoặc vị trí 0 nếu offset không được sử dụng.

***2.5.2.2 Các phương thức quan trọng được sử dụng trong lớp DatagramPacket.***

|  |  |
| --- | --- |
| **Phương thức** | **Mô tả** |
| getAddress() | Trả về một đối tượng InetAddress chứa địa chỉ IP của host ở xa. |
| getPort() | Trả về một số nguyên xác định cổng trên host ở xa. |
| SocketAddress() | Trả về một đối tượng SocketAddress chứa địa chỉ IP và số hiệu cổng của host ở xa. |
| getData() | Trả về một mảng byte chứa dữ liệu từ datagram. |
| getLength() | Trả về số bytes dữ liệu có trong một datagram. |
| getOffset() | Trả về vị trí trong mảng được trả về bởi phương thức getData() mà từ đó dữ liệu trong datagram xuất phát. |
| setData(byte[] b) | Thay đổi dữ liệu của datagram. |
| setData(byte[] b, int offset, int length) | Gửi một khối lượng dữ liệu lớn |
| setAddress(InetAddress dc) | Thay đổi địa chỉ của máy mà ta sẽ gửi gói tin tới. |
| setPort(int port) | Thay đổi số hiệu cổng gửi tới của gói tin. |
| setLength(int length) | Thay đổi số byte dữ liệu có thể đặt trong vùng đệm. |

Bảng 3: Bảng các phương thức được quan trọng trong lớp DatagramPacket.

**2.6 Lý thuyết cho Socket của TCP.**

***2.6.1 Tổng quan về TCP socket.***

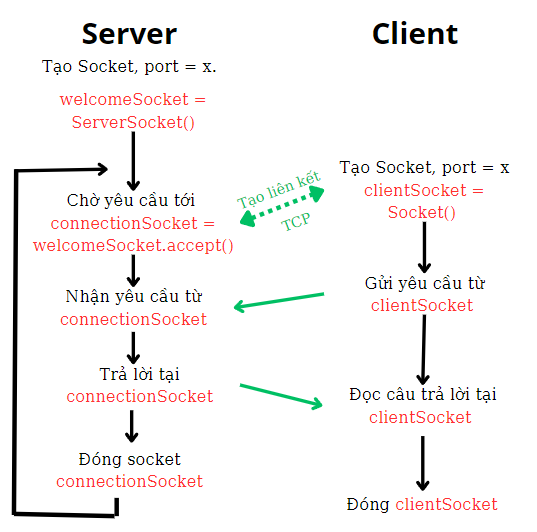
- Theo đúng bản chất của TCP, cần có một kết nối hai chiều trước khi server và client có thể trao đổi thông điệp với nhau.

- Đầu tiên, server tạo một Socket gắn với một cổng (port number) và chờ nhận yêu cầu từ client.

- Tiếp đó, client gửi yêu cầu tới server bằng cách tạo một Socket TCP, kèm theo địa chỉ IP và port number của tiến trình tương ứng trên máy chủ. Khi client tạo Socket, TCP của client thiết lập kết nối với TCP của server và chờ server chấp nhận kết nối.

- TCP cung cấp dịch vụ truyền dữ liệu tin cậy và có thứ tự giữa client và server. Giữa máy chủ và máy nhận chỉ có một địa chỉ IP duy nhất. Hơn nữa, mỗi thông điệp được truyền đi đều có xác nhận trả về.

- Sau đây là quy trình của hoạt động của TCP:



Hình 5: Quy trình hoạt động TCP socket.

***2.6.2 Lớp ServerSocket.***

***2.6.2.1 Constructor của lớp ServerSocket*.**

- Public ServerSocket(int port) throws BindException, IOException

- Public ServerSocket(int port, int queueLength) throws IOException

- public ServerSocket( ) throws IOException

***2.6.2.2 Chấp nhận và ngắt kết nối.***

- public Socket accept() throws IOException

- Khi quá trình thiết lập kết nối hoàn tất và chúng ta đã sẵn sàng chấp nhận kết nối, cần gọi phương thức accept() của lớp ServerSocket. Phương thức này sẽ chặn lại; nó dừng quá trình xử lý và đợi cho đến khi có một client kết nối. Khi client thực sự kết nối, phương thức accept() sẽ trả về một đối tượng Socket. Chúng ta sử dụng các phương thức getInputStream() và getOutputStream() của đối tượng Socket này để truyền dữ liệu với client.

- public void close( ) throws IOException

- Khi đã hoàn thành công việc với một ServerSocket, chúng ta cần phải đóng nó lại, đặc biệt nếu chương trình của chúng ta vẫn tiếp tục chạy. Điều này giúp các chương trình khác có thể sử dụng cổng đó. Đóng một ServerSocket không giống với việc đóng một Socket.

***2.6.2.3 Các phương thức quan trọng được sử dụng trong lớp ServerSocket.***

|  |  |
| --- | --- |
| **Phương thức** | **Mô tả** |
| getInetAddress() | Trả về địa chỉ được sử dụng bởi server (localhost) |
| getLocalHost() | Phương thức này cho phép ta tìm ra cổng mà server đang nghe |

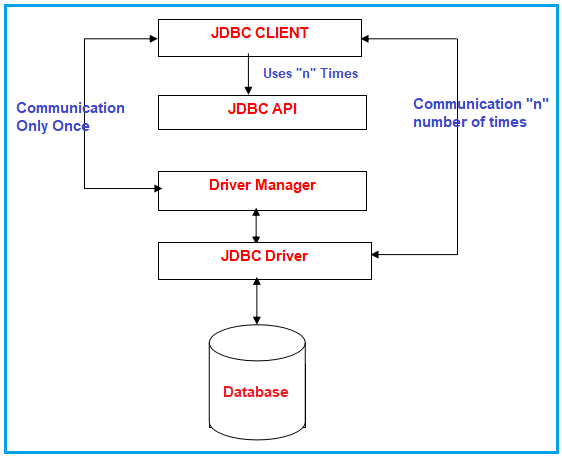
Bảng 4: Bảng các phương thức được quan trọng trong lớp ServerSocket.

**2.7 JDBC cơ bản.**

***2.7.1 Tổng quan về JDBC.***

- JDBC (Java Database Connectivity) là một API (Application Programming Interface) trong Java, cung cấp các phương thức và chuẩn mực để kết nối và tương tác với các cơ sở dữ liệu. JDBC cho phép các ứng dụng Java thực hiện các thao tác như gửi các câu lệnh SQL, truy vấn dữ liệu, và xử lý kết quả truy vấn từ các cơ sở dữ liệu. Nó hỗ trợ nhiều loại cơ sở dữ liệu khác nhau như MySQL, PostgreSQL, Oracle, SQL Server và nhiều hệ quản trị cơ sở dữ liệu khác.

***2.7.2 Kiến trúc JDBC.***



Hình 6: Kiến trúc JDBC.

- JDBC API: Cho phép chương trình Java thực thi các câu lệnh SQL và truy xuất kết quả.

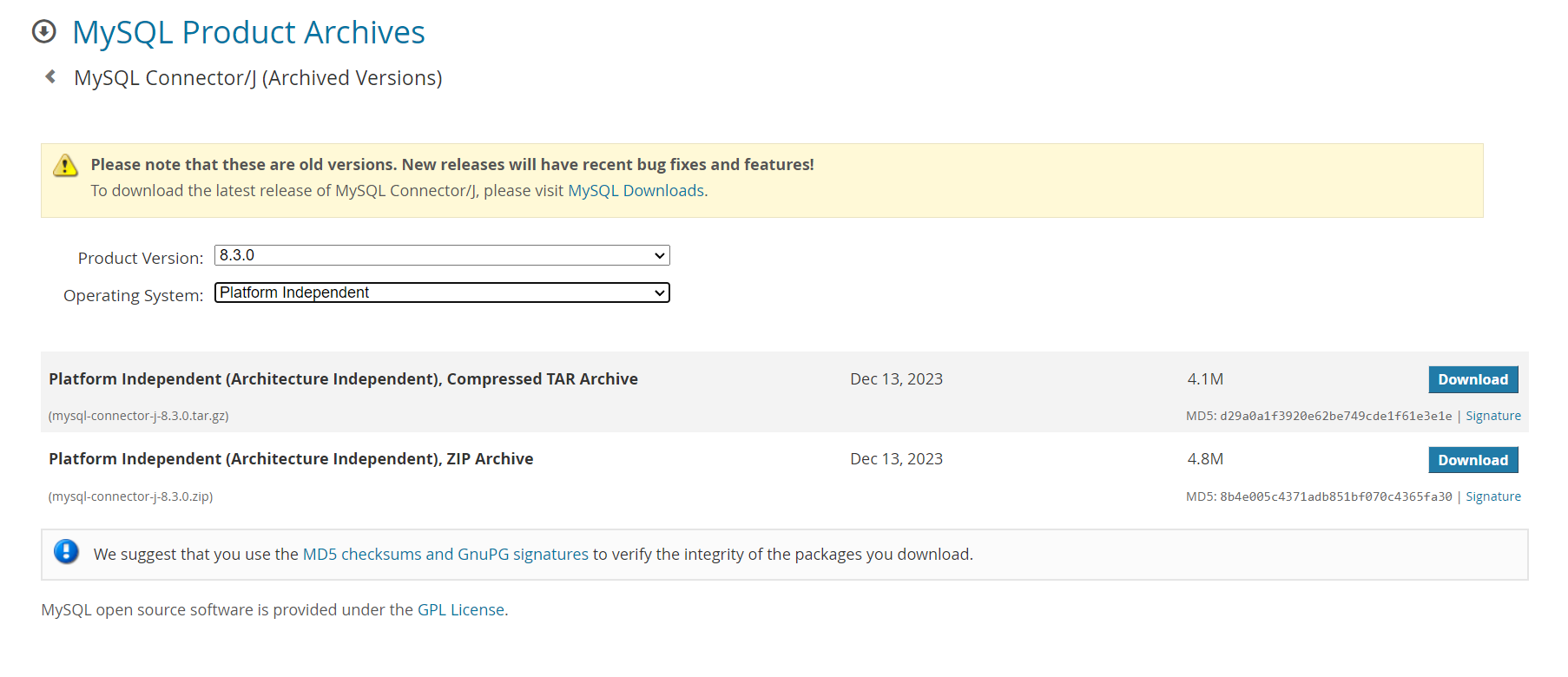
- Driver Manager: Đóng vai trò quan trọng trong kiến trúc JDBC. Nó sử dụng một số quy trình điều khiển dành riêng cho cơ sở dữ liệu để kết nối hiệu quả các ứng dụng doanh nghiệp với database

- JDBC Driver: Để giao tiếp với nguồn dữ liệu thông qua JDBC, bạn cần JDBC Driver để giao tiếp với nguồn dữ liệu tương ứng.

***2.7.3 Các bước kết nối cơ sở dữ liệu (MySQL Workbench 8.0 CE) với dự án Java trên Eclipse.***

**Bước 1:** Cài đặt MySQL Connector/J.

- Truy cập trang web MySQL và tải xuống MySQL Connector/J phù hợp với phiên bản MySQL của bạn.



Hình 7: Tải xuống MySQL Connector/J.

- Giải nén file tải về và lưu trữ thư viện mysql-connector-java-x.x.x.jar ở một vị trí dễ tìm.

**Bước 2:** Thêm MySQL Connector/J vào dự án Eclipse.

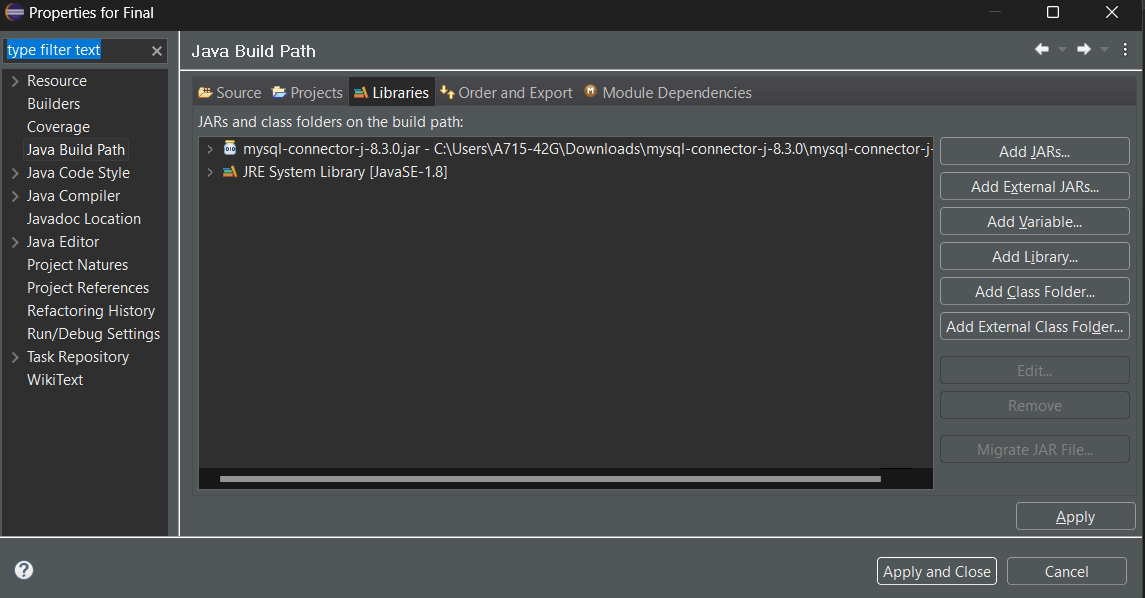
- Mở Eclipse và dự án Java của bạn.

- Nhấp chuột phải vào dự án, chọn **Build Path** > **Configure Build Path**.

- Chọn **Libraries** và nhấp vào **Add External JARs**.

- Duyệt và chọn file mysql-connector-java-x.x.x.jar vừa tải về và nhấp Open.

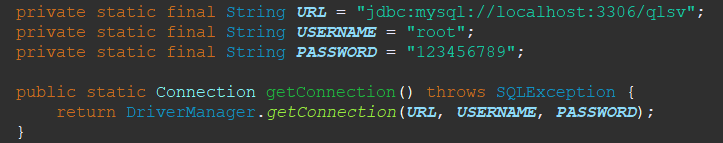
- Nhấp Apply and Close để thêm thư viện vào dự án.



Hình 8: Giao diện sau khi thêm thành công MYSQL Connector/J.

**Bước 3:** Tạo cơ sở dữ liệu hoặc thêm cơ sở dữ liệu có sẵn.

**Bước 4:** Viết mã Java để kết nối với cơ sở dữ liệu.



Hình 9: Ví dụ mã cho kết nối với cơ sở dữ liệu qlsv.

- API: “jdbc”.

- Hệ cơ sở dữ liệu: “mysql”.

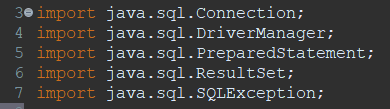
- Tên server đang chạy: “localhost”.

- Cổng kết nối: 3306.

- Tên cơ sở dữ liệu: “qlsv”.

- Tên đăng nhập: “root”.

- Mật khẩu: “123456789”.

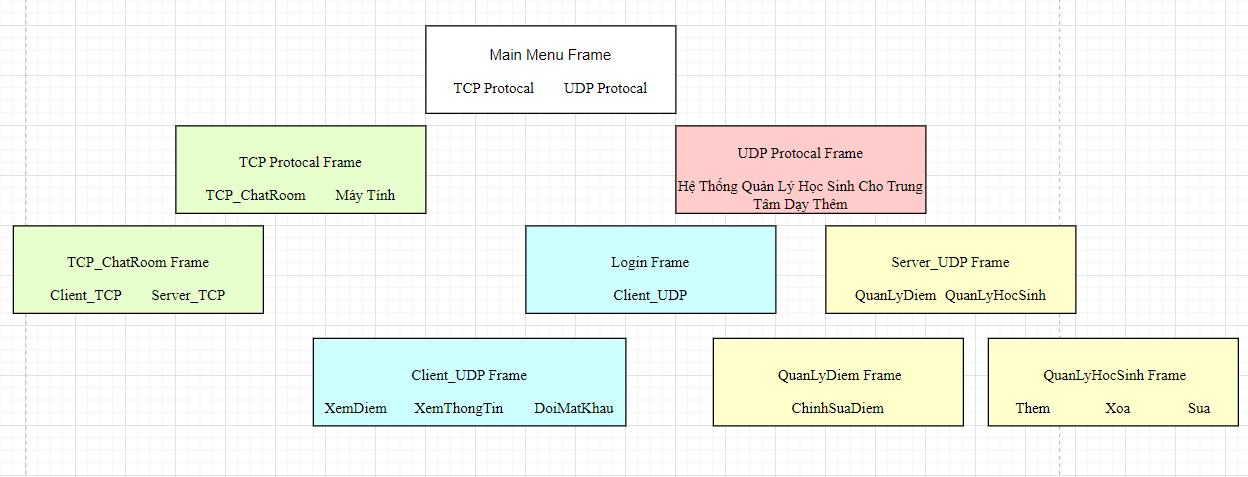


Hình 10: Các thư viện danh cho JDBC.

CHƯƠNG 3 – THIẾT KẾ.

1. Mô hình đề suất
2. Mô hình hệ thống tổng quan

3.1.1.1 B**iểu đồ tổng quát**

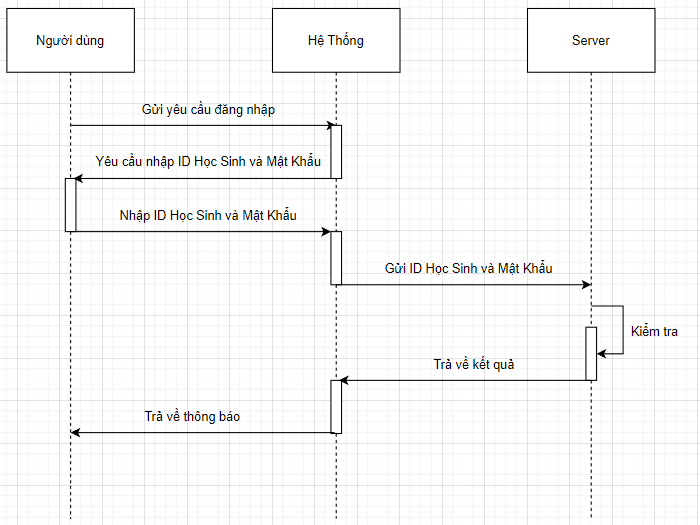


Hình 11: Biểu đồ tổng quát của đề tài.

3.1.1.2 B**iểu đồ hoạt động từng chức năng.**

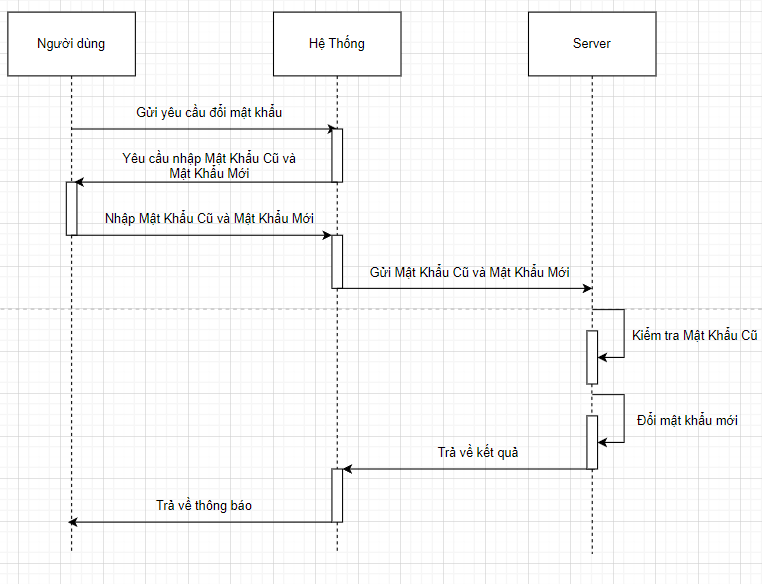
Em sẽ sử dụng biếu đồ tuần tự để biểu diễn các chức năng trong chương trình.

* Đăng nhập:



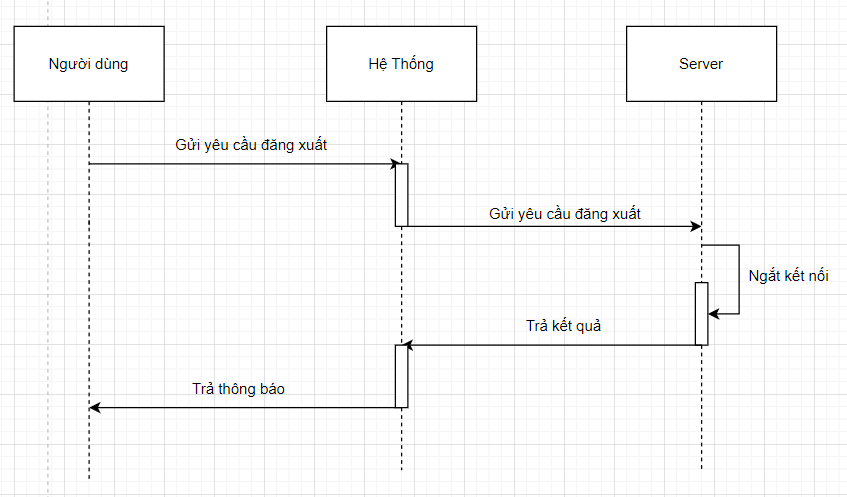
Hình 12: Biểu đồ chức năng đăng nhập.

* Đổi mật khẩu:



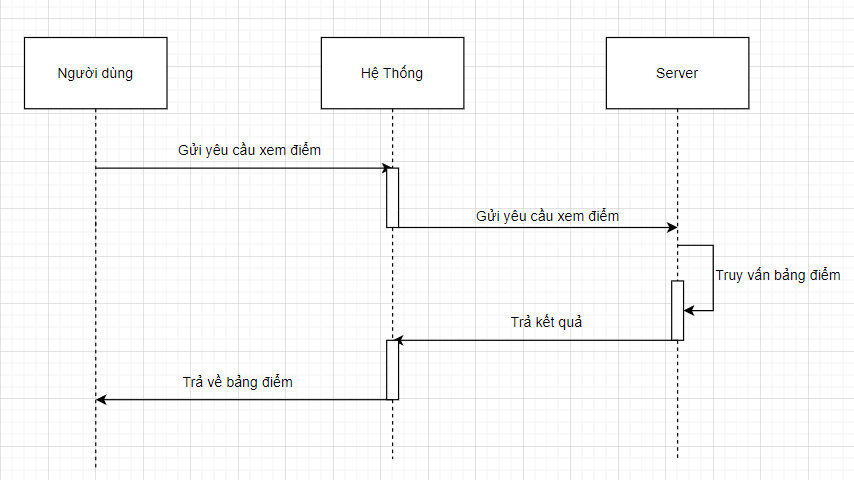
Hình 13: Biểu đồ chức năng đổi mật khẩu.

* Đăng xuất:



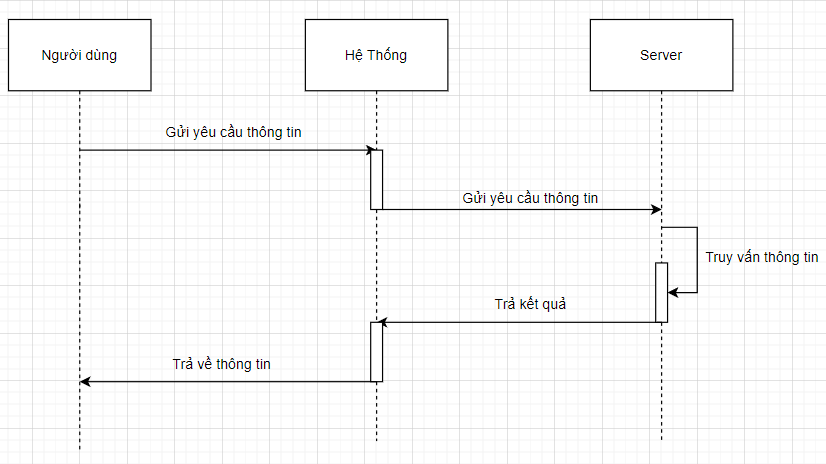
Hình 14: Biểu đồ chức năng đăng xuất.

* Xem điểm:



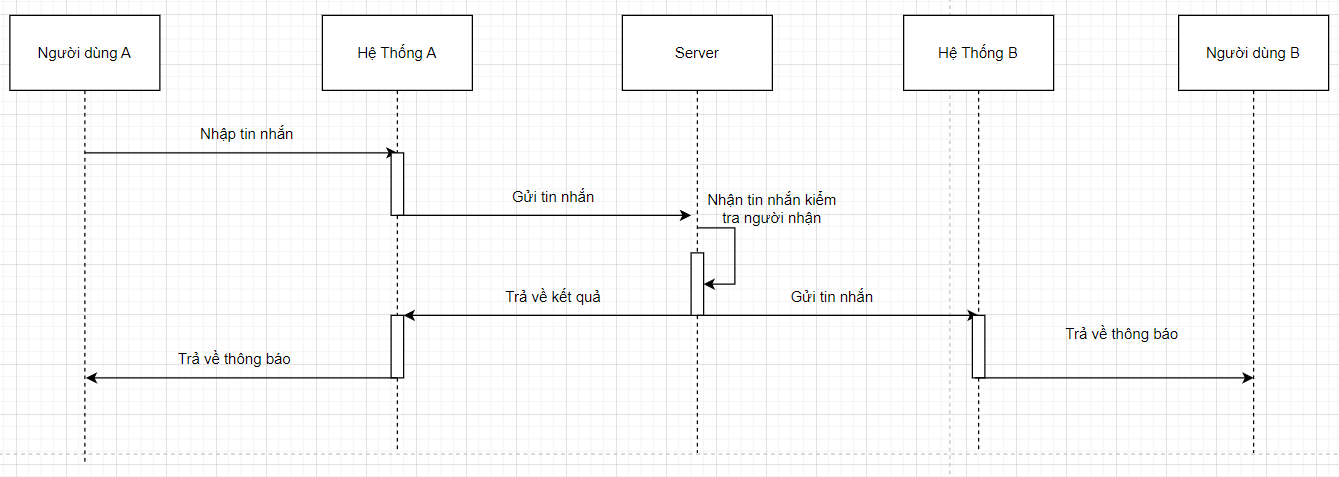
Hình 15: Biểu đồ chức năng xem điểm.

* Xem thông tin:



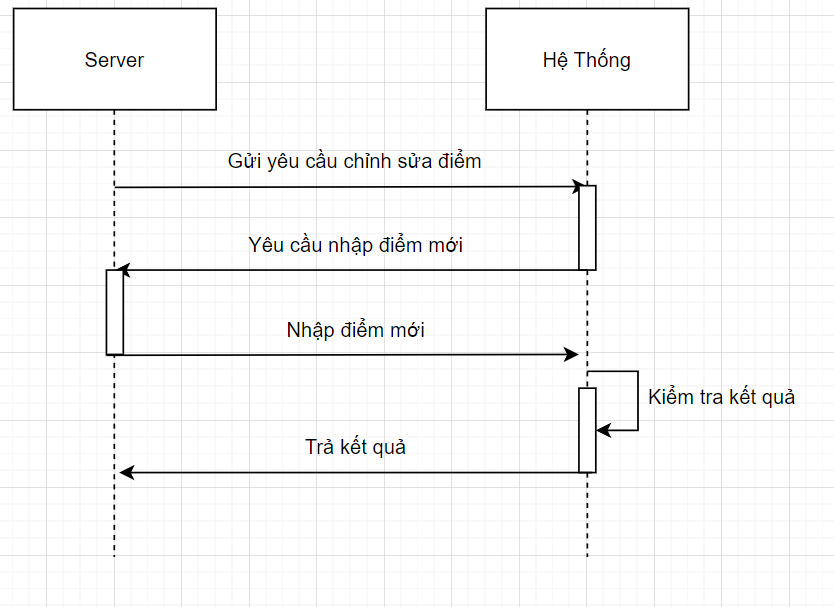
Hình 16: Biểu đồ chức năng xem thông tin.

* Gửi tin nhắn:



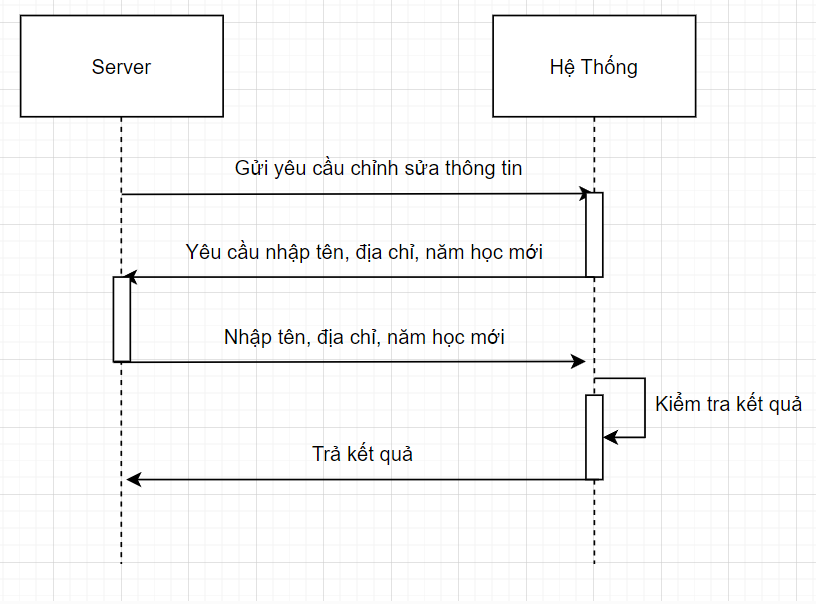
Hình 17: Biểu đồ chức năng gửi tin nhắn.

* Chỉnh sửa điểm:



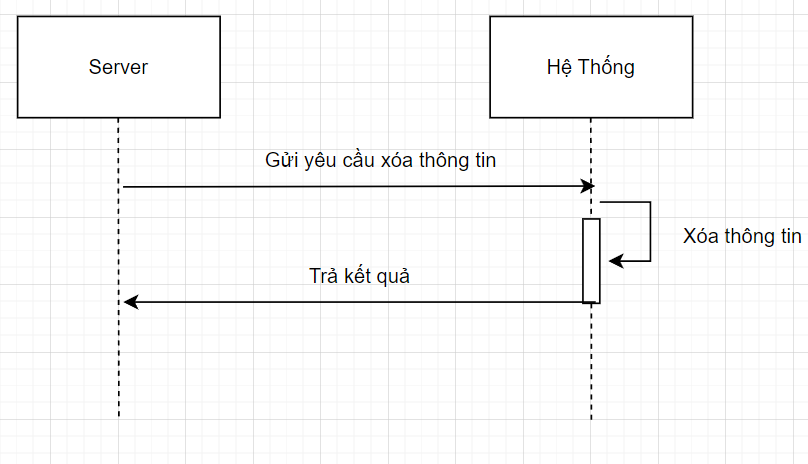
Hình 18: Biểu đồ chức năng sửa điểm.

* Chỉnh sửa thông tin:



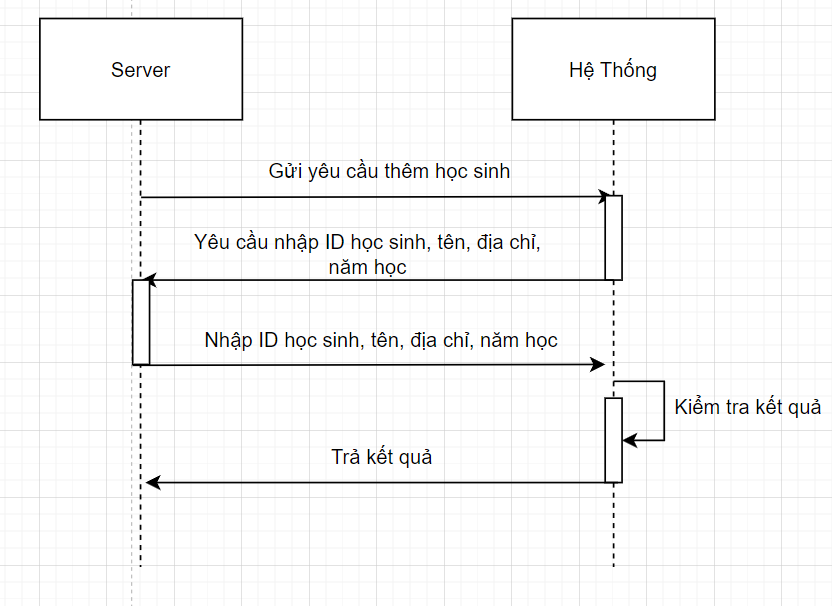
Hình 19: Biểu đồ chức năng sửa thông tin

* Xóa thông tin:



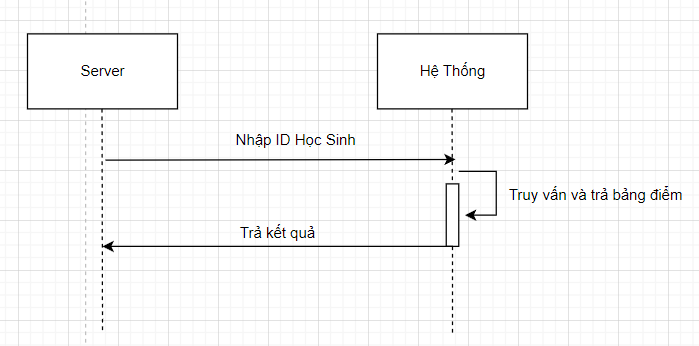
Hình 20: Biểu đồ chức năng xóa thông tin.

* Thêm học sinh:



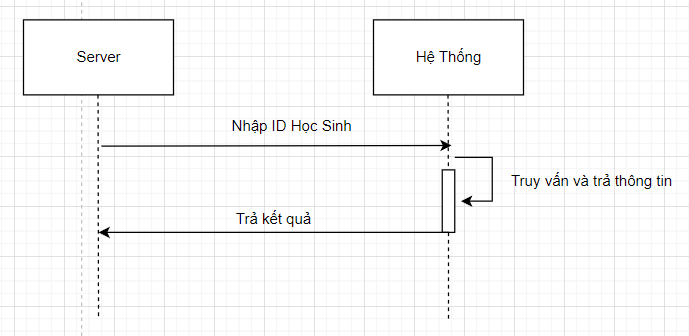
Hình 21: Biểu đồ chức năng thêm học sinh.

* Tra cứu điểm theo ID học sinh:



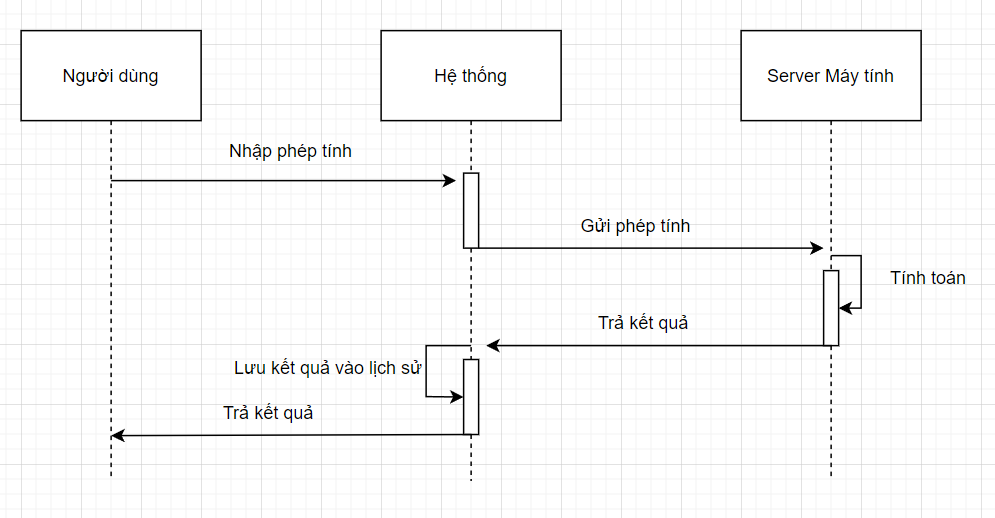
Hình 22: Biểu đồ chức năng tra cứu điểm theo ID Học Sinh.

* Tra cứu thông tin theo ID học sinh:



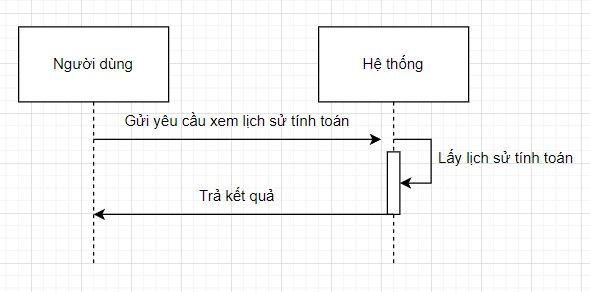
Hình 23: Biểu đồ chức năng tra cứu thông tin theo ID Học Sinh.

* Tính toán:



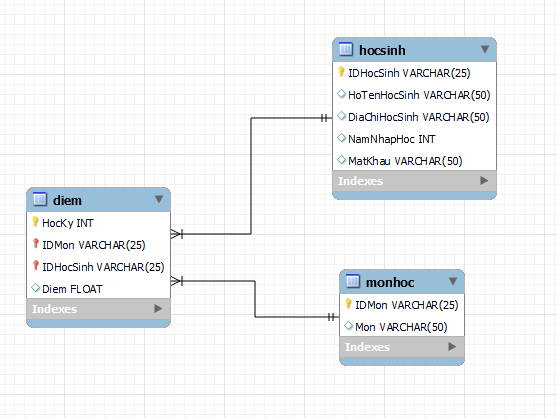
Hình 24: Biểu đồ chức năng tính toán của máy tính.

* Lịch sử tính toán:



Hình 25: Biểu đồ chức năng lịch sử tính toán.

3.1.1.3 Mô hình thực thể liên kết cơ sở dữ liệu quản lý sinh viên**.**



Hình 26: Mô hình thực thể kiên kết.

3.1.2 Xây dựng chương trình

- Đầu tiên em sẽ xây dựng ba lớp đối tượng là Diem, HocSinh và MonHoc. Tất cả các lớp đều được thêm đầy đủ các phương thức constructors, getters và setters.

Diem.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50 | package OOP;  public class Diem {  private int hocKy;  private String idMon;  private String idHocSinh;  private double diem;  public Diem() {  }  public Diem(int hocKy, String idMon, String idHocSinh, double diem) {  this.hocKy = hocKy;  this.idMon = idMon;  this.idHocSinh = idHocSinh;  this.diem = diem;  }  // Getters and setters  public int getHocKy() {  return hocKy;  }  public void setHocKy(int hocKy) {  this.hocKy = hocKy;  }  public String getIdMon() {  return idMon;  }  public void setIdMon(String idMon) {  this.idMon = idMon;  }  public String getIdHocSinh() {  return idHocSinh;  }  public void setIdHocSinh(String idHocSinh) {  this.idHocSinh = idHocSinh;  }  public double getDiem() {  return diem;  }  public void setDiem(double diem) {  this.diem = diem;  }  } |

HocSinh.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62 | package OOP;  public class HocSinh {  private String IDHocSinh;  private String HoTenHocSinh;  private String DiaChiHocSinh;  private int NamNhapHoc;  private String MatKhau;  // Constructors  public HocSinh(String IDHocSinh, String HoTenHocSinh, String DiaChiHocSinh, int NamNhapHoc, String MatKhau) {  this.IDHocSinh = IDHocSinh;  this.HoTenHocSinh = HoTenHocSinh;  this.DiaChiHocSinh = DiaChiHocSinh;  this.NamNhapHoc = NamNhapHoc;  this.MatKhau = MatKhau;  }    // Getters and setters  public String getIDHocSinh() {  return IDHocSinh;  }  public void setIDHocSinh(String IDHocSinh) {  this.IDHocSinh = IDHocSinh;  }  public String getHoTenHocSinh() {  return HoTenHocSinh;  }  public void setHoTenHocSinh(String HoTenHocSinh) {  this.HoTenHocSinh = HoTenHocSinh;  }  public String getDiaChiHocSinh() {  return DiaChiHocSinh;  }  public void setDiaChiHocSinh(String DiaChiHocSinh) {  this.DiaChiHocSinh = DiaChiHocSinh;  }  public int getNamNhapHoc() {  return NamNhapHoc;  }  public void setNamNhapHoc(int NamNhapHoc) {  this.NamNhapHoc = NamNhapHoc;  }  public String getMatKhau() {  return MatKhau;  }  public void setMatKhau(String MatKhau) {  this.MatKhau = MatKhau;  }    } |

MonHoc.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27 | package OOP;  public class MonHoc {  private String IDMon;  private String mon;  public MonHoc(String IDMon, String mon) {  this.IDMon = IDMon;  this.mon = mon;  }  // Getters và setters  public String getIDMon() {  return IDMon;  }  public void setIDMon(String IDMon) {  this.IDMon = IDMon;  }  public String getMon() {  return mon;  }  public void setMon(String mon) {  this.mon = mon;  }  } |

- Tiếp theo em sẽ xây dựng ba file DataAccessObject (DAO) cho việc triển khai truy cập nguồn dữ liệu cho các chương trình. Ba file bao gồm DiemDAO, HocSingDAO, MonHocDAO.

DiemDAO.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146 | package DAO;  import java.sql.Connection;import java.sql.DriverManager;import java.sql.PreparedStatement;import java.sql.ResultSet;import java.sql.SQLException;import java.util.HashMap;import java.util.Map;  import OOP.Diem;  public class DiemDAO {  private static final String URL = "jdbc:mysql://localhost:3306/qlsv";  private static final String USERNAME = "root";  private static final String PASSWORD = "123456789";  public static Connection getConnection() throws SQLException {  return DriverManager.getConnection(URL, USERNAME, PASSWORD);  }  public static Diem getByID(int hocKy, String idMon, String idHocSinh) {  String sql = "SELECT \* FROM Diem WHERE HocKy = ? AND IDMon = ? AND IDHocSinh = ?";  try (Connection conn = getConnection();  PreparedStatement stmt = conn.prepareStatement(sql)) {  stmt.setInt(1, hocKy);  stmt.setString(2, idMon);  stmt.setString(3, idHocSinh);  try (ResultSet rs = stmt.executeQuery()) {  if (rs.next()) {  return new Diem(  rs.getInt("HocKy"),  rs.getString("IDMon"),  rs.getString("IDHocSinh"),  rs.getDouble("Diem")  );  }  }  } catch (SQLException e) {  e.printStackTrace();  }  return null;  }    private static String getAllDiem() {  String sql = "SELECT d.HocKy, d.IDMon, m.Mon, d.IDHocSinh, d.Diem, h.HoTenHocSinh " +  "FROM Diem d " +  "JOIN MonHoc m ON d.IDMon = m.IDMon " +  "JOIN HocSinh h ON d.IDHocSinh = h.IDHocSinh";  String result = "";  try (Connection conn = getConnection();  PreparedStatement stmt = conn.prepareStatement(sql);  ResultSet rs = stmt.executeQuery()) {  while (rs.next()) {  result += rs.getInt("HocKy") + "," + rs.getString("IDMon") + "," +rs.getString("Mon") + "," +rs.getString("IDHocSinh") + "," + rs.getString("HoTenHocSinh") + "," + rs.getDouble("Diem") + "\n";  }  } catch (SQLException e) {  e.printStackTrace();  }  return result;  }    public static String BangDiemAll() {  String result = "";  String getAllDiem1 = getAllDiem();  String[] lines = getAllDiem1.split("\n");  String[][] temp = new String[lines.length][];  for (int i = 0; i < lines.length; i++) {  temp[i] = lines[i].split(",");  }  Map<String, Map<String, Double>> hocSinhDiemMap = new HashMap<>();  for (String[] item : temp) {  String hocSinh = item[4];  String monHoc = item[2];  Double diem = Double.parseDouble(item[5]);  if (!hocSinhDiemMap.containsKey(hocSinh)) {  hocSinhDiemMap.put(hocSinh, new HashMap<String, Double>());  }  hocSinhDiemMap.get(hocSinh).put(monHoc, diem);  }  for (Map.Entry<String, Map<String, Double>> entry : hocSinhDiemMap.entrySet()) {  String hocSinh = entry.getKey();  Map<String, Double> diemMap = entry.getValue();  result += "================================\n";  result += "Họ và tên: " + hocSinh + "\n";  for (Map.Entry<String, Double> diemEntry : diemMap.entrySet()) {  String monHoc = diemEntry.getKey();  Double diem = diemEntry.getValue();  result += monHoc + ": " + diem + "\n";  }  result += "\n";  }  return result;  }  public static String TimDiemTheoIDHocSinh(String IDHocSinh) {  String getAllDiem1 = getAllDiem();  String[] lines = getAllDiem1.split("\n");  String[][] temp = new String[lines.length][];  boolean found = false;  String result="";    for (int i = 0; i < lines.length; i++) {  temp[i] = lines[i].split(",");  }  for (int i = 0; i < temp.length; i++) {  if (temp[i][3].equals(IDHocSinh)) {  if (!found) {  result += "================================\n";  result += "Họ và tên: " + temp[i][4]+"\n";  found = true;  }  result += temp[i][2] + ": " + temp[i][5]+"\n";  }  }  return result;  }  public static boolean updateDiem(String hocSinhID, String monHocID, double diem) {  String sql = "UPDATE Diem SET Diem = ? WHERE IDHocSinh = ? AND IDMon = ?";  try (Connection conn = getConnection();  PreparedStatement stmt = conn.prepareStatement(sql)) {  stmt.setDouble(1, diem);  stmt.setString(2, hocSinhID);  stmt.setString(3, monHocID);    int rowsAffected = stmt.executeUpdate();  if (rowsAffected > 0) {  return true;  } else {  return false;  }  } catch (SQLException e) {  e.printStackTrace();  }  return false;  }  } |

Phương thức **getAllDiem()** trả về toàn bộ danh sách điểm của tất cả các học sinh.

Phương thức **BangDiemAll()** trả về toàn bộ danh sách điểm của tất cả các học sinh nhưng được chỉnh sửa lại phù hợp mới giao diện người dùng hơn.

Phương thức **TimDiemTheoIDHocSinh(String IDHocSinh)** trả về chuỗi thông tin như họ tên và điểm các môn của học sinh được tìm.

Phương thức **updateDiem(String hocSinhID, String monHocID, double diem**) kiểu dữ liệu boolean. Nếu đúng thì đã chỉnh sửa điểm thành công, nếu sai thì chỉnh sửa điểm thất bại.

Phương thức **getConnection() throws SQLException** trả về đối tượng Connection và phương thức này dùng để kết nối hệ cơ sở dữ liệu.

HocSingDAO.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149  150  151  152  153  154  155  156  157  158  159  160  161  162  163  164  165  166  167  168  169  170  171  172  173  174  175  176  177  178  179  180  181  182  183  184  185  186  187  188  189  190  191  192  193  194  195  196  197  198  199  200  201  202  203  204  205  206  207  208  209  210  211  212  213  214  215  216  217  218  219  220  221  222 | package DAO;import OOP.HocSinh;import java.sql.Connection;import java.sql.DriverManager;import java.sql.PreparedStatement;import java.sql.ResultSet;import java.sql.SQLException;import java.util.Arrays;  public class HocSinhDAO {  private static final String URL = "jdbc:mysql://localhost:3306/qlsv";  private static final String USERNAME = "root";  private static final String PASSWORD = "123456789";  public static Connection getConnection() throws SQLException {  return DriverManager.getConnection(URL, USERNAME, PASSWORD);  }  public static HocSinh getByID(String ID) {  String sql = "SELECT \* FROM HocSinh WHERE IDHocSinh = ?";  try (Connection conn = getConnection();  PreparedStatement stmt = conn.prepareStatement(sql)) {  stmt.setString(1, ID);  try (ResultSet rs = stmt.executeQuery()) {  if (rs.next()) {  return new HocSinh(  rs.getString("IDHocSinh"),  rs.getString("HoTenHocSinh"),  rs.getString("DiaChiHocSinh"),  rs.getInt("NamNhapHoc"),  rs.getString("MatKhau")  );  }  }  } catch (SQLException e) {  e.printStackTrace();  }  return null;  }  public static String printAll() {  String sql = "SELECT \* FROM HocSinh";  String result = "";  try (Connection conn = getConnection();  PreparedStatement stmt = conn.prepareStatement(sql);  ResultSet rs = stmt.executeQuery()) {  result += "Danh sách các học sinh:\n";  while (rs.next()) {  result += "\nID Học Sinh: " + rs.getString("IDHocSinh");  result += "\nHọ và Tên: " + rs.getString("HoTenHocSinh");  result += "\nĐịa chỉ: " + rs.getString("DiaChiHocSinh");  result += "\nNăm nhập học: " + rs.getInt("NamNhapHoc");  result += "\n--------------------";  }  } catch (SQLException e) {  e.printStackTrace();  }  return result;  }  public static String TimHocSinhTheoIDHocSinh(String ID) {  String result = "";  HocSinh hocSinh = getByID(ID);  if (hocSinh != null) {  result += "ID Học Sinh: " + hocSinh.getIDHocSinh() + "\n";  result += "Họ và Tên: " + hocSinh.getHoTenHocSinh() + "\n";  result += "Địa Chỉ: " + hocSinh.getDiaChiHocSinh() + "\n";  result += "Năm Nhập Học: " + hocSinh.getNamNhapHoc() + "\n";  } else {  result = "Không tìm thấy học sinh có ID " + ID;  }  return result;  }  public static boolean themHocSinh(String IDHocSinh, String HoTen, String DiaChi, String NamHoc, String MatKhau) {  int namHocInt = Integer.parseInt(NamHoc);  String sqlThemHocSinh = "INSERT INTO HocSinh (IDHocSinh, HoTenHocSinh, DiaChiHocSinh, NamNhapHoc, MatKhau) VALUES (?, ?, ?, ?, ?)";  String sqlThemDiemMacDinh = "INSERT INTO Diem (HocKy, IDMon, IDHocSinh, Diem) VALUES (?, ?, ?, ?)";  try (Connection conn = getConnection();  PreparedStatement stmtThemHocSinh = conn.prepareStatement(sqlThemHocSinh);  PreparedStatement stmtThemDiem = conn.prepareStatement(sqlThemDiemMacDinh)) {  stmtThemHocSinh.setString(1, IDHocSinh);  stmtThemHocSinh.setString(2, HoTen);  stmtThemHocSinh.setString(3, DiaChi);  stmtThemHocSinh.setInt(4, namHocInt);  stmtThemHocSinh.setString(5, MatKhau);  int rowsAffectedHocSinh = stmtThemHocSinh.executeUpdate();  int hocKyMacDinh = 1;  String[] monHocMacDinh = {"MH001", "MH002", "MH003"};  for (String monHoc : monHocMacDinh) {  stmtThemDiem.setInt(1, hocKyMacDinh);  stmtThemDiem.setString(2, monHoc);  stmtThemDiem.setString(3, IDHocSinh);  stmtThemDiem.setDouble(4, 0);  stmtThemDiem.addBatch();  }  int[] rowsAffectedDiem = stmtThemDiem.executeBatch();  boolean hocSinhThemThanhCong = rowsAffectedHocSinh > 0;  boolean diemThemThanhCong = Arrays.stream(rowsAffectedDiem).allMatch(count -> count == 1);    return hocSinhThemThanhCong && diemThemThanhCong;  } catch (SQLException e) {  e.printStackTrace();  return false;  }  }  public static boolean xoaHocSinh(String IDHocSinh) {  String sqlDiem = "DELETE FROM Diem WHERE IDHocSinh = ?";  String sqlHocSinh = "DELETE FROM HocSinh WHERE IDHocSinh = ?";  try (Connection conn = getConnection();  PreparedStatement stmtDiem = conn.prepareStatement(sqlDiem);  PreparedStatement stmtHocSinh = conn.prepareStatement(sqlHocSinh)) {  conn.setAutoCommit(false);  stmtDiem.setString(1, IDHocSinh);  int rowsAffectedDiem = stmtDiem.executeUpdate();  stmtHocSinh.setString(1, IDHocSinh);  int rowsAffectedHocSinh = stmtHocSinh.executeUpdate();  if (rowsAffectedDiem > 0 && rowsAffectedHocSinh > 0) {  conn.commit();  return true;  } else {  conn.rollback();  return false;  }  } catch (SQLException e) {  e.printStackTrace();  return false;  }  }  public static boolean editHoTenHocSinh(String IDHocSinh, String HoTenMoi) {  String sql = "UPDATE HocSinh SET HoTenHocSinh = ? WHERE IDHocSinh = ?";  try (Connection conn = getConnection();  PreparedStatement stmt = conn.prepareStatement(sql)) {  stmt.setString(1, HoTenMoi);  stmt.setString(2, IDHocSinh);  int rowsAffected = stmt.executeUpdate();  return rowsAffected > 0;  } catch (SQLException e) {  e.printStackTrace();  return false;  }  }  public static boolean editDiaChiHocSinh(String IDHocSinh, String DiaChiMoi) {  String sql = "UPDATE HocSinh SET DiaChiHocSinh = ? WHERE IDHocSinh = ?";  try (Connection conn = getConnection();  PreparedStatement stmt = conn.prepareStatement(sql)) {  stmt.setString(1, DiaChiMoi);  stmt.setString(2, IDHocSinh);  int rowsAffected = stmt.executeUpdate();  return rowsAffected > 0;  } catch (SQLException e) {  e.printStackTrace();  return false;  }  }  public static boolean editNamHocHocSinh(String IDHocSinh, int NamHocMoi) {  String sql = "UPDATE HocSinh SET NamNhapHoc = ? WHERE IDHocSinh = ?";  try (Connection conn = getConnection();  PreparedStatement stmt = conn.prepareStatement(sql)) {  stmt.setInt(1, NamHocMoi);  stmt.setString(2, IDHocSinh);  int rowsAffected = stmt.executeUpdate();  return rowsAffected > 0;  } catch (SQLException e) {  e.printStackTrace();  return false;  }  }  public static boolean checkLogin(String IDHocSinh, String password) {  String sql = "SELECT \* FROM HocSinh WHERE IDHocSinh = ? AND MatKhau = ?";  try (Connection conn = getConnection();  PreparedStatement stmt = conn.prepareStatement(sql)) {  stmt.setString(1, IDHocSinh);  stmt.setString(2, password);  try (ResultSet rs = stmt.executeQuery()) {  return rs.next();  }  } catch (SQLException e) {  e.printStackTrace();  return false;  }  }  public static boolean changePass(String IDHocSinh, String matkhaucu, String matkhaumoi) {  if (!checkLogin(IDHocSinh, matkhaucu)) {  System.out.println("Mật khẩu cũ không chính xác.");  return false;  }  String sql = "UPDATE HocSinh SET MatKhau = ? WHERE IDHocSinh = ?";  try (Connection conn = getConnection();  PreparedStatement stmt = conn.prepareStatement(sql)) {  stmt.setString(1, matkhaumoi);  stmt.setString(2, IDHocSinh);  int rowsAffected = stmt.executeUpdate();  if (rowsAffected > 0) {  return true;  } else {  return false;  }  } catch (SQLException e) {  e.printStackTrace();  return false;  }  }  } |

Phương thức **printAll()** trả về chuỗi tất cả các thông tin học sinh.

Phương thức **TimHocSinhTheoIDHocSinh(String ID)** trả về chuỗi các thông tin như ID Học sinh, họ tên, địa chỉ, năm nhập học của học sinh được tìm.

Phương thức **themHocSinh(String IDHocSinh, String HoTen, String DiaChi, String NamHoc, String MatKhau)** trả về kiểu dữ liệu boolean, nếu đúng thì thêm học sinh thành côngm nếu sai thì thêm học sinh thất bại. Phương thức không chỉ thêm học sinh với các giá trị truyền vào mà còn thêm mặc định các môn học và điểm mặc định là 0 điểm.

Phương thức **xoaHocSinh(String IDHocSinh)** trả về kiểu dữ liệu boolean, nếu đúng thì xóa thành công, sai thì xóa thất bại. Phương thức không chỉ xóa các thông tin học sinh mà còn xóa luôn các điểm của các môn.

Phương thức **editHoTenHocSinh(String IDHocSinh, String HoTenMoi)** trả về kiểu dữ liệu boolean, nếu đúng thì chỉnh sửa tên thành công, nếu sai thì chỉnh sủa tên thất bại.

Phương thức **editDiaChiHocSinh(String IDHocSinh, String DiaChiMoi)** trả về kiểu dữ liệu boolean, nếu đúng thì chỉnh sửa địa chỉ thành công, nếu sai thì chỉnh sủa địa chỉ thất bại.

Phương thức **editNamHocHocSinh(String IDHocSinh, int NamHocMoi)** trả về kiểu dữ liệu boolean, nếu đúng thì chỉnh sửa năm nhập học thành công, nếu sai thì chỉnh sủa năm nhập học thất bại.

Phương thức **changePass(String IDHocSinh, String matkhaucu, String matkhaumoi)** trả về kiểu dữ liệu boolean, nếu đúng thì đổi mật khẩu thành công, nếu sai thì thông báo sai mật khẩu.

Phương thức

Phương thức **getConnection() throws SQLException** trả về đối tượng Connection và phương thức này dùng để kết nối hệ cơ sở dữ liệu.

MonHocDAO.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41 | package DAO;  import java.sql.Connection;import java.sql.DriverManager;import java.sql.PreparedStatement;import java.sql.ResultSet;import java.sql.SQLException;import OOP.MonHoc;  public class MonHocDAO {  private static final String URL = "jdbc:mysql://localhost:3306/qlsv";  private static final String USERNAME = "root";  private static final String PASSWORD = "123456789";  public static Connection getConnection() throws SQLException {  return DriverManager.getConnection(URL, USERNAME, PASSWORD);  }  public static void insert(MonHoc monHoc) {  String sql = "INSERT INTO MonHoc (IDMon, Mon) VALUES (?, ?)";  try (Connection conn = getConnection();  PreparedStatement stmt = conn.prepareStatement(sql)) {  stmt.setString(1, monHoc.getIDMon());  stmt.setString(2, monHoc.getMon());  stmt.executeUpdate();  } catch (SQLException e) {  e.printStackTrace();  }  }    public static void printAll() {  String sql = "SELECT \* FROM MonHoc";  try (Connection conn = getConnection();  PreparedStatement stmt = conn.prepareStatement(sql);  ResultSet rs = stmt.executeQuery()) {  System.out.println("Danh sách môn học:");  while (rs.next()) {  System.out.println("IDMon: " + rs.getString("IDMon"));  System.out.println("Môn học: " + rs.getString("Mon"));  System.out.println();  }  } catch (SQLException e) {  e.printStackTrace();  }  }  } |

Phương thức **insert(MonHoc monHoc)** dùng để thêm môn học mới.

Phương thức **printAll()** dùng để in toàn bộ danh sách môn học.

Phương thức **getConnection() throws SQLException** trả về đối tượng Connection và phương thức này dùng để kết nối hệ cơ sở dữ liệu.

-Sau khi cài em cài đặt các đối tượng cũng như các phương thức truy vấn dữ liệu xong. Em sẽ tiến hành viết ứng dụng như đã thiết kế ở (mục 3.1.1).

MainMenu.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81 | package Main;  import java.awt.EventQueue;  import javax.swing.JFrame;import javax.swing.JPanel;import javax.swing.border.EmptyBorder;import javax.swing.JLabel;import java.awt.Font;import javax.swing.JButton;import java.awt.event.ActionListener;import java.awt.event.ActionEvent;import java.awt.Color;  public class MainMenu extends JFrame {  private static final long serialVersionUID = 1L;  private JPanel contentPane;  /\*\* \* Launch the application. \*/  public static void main(String[] args) {  EventQueue.invokeLater(new Runnable() {  public void run() {  try {  MainMenu frame = new MainMenu();  frame.setVisible(true);  } catch (Exception e) {  e.printStackTrace();  }  }  });  }  /\*\* \* Create the frame. \*/  public MainMenu() {  setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  setBounds(100, 100, 450, 300);  contentPane = new JPanel();  contentPane.setBackground(new Color(62, 87, 64));  contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));  setContentPane(contentPane);  contentPane.setLayout(null);    JLabel lblNewLabel = new JLabel("Main Menu");  lblNewLabel.setForeground(new Color(255, 255, 255));  lblNewLabel.setBackground(new Color(255, 255, 255));  lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 20));  lblNewLabel.setBounds(161, 25, 116, 33);  contentPane.add(lblNewLabel);    JButton btnNewButton = new JButton("TCP Protocal");  btnNewButton.setForeground(new Color(255, 255, 255));  btnNewButton.setBackground(new Color(244, 204, 11));  btnNewButton.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  TCP\_Protocal frame = new TCP\_Protocal();  frame.setVisible(true);  setVisible(false);  }  });  btnNewButton.setBounds(21, 106, 131, 51);  contentPane.add(btnNewButton);    JButton btnUdpProtocal = new JButton("UDP Protocal");  btnUdpProtocal.setForeground(new Color(255, 255, 255));  btnUdpProtocal.setBackground(new Color(244, 204, 11));  btnUdpProtocal.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  UDP\_Protocal frame = new UDP\_Protocal();  frame.setVisible(true);  setVisible(false);  }  });  btnUdpProtocal.setBounds(267, 106, 131, 51);  contentPane.add(btnUdpProtocal);    JButton btnNewButton\_1 = new JButton("Thoát");  btnNewButton\_1.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  dispose();  }  });  btnNewButton\_1.setForeground(new Color(198, 54, 0));  btnNewButton\_1.setBounds(341, 232, 85, 21);  contentPane.add(btnNewButton\_1);  }  } |

Mô tả: Ứng dụng này hiển thị một cửa sổ chính (Main Menu) với các nút để chuyển sang các cửa sổ khác và một nút để thoát ứng dụng.

Giải thích chi tiết:

1. Khai báo package và import các thư viện cần thiết:

- ‘package Main;’ : Đặt lớp vào package Main.

- Các import từ java.awt, javax.swing để sử dụng các thành phần giao diện Swing.

1. Khai báo lớp MainMenu kế thừa từ JFrame:

- ‘private static final long serialVersionUID = 1L;’ : Định nghĩa ‘serialVersionUID’ để đảm bảo tính tương thích khi lớp được tuần tự hóa.

1. Phương thức main:

‘EventQueue.invokeLater’ : Đảm bảo rằng việc khởi tạo giao diện người dùng xảy ra trên luồng sự kiện AWT.

‘MainMenu frame = new MainMenu();’ : Tạo một đối tượng MainMenu.

‘frame.setVisible(true);’ : Hiển thị cửa sổ chính.

1. Constructor MainMenu:

Thiết lập thuộc tính của JFrame (setDefaultCloseOperation, setBounds).

Tạo và cấu hình JPanel (contentPane) để chứa các thành phần giao diện khác.

Thêm JLabel với tiêu đề "Main Menu".

Thêm nút "TCP Protocal" với sự kiện ActionListener để mở cửa sổ TCP\_Protocal.

Thêm nút "UDP Protocal" với sự kiện ActionListener để mở cửa sổ UDP\_Protocal.

Thêm nút "Thoát" với sự kiện ActionListener để đóng ứng dụng.

TCP\_Protocal.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103 | package Main;  import java.awt.EventQueue;  import javax.swing.JFrame;import javax.swing.JPanel;import javax.swing.border.EmptyBorder;  import Client.Client\_TCP;import Client.MayTinh;import Server.Server\_TCP;import Server.Server\_TCP\_MayTinh;  import javax.swing.JLabel;import java.awt.Font;import javax.swing.JButton;import java.awt.event.ActionListener;import java.awt.event.ActionEvent;import java.awt.Color;  public class TCP\_Protocal extends JFrame {  private static final long serialVersionUID = 1L;  private JPanel contentPane;  /\*\* \* Launch the application. \*/  public static void main(String[] args) {  EventQueue.invokeLater(new Runnable() {  public void run() {  try {  TCP\_Protocal frame = new TCP\_Protocal();  frame.setVisible(true);  } catch (Exception e) {  e.printStackTrace();  }  }  });  }  /\*\* \* Create the frame. \*/  public TCP\_Protocal() {  setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  setBounds(100, 100, 450, 300);  contentPane = new JPanel();  contentPane.setBackground(new Color(62, 87, 64));  contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));  setContentPane(contentPane);  contentPane.setLayout(null);    JLabel lblTcpProtocal = new JLabel("TCP Protocal");  lblTcpProtocal.setForeground(new Color(255, 255, 255));  lblTcpProtocal.setBackground(new Color(255, 255, 255));  lblTcpProtocal.setFont(new Font("Tahoma", Font.BOLD, 20));  lblTcpProtocal.setBounds(151, 26, 133, 33);  contentPane.add(lblTcpProtocal);    JButton btnNewButton = new JButton("Chat Room");  btnNewButton.setForeground(new Color(255, 255, 255));  btnNewButton.setBackground(new Color(244, 204, 11));  btnNewButton.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  TCP\_ChatRoom frame = new TCP\_ChatRoom();  frame.setVisible(true);  setVisible(false);  }  });  btnNewButton.setBounds(25, 118, 133, 45);  contentPane.add(btnNewButton);    JButton btnGame = new JButton("Máy tính");  btnGame.setForeground(new Color(255, 255, 255));  btnGame.setBackground(new Color(244, 204, 11));  btnGame.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  Server\_TCP\_MayTinh frame1 = new Server\_TCP\_MayTinh();  frame1.setVisible(true);  frame1.startServer();  MayTinh frame = new MayTinh();  frame.setVisible(true);  frame.connectToServer();  }  });  btnGame.setBounds(262, 118, 133, 45);  contentPane.add(btnGame);    JButton btnNewButton\_1 = new JButton("Trở về");  btnNewButton\_1.setBackground(new Color(255, 255, 255));  btnNewButton\_1.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  MainMenu frame = new MainMenu();  frame.setVisible(true);  setVisible(false);  }  });  btnNewButton\_1.setBounds(10, 232, 85, 21);  contentPane.add(btnNewButton\_1);    JButton btnThot = new JButton("Thoát");  btnThot.setBackground(new Color(255, 255, 255));  btnThot.setForeground(new Color(198, 54, 0));  btnThot.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  dispose();  }  });  btnThot.setBounds(341, 232, 85, 21);  contentPane.add(btnThot);  }  } |

Mô tả: Ứng dụng này hiển thị một cửa sổ (TCP Protocal) với các nút để mở các chức năng khác như "Chat Room" và "Máy tính", và các nút để trở về menu chính hoặc thoát ứng dụng.

Giải thích chi tiết:

1. Khai báo package và import các thư viện cần thiết:

‘package Main;’ : Đặt lớp vào package Main.

Các import từ java.awt, javax.swing, và các lớp từ package Client và Server để sử dụng các thành phần giao diện và chức năng mạng.

1. Khai báo lớp TCP\_Protocal kế thừa từ JFrame:

‘private static final long serialVersionUID = 1L;’ : Định nghĩa ‘serialVersionUID’ để đảm bảo tính tương thích khi lớp được tuần tự hóa.

1. Phương thức main:

‘EventQueue.invokeLater’ : Đảm bảo rằng việc khởi tạo giao diện người dùng xảy ra trên luồng sự kiện AWT.

‘TCP\_Protocal frame = new TCP\_Protocal();’ : Tạo một đối tượng TCP\_Protocal.

‘frame.setVisible(true);’ : Hiển thị cửa sổ.

1. Constructor TCP\_Protocal:

Thiết lập thuộc tính của JFrame (setDefaultCloseOperation, setBounds).

Tạo và cấu hình JPanel (contentPane) để chứa các thành phần giao diện khác.

Thêm JLabel với tiêu đề "TCP Protocal".

Thêm nút "Chat Room" với sự kiện ActionListener để mở cửa sổ TCP\_ChatRoom và ẩn cửa sổ hiện tại.

Thêm nút "Máy tính" với sự kiện ActionListener để mở và khởi động Server\_TCP\_MayTinh, sau đó mở cửa sổ MayTinh và kết nối đến server.

Thêm nút "Trở về" với sự kiện ActionListener để mở cửa sổ MainMenu và ẩn cửa sổ hiện tại.

Thêm nút "Thoát" với sự kiện ActionListener để đóng ứng dụng.

UDP\_Protocal.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84 | package Main;  import java.awt.EventQueue;  import javax.swing.JFrame;import javax.swing.JPanel;import javax.swing.border.EmptyBorder;  import Client.Login;import Server.Server\_UDP;  import javax.swing.JLabel;import java.awt.Font;import javax.swing.JButton;import java.awt.event.ActionListener;import java.awt.event.ActionEvent;import java.awt.Color;  public class UDP\_Protocal extends JFrame {  private static final long serialVersionUID = 1L;  private JPanel contentPane;  /\*\* \* Launch the application. \*/  public static void main(String[] args) {  EventQueue.invokeLater(new Runnable() {  public void run() {  try {  UDP\_Protocal frame = new UDP\_Protocal();  frame.setVisible(true);  } catch (Exception e) {  e.printStackTrace();  }  }  });  }  /\*\* \* Create the frame. \*/  public UDP\_Protocal() {  setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  setBounds(100, 100, 450, 300);  contentPane = new JPanel();  contentPane.setBackground(new Color(62, 87, 64));  contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));  setContentPane(contentPane);  contentPane.setLayout(null);    JLabel lblUdpProtocal = new JLabel("UDP Protocal");  lblUdpProtocal.setForeground(new Color(255, 255, 255));  lblUdpProtocal.setFont(new Font("Tahoma", Font.BOLD, 20));  lblUdpProtocal.setBounds(153, 10, 133, 33);  contentPane.add(lblUdpProtocal);    JButton btnHThngQun = new JButton("Hệ thống quản lý học sinh");  btnHThngQun.setForeground(new Color(255, 255, 255));  btnHThngQun.setBackground(new Color(244, 204, 11));  btnHThngQun.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  Login frame = new Login();  Server\_UDP frame2 = new Server\_UDP();  frame.setVisible(true);  frame2.setVisible(true);  frame2.startUDPServer();  }  });  btnHThngQun.setBounds(139, 103, 157, 51);  contentPane.add(btnHThngQun);    JButton btnNewButton = new JButton("Trở về");  btnNewButton.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  MainMenu frame = new MainMenu();  frame.setVisible(true);  setVisible(false);  }  });  btnNewButton.setBounds(10, 232, 85, 21);  contentPane.add(btnNewButton);    JButton btnThot = new JButton("Thoát");  btnThot.setForeground(new Color(198, 54, 0));  btnThot.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  dispose();  }  });  btnThot.setBounds(341, 232, 85, 21);  contentPane.add(btnThot);  }  } |

Mô tả: Ứng dụng này hiển thị một cửa sổ (UDP Protocal) với các nút để mở các chức năng như "Hệ thống quản lý học sinh", và các nút để trở về menu chính hoặc thoát ứng dụng.

Giải thích chi tiết:

1. Khai báo package và import các thư viện cần thiết:

‘package Main;’ : Đặt lớp vào package Main.

Các import từ java.awt, javax.swing, và các lớp từ package Client và Server để sử dụng các thành phần giao diện và chức năng mạng.

1. Khai báo lớp UDP\_Protocal kế thừa từ JFrame:

‘private static final long serialVersionUID = 1L;’ : Định nghĩa ‘serialVersionUID’ để đảm bảo tính tương thích khi lớp được tuần tự hóa.

1. Phương thức main:

‘EventQueue.invokeLater’ : Đảm bảo rằng việc khởi tạo giao diện người dùng xảy ra trên luồng sự kiện AWT.

‘UDP\_Protocal frame = new UDP\_Protocal();’ : Tạo một đối tượng UDP\_Protocal.

‘frame.setVisible(true);’ : Hiển thị cửa sổ.

1. Constructor UDP\_Protocal:

Thiết lập thuộc tính của JFrame (setDefaultCloseOperation, setBounds).

Tạo và cấu hình JPanel (contentPane) để chứa các thành phần giao diện khác.

Thêm JLabel với tiêu đề "UDP Protocal".

Thêm nút "Hệ thống quản lý học sinh" với sự kiện ActionListener để mở cửa sổ Login, khởi động và hiển thị Server\_UDP.

Thêm nút "Trở về" với sự kiện ActionListener để mở cửa sổ MainMenu và ẩn cửa sổ hiện tại.

Thêm nút "Thoát" với sự kiện ActionListener để đóng ứng dụng.

TCP\_ChatRoom.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105 | package Main;  import java.awt.EventQueue;  import javax.swing.JFrame;import javax.swing.JPanel;import javax.swing.border.EmptyBorder;  import Client.Client\_TCP;import Server.Server\_TCP;  import javax.swing.JButton;import javax.swing.JLabel;import java.awt.Font;import java.awt.event.ActionListener;import java.awt.event.ActionEvent;import java.awt.Color;  public class TCP\_ChatRoom extends JFrame {  private static final long serialVersionUID = 1L;  private JPanel contentPane;  /\*\* \* Launch the application. \*/  public static void main(String[] args) {  EventQueue.invokeLater(new Runnable() {  public void run() {  try {  TCP\_ChatRoom frame = new TCP\_ChatRoom();  frame.setVisible(true);  } catch (Exception e) {  e.printStackTrace();  }  }  });  }  /\*\* \* Create the frame. \*/  public TCP\_ChatRoom() {  setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  setBounds(100, 100, 450, 300);  contentPane = new JPanel();  contentPane.setBackground(new Color(62, 87, 64));  contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));  setContentPane(contentPane);  contentPane.setLayout(null);    JButton btnNewButton = new JButton("Client");  btnNewButton.setBackground(new Color(244, 204, 11));  btnNewButton.setForeground(new Color(255, 255, 255));  btnNewButton.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  Client\_TCP frame = new Client\_TCP();  frame.setVisible(true);  }  });  btnNewButton.setBounds(10, 99, 148, 51);  contentPane.add(btnNewButton);    JButton btnServer = new JButton("Server");  btnServer.setBackground(new Color(244, 204, 11));  btnServer.setForeground(new Color(255, 255, 255));  btnServer.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  Server\_TCP frame1 = new Server\_TCP();  frame1.setVisible(true);  }  });  btnServer.setBounds(278, 99, 148, 51);  contentPane.add(btnServer);    JLabel lblNewLabel = new JLabel("Có thể nhiều Client");  lblNewLabel.setForeground(new Color(255, 255, 255));  lblNewLabel.setLabelFor(this);  lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 13));  lblNewLabel.setBounds(20, 164, 148, 27);  contentPane.add(lblNewLabel);    JLabel lblNewLabel\_1 = new JLabel("Chỉ 1 Server được chạy");  lblNewLabel\_1.setForeground(new Color(255, 255, 255));  lblNewLabel\_1.setFont(new Font("Tahoma", Font.BOLD, 13));  lblNewLabel\_1.setBounds(278, 166, 156, 22);  contentPane.add(lblNewLabel\_1);    JLabel lblNewLabel\_2 = new JLabel("Chat Room");  lblNewLabel\_2.setForeground(new Color(255, 255, 255));  lblNewLabel\_2.setFont(new Font("Tahoma", Font.BOLD, 20));  lblNewLabel\_2.setBounds(167, 10, 110, 34);  contentPane.add(lblNewLabel\_2);    JButton btnNewButton\_1 = new JButton("Trở về");  btnNewButton\_1.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  TCP\_Protocal frame = new TCP\_Protocal();  frame.setVisible(true);  setVisible(false);  }  });  btnNewButton\_1.setBounds(10, 232, 85, 21);  contentPane.add(btnNewButton\_1);    JButton btnNewButton\_2 = new JButton("Thoát");  btnNewButton\_2.setForeground(new Color(198, 54, 0));  btnNewButton\_2.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  dispose();  }  });  btnNewButton\_2.setBounds(341, 232, 85, 21);  contentPane.add(btnNewButton\_2);  }  } |

Mô tả: Ứng dụng này hiển thị một cửa sổ (TCP Chat Room) với các nút để mở các chức năng liên quan đến Client và Server của giao thức TCP, và các nút để trở về menu trước hoặc thoát ứng dụng.

Giải thích chi tiết:

1. Khai báo package và import các thư viện cần thiết:

‘package Main;’: Đặt lớp vào package Main.

Các import từ java.awt, javax.swing, và các lớp từ package Client và Server để sử dụng các thành phần giao diện và chức năng mạng.

1. Khai báo lớp TCP\_ChatRoom kế thừa từ JFrame:

‘private static final long serialVersionUID = 1L;’: Định nghĩa serialVersionUID để đảm bảo tính tương thích khi lớp được tuần tự hóa.

1. Phương thức main:

‘EventQueue.invokeLater’: Đảm bảo rằng việc khởi tạo giao diện người dùng xảy ra trên luồng sự kiện AWT.

‘TCP\_ChatRoom frame = new TCP\_ChatRoom();’: Tạo một đối tượng TCP\_ChatRoom.

‘frame.setVisible(true);’: Hiển thị cửa sổ.

1. Constructor TCP\_ChatRoom:

Thiết lập thuộc tính của JFrame (setDefaultCloseOperation, setBounds).

Tạo và cấu hình JPanel (contentPane) để chứa các thành phần giao diện khác.

Thêm JButton với nhãn "Client" và sự kiện ActionListener để mở cửa sổ Client\_TCP.

Thêm JButton với nhãn "Server" và sự kiện ActionListener để mở cửa sổ Server\_TCP.

Thêm JLabel với các nhãn thông báo "Có thể nhiều Client" và "Chỉ 1 Server được chạy" để hướng dẫn người dùng.

Thêm JLabel với tiêu đề "Chat Room".

Thêm nút "Trở về" với sự kiện ActionListener để mở cửa sổ TCP\_Protocal và ẩn cửa sổ hiện tại.

Thêm nút "Thoát" với sự kiện ActionListener để đóng ứng dụng.

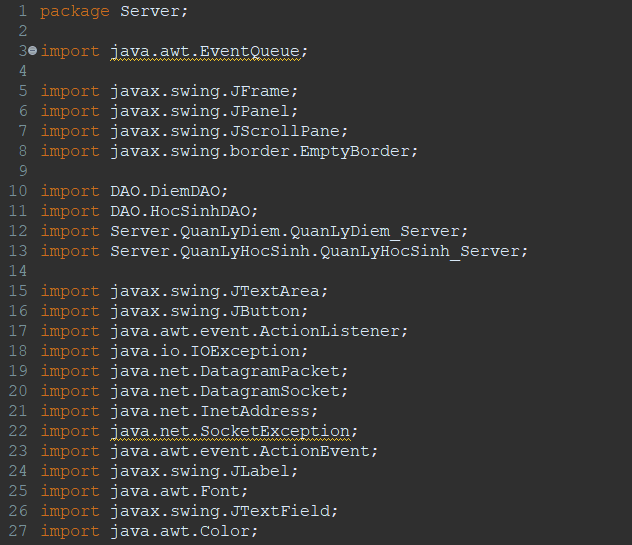
Server\_UDP.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149  150  151  152  153  154  155  156  157  158  159  160  161  162  163  164  165  166  167  168  169  170  171  172  173  174  175  176  177  178  179  180  181  182  183  184  185  186  187  188  189  190  191  192  193  194  195  196  197  198  199  200  201  202  203  204  205  206  207  208  209  210  211  212  213  214  215  216  217  218  219  220  221  222  223  224  225  226  227  228  229  230  231  232  233  234  235  236  237  238  239  240  241  242  243  244  245  246  247  248  249  250  251  252  253  254  255  256  257  258  259  260  261  262  263  264  265  266  267  268  269  270  271  272  273  274  275  276  277  278  279  280  281  282  283  284  285  286  287  288  289  290  291  292  293  294  295  296  297  298  299  300  301  302  303  304  305 | package Server;  import java.awt.EventQueue;  import javax.swing.JFrame;import javax.swing.JPanel;import javax.swing.JScrollPane;import javax.swing.border.EmptyBorder;  import DAO.DiemDAO;import DAO.HocSinhDAO;import Server.QuanLyDiem.QuanLyDiem\_Server;import Server.QuanLyHocSinh.QuanLyHocSinh\_Server;  import javax.swing.JTextArea;import javax.swing.JButton;import java.awt.event.ActionListener;import java.io.IOException;import java.net.DatagramPacket;import java.net.DatagramSocket;import java.net.InetAddress;import java.net.SocketException;import java.awt.event.ActionEvent;import javax.swing.JLabel;import java.awt.Font;import javax.swing.JTextField;import java.awt.Color;  public class Server\_UDP extends JFrame {  private static final long serialVersionUID = 1L;  private JPanel contentPane;  private JTextArea activityHistory;  private DatagramSocket socket;  private JTextField textField;  private InetAddress clientAddress1;  private int clientPort1;  private boolean running = true;  public static void main(String[] args) {  Server\_UDP frame = new Server\_UDP();  frame.setVisible(true);  frame.startUDPServer();  }  public Server\_UDP() {  setBackground(new Color(0, 0, 53));  setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  setBounds(100, 100, 450, 300);  contentPane = new JPanel();  contentPane.setBackground(new Color(0, 0, 53));  contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));  setContentPane(contentPane);  contentPane.setLayout(null);  activityHistory = new JTextArea();  JScrollPane scrollPane = new JScrollPane(activityHistory);  scrollPane.setBounds(10, 63, 276, 141);  contentPane.add(scrollPane);  JButton qlDiem\_bt = new JButton("Quản lý điểm");  qlDiem\_bt.setBackground(new Color(255, 255, 255));  qlDiem\_bt.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  QuanLyDiem\_Server frame = new QuanLyDiem\_Server();  frame.setVisible(true);  frame.clearDiem();  frame.updateDiem();  }  });  qlDiem\_bt.setBounds(296, 66, 130, 21);  contentPane.add(qlDiem\_bt);  JButton qlHs\_bt = new JButton("Quản lý học sinh");  qlHs\_bt.setBackground(new Color(255, 255, 255));  qlHs\_bt.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  QuanLyHocSinh\_Server frame = new QuanLyHocSinh\_Server();  frame.setVisible(true);  frame.clearDanhSach();  frame.UpdateDanhSach();  }  });  qlHs\_bt.setBounds(296, 97, 130, 21);  contentPane.add(qlHs\_bt);  JLabel lblNewLabel = new JLabel("Server Main");  lblNewLabel.setForeground(new Color(255, 255, 255));  lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 22));  lblNewLabel.setBounds(152, 10, 134, 27);  contentPane.add(lblNewLabel);  textField = new JTextField();  textField.setBounds(10, 214, 207, 19);  contentPane.add(textField);  textField.setColumns(10);  JButton btnNewButton = new JButton("Gửi");  btnNewButton.setBackground(new Color(255, 255, 255));  btnNewButton.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  String mess = textField.getText();  String message = "message,Manager," + mess;  byte[] sendData = message.getBytes();  if (clientAddress1 != null && clientPort1 != 0) { // Kiểm tra clientAddress1 và clientPort1 trước khi gửi  DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length, clientAddress1, clientPort1);  try {  socket.send(sendPacket);  textField.setText("");  appendToActivityHistory("You: " + mess);  } catch (IOException e1) {  e1.printStackTrace();  }  } else {  appendToActivityHistory("Không có máy client nào được kết nối.");  }  }  });  btnNewButton.setBounds(227, 214, 59, 21);  contentPane.add(btnNewButton);    JButton btnNewButton\_1 = new JButton("Thoát");  btnNewButton\_1.setBackground(new Color(255, 255, 255));  btnNewButton\_1.setForeground(new Color(198, 54, 0));  btnNewButton\_1.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  running = false; // Dừng vòng lặp trong startUDPServer  socket.close(); // Đóng socket  dispose(); // Đóng cửa sổ  }  });  btnNewButton\_1.setBounds(341, 232, 85, 21);  contentPane.add(btnNewButton\_1);  }  public void startUDPServer() {  new Thread(new Runnable() {  @Override  public void run() {  try {  socket = new DatagramSocket(9876);  appendToActivityHistory("Server đã khởi chạy, đang đợi các máy Client..");    while (running) {  byte[] receiveData = new byte[1024];  DatagramPacket receivePacket = new DatagramPacket(receiveData, receiveData.length);  socket.receive(receivePacket);  String receivedMessage = new String(receivePacket.getData()).trim();  String[] info = receivedMessage.split(",");  clientAddress1 = receivePacket.getAddress();  clientPort1 = receivePacket.getPort();  if (info[0].endsWith("Login")) {  processLoginRequest(receivedMessage, receivePacket.getAddress(), receivePacket.getPort());  } else if (info[0].endsWith("User")) {  userConectingRequest(receivedMessage, receivePacket.getAddress(), receivePacket.getPort());  } else if (info[0].endsWith("Pass")) {  changePassRequest(receivedMessage, receivePacket.getAddress(), receivePacket.getPort());  } else if (info[0].endsWith("Disconect")) {  disconectRequest(receivedMessage, receivePacket.getAddress(), receivePacket.getPort());  } else if (info[0].endsWith("Point")) {  pointRequest(receivedMessage, receivePacket.getAddress(), receivePacket.getPort());  } else if (info[0].endsWith("Infomation")) {  infoRequest(receivedMessage, receivePacket.getAddress(), receivePacket.getPort());  }  }  } catch (IOException e) {  e.printStackTrace();  } finally {  if (socket != null && !socket.isClosed()) {  socket.close();  }  }  }  }).start();  }  private void appendToActivityHistory(String message) {  activityHistory.append(message + "\n");  }  private void infoRequest(String receivedMessage, InetAddress clientAddress, int clientPort) {  String[] Info = receivedMessage.split(",");  String ID = Info[1];  String DanhSach = HocSinhDAO.TimHocSinhTheoIDHocSinh(ID);    String message = "info-" + DanhSach;  byte[] sendData = message.getBytes();  DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length, clientAddress, clientPort);  try {  socket.send(sendPacket);  appendToActivityHistory("Đã gửi thông tin cho!" + ID);  } catch (IOException e) {  e.printStackTrace();  }  }  private void pointRequest(String receivedMessage, InetAddress clientAddress, int clientPort) {  String[] Info = receivedMessage.split(",");  String ID = Info[1];  String Diem = DiemDAO.TimDiemTheoIDHocSinh(ID);  String message = "point," + Diem;  byte[] sendData = message.getBytes();  DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length, clientAddress, clientPort);  try {  socket.send(sendPacket);  appendToActivityHistory("Đã gửi bảng điểm cho!" + ID);  } catch (IOException e) {  e.printStackTrace();  }  }  private void disconectRequest(String receivedMessage, InetAddress clientAddress, int clientPort) {  String[] Info = receivedMessage.split(",");  String user = Info[1];  appendToActivityHistory(user + " đã ngắt kết nối với Server!");  }  private void userConectingRequest(String receivedMessage, InetAddress clientAddress, int clientPort) {  String[] Info = receivedMessage.split(",");  String con = Info[1];  String user = Info[2];  String mess = Info[3];  if (con.equals("connect")) {  String message = "connect,success," + user;  byte[] sendData = message.getBytes();  DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length, clientAddress, clientPort);  try {  socket.send(sendPacket);  appendToActivityHistory(user + " đã kết nối đến Server!");  } catch (IOException e) {  e.printStackTrace();  }  } else if (con.equals("message")) {  appendToActivityHistory(user + ": " + mess);  }  }    private void changePassRequest(String receivedMessage, InetAddress clientAddress, int clientPort) {  String[] Info = receivedMessage.split(",");  String user = Info[1];  String cu = Info[2];  String moi = Info[3];  boolean temp = HocSinhDAO.changePass(user, cu, moi);    if(temp) {  byte[] sendData = "success".getBytes();  DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length, clientAddress, clientPort);  try {  socket.send(sendPacket);  appendToActivityHistory(user+" đã đổi mật khẩu!");  } catch (IOException e) {  e.printStackTrace();  }  }else {  byte[] sendData = "failure".getBytes();  DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length, clientAddress, clientPort);  try {  socket.send(sendPacket);  } catch (IOException e) {  e.printStackTrace();  }  }  }  private void processLoginRequest(String receivedMessage, InetAddress clientAddress, int clientPort) {  String[] loginInfo = receivedMessage.split(",");  if (loginInfo.length >= 3) { // Kiểm tra độ dài của mảng  String username = loginInfo[1];  String message = loginInfo[2];  // Example validation (replace with your own authentication logic)  if (HocSinhDAO.checkLogin(username, message)) {  byte[] sendData = "success".getBytes();  DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length, clientAddress, clientPort);  try {  socket.send(sendPacket);  } catch (IOException e) {  e.printStackTrace();  }  } else {  byte[] sendData = "failure".getBytes();  DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length, clientAddress, clientPort);  try {  socket.send(sendPacket);  } catch (IOException e) {  e.printStackTrace();  }  }  } else {  // Xử lý lỗi hoặc báo cáo một cách phù hợp nếu thông điệp không đủ dài  }  }  } |

Mô tả: Ứng dụng Server\_UDP là một ứng dụng server sử dụng giao thức UDP để giao tiếp với các client. Nó có giao diện người dùng bằng Java Swing để hiển thị hoạt động của server và cho phép người dùng tương tác với các chức năng quản lý điểm và học sinh.

Giải thích chi tiết:

1. Khai báo package và import các thư viện cần thiết:



Hình 27: Khai báo thư viện Server\_UDP.java.

1. Khai báo lớp Server\_UDP kế thừa từ JFrame:

‘private static final long serialVersionUID = 1L;’: Định nghĩa ‘serialVersionUID’ để đảm bảo tính tương thích khi lớp được tuần tự hóa.

Các biến thành viên để quản lý giao diện và chức năng UDP.

1. Phương thức main:

Khởi tạo và hiển thị giao diện của server.

Gọi phương thức ‘startUDPServer’ để bắt đầu server UDP.

1. Constructor Server\_UDP:

Thiết lập thuộc tính của JFrame (setDefaultCloseOperation, setBounds).

Tạo và cấu hình JPanel (contentPane) để chứa các thành phần giao diện khác.

Thêm JTextArea (activityHistory) để hiển thị lịch sử hoạt động của server.

Thêm các nút JButton để quản lý điểm và học sinh, với các sự kiện ActionListener tương ứng.

Thêm các nhãn JLabel để hiển thị tiêu đề và thông tin khác.

Thêm JTextField để nhập tin nhắn gửi tới client.

Thêm nút Gửi với sự kiện ActionListener để gửi tin nhắn tới client.

Thêm nút Thoát để dừng server và đóng ứng dụng.

1. Phương thức startUDPServer:

Tạo một thread mới để chạy server UDP.

Sử dụng DatagramSocket để lắng nghe các gói tin từ client.

Phân tích các thông điệp nhận được và xử lý các yêu cầu như login, kết nối, thay đổi mật khẩu, điểm số, thông tin học sinh, và ngắt kết nối.

1. Các phương thức hỗ trợ:

appendToActivityHistory(String message): Thêm một thông điệp vào activityHistory.

infoRequest, pointRequest, disconectRequest, userConectingRequest, changePassRequest, processLoginRequest: Các phương thức để xử lý các yêu cầu cụ thể từ client, bao gồm xử lý thông tin học sinh, điểm số, kết nối, ngắt kết nối và thay đổi mật khẩu.

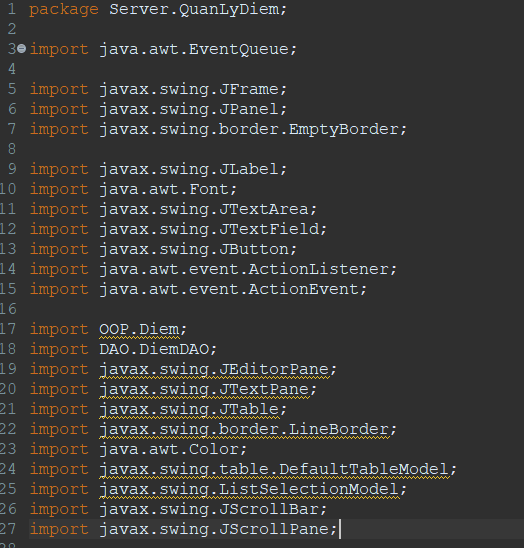
QuanLyDiem\_Server.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133 | package Server.QuanLyDiem;  import java.awt.EventQueue;  import javax.swing.JFrame;import javax.swing.JPanel;import javax.swing.border.EmptyBorder;  import javax.swing.JLabel;import java.awt.Font;import javax.swing.JTextArea;import javax.swing.JTextField;import javax.swing.JButton;import java.awt.event.ActionListener;import java.awt.event.ActionEvent;  import OOP.Diem;import DAO.DiemDAO;import javax.swing.JEditorPane;import javax.swing.JTextPane;import javax.swing.JTable;import javax.swing.border.LineBorder;import java.awt.Color;import javax.swing.table.DefaultTableModel;import javax.swing.ListSelectionModel;import javax.swing.JScrollBar;import javax.swing.JScrollPane;  public class QuanLyDiem\_Server extends JFrame {  private static final long serialVersionUID = 1L;  private JPanel contentPane;  private JTextField IDHocSinh;  private static JTextArea textArea = new JTextArea();  /\*\* \* Launch the application. \*/  public static void main(String[] args) {  EventQueue.invokeLater(new Runnable() {  public void run() {  try {  QuanLyDiem\_Server frame = new QuanLyDiem\_Server();  frame.setVisible(true);  } catch (Exception e) {  e.printStackTrace();  }  }  });  updateDiem();  }  /\*\* \* Create the frame. \*/  public static void updateDiem() {  String Diem = DiemDAO.BangDiemAll();  appendToTextArea(Diem);  }  public static void clearDiem() {  textArea.setText("");  }  public QuanLyDiem\_Server() {  setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  setBounds(100, 100, 450, 300);  contentPane = new JPanel();  contentPane.setBackground(new Color(0, 0, 53));  contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));  setContentPane(contentPane);  contentPane.setLayout(null);    JLabel lblNewLabel = new JLabel("Trang Quản Lý Điểm");  lblNewLabel.setForeground(new Color(255, 255, 255));  lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 20));  lblNewLabel.setBounds(116, 10, 208, 25);  contentPane.add(lblNewLabel);    JScrollPane scrollPane = new JScrollPane(textArea);  scrollPane.setBounds(10, 93, 418, 133);  contentPane.add(scrollPane);    IDHocSinh = new JTextField();  IDHocSinh.setText("HS001");  IDHocSinh.setBounds(195, 64, 96, 19);  contentPane.add(IDHocSinh);  IDHocSinh.setColumns(10);    JLabel lblNewLabel\_1 = new JLabel("Tìm kiếm theo IDHocSinh:");  lblNewLabel\_1.setForeground(new Color(255, 255, 255));  lblNewLabel\_1.setBounds(37, 67, 148, 13);  contentPane.add(lblNewLabel\_1);    JButton btnNewButton = new JButton("Tìm");  btnNewButton.setBackground(new Color(255, 255, 255));  btnNewButton.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  String ID = IDHocSinh.getText();  textArea.setText("");  String Diem = DiemDAO.TimDiemTheoIDHocSinh(ID);  appendToTextArea(Diem);  }  });  btnNewButton.setBounds(297, 62, 57, 21);  contentPane.add(btnNewButton);    JButton btnNewButton\_1 = new JButton("Thoát");  btnNewButton\_1.setBackground(new Color(255, 255, 255));  btnNewButton\_1.setForeground(new Color(198, 54, 0));  btnNewButton\_1.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  dispose();  }  });  btnNewButton\_1.setBounds(341, 236, 85, 21);  contentPane.add(btnNewButton\_1);    JButton btnNewButton\_2 = new JButton("Chỉnh sửa điểm");  btnNewButton\_2.setBackground(new Color(255, 255, 255));  btnNewButton\_2.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  ChinhSuaDiem frame = new ChinhSuaDiem();  frame.setVisible(true);  setVisible(false);  }  });  btnNewButton\_2.setBounds(18, 236, 132, 21);  contentPane.add(btnNewButton\_2);    JScrollBar scrollBar = new JScrollBar();  scrollBar.setBounds(411, 93, 17, 133);  contentPane.add(scrollBar);    JButton btnNewButton\_3 = new JButton("Danh sách tổng");  btnNewButton\_3.setBackground(new Color(255, 255, 255));  btnNewButton\_3.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  clearDiem();  updateDiem();  }  });  btnNewButton\_3.setBounds(160, 236, 131, 21);  contentPane.add(btnNewButton\_3);  }  private static void appendToTextArea(String text) {  textArea.append(text + "\n");  textArea.setLineWrap(true);  textArea.setWrapStyleWord(true);  }  } |

Mô tả: Lớp QuanLyDiem\_Server là một phần của module quản lý điểm trong ứng dụng server sử dụng Java Swing để tạo giao diện người dùng. Lớp này cho phép quản lý điểm số của học sinh, tìm kiếm điểm theo ID học sinh, chỉnh sửa điểm và hiển thị danh sách tổng các điểm số.

Giải thích chi tiết:

1. Khai báo package và import các thư viện cần thiết:



Hình 28: Khai báo thư viện QuanLyDiem\_Server.java.

1. Khai báo lớp QuanLyDiem\_Server kế thừa từ JFrame:

‘private static final long serialVersionUID = 1L;’ : Định nghĩa serialVersionUID để đảm bảo tính tương thích khi lớp được tuần tự hóa.

Các biến thành viên để quản lý giao diện và chức năng liên quan đến điểm.

1. Phương thức main:

Khởi tạo và hiển thị giao diện của quản lý điểm.

Gọi phương thức ‘updateDiem’ để cập nhật điểm khi ứng dụng khởi động.

1. Constructor QuanLyDiem\_Server:

Thiết lập thuộc tính của JFrame (setDefaultCloseOperation, setBounds).

Tạo và cấu hình JPanel (contentPane) để chứa các thành phần giao diện khác.

Thêm JTextArea (textArea) để hiển thị điểm số.

Thêm các nhãn JLabel để hiển thị tiêu đề và thông tin khác.

Thêm JTextField để nhập ID học sinh.

Thêm các nút JButton để thực hiện các chức năng như tìm kiếm, thoát, chỉnh sửa điểm và hiển thị danh sách tổng, với các sự kiện ActionListener tương ứng.

1. Phương thức updateDiem và clearDiem:

updateDiem(): Gọi hàm BangDiemAll từ DiemDAO để lấy danh sách tổng các điểm số và hiển thị trong textArea.

clearDiem(): Xóa toàn bộ nội dung trong textArea.

1. Phương thức appendToTextArea:

Thêm một đoạn văn bản vào textArea và thiết lập chế độ tự động xuống dòng.

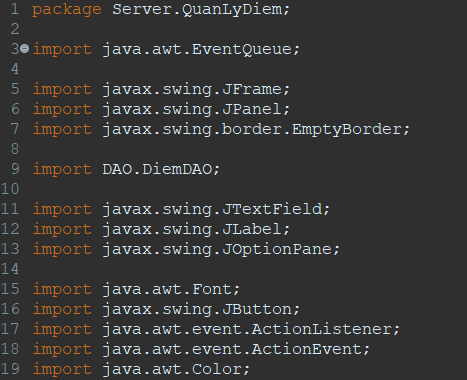
ChinhSuaDiem.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149  150  151  152  153  154  155  156  157  158  159  160  161  162  163  164  165  166  167  168  169  170  171  172  173  174  175  176  177  178  179  180  181  182  183  184  185  186  187  188  189  190  191  192  193  194  195  196  197  198  199  200  201  202  203  204  205  206  207  208  209  210  211  212  213  214  215  216  217  218  219  220  221  222  223  224  225  226  227  228  229  230  231  232  233  234  235  236  237  238  239  240  241  242  243  244  245  246  247  248  249  250  251  252  253  254  255  256  257  258  259  260  261  262  263  264  265  266  267  268  269  270  271  272  273  274  275  276  277  278  279  280  281  282  283  284  285  286  287  288  289  290  291  292  293  294  295  296  297  298  299  300  301  302  303  304  305  306  307  308  309  310  311  312  313  314  315  316  317  318  319  320  321  322  323  324  325  326  327  328  329  330  331  332  333  334  335  336  337  338  339  340  341  342  343  344  345  346  347  348  349  350  351  352  353  354  355  356  357  358  359  360  361  361  363  364  365  366  367  368  369  370  371  372  373  374  375  376  377  378  379  380  381 | package Server.QuanLyDiem;  import java.awt.EventQueue;  import javax.swing.JFrame;import javax.swing.JPanel;import javax.swing.border.EmptyBorder;  import DAO.DiemDAO;  import javax.swing.JTextField;import javax.swing.JLabel;import javax.swing.JOptionPane;  import java.awt.Font;import javax.swing.JButton;import java.awt.event.ActionListener;import java.awt.event.ActionEvent;import java.awt.Color;  public class ChinhSuaDiem extends JFrame {  private static final long serialVersionUID = 1L;  private JPanel contentPane;  private JTextField IDHocSinh;  private JTextField MHAnh;  private JTextField MHToan;  private JTextField MHVan;  private JTextField HoTen;  /\*\* \* Launch the application. \*/  public static void main(String[] args) {  EventQueue.invokeLater(new Runnable() {  public void run() {  try {  ChinhSuaDiem frame = new ChinhSuaDiem();  frame.setVisible(true);  } catch (Exception e) {  e.printStackTrace();  }  }  });  }  /\*\* \* Create the frame. \*/  public static String[][] parseInput(String input){  String[][] temp = new String[4][2];      String[] lines = input.split("\n");  // Khởi tạo mảng hai chiều để lưu trữ thông tin  temp = new String[lines.length + 1][2];  // Tìm vị trí dòng bắt đầu thông tin điểm  int start = 0;  for (int i = 0; i < lines.length; i++) {  if (lines[i].startsWith("Họ và tên:")) {  start = i;  break;  }  }  // Tách họ tên từ dòng bắt đầu thông tin điểm  String[] hoTen = lines[start].split(":");  temp[0][0] = hoTen[0].trim(); // "Họ và tên"  temp[0][1] = hoTen[1].trim(); // "Nguyễn Văn A"  // Bắt đầu từ dòng thứ start + 1  for (int i = start + 1; i < lines.length; i++) {  // Tách tên môn và điểm từ mỗi dòng  String[] parts = lines[i].split(":");  // Xóa khoảng trắng ở đầu của điểm (nếu có)  String monHoc = parts[0].trim();  String diem = parts[1].trim();  // Lưu tên môn vào cột 0 và điểm vào cột 1  temp[i - start][0] = monHoc;  temp[i - start][1] = diem;  }  return temp;  }  public static String[][] conf(String[][] parseInput) {  String[][] temp = new String[3][2];  for (int i = 0; i < parseInput.length; i++) {  // Kiểm tra parseInput[i] không phải là null  if (parseInput[i] != null) {  for (int j = 0; j < parseInput[i].length; j++) {  // Kiểm tra parseInput[i][0] không phải là null trước khi so sánh  if (parseInput[i][0] != null) {  if (parseInput[i][0].equals("Anh")) {  temp[0][0] = "Anh";  temp[0][1] = parseInput[i][1];  } else if (parseInput[i][0].equals("Toán")) {  temp[1][0] = "Toán";  temp[1][1] = parseInput[i][1];  } else if (parseInput[i][0].equals("Văn")) {  temp[2][0] = "Văn";  temp[2][1] = parseInput[i][1];  }  }  }  }  }  return temp;  }  public ChinhSuaDiem() {  setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  setBounds(100, 100, 450, 300);  contentPane = new JPanel();  contentPane.setBackground(new Color(0, 0, 53));  contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));  setContentPane(contentPane);  contentPane.setLayout(null);    IDHocSinh = new JTextField();  IDHocSinh.setText("HS001");  IDHocSinh.setBounds(112, 55, 96, 19);  contentPane.add(IDHocSinh);  IDHocSinh.setColumns(10);    JLabel lblNewLabel = new JLabel("Nhập IDHocSinh:");  lblNewLabel.setForeground(new Color(255, 255, 255));  lblNewLabel.setBounds(10, 58, 117, 13);  contentPane.add(lblNewLabel);    JLabel lblNewLabel\_1 = new JLabel("Chỉnh Sửa Điểm");  lblNewLabel\_1.setForeground(new Color(255, 255, 255));  lblNewLabel\_1.setFont(new Font("Tahoma", Font.BOLD, 20));  lblNewLabel\_1.setBounds(120, 10, 167, 24);  contentPane.add(lblNewLabel\_1);    JButton btnNewButton = new JButton("Xem");  btnNewButton.setBackground(new Color(255, 255, 255));  btnNewButton.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  String ID = IDHocSinh.getText();  MHAnh.setText("");  MHToan.setText("");  MHVan.setText("");  String Diem = DiemDAO.TimDiemTheoIDHocSinh(ID);  String[][] temp =parseInput(Diem);  String[][] temp1 =conf(temp);  HoTen.setText(temp[0][1]);  MHAnh.setText(temp1[0][1]);  MHToan.setText(temp1[1][1]);  MHVan.setText(temp1[2][1]);  }  });  btnNewButton.setBounds(223, 54, 64, 21);  contentPane.add(btnNewButton);    JLabel Anh = new JLabel("Anh:");  Anh.setForeground(new Color(255, 255, 255));  Anh.setBounds(86, 117, 45, 13);  contentPane.add(Anh);    JLabel lblNewLabel\_2 = new JLabel("Toán:");  lblNewLabel\_2.setForeground(new Color(255, 255, 255));  lblNewLabel\_2.setBounds(82, 154, 45, 13);  contentPane.add(lblNewLabel\_2);    JLabel lblNewLabel\_3 = new JLabel("Văn:");  lblNewLabel\_3.setForeground(new Color(255, 255, 255));  lblNewLabel\_3.setBounds(86, 192, 45, 13);  contentPane.add(lblNewLabel\_3);    MHAnh = new JTextField();  MHAnh.setColumns(10);  MHAnh.setBounds(127, 114, 45, 19);  contentPane.add(MHAnh);    MHToan = new JTextField();  MHToan.setColumns(10);  MHToan.setBounds(127, 151, 45, 19);  contentPane.add(MHToan);    MHVan = new JTextField();  MHVan.setColumns(10);  MHVan.setBounds(127, 189, 45, 19);  contentPane.add(MHVan);    JButton btnNewButton\_1 = new JButton("Trở về");  btnNewButton\_1.setBackground(new Color(255, 255, 255));  btnNewButton\_1.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  QuanLyDiem\_Server frame = new QuanLyDiem\_Server();  frame.setVisible(true);  frame.clearDiem();  frame.updateDiem();  setVisible(false);  }  });  btnNewButton\_1.setBounds(10, 232, 85, 21);  contentPane.add(btnNewButton\_1);    JButton btnNewButton\_2 = new JButton("Thoát");  btnNewButton\_2.setBackground(new Color(255, 255, 255));  btnNewButton\_2.setForeground(new Color(198, 54, 0));  btnNewButton\_2.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  dispose();  }  });  btnNewButton\_2.setBounds(341, 232, 85, 21);  contentPane.add(btnNewButton\_2);    JButton btnNewButton\_3 = new JButton("Chỉnh sửa");  btnNewButton\_3.setBackground(new Color(255, 255, 255));  btnNewButton\_3.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  String ID = IDHocSinh.getText();  String Anh = MHAnh.getText();  String Toan = MHToan.getText();  String Van = MHVan.getText();  String Name = HoTen.getText();    if(Name.equals("")) {  JOptionPane.showMessageDialog(null, "Vui lòng xem học sinh để chỉnh sửa!");  }  else {  String Diem = DiemDAO.TimDiemTheoIDHocSinh(ID);    String[][] temp =parseInput(Diem);  String[][] temp1 =conf(temp);    // So sánh xem cái nào được chỉnh sửa  if(Anh.equals(temp1[0][1]) && Toan.equals(temp1[1][1]) && Van.equals(temp1[2][1]) ) {  JOptionPane.showMessageDialog(null, "Bạn không chỉnh sửa.");  }  if(Anh.equals(temp1[0][1]) == false && Toan.equals(temp1[1][1]) && Van.equals(temp1[2][1])) {  boolean check = DiemDAO.updateDiem(ID, "MH003", Double.parseDouble(Anh));  if(check) {  JOptionPane.showMessageDialog(null, "Chỉnh sửa môn Anh thành công.");  }  else {  JOptionPane.showMessageDialog(null, "Chỉnh sửa môn Anh thất bại.");  }  }  if(Anh.equals(temp1[0][1]) && Toan.equals(temp1[1][1]) == false && Van.equals(temp1[2][1])) {  boolean check = DiemDAO.updateDiem(ID, "MH001", Double.parseDouble(Toan));  if(check) {  JOptionPane.showMessageDialog(null, "Chỉnh sửa môn Toán thành công.");  }  else {  JOptionPane.showMessageDialog(null, "Chỉnh sửa môn Toán thất bại.");  }  }  if(Anh.equals(temp1[0][1]) && Toan.equals(temp1[1][1]) && Van.equals(temp1[2][1]) == false) {  boolean check = DiemDAO.updateDiem(ID, "MH002", Double.parseDouble(Van));  if(check) {  JOptionPane.showMessageDialog(null, "Chỉnh sửa môn Văn thành công.");  }  else {  JOptionPane.showMessageDialog(null, "Chỉnh sửa môn Văn thất bại.");  }  }  if(Anh.equals(temp1[0][1])== false && Toan.equals(temp1[1][1])== false && Van.equals(temp1[2][1])) {  boolean check = DiemDAO.updateDiem(ID, "MH003", Double.parseDouble(Anh));  boolean check1 = DiemDAO.updateDiem(ID, "MH001", Double.parseDouble(Toan));  if(check && check1) {  JOptionPane.showMessageDialog(null, "Chỉnh sửa môn Anh và Toán thành công.");  }  else {  JOptionPane.showMessageDialog(null, "Chỉnh sửa môn Anh và Toán thất bại.");  }  }  if(Anh.equals(temp1[0][1]) && Toan.equals(temp1[1][1])== false && Van.equals(temp1[2][1])== false) {  boolean check1 = DiemDAO.updateDiem(ID, "MH001", Double.parseDouble(Toan));  boolean check = DiemDAO.updateDiem(ID, "MH002", Double.parseDouble(Van));  if(check && check1) {  JOptionPane.showMessageDialog(null, "Chỉnh sửa môn Toán và Văn thành công.");  }  else {  JOptionPane.showMessageDialog(null, "Chỉnh sửa môn Toán và Văn thất bại.");  }  }  if(Anh.equals(temp1[0][1])== false && Toan.equals(temp1[1][1]) && Van.equals(temp1[2][1])== false) {  boolean check = DiemDAO.updateDiem(ID, "MH003", Double.parseDouble(Anh));  boolean check1 = DiemDAO.updateDiem(ID, "MH002", Double.parseDouble(Van));  if(check && check1) {  JOptionPane.showMessageDialog(null, "Chỉnh sửa môn Anh và Văn thành công.");  }  else {  JOptionPane.showMessageDialog(null, "Chỉnh sửa môn Anh và Văn thất bại.");  }  }  if(Anh.equals(temp1[0][1])== false && Toan.equals(temp1[1][1])== false && Van.equals(temp1[2][1])== false) {  boolean check = DiemDAO.updateDiem(ID, "MH003", Double.parseDouble(Anh));  boolean check1 = DiemDAO.updateDiem(ID, "MH002", Double.parseDouble(Van));  boolean check2 = DiemDAO.updateDiem(ID, "MH001", Double.parseDouble(Toan));  if(check && check1 && check2) {  JOptionPane.showMessageDialog(null, "Chỉnh sửa môn 3 môn thành công.");  }  else {  JOptionPane.showMessageDialog(null, "Chỉnh sửa môn 3 môn thất bại.");  }  }    String ID1 = IDHocSinh.getText();  MHAnh.setText("");  MHToan.setText("");  MHVan.setText("");  String Diem1 = DiemDAO.TimDiemTheoIDHocSinh(ID1);  String[][] temp2 =parseInput(Diem1);  String[][] temp11 =conf(temp2);  HoTen.setText(temp2[0][1]);  MHAnh.setText(temp11[0][1]);  MHToan.setText(temp11[1][1]);  MHVan.setText(temp11[2][1]);  }    }  });  btnNewButton\_3.setBounds(223, 150, 106, 21);  contentPane.add(btnNewButton\_3);    JLabel lblNewLabel\_4 = new JLabel("Họ và tên:");  lblNewLabel\_4.setForeground(new Color(255, 255, 255));  lblNewLabel\_4.setBounds(59, 84, 96, 13);  contentPane.add(lblNewLabel\_4);    HoTen = new JTextField();  HoTen.setBounds(127, 81, 149, 19);  contentPane.add(HoTen);  HoTen.setColumns(10);  }  } |

Mô tả: Lớp ChinhSuaDiem là một giao diện chỉnh sửa điểm số học sinh trong ứng dụng quản lý điểm, được tạo bằng Java Swing. Giao diện này cho phép người dùng nhập ID học sinh để xem và chỉnh sửa điểm số của các môn học, bao gồm Anh, Toán và Văn. Khi người dùng thực hiện chỉnh sửa, lớp sẽ cập nhật thông tin điểm số vào cơ sở dữ liệu thông qua DAO (Data Access Object).

Giải thích chi tiết:

1. Khai báo package và import các thư viện cần thiết:



Hình 29: Khai báo thư viện ChinhSuaDiem.java.

1. Khai báo lớp ChinhSuaDiem kế thừa từ JFrame:

‘private static final long serialVersionUID = 1L;’: Định nghĩa serialVersionUID để đảm bảo tính tương thích khi lớp được tuần tự hóa.vb

Các biến thành viên để quản lý giao diện và chức năng liên quan đến chỉnh sửa điểm số.

1. Phương thức main:

Khởi tạo và hiển thị giao diện chỉnh sửa điểm.

1. Phương thức parseInput:

Phân tích chuỗi đầu vào để tách thông tin về học sinh và điểm số các môn học thành mảng hai chiều.

1. Phương thức conf:

Lọc ra điểm số của các môn Anh, Toán, Văn từ mảng kết quả của phương thức parseInput.

1. Constructor ChinhSuaDiem:

Thiết lập thuộc tính của JFrame (setDefaultCloseOperation, setBounds).

Tạo và cấu hình JPanel (contentPane) để chứa các thành phần giao diện khác.

Thêm các JTextField, JLabel và JButton để nhập ID học sinh, xem và chỉnh sửa điểm số.

1. Cấu trúc giao diện:

Nhập ID học sinh: JTextField để nhập ID học sinh.

Xem điểm: JButton để xem điểm của học sinh theo ID đã nhập.

Chỉnh sửa điểm: Các JTextField để chỉnh sửa điểm của các môn học và JButton để lưu lại thay đổi.

Trở về và Thoát: Các nút JButton để trở về giao diện quản lý điểm và thoát khỏi giao diện chỉnh sửa.

1. Xử lý sự kiện:

Xem điểm: Khi nhấn nút "Xem", hệ thống sẽ tìm điểm theo ID học sinh, phân tích kết quả và hiển thị thông tin học sinh cùng điểm số các môn học.

Chỉnh sửa điểm: Khi nhấn nút "Chỉnh sửa", hệ thống sẽ so sánh điểm số hiện tại với điểm số nhập vào. Nếu có thay đổi, hệ thống sẽ cập nhật điểm số vào cơ sở dữ liệu thông qua phương thức updateDiem của DiemDAO.

QuanLyHocSinh\_Server.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145 | package Server.QuanLyHocSinh;  import java.awt.EventQueue;  import javax.swing.JFrame;import javax.swing.JPanel;import javax.swing.border.EmptyBorder;  import DAO.HocSinhDAO;  import javax.swing.JLabel;import java.awt.Font;import javax.swing.JSeparator;import javax.swing.JList;import javax.swing.JSpinner;import javax.swing.JTree;import javax.swing.JProgressBar;import javax.swing.JScrollBar;import javax.swing.JScrollPane;import javax.swing.JToggleButton;import java.awt.Scrollbar;import javax.swing.JButton;import javax.swing.JTextArea;import javax.swing.JTextField;import java.awt.event.ActionListener;import java.awt.event.ActionEvent;import java.awt.Color;  public class QuanLyHocSinh\_Server extends JFrame {  private static final long serialVersionUID = 1L;  private JPanel contentPane;  private JTextField IDHocSinh;  private static JTextArea textArea = new JTextArea();  public static void main(String[] args) {  EventQueue.invokeLater(new Runnable() {  public void run() {  try {  QuanLyHocSinh\_Server frame = new QuanLyHocSinh\_Server();  frame.setVisible(true);  UpdateDanhSach();  } catch (Exception e) {  e.printStackTrace();  }  }  });  }  public static void appendToTextArea(String text) {  textArea.append(text + "\n");  textArea.setLineWrap(true);  textArea.setWrapStyleWord(true);  }    public static void clearDanhSach() {  textArea.setText("");  }  public static void UpdateDanhSach() {  String DanhSach = HocSinhDAO.printAll();  appendToTextArea(DanhSach);  }  public QuanLyHocSinh\_Server() {  setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  setBounds(100, 100, 450, 300);  contentPane = new JPanel();  contentPane.setBackground(new Color(0, 0, 53));  contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));  setContentPane(contentPane);  contentPane.setLayout(null);  JLabel lblTrangQunL = new JLabel("Trang Quản Lý Học Sinh");  lblTrangQunL.setForeground(new Color(255, 255, 255));  lblTrangQunL.setBackground(new Color(255, 255, 255));  lblTrangQunL.setFont(new Font("Tahoma", Font.BOLD, 20));  lblTrangQunL.setBounds(95, 10, 251, 25);  contentPane.add(lblTrangQunL);  JButton btnNewButton = new JButton("Thêm");  btnNewButton.setBackground(new Color(255, 255, 255));  btnNewButton.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  Them frame = new Them();  frame.setVisible(true);  setVisible(false);  }  });  btnNewButton.setBounds(10, 232, 70, 21);  contentPane.add(btnNewButton);  JButton btnNewButton\_1 = new JButton("Thoát");  btnNewButton\_1.setBackground(new Color(255, 255, 255));  btnNewButton\_1.setForeground(new Color(198, 54, 0));  btnNewButton\_1.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  dispose();  }  });  btnNewButton\_1.setBounds(341, 232, 85, 21);  contentPane.add(btnNewButton\_1);  JButton btnNewButton\_2 = new JButton("Xóa");  btnNewButton\_2.setBackground(new Color(255, 255, 255));  btnNewButton\_2.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  Xoa frame = new Xoa();  frame.setVisible(true);  setVisible(false);  }  });  btnNewButton\_2.setBounds(95, 232, 70, 21);  contentPane.add(btnNewButton\_2);  JButton btnNewButton\_3 = new JButton("Sửa");  btnNewButton\_3.setBackground(new Color(255, 255, 255));  btnNewButton\_3.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  ChinhSua frame = new ChinhSua();  frame.setVisible(true);  setVisible(false);  }  });  btnNewButton\_3.setBounds(184, 232, 70, 21);  contentPane.add(btnNewButton\_3);  JScrollPane scrollPane = new JScrollPane(textArea);  scrollPane.setBounds(10, 73, 416, 149);  contentPane.add(scrollPane);  JLabel lblNewLabel\_1 = new JLabel("Tìm kiếm theo IDHocSinh:");  lblNewLabel\_1.setForeground(new Color(255, 255, 255));  lblNewLabel\_1.setBackground(new Color(255, 255, 255));  lblNewLabel\_1.setBounds(10, 50, 148, 13);  contentPane.add(lblNewLabel\_1);  IDHocSinh = new JTextField();  IDHocSinh.setText("HS001");  IDHocSinh.setColumns(10);  IDHocSinh.setBounds(168, 47, 96, 19);  contentPane.add(IDHocSinh);  JButton btnNewButton\_4 = new JButton("Tìm");  btnNewButton\_4.setBackground(new Color(255, 255, 255));  btnNewButton\_4.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  String ID = IDHocSinh.getText();  String DanhSach = HocSinhDAO.TimHocSinhTheoIDHocSinh(ID);  textArea.setText("");  appendToTextArea(DanhSach);  }  });  btnNewButton\_4.setBounds(270, 45, 57, 21);  contentPane.add(btnNewButton\_4);  }  } |

Mô tả: Mã nguồn này tạo ra một giao diện người dùng cho phép người dùng thực hiện các hoạt động quản lý cơ bản liên quan đến học sinh như thêm, xóa, sửa và tìm kiếm.

Giải thích chi tiết:

1. Khai báo package và import các thư viện cần thiết:

‘package Server.QuanLyHocSinh;’: Lớp được đặt trong package Server.QuanLyHocSinh.

Import các thư viện và lớp cần thiết từ javax.swing, java.awt, java.awt.event, và DAO.HocSinhDAO.

1. Khai báo package và import các thư viện cần thiết:

‘private static final long serialVersionUID = 1L;’: Đảm bảo tính tương thích của serialization.

1. Phương thức main:

Tạo một đối tượng QuanLyHocSinh\_Server và hiển thị nó trên giao diện người dùng.

Gọi phương thức UpdateDanhSach() để cập nhật danh sách học sinh khi ứng dụng được khởi động.

1. Phương thức appendToTextArea:

Thêm văn bản vào JTextArea để hiển thị thông tin.

1. Phương thức clearDanhSach:

Xóa nội dung của JTextArea để làm sạch danh sách học sinh.

1. Phương thức UpdateDanhSach:

Gọi phương thức từ lớp HocSinhDAO để lấy danh sách học sinh và hiển thị chúng trên JTextArea bằng cách gọi appendToTextArea.

1. Constructor QuanLyHocSinh\_Server:

Thiết lập các thuộc tính của JFrame và contentPane.

Thêm các thành phần giao diện như JLabel, JButton, JTextField, và JTextArea.

Xử lý các sự kiện khi người dùng tương tác với các thành phần giao diện như nút "Thêm", "Thoát", "Xóa", "Sửa", và "Tìm".

Sử dụng JScrollPane để thêm thanh cuộn cho JTextArea nếu cần.

ChinhSua.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149  150  151  152  153  154  155  156  157  158  159  160  161  162  163  164  165  166  167  168  169  170  171  172  173  174  175  176  177  178  179  180  181  182  183  184  185  186  187  188  189  190  191  192  193  194  195  196  197  198  199  200  201  202  203  204  205  206  207  208  209  210  211  212  213  214  215  216  217  218  219  220  221  222  223  224  225  226  227  228  229  230  231  232  233  234  235  236  237  238  239  240  241  242  243  244  245  246  247  248  249  250  251  252  253  254  255 | package Server.QuanLyHocSinh;  import java.awt.EventQueue;  import javax.swing.JFrame;import javax.swing.JPanel;import javax.swing.border.EmptyBorder;  import DAO.HocSinhDAO;  import javax.swing.JLabel;import javax.swing.JOptionPane;  import java.awt.Font;import javax.swing.JTextField;import javax.swing.JButton;import java.awt.event.ActionListener;import java.awt.event.ActionEvent;import java.awt.Color;  public class ChinhSua extends JFrame {  private static final long serialVersionUID = 1L;  private JPanel contentPane;  private JTextField IDHocSinh;  private JTextField textField\_NamHoc;  private JTextField textField\_DiaChi;  private JTextField textField\_HoTen;  /\*\* \* Launch the application. \*/  public static void main(String[] args) {  EventQueue.invokeLater(new Runnable() {  public void run() {  try {  ChinhSua frame = new ChinhSua();  frame.setVisible(true);  } catch (Exception e) {  e.printStackTrace();  }  }  });  }  /\*\* \* Create the frame. \*/  public static String[][] convertTo2DArray(String info) {  // Tách thông tin thành các dòng  String[] lines = info.split("\n");  // Khởi tạo mảng hai chiều để lưu trữ thông tin  String[][] result = new String[lines.length][2];  // Lặp qua từng dòng và chuyển đổi thành cặp key-value  for (int i = 0; i < lines.length; i++) {  String[] parts = lines[i].split(": ");  result[i][0] = parts[0]; // Key  result[i][1] = parts[1]; // Value  }  return result;  }  public ChinhSua() {  setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  setBounds(100, 100, 450, 300);  contentPane = new JPanel();  contentPane.setBackground(new Color(0, 0, 53));  contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));  setContentPane(contentPane);  contentPane.setLayout(null);    JLabel lblNewLabel\_1 = new JLabel("Chỉnh Sửa Học Sinh");  lblNewLabel\_1.setForeground(new Color(255, 255, 255));  lblNewLabel\_1.setFont(new Font("Tahoma", Font.BOLD, 20));  lblNewLabel\_1.setBounds(115, 10, 199, 24);  contentPane.add(lblNewLabel\_1);    JLabel lblNewLabel\_1\_1 = new JLabel("Tìm kiếm theo IDHocSinh:");  lblNewLabel\_1\_1.setForeground(new Color(255, 255, 255));  lblNewLabel\_1\_1.setBounds(10, 49, 148, 13);  contentPane.add(lblNewLabel\_1\_1);    IDHocSinh = new JTextField();  IDHocSinh.setText("HS001");  IDHocSinh.setColumns(10);  IDHocSinh.setBounds(168, 46, 96, 19);  contentPane.add(IDHocSinh);    JButton btnNewButton\_4 = new JButton("Xem");  btnNewButton\_4.setBackground(new Color(255, 255, 255));  btnNewButton\_4.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  String ID = IDHocSinh.getText();  String DanhSach = HocSinhDAO.TimHocSinhTheoIDHocSinh(ID);  String[][] temp = convertTo2DArray(DanhSach);    textField\_HoTen.setText(temp[1][1]);  textField\_DiaChi.setText(temp[2][1]);  textField\_NamHoc.setText(temp[3][1]);  }  });  btnNewButton\_4.setBounds(270, 44, 71, 21);  contentPane.add(btnNewButton\_4);    JLabel lblNewLabel\_4 = new JLabel("Năm nhập học:");  lblNewLabel\_4.setForeground(new Color(255, 255, 255));  lblNewLabel\_4.setBounds(62, 134, 85, 13);  contentPane.add(lblNewLabel\_4);    textField\_NamHoc = new JTextField();  textField\_NamHoc.setColumns(10);  textField\_NamHoc.setBounds(160, 131, 55, 19);  contentPane.add(textField\_NamHoc);    textField\_DiaChi = new JTextField();  textField\_DiaChi.setColumns(10);  textField\_DiaChi.setBounds(160, 105, 266, 19);  contentPane.add(textField\_DiaChi);    JLabel lblNewLabel\_3 = new JLabel("Địa Chỉ:");  lblNewLabel\_3.setForeground(new Color(255, 255, 255));  lblNewLabel\_3.setBounds(62, 111, 64, 13);  contentPane.add(lblNewLabel\_3);    JLabel lblNewLabel\_1\_2 = new JLabel("Họ Tên:");  lblNewLabel\_1\_2.setForeground(new Color(255, 255, 255));  lblNewLabel\_1\_2.setBounds(62, 85, 64, 13);  contentPane.add(lblNewLabel\_1\_2);    textField\_HoTen = new JTextField();  textField\_HoTen.setColumns(10);  textField\_HoTen.setBounds(160, 82, 104, 19);  contentPane.add(textField\_HoTen);    JButton btnNewButton = new JButton("Trở về");  btnNewButton.setBackground(new Color(255, 255, 255));  btnNewButton.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  QuanLyHocSinh\_Server frame = new QuanLyHocSinh\_Server();  frame.setVisible(true);  frame.clearDanhSach();  frame.UpdateDanhSach();  setVisible(false);  }  });  btnNewButton.setBounds(10, 232, 85, 21);  contentPane.add(btnNewButton);    JButton btnNewButton\_1 = new JButton("Thoát");  btnNewButton\_1.setBackground(new Color(255, 255, 255));  btnNewButton\_1.setForeground(new Color(198, 54, 0));  btnNewButton\_1.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  dispose();  }  });  btnNewButton\_1.setBounds(341, 232, 85, 21);  contentPane.add(btnNewButton\_1);    JButton btnNewButton\_2 = new JButton("Chỉnh sửa");  btnNewButton\_2.setBackground(new Color(255, 255, 255));  btnNewButton\_2.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  String HoTen = textField\_HoTen.getText();  String DiaChi = textField\_DiaChi.getText();  String NamHoc = textField\_NamHoc.getText();    if(HoTen.equals("") && DiaChi.equals("") && NamHoc.equals("")) {  JOptionPane.showMessageDialog(null, "Vui lòng xem học sinh để bắt đầu chỉnh sửa!");  }else {  String ID = IDHocSinh.getText();  String DanhSach = HocSinhDAO.TimHocSinhTheoIDHocSinh(ID);  String[][] temp = convertTo2DArray(DanhSach);  if(HoTen.equals(temp[1][1]) && DiaChi.equals(temp[2][1]) && NamHoc.equals(temp[3][1])) {  JOptionPane.showMessageDialog(null, "Bạn chưa chỉnh sửa!");  }else if(HoTen.equals(temp[1][1]) == false && DiaChi.equals(temp[2][1]) && NamHoc.equals(temp[3][1])) {  HocSinhDAO.editHoTenHocSinh(ID, HoTen);  JOptionPane.showMessageDialog(null, "Bạn đã chỉnh sửa Họ Tên!");  }else if(HoTen.equals(temp[1][1]) && DiaChi.equals(temp[2][1]) == false && NamHoc.equals(temp[3][1])) {  HocSinhDAO.editDiaChiHocSinh(ID, DiaChi);  JOptionPane.showMessageDialog(null, "Bạn đã chỉnh sửa Địa Chỉ!");  }else if(HoTen.equals(temp[1][1]) && DiaChi.equals(temp[2][1]) && NamHoc.equals(temp[3][1]) == false) {  HocSinhDAO.editNamHocHocSinh(ID, Integer.parseInt(NamHoc));  JOptionPane.showMessageDialog(null, "Bạn đã chỉnh sửa Năm Học!");  }else if(HoTen.equals(temp[1][1]) == false && DiaChi.equals(temp[2][1]) == false && NamHoc.equals(temp[3][1])) {  HocSinhDAO.editHoTenHocSinh(ID, HoTen);  HocSinhDAO.editDiaChiHocSinh(ID, DiaChi);  JOptionPane.showMessageDialog(null, "Bạn đã chỉnh sửa Họ Tên và Địa Chỉ!");  }else if(HoTen.equals(temp[1][1]) == false && DiaChi.equals(temp[2][1]) && NamHoc.equals(temp[3][1]) == false) {  HocSinhDAO.editHoTenHocSinh(ID, HoTen);  HocSinhDAO.editNamHocHocSinh(ID, Integer.parseInt(NamHoc));  JOptionPane.showMessageDialog(null, "Bạn đã chỉnh sửa Họ Tên và Năm Học!");  }else if(HoTen.equals(temp[1][1]) && DiaChi.equals(temp[2][1]) == false && NamHoc.equals(temp[3][1]) == false) {  HocSinhDAO.editDiaChiHocSinh(ID, DiaChi);  HocSinhDAO.editNamHocHocSinh(ID, Integer.parseInt(NamHoc));  JOptionPane.showMessageDialog(null, "Bạn đã chỉnh sửa Địa Chỉ và Năm Học!");  }else if(HoTen.equals(temp[1][1])== false && DiaChi.equals(temp[2][1]) == false && NamHoc.equals(temp[3][1]) == false) {  HocSinhDAO.editHoTenHocSinh(ID, HoTen);  HocSinhDAO.editDiaChiHocSinh(ID, DiaChi);  HocSinhDAO.editNamHocHocSinh(ID, Integer.parseInt(NamHoc));  JOptionPane.showMessageDialog(null, "Bạn đã chỉnh sửa cả 3 thông tin!");  }  String ID1 = IDHocSinh.getText();  String DanhSach1 = HocSinhDAO.TimHocSinhTheoIDHocSinh(ID1);  String[][] temp1 = convertTo2DArray(DanhSach1);    textField\_HoTen.setText(temp1[1][1]);  textField\_DiaChi.setText(temp1[2][1]);  textField\_NamHoc.setText(temp1[3][1]);  }  }  });  btnNewButton\_2.setBounds(115, 173, 96, 21);  contentPane.add(btnNewButton\_2);  }  } |

Mô tả: Đây là một ứng dụng Java Swing cho phép người dùng chỉnh sửa thông tin của học sinh.

Giải thích chi tiết:

1. Khai báo package và import các thư viện cần thiết:

‘package Server.QuanLyHocSinh;’ :Lớp được đặt trong package Server.QuanLyHocSinh.

Import các thư viện và lớp cần thiết từ javax.swing, java.awt, java.awt.event, và DAO.HocSinhDAO.

1. Khai báo lớp ChinhSua kế thừa từ JFrame:

‘private static final long serialVersionUID = 1L;’: Đảm bảo tính tương thích của serialization.

1. Phương thức main:

Tạo một đối tượng ChinhSua và hiển thị nó trên giao diện người dùng.

1. Phương thức convertTo2DArray:

Chuyển đổi thông tin học sinh từ chuỗi văn bản sang mảng hai chiều.

1. Constructor ChinhSua:

Thiết lập các thuộc tính của JFrame và contentPane.

Thêm các thành phần giao diện như JLabel, JButton, và JTextField.

Xử lý sự kiện khi người dùng tương tác với các thành phần giao diện như nút "Xem", "Trở về", "Thoát", và "Chỉnh sửa".

Gọi các phương thức từ lớp HocSinhDAO để tìm và chỉnh sửa thông tin học sinh dựa trên ID Hoc Sinh. Hiển thị thông báo phản hồi tương ứng với kết quả chỉnh sửa.

Cập nhật thông tin học sinh sau khi chỉnh sửa thành công.

Them.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149  150  151  152  153  154  155  156  157 | package Server.QuanLyHocSinh;  import java.awt.EventQueue;  import javax.swing.JFrame;import javax.swing.JPanel;import javax.swing.border.EmptyBorder;  import DAO.HocSinhDAO;import Server.QuanLyDiem.QuanLyDiem\_Server;  import javax.swing.JLabel;import javax.swing.JOptionPane;  import java.awt.Font;import javax.swing.JTextField;import javax.swing.JButton;import java.awt.event.ActionListener;import java.awt.event.ActionEvent;import java.awt.Color;  public class Them extends JFrame {  private static final long serialVersionUID = 1L;  private JPanel contentPane;  private JTextField IDHocSinh;  private JTextField HoTen;  private JTextField DiaChi;  private JTextField Nam;  /\*\* \* Launch the application. \*/  public static void main(String[] args) {  EventQueue.invokeLater(new Runnable() {  public void run() {  try {  Them frame = new Them();  frame.setVisible(true);  } catch (Exception e) {  e.printStackTrace();  }  }  });  }  /\*\* \* Create the frame. \*/  public Them() {  setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  setBounds(100, 100, 450, 300);  contentPane = new JPanel();  contentPane.setBackground(new Color(0, 0, 53));  contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));  setContentPane(contentPane);  contentPane.setLayout(null);    JLabel lblTrangThmHc = new JLabel("Thêm Học Sinh");  lblTrangThmHc.setForeground(new Color(255, 255, 255));  lblTrangThmHc.setFont(new Font("Tahoma", Font.BOLD, 20));  lblTrangThmHc.setBounds(139, 10, 222, 25);  contentPane.add(lblTrangThmHc);    JLabel lblNewLabel\_1 = new JLabel("Họ Tên:");  lblNewLabel\_1.setForeground(new Color(255, 255, 255));  lblNewLabel\_1.setBounds(54, 92, 64, 13);  contentPane.add(lblNewLabel\_1);    JLabel lblNewLabel\_2 = new JLabel("IDHocSinh:");  lblNewLabel\_2.setForeground(new Color(255, 255, 255));  lblNewLabel\_2.setBounds(54, 69, 64, 13);  contentPane.add(lblNewLabel\_2);    JLabel lblNewLabel\_3 = new JLabel("Địa Chỉ:");  lblNewLabel\_3.setForeground(new Color(255, 255, 255));  lblNewLabel\_3.setBounds(54, 115, 64, 13);  contentPane.add(lblNewLabel\_3);    JLabel lblNewLabel\_4 = new JLabel("Năm nhập học:");  lblNewLabel\_4.setForeground(new Color(255, 255, 255));  lblNewLabel\_4.setBounds(54, 141, 85, 13);  contentPane.add(lblNewLabel\_4);    IDHocSinh = new JTextField();  IDHocSinh.setBounds(152, 66, 55, 19);  contentPane.add(IDHocSinh);  IDHocSinh.setColumns(10);    HoTen = new JTextField();  HoTen.setBounds(152, 89, 154, 19);  contentPane.add(HoTen);  HoTen.setColumns(10);    DiaChi = new JTextField();  DiaChi.setBounds(152, 112, 154, 19);  contentPane.add(DiaChi);  DiaChi.setColumns(10);    Nam = new JTextField();  Nam.setBounds(152, 138, 55, 19);  contentPane.add(Nam);  Nam.setColumns(10);    JButton btnNewButton = new JButton("Trở về");  btnNewButton.setBackground(new Color(255, 255, 255));  btnNewButton.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  QuanLyHocSinh\_Server frame = new QuanLyHocSinh\_Server();  frame.setVisible(true);  frame.clearDanhSach();  frame.UpdateDanhSach();  setVisible(false);  }  });  btnNewButton.setBounds(10, 232, 85, 21);  contentPane.add(btnNewButton);    JButton btnNewButton\_1 = new JButton("Thoát");  btnNewButton\_1.setBackground(new Color(255, 255, 255));  btnNewButton\_1.setForeground(new Color(198, 54, 0));  btnNewButton\_1.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  dispose();  }  });  btnNewButton\_1.setBounds(341, 232, 85, 21);  contentPane.add(btnNewButton\_1);    JButton btnNewButton\_2 = new JButton("Thêm");  btnNewButton\_2.setBackground(new Color(255, 255, 255));  btnNewButton\_2.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  String ID = IDHocSinh.getText();  String Name = HoTen.getText();  String Address = DiaChi.getText();  String Year = Nam.getText();// System.out.println();  if(ID.equals("")) {  JOptionPane.showMessageDialog(null, "Bạn để trống ID học sinh");  }else if(Name.equals("")) {  JOptionPane.showMessageDialog(null, "Bạn để trống họ tên học sinh");  }else if(Address.equals("")) {  JOptionPane.showMessageDialog(null, "Bạn để trống địa chỉ học sinh");  }else if(Year.equals("")) {  JOptionPane.showMessageDialog(null, "Bạn để trống năm nhập học của học sinh");  }else {  if(HocSinhDAO.themHocSinh(ID, Name, Address, Year, "123")) {  JOptionPane.showMessageDialog(null, "Thêm học sinh (" + Name+ ") thành công!");  }  else {  JOptionPane.showMessageDialog(null, "Thêm học sinh (" + Name+ ") thất bại!");  }  }  }  });  btnNewButton\_2.setBounds(163, 178, 85, 21);  contentPane.add(btnNewButton\_2);  }  } |

Mô tả: Đây là một ứng dụng Java Swing cho phép người dùng thêm học sinh.

Giải thích chi tiết:

1. Khai báo package và import các thư viện cần thiết:

‘package Server.QuanLyHocSinh;’ : Lớp được đặt trong package Server.QuanLyHocSinh.

Import các thư viện và lớp cần thiết từ javax.swing, java.awt, java.awt.event, và DAO.HocSinhDAO.

1. Khai báo lớp Them kế thừa từ JFrame:

‘private static final long serialVersionUID = 1L;’: Đảm bảo tính tương thích của serialization.

1. Phương thức main:

Tạo một đối tượng Them và hiển thị nó trên giao diện người dùng.

1. Constructor Them:

Thiết lập các thuộc tính của JFrame và contentPane.

Thêm các thành phần giao diện như JLabel, JButton, và JTextField để người dùng nhập thông tin học sinh mới.

Xử lý sự kiện khi người dùng tương tác với các thành phần giao diện như nút "Trở về", "Thoát", và "Thêm".

Gọi phương thức themHocSinh từ lớp HocSinhDAO để thêm một học sinh mới vào cơ sở dữ liệu, dựa trên thông tin nhập vào từ người dùng. Hiển thị thông báo phản hồi tương ứng với kết quả thêm mới học sinh.

Xoa.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149 | package Server.QuanLyHocSinh;  import java.awt.EventQueue;  import javax.swing.JFrame;import javax.swing.JPanel;import javax.swing.border.EmptyBorder;  import DAO.HocSinhDAO;  import javax.swing.JLabel;import javax.swing.JOptionPane;  import java.awt.Font;import javax.swing.JTextField;import javax.swing.JButton;import javax.swing.JTextArea;import java.awt.event.ActionListener;import java.awt.event.ActionEvent;import java.awt.Color;  public class Xoa extends JFrame {  private static final long serialVersionUID = 1L;  private JPanel contentPane;  private JTextField IDHocSinh;  static JTextArea textArea = new JTextArea();  /\*\* \* Launch the application. \*/  public static void main(String[] args) {  EventQueue.invokeLater(new Runnable() {  public void run() {  try {  Xoa frame = new Xoa();  frame.setVisible(true);  } catch (Exception e) {  e.printStackTrace();  }  }  });  }  /\*\* \* Create the frame. \*/  public static void appendToTextArea(String text) {  textArea.append(text + "\n");  textArea.setLineWrap(true);  textArea.setWrapStyleWord(true);  }  public Xoa() {  setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  setBounds(100, 100, 450, 300);  contentPane = new JPanel();  contentPane.setBackground(new Color(0, 0, 53));  contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));  setContentPane(contentPane);  contentPane.setLayout(null);    JLabel lblTrangXaHc = new JLabel("Xóa Học Sinh");  lblTrangXaHc.setForeground(new Color(255, 255, 255));  lblTrangXaHc.setFont(new Font("Tahoma", Font.BOLD, 20));  lblTrangXaHc.setBounds(144, 10, 222, 25);  contentPane.add(lblTrangXaHc);    JLabel lblNewLabel\_1 = new JLabel("Tìm kiếm theo IDHocSinh:");  lblNewLabel\_1.setForeground(new Color(255, 255, 255));  lblNewLabel\_1.setBounds(10, 50, 148, 13);  contentPane.add(lblNewLabel\_1);    IDHocSinh = new JTextField();  IDHocSinh.setText("HS001");  IDHocSinh.setColumns(10);  IDHocSinh.setBounds(168, 47, 96, 19);  contentPane.add(IDHocSinh);    JButton btnNewButton\_4 = new JButton("Tìm");  btnNewButton\_4.setBackground(new Color(255, 255, 255));  btnNewButton\_4.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  String ID = IDHocSinh.getText();  String DanhSach = HocSinhDAO.TimHocSinhTheoIDHocSinh(ID);  textArea.setText("");  appendToTextArea(DanhSach);  }  });  btnNewButton\_4.setBounds(270, 45, 57, 21);  contentPane.add(btnNewButton\_4);      textArea.setBounds(11, 74, 415, 83);  contentPane.add(textArea);    JButton btnNewButton = new JButton("Xóa Học Sinh Này");  btnNewButton.setBackground(new Color(255, 255, 255));  btnNewButton.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  String ID = IDHocSinh.getText();  String text = textArea.getText();  if(text.equals("")) {  JOptionPane.showMessageDialog(null, "Mời bạn nhấn nút (Tìm) để xác nhận học sinh cần xóa");  }else {  if(HocSinhDAO.xoaHocSinh(ID)) {  JOptionPane.showMessageDialog(null, "Xóa học sinh có ID ("+ ID +") thành công!");  textArea.setText("");  }else {  JOptionPane.showMessageDialog(null, "Xóa học sinh có ID ("+ ID +") thất bại!");  }  }  }  });  btnNewButton.setBounds(10, 167, 148, 21);  contentPane.add(btnNewButton);    JButton btnNewButton\_1 = new JButton("Chỉnh Sửa");  btnNewButton\_1.setBackground(new Color(255, 255, 255));  btnNewButton\_1.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  ChinhSua frame = new ChinhSua();  frame.setVisible(true);  setVisible(false);  }  });  btnNewButton\_1.setBounds(168, 167, 96, 21);  contentPane.add(btnNewButton\_1);    JButton btnNewButton\_2 = new JButton("Trở về");  btnNewButton\_2.setBackground(new Color(255, 255, 255));  btnNewButton\_2.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  QuanLyHocSinh\_Server frame = new QuanLyHocSinh\_Server();  frame.setVisible(true);  frame.clearDanhSach();  frame.UpdateDanhSach();  setVisible(false);  }  });  btnNewButton\_2.setBounds(10, 232, 85, 21);  contentPane.add(btnNewButton\_2);    JButton btnNewButton\_3 = new JButton("Thoát");  btnNewButton\_3.setBackground(new Color(255, 255, 255));  btnNewButton\_3.setForeground(new Color(198, 54, 0));  btnNewButton\_3.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  dispose();  }  });  btnNewButton\_3.setBounds(341, 232, 85, 21);  contentPane.add(btnNewButton\_3);  }  } |

Mô tả: Đây là một ứng dụng Java Swing cho phép người dùng xóa thông tin của học sinh.

Giải thích chi tiết:

1. Khai báo package và import các thư viện cần thiết:

‘package Server.QuanLyHocSinh;’ : Lớp được đặt trong package Server.QuanLyHocSinh.

Import các thư viện và lớp cần thiết từ javax.swing, java.awt, java.awt.event, và DAO.HocSinhDAO.

1. Khai báo lớp Xoa kế thừa từ JFrame:

‘private static final long serialVersionUID = 1L;’ : Đảm bảo tính tương thích của serialization.

1. Phương thức main:

Tạo một đối tượng Xoa và hiển thị nó trên giao diện người dùng.

1. Constructor Xoa:

Thiết lập các thuộc tính của JFrame và contentPane.

Thêm các thành phần giao diện như JLabel, JButton, và JTextField để người dùng nhập ID học sinh cần xóa.

Hiển thị kết quả tìm kiếm học sinh theo ID trên JTextArea.

Xử lý sự kiện khi người dùng tương tác với các thành phần giao diện như nút "Tìm", "Xóa Học Sinh Này", "Chỉnh Sửa", "Trở về", và "Thoát".

Gọi phương thức xoaHocSinh từ lớp HocSinhDAO để xóa học sinh theo ID nhập từ người dùng. Hiển thị thông báo phản hồi tương ứng với kết quả xóa học sinh.

Server\_TCP.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149  150  151  152  153  154  155  156  157  158  159  160  161  162  163  164  165  166  167  168  169  170  171  172  173  174  175 | package Server;  import javax.swing.\*;import java.awt.\*;import java.awt.event.ActionEvent;import java.awt.event.ActionListener;import java.io.\*;import java.net.ServerSocket;import java.net.Socket;import java.util.ArrayList;  public class Server\_TCP extends JFrame {  private ServerSocket serverSocket;  private final ArrayList<ClientHandler> clients;  private boolean serverRunning;  private JTextArea messageArea;  public Server\_TCP() {  setTitle("Server");  setSize(400, 300);  setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  clients = new ArrayList<>();  serverRunning = true;  JPanel panel = new JPanel(new BorderLayout());  messageArea = new JTextArea();  messageArea.setEditable(false);  panel.add(new JScrollPane(messageArea), BorderLayout.CENTER);  JPanel bottomPanel = new JPanel(new BorderLayout());  JTextField messageField = new JTextField();  bottomPanel.add(messageField, BorderLayout.CENTER);  JButton sendButton = new JButton("Send");  sendButton.setBackground(new Color(255, 255, 255));  sendButton.addActionListener(new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {  sendMessageToAll("Server: " + messageField.getText());  messageArea.append("You: " + messageField.getText() + "\n");  messageField.setText("");  }  });  bottomPanel.add(sendButton, BorderLayout.EAST);  JButton btnNewButton = new JButton("Thoát");  btnNewButton.setBackground(new Color(255, 255, 255));  btnNewButton.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  serverRunning = false;  try {  serverSocket.close();  for (ClientHandler client : clients) {  client.closeConnection();  }  } catch (IOException e1) {  e1.printStackTrace();  }  dispose();  }  });  btnNewButton.setForeground(new Color(198, 54, 0));  bottomPanel.add(btnNewButton, BorderLayout.WEST);  panel.add(bottomPanel, BorderLayout.SOUTH);  getContentPane().add(panel);  try {  serverSocket = new ServerSocket(9999);  new Thread(new ServerHandler()).start();  messageArea.append("Đã khởi động Server đang đợi client kết nối...\n");  } catch (IOException e) {  e.printStackTrace();  }  }  private void sendMessageToAll(String message) {  for (ClientHandler client : clients) {  client.sendMessage(message);  }  }  private class ServerHandler implements Runnable {  @Override  public void run() {  while (serverRunning) {  try {  Socket clientSocket = serverSocket.accept();  if (!serverRunning) {  clientSocket.close();  break;  }  ClientHandler clientHandler = new ClientHandler(clientSocket);  clients.add(clientHandler);  new Thread(clientHandler).start();  messageArea.append("Địa chỉ: " + clientSocket.getInetAddress().getHostAddress() + " đã kết nối!\n");  } catch (IOException e) {  if (!serverRunning) {  } else {  e.printStackTrace();  }  }  }  }  }  private class ClientHandler implements Runnable {  private Socket clientSocket;  private BufferedReader in;  private PrintWriter out;  private String clientName;  public ClientHandler(Socket clientSocket) {  this.clientSocket = clientSocket;  try {  in = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));  out = new PrintWriter(clientSocket.getOutputStream(), true);  out.println("Hãy nhập tên và bấm nút 'Send' để chat: ");  clientName = in.readLine();  sendMessageToAll(clientName + " đã tham gia kênh chat!");  } catch (IOException e) {  e.printStackTrace();  }  }  public void sendMessage(String message) {  out.println(message);  }  public void sendMessageToAll(String message) {  for (ClientHandler client : clients) {  if (!client.equals(this)) {  client.sendMessage(clientName + ": " + message);  messageArea.append(clientName + ": " + message + "\n"); // In tin nhắn của client ra messageArea  }  }  }  @Override  public void run() {  try {  String message;  while ((message = in.readLine()) != null) {  sendMessageToAll(message);  }  clientSocket.close();  } catch (IOException e) {  e.printStackTrace();  }  }  public void closeConnection() {  try {  if (clientSocket != null && !clientSocket.isClosed()) {  clientSocket.close();  }  } catch (IOException e) {  e.printStackTrace();  }  }  }  public static void main(String[] args) {  SwingUtilities.invokeLater(new Runnable() {  @Override  public void run() {  new Server\_TCP().setVisible(true);  }  });  }  } |

Mô tả: Mã nguồn này cung cấp một cách thức đơn giản và hiệu quả để xây dựng một ứng dụng Chat Server sử dụng giao thức TCP, cho phép nhiều client kết nối và giao tiếp với nhau thông qua một server trung gian.

Giải thích chi tiết:

1. Khai báo package và import các thư viện cần thiết:

‘package Server;’ : Lớp được đặt trong package Server.

Import các thư viện và lớp cần thiết từ javax.swing, java.awt, java.awt.event, java.io, và java.net.

1. Khai báo lớp Server\_TCP kế thừa từ JFrame:

private ServerSocket serverSocket;: Đối tượng ServerSocket để lắng nghe kết nối từ client.

private final ArrayList<ClientHandler> clients;: Danh sách các client đã kết nối.

private boolean serverRunning;: Biến đánh dấu trạng thái của server (đang chạy hoặc không).

private JTextArea messageArea;: TextArea để hiển thị tin nhắn giữa các client.

1. Constructor Server\_TCP:

Khởi tạo cấu trúc của giao diện người dùng với JTextArea, JTextField, và các JButton để nhận và gửi tin nhắn.

Khởi tạo ServerSocket để lắng nghe kết nối từ client.

Tạo một luồng mới để xử lý các kết nối đến từ client.

1. Phương thức sendMessageToAll:

Gửi tin nhắn tới tất cả các client đã kết nối.

1. Lớp ServerHandler:

Một luồng riêng để xử lý các kết nối từ client.

Lắng nghe các kết nối từ client và thêm client mới vào danh sách.

Hiển thị thông báo khi một client mới kết nối thành công.

1. Lớp ClientHandler:

Một luồng riêng để xử lý tin nhắn từ client.

Gửi tin nhắn từ client đến tất cả các client khác.

Đóng kết nối khi client ngắt kết nối.

1. Phương thức main:

Tạo một đối tượng Server\_TCP và hiển thị nó trên giao diện người dùng.

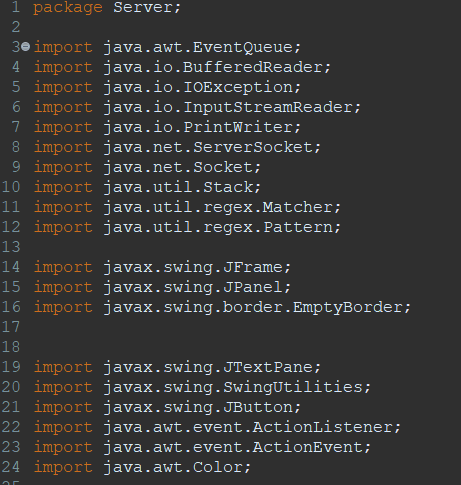
Server\_TCP\_MayTinh.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149  150  151  152  153  154  155  156  157  158  159  160  161  162  163  164  165  166  167  168  169  170  171  172  173  174  175  176  177  178  179  180  181  182  183  184  185  186  187  188  189  190  191  192  193  194  195  196  197  198  199  200  201  202  203  204  205  206  207  208  209  210  211  212  213  214  215  216  217  218  219  220  221  222  223  224  225  226  227  228  229  230  231  232  233  234  235  236  237  238  239  240  241  242  243  244  245  246  247  248  249  250  251  252  253  254  255 | package Server;  import java.awt.EventQueue;import java.io.BufferedReader;import java.io.IOException;import java.io.InputStreamReader;import java.io.PrintWriter;import java.net.ServerSocket;import java.net.Socket;import java.util.Stack;import java.util.regex.Matcher;import java.util.regex.Pattern;  import javax.swing.JFrame;import javax.swing.JPanel;import javax.swing.border.EmptyBorder;  import javax.swing.JTextPane;import javax.swing.SwingUtilities;import javax.swing.JButton;import java.awt.event.ActionListener;import java.awt.event.ActionEvent;import java.awt.Color;  public class Server\_TCP\_MayTinh extends JFrame {  private static final long serialVersionUID = 1L;  private JPanel contentPane;  private JTextPane textPane;  private ServerSocket serverSocket;  private boolean serverRunning;  /\*\* \* Launch the application. \*/  public static void main(String[] args) {  EventQueue.invokeLater(new Runnable() {  public void run() {  try {  Server\_TCP\_MayTinh frame = new Server\_TCP\_MayTinh();  frame.setVisible(true);  frame.startServer();  } catch (Exception e) {  e.printStackTrace();  }  }  });  }  /\*\* \* Create the frame. \*/  public Server\_TCP\_MayTinh() {  setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  setBounds(100, 100, 450, 300);  contentPane = new JPanel();  contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));  setContentPane(contentPane);  contentPane.setLayout(null);    textPane = new JTextPane();  textPane.setBounds(10, 34, 416, 189);  contentPane.add(textPane);    JButton btnNewButton = new JButton("Thoát");  btnNewButton.setForeground(new Color(198, 54, 0));  btnNewButton.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  stopServer();  dispose();  }  });  btnNewButton.setBounds(341, 233, 85, 21);  contentPane.add(btnNewButton);  }  /\*\* \* Start the TCP server. \*/  public void startServer() {  serverRunning = true;  new Thread(new Runnable() {  @Override  public void run() {  try {  serverSocket = new ServerSocket(12345);  updateTextPane("Server đã khởi động, đang đợi kết nối.");  while (serverRunning) {  Socket clientSocket = serverSocket.accept();  updateTextPane("Client đã kết nối.");  // Xử lý client connection  new Thread(new ClientHandler(clientSocket)).start();  }  } catch (IOException e) {  if (!serverRunning) {  updateTextPane("Server đã dừng.");  } else {  e.printStackTrace();  }  } finally {  if (serverSocket != null && !serverSocket.isClosed()) {  try {  serverSocket.close();  } catch (IOException e) {  e.printStackTrace();  }  }  }  }  }).start();  }  /\*\* \* Stop the TCP server. \*/  public void stopServer() {  serverRunning = false;  if (serverSocket != null && !serverSocket.isClosed()) {  try {  serverSocket.close();  } catch (IOException e) {  e.printStackTrace();  }  }  }  /\*\* \* Update the JTextPane with a message. \*/  private void updateTextPane(final String message) {  SwingUtilities.invokeLater(new Runnable() {  @Override  public void run() {  textPane.setText(textPane.getText() + "\n" + message);  }  });  }  public static String convertPercentage(String expression) {  // Regular expression to match percentage values (e.g., 10%)  String regex = "(\\d+\\.?\\d\*)%";  Pattern pattern = Pattern.compile(regex);  Matcher matcher = pattern.matcher(expression);  StringBuffer result = new StringBuffer();  while (matcher.find()) {  String numericPart = matcher.group(1);  double decimalValue = Double.parseDouble(numericPart) / 100;  matcher.appendReplacement(result, String.valueOf(decimalValue));  }  matcher.appendTail(result);  return result.toString();  }  public static double evaluate(String expression) throws Exception {  Stack<Double> values = new Stack<>();  Stack<Character> ops = new Stack<>();  for (int i = 0; i < expression.length(); i++) {  if (expression.charAt(i) == ' ')  continue;  if (expression.charAt(i) >= '0' && expression.charAt(i) <= '9' || expression.charAt(i) == '.') {  StringBuilder sbuf = new StringBuilder();  while (i < expression.length() && (expression.charAt(i) >= '0' && expression.charAt(i) <= '9' || expression.charAt(i) == '.'))  sbuf.append(expression.charAt(i++));  values.push(Double.parseDouble(sbuf.toString()));  i--;  } else if (expression.charAt(i) == '(') {  ops.push(expression.charAt(i));  } else if (expression.charAt(i) == ')') {  while (ops.peek() != '(')  values.push(applyOp(ops.pop(), values.pop(), values.pop()));  ops.pop();  } else if (expression.charAt(i) == '+' || expression.charAt(i) == '-' ||  expression.charAt(i) == 'x' || expression.charAt(i) == ':' ||  expression.charAt(i) == '%') {  while (!ops.empty() && hasPrecedence(expression.charAt(i), ops.peek()))  values.push(applyOp(ops.pop(), values.pop(), values.pop()));  ops.push(expression.charAt(i));  }  }  while (!ops.empty())  values.push(applyOp(ops.pop(), values.pop(), values.pop()));  return values.pop();  }  public static boolean hasPrecedence(char op1, char op2) {  if (op2 == '(' || op2 == ')')  return false;  if ((op1 == 'x' || op1 == ':' || op1 == '%') && (op2 == '+' || op2 == '-'))  return false;  else  return true;  }  public static double applyOp(char op, double b, double a) {  switch (op) {  case '+':  return a + b;  case '-':  return a - b;  case 'x':  return a \* b;  case ':':  if (b == 0)  throw new UnsupportedOperationException("Không thể chia cho 0");  return a / b;  case '%':  return a \* (b / 100);  }  return 0;  }  private class ClientHandler implements Runnable {  private Socket clientSocket;  private BufferedReader in;  private PrintWriter out;  public ClientHandler(Socket clientSocket) {  this.clientSocket = clientSocket;  try {  in = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));  out = new PrintWriter(clientSocket.getOutputStream(), true);  } catch (IOException e) {  e.printStackTrace();  }  }  @Override  public void run() {  try {  String clientMessage;  while ((clientMessage = in.readLine()) != null) {  updateTextPane("Nhận từ client: " + clientMessage);  String convertedExpression = convertPercentage(clientMessage);  double result = evaluate(convertedExpression);  out.println(clientMessage + " = " + result);  }  clientSocket.close();  } catch (Exception e) {  e.printStackTrace();  } finally {  try {  if (clientSocket != null && !clientSocket.isClosed()) {  clientSocket.close();  }  } catch (IOException e) {  e.printStackTrace();  }  }  }  }  } |

Mô tả: Ứng dụng này cung cấp một máy chủ TCP đơn giản có khả năng xử lý các biểu thức toán học từ các client và trả về kết quả.

Giải thích chi tiết:

1. Khai báo Package và Import:



Hình 30: Khai báo thư viện Server\_TCP\_MayTinh.java

1. Kế thừa từ JFrame:

Lớp Server\_TCP\_MayTinh kế thừa từ JFrame, cho phép tạo giao diện người dùng.

1. Phương thức Main:

Sử dụng EventQueue.invokeLater để đảm bảo việc khởi tạo giao diện xảy ra trên luồng sự kiện AWT.

Tạo một đối tượng của lớp Server\_TCP\_MayTinh và hiển thị cửa sổ.

1. Constructor:

Thiết lập các thuộc tính cơ bản của cửa sổ JFrame như kích thước và hành động khi đóng.

Tạo và cấu hình JPanel để chứa các thành phần giao diện khác.

Thêm một JTextPane để hiển thị thông điệp từ client.

Thêm một nút "Thoát" để dừng máy chủ và đóng ứng dụng.

1. Phương thức startServer:

Khởi tạo máy chủ TCP trên cổng 12345 và chờ đợi các kết nối từ client.

Mỗi khi có một client kết nối, một luồng mới được tạo để xử lý kết nối đó.

1. Phương thức stopServer:

Dừng máy chủ TCP bằng cách đặt biến serverRunning thành false.

1. Phương thức updateTextPane:

Cập nhật nội dung hiển thị trên JTextPane trong giao diện người dùng.

1. Các phương thức liên quan đến xử lý biểu thức toán học:

convertPercentage: Chuyển đổi phần trăm sang dạng thập phân.

evaluate: Đánh giá biểu thức toán học.

hasPrecedence: Kiểm tra ưu tiên của các toán tử.

applyOp: Áp dụng một toán tử lên hai toán hạng.

1. Lớp con ClientHandler:

Một luồng riêng được tạo cho mỗi client kết nối để xử lý thông điệp từ client, tính toán và gửi kết quả trở lại.

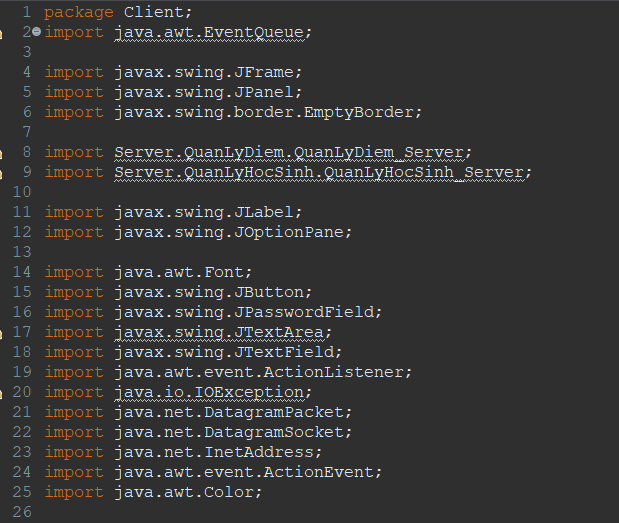
Login.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116 | package Client;import java.awt.EventQueue;  import javax.swing.JFrame;import javax.swing.JPanel;import javax.swing.border.EmptyBorder;  import Server.QuanLyDiem.QuanLyDiem\_Server;import Server.QuanLyHocSinh.QuanLyHocSinh\_Server;  import javax.swing.JLabel;import javax.swing.JOptionPane;  import java.awt.Font;import javax.swing.JButton;import javax.swing.JPasswordField;import javax.swing.JTextArea;import javax.swing.JTextField;import java.awt.event.ActionListener;import java.io.IOException;import java.net.DatagramPacket;import java.net.DatagramSocket;import java.net.InetAddress;import java.awt.event.ActionEvent;import java.awt.Color;  public class Login extends JFrame {  private static final long serialVersionUID = 1L;  private JPanel contentPane;  private JPasswordField passwordField;  private JTextField txtHs;  public static void main(String[] args) {  Login frame = new Login();  frame.setVisible(true);  }  private void sendLoginData(String username, String password) {  try {  DatagramSocket socket = new DatagramSocket();  InetAddress serverAddress = InetAddress.getByName("localhost");  int serverPort = 9876;  byte[] sendData = ("Login" + "," + username + "," + password).getBytes();  DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length, serverAddress, serverPort);  socket.send(sendPacket);  byte[] receiveData = new byte[1024];  DatagramPacket receivePacket = new DatagramPacket(receiveData, receiveData.length);  socket.receive(receivePacket);  String result = new String(receivePacket.getData(), 0, receivePacket.getLength());  if ("success".equals(result)) {  setVisible(false);  Client\_UDP frame = new Client\_UDP(username);  frame.setVisible(true);  frame.initializeSocket();  } else {  JOptionPane.showMessageDialog(this, "Kiểm tra lại thông tin đăng nhập", "Đăng nhập không thành công", JOptionPane.ERROR\_MESSAGE);  }  socket.close();  } catch (Exception ex) {  ex.printStackTrace();  }  }  public Login() {  setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  setBounds(100, 100, 450, 300);  contentPane = new JPanel();  contentPane.setBackground(new Color(0, 0, 53));  contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));  setContentPane(contentPane);  contentPane.setLayout(null);  JLabel lblNewLabel = new JLabel("Tên đăng nhập:");  lblNewLabel.setForeground(new Color(255, 255, 255));  lblNewLabel.setBounds(58, 80, 110, 17);  contentPane.add(lblNewLabel);  JLabel lblNewLabel\_1 = new JLabel("Mật khẩu:");  lblNewLabel\_1.setForeground(new Color(255, 255, 255));  lblNewLabel\_1.setBounds(58, 126, 110, 17);  contentPane.add(lblNewLabel\_1);  JButton btnNewButton = new JButton("Đăng nhập");  btnNewButton.setBackground(new Color(255, 255, 255));  btnNewButton.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  String username = txtHs.getText();  String password = new String(passwordField.getPassword());  sendLoginData(username, password);  }  });  btnNewButton.setBounds(156, 178, 110, 21);  contentPane.add(btnNewButton);  passwordField = new JPasswordField();  passwordField.setBounds(178, 124, 203, 19);  contentPane.add(passwordField);  txtHs = new JTextField();  txtHs.setText("HS001");  txtHs.setBounds(178, 78, 203, 19);  contentPane.add(txtHs);  JLabel lblNewLabel\_2 = new JLabel("Đăng Nhập");  lblNewLabel\_2.setForeground(new Color(255, 255, 255));  lblNewLabel\_2.setFont(new Font("Tahoma", Font.BOLD, 20));  lblNewLabel\_2.setBounds(156, 10, 117, 25);  contentPane.add(lblNewLabel\_2);    JButton btnNewButton\_1 = new JButton("Thoát");  btnNewButton\_1.setBackground(new Color(255, 255, 255));  btnNewButton\_1.setForeground(new Color(198, 54, 0));  btnNewButton\_1.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  dispose();  }  });  btnNewButton\_1.setBounds(341, 232, 85, 21);  contentPane.add(btnNewButton\_1);  }  } |

Mô tả: Lớp Login là một giao diện người dùng trong ứng dụng client, cho phép người dùng đăng nhập vào hệ thống bằng tên đăng nhập và mật khẩu.

Giải thích chi tiết:

1. Khai báo và Import:



Hình 31: Khai báo thư viện Login.java

1. Phương thức Main:

Tạo một đối tượng của lớp Login và hiển thị cửa sổ.

1. Phương thức sendLoginData:

Gửi dữ liệu đăng nhập (tên đăng nhập và mật khẩu) đến máy chủ qua giao thức UDP.

Nhận kết quả từ máy chủ và hiển thị cửa sổ chính của ứng dụng nếu đăng nhập thành công.

1. Constructor:

Thiết lập các thuộc tính cơ bản của cửa sổ JFrame như kích thước và hành động khi đóng.

Tạo và cấu hình JPanel để chứa các thành phần giao diện khác.

Thêm các thành phần như nhãn, ô nhập liệu, nút "Đăng nhập" và nút "Thoát".

1. Sự kiện ActionListener cho nút "Đăng nhập":

Khi người dùng nhấn nút "Đăng nhập", lấy dữ liệu từ các ô nhập liệu và gọi phương thức sendLoginData để xử lý việc đăng nhập.

1. Sự kiện ActionListener cho nút "Thoát":

Khi người dùng nhấn nút "Thoát", đóng cửa sổ đăng nhập.

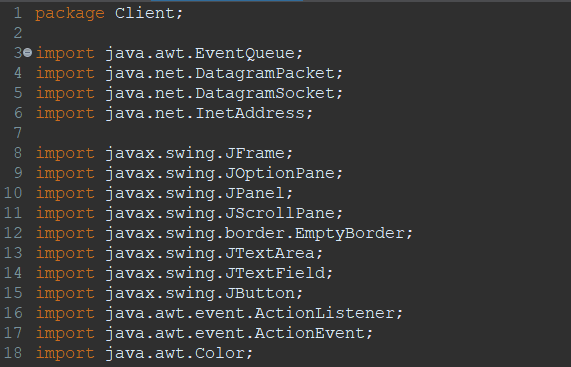
Client\_UDP.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149  150  151  152  153  154  155  156  157  158  159  160  161  162  163  164  165  166  167  168  169  170  171  172  173  174  175  176  177  178  179  180  181  182  183  184  185  186  187  188  189  190  191  192  193  194  195  196  197  198  199  200  201  202  203  204  205  206  207  208  209  210  211  212  213  214  215  216  217  218  219  220  221  222  223  224  225  226  227  228  229  230  231  232  233  234  235  236  237  238  239  240  241  242  243  244  245  246  247  248  249  250  251  252  253  254  255  256  257  258  259  260  261  262  263  264  265  266  267  268  269  270  271  272  273  274  275  276 | package Client;  import java.awt.EventQueue;import java.net.DatagramPacket;import java.net.DatagramSocket;import java.net.InetAddress;  import javax.swing.JFrame;import javax.swing.JOptionPane;import javax.swing.JPanel;import javax.swing.JScrollPane;import javax.swing.border.EmptyBorder;import javax.swing.JTextArea;import javax.swing.JTextField;import javax.swing.JButton;import java.awt.event.ActionListener;import java.awt.event.ActionEvent;import java.awt.Color;  public class Client\_UDP extends JFrame {  private static final long serialVersionUID = 1L;  private JPanel contentPane;  private JTextField textField;  private static DatagramSocket socket;  private static String user;  static JTextArea textArea = new JTextArea();  private JButton btnNewButton\_1;  private JButton btnNewButton\_2;  private JButton btnNewButton\_3;  /\*\* \* Launch the application. \*/  public static void main(String[] args) {  EventQueue.invokeLater(new Runnable() {  public void run() {  try {  Client\_UDP frame = new Client\_UDP("HS001");  frame.setVisible(true);  initializeSocket();  } catch (Exception e) {  e.printStackTrace();  }  }  });  }  /\*\* \* Create the frame. \*/  public Client\_UDP(String username) {  user = username;  setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  setBounds(100, 100, 450, 300);  contentPane = new JPanel();  contentPane.setBackground(new Color(0, 0, 53));  contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));  setContentPane(contentPane);  contentPane.setLayout(null);  JScrollPane scrollPane = new JScrollPane();  scrollPane.setBounds(10, 10, 289, 170);  contentPane.add(scrollPane);  scrollPane.setViewportView(textArea);  textField = new JTextField();  textField.setBounds(10, 183, 289, 19);  contentPane.add(textField);  textField.setColumns(10);  JButton btnNewButton = new JButton("Gửi");  btnNewButton.setBackground(new Color(255, 255, 255));  btnNewButton.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  String message = textField.getText();  String fullMessage = "User,message," + user + "," + message;  try {  // Gửi thông điệp tới ServerMain  byte[] sendData = fullMessage.getBytes();  DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length, InetAddress.getByName("localhost"), 9876);  socket.send(sendPacket);  textArea.append("You: " + message + "\n");  textField.setText("");  } catch (Exception ex) {  ex.printStackTrace();  }  }  });  btnNewButton.setBounds(309, 182, 85, 21);  contentPane.add(btnNewButton);    btnNewButton\_1 = new JButton("Đổi mật khẩu");  btnNewButton\_1.setBackground(new Color(255, 255, 255));  btnNewButton\_1.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  DoiMatKhau frame = new DoiMatKhau(user);  frame.setVisible(true);  setVisible(false);  }  });  btnNewButton\_1.setBounds(115, 232, 117, 21);  contentPane.add(btnNewButton\_1);    btnNewButton\_2 = new JButton("Xem điểm");  btnNewButton\_2.setBackground(new Color(255, 255, 255));  btnNewButton\_2.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  try {  DatagramSocket socket = new DatagramSocket();  InetAddress serverAddress = InetAddress.getByName("localhost");  int serverPort = 9876;  byte[] sendData = ("Point" + ","+ user).getBytes();  DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length, serverAddress, serverPort);  socket.send(sendPacket);  byte[] receiveData = new byte[1024];  DatagramPacket receivePacket = new DatagramPacket(receiveData, receiveData.length);  socket.receive(receivePacket);  String result = new String(receivePacket.getData(), 0, receivePacket.getLength());  String[] mess = result.split(",");  if ("point".equals(mess[0])) {  //Nếu thành công code ở đây  XemDiem frame= new XemDiem(mess[1], user);  frame.setVisible(true);  setVisible(false);  }  }catch (Exception ex){  JOptionPane.showMessageDialog(null, "Server đang bận xem điểm sau!");  }  }  });  btnNewButton\_2.setBounds(309, 13, 117, 21);  contentPane.add(btnNewButton\_2);    btnNewButton\_3 = new JButton("Xem thông tin");  btnNewButton\_3.setBackground(new Color(255, 255, 255));  btnNewButton\_3.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  try {  DatagramSocket socket = new DatagramSocket();  InetAddress serverAddress = InetAddress.getByName("localhost");  int serverPort = 9876;  byte[] sendData = ("Infomation" + ","+ user).getBytes();  DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length, serverAddress, serverPort);  socket.send(sendPacket);  byte[] receiveData = new byte[4096];  DatagramPacket receivePacket = new DatagramPacket(receiveData, receiveData.length);  socket.receive(receivePacket);  String result = new String(receivePacket.getData(), 0, receivePacket.getLength());  String[] mess = result.split("-");  if ("info".equals(mess[0])) {  //Nếu thành công code ở đây  XemThongTin frame= new XemThongTin(mess[1], user);  frame.setVisible(true);  setVisible(false);  }  }catch (Exception ex){  JOptionPane.showMessageDialog(null, "Server đang bận xem điểm sau!");  }  }  });  btnNewButton\_3.setBounds(309, 44, 117, 21);  contentPane.add(btnNewButton\_3);    JButton btnNewButton\_4 = new JButton("Thoát");  btnNewButton\_4.setBackground(new Color(255, 255, 255));  btnNewButton\_4.setForeground(new Color(198, 54, 0));  btnNewButton\_4.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  String message = textField.getText();  String fullMessage = "Disconect," + user ;  try {  // Gửi thông điệp tới ServerMain  byte[] sendData = fullMessage.getBytes();  DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length, InetAddress.getByName("localhost"), 9876);  socket.send(sendPacket);  dispose();  } catch (Exception ex) {  ex.printStackTrace();  }  }  });  btnNewButton\_4.setBounds(341, 232, 85, 21);  contentPane.add(btnNewButton\_4);    JButton btnNewButton\_5 = new JButton("Đăng xuất");  btnNewButton\_5.setForeground(new Color(198, 54, 0));  btnNewButton\_5.setBackground(new Color(255, 255, 255));  btnNewButton\_5.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  String message = textField.getText();  String fullMessage = "Disconect," + user ;  try {  // Gửi thông điệp tới ServerMain  byte[] sendData = fullMessage.getBytes();  DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length, InetAddress.getByName("localhost"), 9876);  socket.send(sendPacket);  Login frame = new Login();  frame.setVisible(true);  dispose();  } catch (Exception ex) {  ex.printStackTrace();  }  }  });  btnNewButton\_5.setBounds(10, 232, 95, 21);  contentPane.add(btnNewButton\_5);  }  static void initializeSocket() {  try {  socket = new DatagramSocket();  // Gửi thông báo kết nối thành công đến ServerMain  String message = "User,connect," + user + ",x";  byte[] sendData = message.getBytes();  DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length, InetAddress.getByName("localhost"), 9876);  socket.send(sendPacket);  // Khởi tạo một luồng để nhận phản hồi từ server  new Thread(new Runnable() {  @Override  public void run() {  try {  while (true) {  // Nhận phản hồi từ ServerMain  byte[] receiveData = new byte[1024];  DatagramPacket receivePacket = new DatagramPacket(receiveData, receiveData.length);  socket.receive(receivePacket);  String response = new String(receivePacket.getData(), 0, receivePacket.getLength());  String[] mess = response.split(",");  //mess[0]="connect",mess[1]="success"  if (mess[0].equals("connect")) {  if (mess[1].equals("success")) {  textArea.append("Kết nối thành công với Server\n");  }  if (mess[1].equals("success") == false) {  textArea.append("Kết nối thất bại\n");  }  }  if (mess[0].equals("message")) {  if (mess[1].equals(user)) {  textArea.append("You: " + mess[2] + "\n");  }  if (mess[1].equals(user) == false) {  textArea.append(mess[1] + ": " + mess[2] + "\n");  }  }  }  } catch (Exception e) {  e.printStackTrace();  }  }  }).start();  } catch (Exception e) {  e.printStackTrace();  }  }  } |

Mô tả: Lớp Client\_UDP là giao diện người dùng cho phép gửi và nhận tin nhắn thông qua giao thức UDP.

Giải thích chi tiết:

1. Khai báo và Import:



Hình 32: Khai báo thư viện Client\_UDP.java

1. Phương thức Main:

Tạo một đối tượng của lớp Client\_UDP và hiển thị cửa sổ.

Khởi tạo socket UDP bằng cách gọi phương thức initializeSocket().

1. Constructor:

Thiết lập các thuộc tính cơ bản của cửa sổ JFrame như kích thước và hành động khi đóng.

Tạo và cấu hình JPanel để chứa các thành phần giao diện khác.

Thêm các thành phần như ô nhập liệu, nút "Gửi", nút "Đổi mật khẩu", nút "Xem điểm", nút "Xem thông tin", nút "Thoát" và nút "Đăng xuất".

1. Sự kiện ActionListener cho nút "Gửi":

Khi người dùng nhấn nút "Gửi", lấy tin nhắn từ ô nhập liệu, gửi tin nhắn đến server qua giao thức UDP và hiển thị tin nhắn trong textArea.

1. Sự kiện ActionListener cho các nút "Đổi mật khẩu", "Xem điểm", "Xem thông tin", "Thoát" và "Đăng xuất":

Khi người dùng nhấn các nút này, gửi yêu cầu tương ứng đến server qua giao thức UDP.

Xử lý phản hồi từ server và hiển thị các giao diện tương ứng.

1. Phương thức initializeSocket():

Khởi tạo socket UDP.

Gửi thông điệp kết nối thành công đến server.

Khởi tạo một luồng để nhận phản hồi từ server và hiển thị trên textArea.

DoiMatKhau.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149  150 | package Client;  import java.awt.EventQueue;  import javax.swing.JFrame;import javax.swing.JPanel;import javax.swing.border.EmptyBorder;import javax.swing.JLabel;import javax.swing.JOptionPane;  import java.awt.Font;import javax.swing.JButton;import javax.swing.JPasswordField;import javax.swing.JTextField;import java.awt.event.ActionListener;import java.net.DatagramPacket;import java.net.DatagramSocket;import java.net.InetAddress;import java.awt.event.ActionEvent;import java.awt.Color;  public class DoiMatKhau extends JFrame {  private static final long serialVersionUID = 1L;  private JPanel contentPane;  private JPasswordField pass\_kiemTra;  private JPasswordField pass\_moi;  private JPasswordField pass\_cu;  private String user;  /\*\* \* Launch the application. \*/  public static void main(String[] args) {  EventQueue.invokeLater(new Runnable() {  public void run() {  try {  DoiMatKhau frame = new DoiMatKhau("HS001");  frame.setVisible(true);  } catch (Exception e) {  e.printStackTrace();  }  }  });  }  /\*\* \* Create the frame. \*/  private void sendChangePassData(String password\_cu, String password\_moi) {  try {  DatagramSocket socket = new DatagramSocket();  InetAddress serverAddress = InetAddress.getByName("localhost");  int serverPort = 9876;  byte[] sendData = ("Pass" + ","+ user + "," + password\_cu + "," + password\_moi).getBytes();  DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length, serverAddress, serverPort);  socket.send(sendPacket);  byte[] receiveData = new byte[1024];  DatagramPacket receivePacket = new DatagramPacket(receiveData, receiveData.length);  socket.receive(receivePacket);  String result = new String(receivePacket.getData(), 0, receivePacket.getLength());  if ("success".equals(result)) {  //Nếu thành công code ở đây  JOptionPane.showMessageDialog(null, "Đổi mật khẩu thành công, bạn sẽ quay lại trang chính!");  Client\_UDP frame = new Client\_UDP(user);  frame.setVisible(true);  setVisible(false);  } else {  JOptionPane.showMessageDialog(null, "Nhập sai mật khẩu cũ!");  }  socket.close();  } catch (Exception ex) {  ex.printStackTrace();  }  }  public DoiMatKhau(String username) {  user = username;  setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  setBounds(100, 100, 450, 300);  contentPane = new JPanel();  contentPane.setBackground(new Color(0, 0, 53));  contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));  setContentPane(contentPane);  contentPane.setLayout(null);    JLabel lblNewLabel\_2 = new JLabel("Đổi Mật Khẩu");  lblNewLabel\_2.setForeground(new Color(255, 255, 255));  lblNewLabel\_2.setFont(new Font("Tahoma", Font.BOLD, 20));  lblNewLabel\_2.setBounds(140, 10, 143, 25);  contentPane.add(lblNewLabel\_2);    JButton btniMtKhu = new JButton("Đổi mật khẩu");  btniMtKhu.setBackground(new Color(255, 255, 255));  btniMtKhu.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  String cu = new String(pass\_cu.getPassword());  String moi = new String(pass\_moi.getPassword());  String kiemTra = new String(pass\_kiemTra.getPassword());  if(moi.equals(kiemTra)==false) {  JOptionPane.showMessageDialog(null, "Mật khẩu mới không khớp với mật khẩu nhập lại!");  }else {  sendChangePassData(cu,moi);  }    }  });  btniMtKhu.setBounds(171, 181, 143, 21);  contentPane.add(btniMtKhu);    pass\_kiemTra = new JPasswordField();  pass\_kiemTra.setBounds(171, 152, 143, 19);  contentPane.add(pass\_kiemTra);    JLabel lblNewLabel\_1 = new JLabel("Nhập lại mật khẩu:");  lblNewLabel\_1.setForeground(new Color(255, 255, 255));  lblNewLabel\_1.setBounds(51, 154, 110, 17);  contentPane.add(lblNewLabel\_1);    pass\_moi = new JPasswordField();  pass\_moi.setBounds(171, 122, 143, 19);  contentPane.add(pass\_moi);    JLabel lblNewLabel\_1\_1 = new JLabel("Mật khẩu mới:");  lblNewLabel\_1\_1.setForeground(new Color(255, 255, 255));  lblNewLabel\_1\_1.setBounds(51, 125, 110, 17);  contentPane.add(lblNewLabel\_1\_1);    JLabel lblNewLabel\_1\_1\_1 = new JLabel("Mật khẩu cũ:");  lblNewLabel\_1\_1\_1.setForeground(new Color(255, 255, 255));  lblNewLabel\_1\_1\_1.setBounds(51, 95, 110, 17);  contentPane.add(lblNewLabel\_1\_1\_1);    pass\_cu = new JPasswordField();  pass\_cu.setBounds(171, 92, 143, 19);  contentPane.add(pass\_cu);    JButton btnTrV = new JButton("Trở về");  btnTrV.setBackground(new Color(255, 255, 255));  btnTrV.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  Client\_UDP frame = new Client\_UDP(user);  frame.setVisible(true);  setVisible(false);  }  });  btnTrV.setBounds(10, 232, 75, 21);  contentPane.add(btnTrV);  }  } |

Mô tả:Lớp DoiMatKhau là giao diện cho phép người dùng thay đổi mật khẩu.

Giải thích chi tiết:

1. Phương thức Main:

Tạo một đối tượng của lớp DoiMatKhau và hiển thị cửa sổ.

1. Constructor:

Thiết lập các thuộc tính cơ bản của cửa sổ JFrame như kích thước và hành động khi đóng.

Tạo và cấu hình JPanel để chứa các thành phần giao diện khác.

Thêm các thành phần như ô nhập liệu cho mật khẩu cũ, mật khẩu mới và nhập lại mật khẩu mới.

Thêm nút "Đổi mật khẩu" và nút "Trở về".

1. Sự kiện ActionListener cho nút "Đổi mật khẩu":

Khi người dùng nhấn nút "Đổi mật khẩu", lấy thông tin mật khẩu cũ, mật khẩu mới và nhập lại mật khẩu mới.

Kiểm tra xem mật khẩu mới và nhập lại mật khẩu mới có khớp nhau không.

Nếu khớp, gửi thông tin mật khẩu cũ và mật khẩu mới đến server để thay đổi mật khẩu.

Xử lý phản hồi từ server và hiển thị thông báo tương ứng.

1. Sự kiện ActionListener cho nút "Trở về":

Khi người dùng nhấn nút "Trở về", quay lại giao diện chính của ứng dụng.

1. Phương thức sendChangePassData():

Gửi yêu cầu thay đổi mật khẩu đến server.

Xử lý phản hồi từ server và hiển thị thông báo tương ứng.

XemThongTin.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125 | package Client;  import java.awt.EventQueue;  import javax.swing.JFrame;import javax.swing.JPanel;import javax.swing.border.EmptyBorder;import javax.swing.JLabel;import javax.swing.JTextField;import java.awt.Font;import javax.swing.JButton;import java.awt.event.ActionListener;import java.awt.event.ActionEvent;import java.awt.Color;  public class XemThongTin extends JFrame {  private static final long serialVersionUID = 1L;  private JPanel contentPane;  private JTextField textField\_ID;  private JTextField textField\_NamHoc;  private JTextField textField\_DiaChi;  private JTextField textField\_HoTen;  /\*\* \* Launch the application. \*/  public static void main(String[] args) {  EventQueue.invokeLater(new Runnable() {  public void run() {  try {  XemThongTin frame = new XemThongTin("null", "null");  frame.setVisible(true);  } catch (Exception e) {  e.printStackTrace();  }  }  });  }  /\*\* \* Create the frame. \*/  public XemThongTin(String data, String user) {  setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  setBounds(100, 100, 450, 300);  contentPane = new JPanel();  contentPane.setBackground(new Color(0, 0, 53));  contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));  setContentPane(contentPane);  contentPane.setLayout(null);    JLabel lblNewLabel\_1\_1 = new JLabel("IDHocSinh:");  lblNewLabel\_1\_1.setForeground(new Color(255, 255, 255));  lblNewLabel\_1\_1.setBounds(62, 92, 92, 13);  contentPane.add(lblNewLabel\_1\_1);    textField\_ID = new JTextField();  textField\_ID.setColumns(10);  textField\_ID.setBounds(160, 89, 55, 19);  contentPane.add(textField\_ID);    JLabel lblNewLabel\_1\_2 = new JLabel("Họ Tên:");  lblNewLabel\_1\_2.setForeground(new Color(255, 255, 255));  lblNewLabel\_1\_2.setBounds(62, 115, 116, 13);  contentPane.add(lblNewLabel\_1\_2);    JLabel lblNewLabel\_3 = new JLabel("Địa Chỉ:");  lblNewLabel\_3.setForeground(new Color(255, 255, 255));  lblNewLabel\_3.setBounds(62, 141, 92, 13);  contentPane.add(lblNewLabel\_3);    JLabel lblNewLabel\_4 = new JLabel("Năm nhập học:");  lblNewLabel\_4.setForeground(new Color(255, 255, 255));  lblNewLabel\_4.setBounds(62, 164, 116, 13);  contentPane.add(lblNewLabel\_4);    textField\_NamHoc = new JTextField();  textField\_NamHoc.setColumns(10);  textField\_NamHoc.setBounds(160, 161, 55, 19);  contentPane.add(textField\_NamHoc);    textField\_DiaChi = new JTextField();  textField\_DiaChi.setColumns(10);  textField\_DiaChi.setBounds(160, 135, 266, 19);  contentPane.add(textField\_DiaChi);    textField\_HoTen = new JTextField();  textField\_HoTen.setColumns(10);  textField\_HoTen.setBounds(160, 112, 104, 19);  contentPane.add(textField\_HoTen);    JLabel lblNewLabel\_2\_1 = new JLabel("Thông Tin");  lblNewLabel\_2\_1.setForeground(new Color(255, 255, 255));  lblNewLabel\_2\_1.setFont(new Font("Tahoma", Font.BOLD, 20));  lblNewLabel\_2\_1.setBounds(160, 10, 104, 25);  contentPane.add(lblNewLabel\_2\_1);    JButton btnNewButton\_1 = new JButton("Trở về");  btnNewButton\_1.setBackground(new Color(255, 255, 255));  btnNewButton\_1.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  Client\_UDP frame = new Client\_UDP(user);  frame.setVisible(true);  setVisible(false);  }  });  btnNewButton\_1.setBounds(10, 232, 85, 21);  contentPane.add(btnNewButton\_1);    String[][] temp = convertTo2DArray(data);    textField\_ID.setText(temp[0][1]);  textField\_HoTen.setText(temp[1][1]);  textField\_DiaChi.setText(temp[2][1]);  textField\_NamHoc.setText(temp[3][1]);  }  public static String[][] convertTo2DArray(String info) {  // Tách thông tin thành các dòng  String[] lines = info.split("\n");  // Khởi tạo mảng hai chiều để lưu trữ thông tin  String[][] result = new String[lines.length][2];  // Lặp qua từng dòng và chuyển đổi thành cặp key-value  for (int i = 0; i < lines.length; i++) {  String[] parts = lines[i].split(": ");  result[i][0] = parts[0]; // Key  result[i][1] = parts[1]; // Value  }  return result;  }  } |

Mô tả: Lớp XemThongTin là giao diện cho phép người dùng xem thông tin của học sinh.

Giải thích chi tiết:

1. Phương thức Main:

Tạo một đối tượng của lớp XemThongTin và hiển thị cửa sổ.

1. Constructor:

Thiết lập các thuộc tính cơ bản của cửa sổ JFrame như kích thước và hành động khi đóng.

Tạo và cấu hình JPanel để chứa các thành phần giao diện khác.

Thêm các nhãn và ô nhập liệu để hiển thị thông tin của học sinh.

Thêm nút "Trở về" để quay lại giao diện chính của ứng dụng.

1. Phương thức convertTo2DArray():

Nhận thông tin về học sinh dưới dạng một chuỗi.

Tách thông tin thành các dòng và chuyển đổi thành một mảng hai chiều để dễ dàng hiển thị trên giao diện.

Mỗi hàng của mảng chứa một cặp key-value, trong đó key là loại thông tin và value là giá trị tương ứng.

1. Khởi tạo dữ liệu:

Khi khởi tạo lớp XemThongTin nhận thông tin về học sinh dưới dạng một chuỗi và tên người dùng (user).

Thực hiện phân tích chuỗi thông tin để lấy các giá trị tương ứng và hiển thị trên giao diện.

XemDiem.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149  150  151  152  153  154  155  156  157  158  159  160  161  162  163  164  165  166  167  168  169  170  171  172  173 | package Client;  import java.awt.EventQueue;  import javax.swing.JFrame;import javax.swing.JPanel;import javax.swing.border.EmptyBorder;import javax.swing.JButton;import javax.swing.JTextField;import javax.swing.JLabel;import java.awt.Font;import java.awt.event.ActionListener;import java.awt.event.ActionEvent;import java.awt.Color;  public class XemDiem extends JFrame {  private static final long serialVersionUID = 1L;  private JPanel contentPane;  private JTextField HoTen;  private JTextField MHAnh;  private JTextField MHToan;  private JTextField MHVan;  /\*\* \* Launch the application. \*/  public static void main(String[] args) {  EventQueue.invokeLater(new Runnable() {  public void run() {  try {  XemDiem frame = new XemDiem("null", "null");  frame.setVisible(true);  } catch (Exception e) {  e.printStackTrace();  }  }  });  }  /\*\* \* Create the frame. \*/  public XemDiem(String data\_Diem, String user) {  setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  setBounds(100, 100, 450, 300);  contentPane = new JPanel();  contentPane.setBackground(new Color(0, 0, 53));  contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));  setContentPane(contentPane);  contentPane.setLayout(null);    HoTen = new JTextField();  HoTen.setColumns(10);  HoTen.setBounds(169, 80, 149, 19);  contentPane.add(HoTen);    JLabel lblNewLabel\_4 = new JLabel("Họ và tên:");  lblNewLabel\_4.setForeground(new Color(255, 255, 255));  lblNewLabel\_4.setBounds(101, 83, 96, 13);  contentPane.add(lblNewLabel\_4);    JLabel Anh = new JLabel("Anh:");  Anh.setForeground(new Color(255, 255, 255));  Anh.setBounds(128, 116, 45, 13);  contentPane.add(Anh);    MHAnh = new JTextField();  MHAnh.setColumns(10);  MHAnh.setBounds(169, 113, 45, 19);  contentPane.add(MHAnh);    JLabel lblNewLabel\_2 = new JLabel("Toán:");  lblNewLabel\_2.setForeground(new Color(255, 255, 255));  lblNewLabel\_2.setBounds(124, 153, 45, 13);  contentPane.add(lblNewLabel\_2);    MHToan = new JTextField();  MHToan.setColumns(10);  MHToan.setBounds(169, 150, 45, 19);  contentPane.add(MHToan);    MHVan = new JTextField();  MHVan.setColumns(10);  MHVan.setBounds(169, 188, 45, 19);  contentPane.add(MHVan);    JLabel lblNewLabel\_3 = new JLabel("Văn:");  lblNewLabel\_3.setForeground(new Color(255, 255, 255));  lblNewLabel\_3.setBounds(128, 191, 45, 13);  contentPane.add(lblNewLabel\_3);    JButton btnNewButton\_1 = new JButton("Trở về");  btnNewButton\_1.setBackground(new Color(255, 255, 255));  btnNewButton\_1.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  Client\_UDP frame = new Client\_UDP(user);  frame.setVisible(true);  setVisible(false);  }  });  btnNewButton\_1.setBounds(10, 232, 85, 21);  contentPane.add(btnNewButton\_1);    JLabel lblNewLabel\_2\_1 = new JLabel("Bảng Điểm");  lblNewLabel\_2\_1.setForeground(new Color(255, 255, 255));  lblNewLabel\_2\_1.setFont(new Font("Tahoma", Font.BOLD, 20));  lblNewLabel\_2\_1.setBounds(145, 10, 143, 25);  contentPane.add(lblNewLabel\_2\_1);    String Diem = data\_Diem;    String[][] temp =parseInput(Diem);  String[][] temp1 =conf(temp);  HoTen.setText(temp[0][1]);  MHAnh.setText(temp1[0][1]);  MHToan.setText(temp1[1][1]);  MHVan.setText(temp1[2][1]);  }  public static String[][] parseInput(String input){  String[][] temp = new String[4][2];      String[] lines = input.split("\n");  // Khởi tạo mảng hai chiều để lưu trữ thông tin  temp = new String[lines.length + 1][2];  // Tìm vị trí dòng bắt đầu thông tin điểm  int start = 0;  for (int i = 0; i < lines.length; i++) {  if (lines[i].startsWith("Họ và tên:")) {  start = i;  break;  }  }  // Tách họ tên từ dòng bắt đầu thông tin điểm  String[] hoTen = lines[start].split(":");  temp[0][0] = hoTen[0].trim(); // "Họ và tên"  temp[0][1] = hoTen[1].trim(); // "Nguyễn Văn A"  // Bắt đầu từ dòng thứ start + 1  for (int i = start + 1; i < lines.length; i++) {  // Tách tên môn và điểm từ mỗi dòng  String[] parts = lines[i].split(":");  // Xóa khoảng trắng ở đầu của điểm (nếu có)  String monHoc = parts[0].trim();  String diem = parts[1].trim();  // Lưu tên môn vào cột 0 và điểm vào cột 1  temp[i - start][0] = monHoc;  temp[i - start][1] = diem;  }  return temp;  }  public static String[][] conf(String[][] parseInput) {  String[][] temp = new String[3][2];  for (int i = 0; i < parseInput.length; i++) {  // Kiểm tra parseInput[i] không phải là null  if (parseInput[i] != null) {  for (int j = 0; j < parseInput[i].length; j++) {  // Kiểm tra parseInput[i][0] không phải là null trước khi so sánh  if (parseInput[i][0] != null) {  if (parseInput[i][0].equals("Anh")) {  temp[0][0] = "Anh";  temp[0][1] = parseInput[i][1];  } else if (parseInput[i][0].equals("Toán")) {  temp[1][0] = "Toán";  temp[1][1] = parseInput[i][1];  } else if (parseInput[i][0].equals("Văn")) {  temp[2][0] = "Văn";  temp[2][1] = parseInput[i][1];  }  }  }  }  }  return temp;  }  } |

Mô tả: Lớp XemDiem cung cấp giao diện để hiển thị bảng điểm của học sinh.

Giải thích chi tiết:

1. Phương thức Main:

Tạo một đối tượng của lớp XemDiem và hiển thị cửa sổ.

1. Constructor:

Thiết lập các thuộc tính cơ bản của cửa sổ JFrame như kích thước và hành động khi đóng.

Tạo và cấu hình JPanel để chứa các thành phần giao diện khác.

Thêm các nhãn và ô nhập liệu để hiển thị điểm của học sinh theo từng môn.

Thêm nút "Trở về" để quay lại giao diện chính của ứng dụng.

1. Phương thức parseInput():

Nhận dữ liệu điểm ở dạng chuỗi và chuyển đổi thành mảng hai chiều.

Tách họ và tên học sinh từ dòng đầu tiên.

Tách tên môn và điểm từ các dòng tiếp theo và lưu trữ chúng vào mảng.

1. Phương thức conf():

Nhận mảng điểm của học sinh và sắp xếp chúng theo từng môn.

Tạo một mảng mới chứa điểm của từng môn, bỏ qua các môn không có điểm.

1. Khởi tạo dữ liệu:

Khi khởi tạo, lớp XemDiem nhận dữ liệu điểm của học sinh dưới dạng một chuỗi và tên người dùng (user).

Phân tích chuỗi để lấy các giá trị tương ứng và hiển thị chúng trên giao diện.

MayTinh.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149  150  151  152  153  154  155  156  157  158  159  160  161  162  163  164  165  166  167  168  169  170  171  172  173  174  175  176  177  178  179  180  181  182  183  184  185  186  187  188  189  190  191  192  193  194  195  196  197  198  199  200  201  202  203  204  205  206  207  208  209  210  211  212  213  214  215  216  217  218  219  220  221  222  223  224  225  226  227  228  229  230  231  232  233  234  235  236  237  238  239  240  241  242  243  244  245  246  247  248  249  250  251  252  253  254  255  256  257  258  259  260  261  262  263  264  265  266  267  268  269  270  271  272  273  274  275  276  277  278  279  280  281  282  283  284  285  286  287  288  289  290  291  292  293  294  295  296  297  298  299  300  301  302  303  304  305  306  307  308  309  310  311  312  313  314  315  316  317  318  319  320  321  322  323  324  325  326  327  328  329  330  331  332  333  334  335  336  337  338  339  340  341  342  343  344  345  346  347  348  349  350  351  352  353  354  355  356  357  358  359  360  361  362  363  364  365  366  367  368  369  370  371  372  373  374  375  376  377  378  379  380  381  382  383  384  385  386  387  388  389  390  391  392  393  394  395  396  397  398  399  400  401  402  403  404  405  406  407  408  409  410  411  412  413  414  415  416  417  418  419  420  421  422  423  424  425  426  427  428  429  430  431  432  433  434  435  436  437  438  439  440  441  442  443  444  445  446  447  448  449  450  451  452  453  454  455  456  457  458  459  460  461  462  463  464  465  466  467  468  469  470  471  472  473  474  475  476  477  478  479  480  481  482  483  484  485  486  487  488  489  490  491  492  493  494  495  496  497  498  499  500  501  502  503  504 | package Client;  import java.awt.EventQueue;import java.util.ArrayList;import java.io.BufferedReader;import java.io.IOException;import java.io.InputStreamReader;import java.io.PrintWriter;import java.net.Socket;  import javax.swing.JFrame;import javax.swing.JPanel;import javax.swing.JScrollPane;import javax.swing.JTextArea;import javax.swing.border.EmptyBorder;import javax.swing.JTextField;import javax.swing.JButton;import javax.swing.JList;import javax.swing.JToggleButton;import java.awt.event.ActionListener;import java.awt.event.ActionEvent;import java.awt.Font;import java.awt.Color;  public class MayTinh extends JFrame {  private static final long serialVersionUID = 1L;  private JPanel contentPane;  private JTextField textField;  private Socket socket;  private PrintWriter out;  private BufferedReader in;  private ArrayList<String> historyList = new ArrayList<>();  /\*\* \* Launch the application. \*/  public void addToHistory(String data) {  // Thêm dữ liệu vào ArrayList  historyList.add(data);  }    public static void main(String[] args) {  EventQueue.invokeLater(new Runnable() {  public void run() {  try {  MayTinh frame = new MayTinh();  frame.setVisible(true);  frame.connectToServer();  } catch (Exception e) {  e.printStackTrace();  }  }  });  }  /\*\* \* Create the frame. \*/  public MayTinh() {  setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  setBounds(100, 100, 450, 300);  contentPane = new JPanel();  contentPane.setBackground(new Color(65, 65, 65));  contentPane.setForeground(new Color(26, 26, 26));  contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));  setContentPane(contentPane);  contentPane.setLayout(null);    textField = new JTextField();  textField.setFont(new Font("Tahoma", Font.BOLD, 15));  textField.setBounds(35, 44, 296, 32);  contentPane.add(textField);  textField.setColumns(10);    JButton btnNewButton = new JButton("(");  btnNewButton.setBackground(new Color(128, 128, 128));  btnNewButton.setForeground(new Color(255, 255, 255));  btnNewButton.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  textField.setText(textField.getText()+"(");  }  });  btnNewButton.setBounds(35, 86, 85, 21);  contentPane.add(btnNewButton);    JButton btnNewButton\_1 = new JButton(")");  btnNewButton\_1.setBackground(new Color(128, 128, 128));  btnNewButton\_1.setForeground(new Color(255, 255, 255));  btnNewButton\_1.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  textField.setText(textField.getText()+")");  }  });  btnNewButton\_1.setBounds(130, 86, 85, 21);  contentPane.add(btnNewButton\_1);    JButton btnNewButton\_2 = new JButton("%");  btnNewButton\_2.setBackground(new Color(128, 128, 128));  btnNewButton\_2.setForeground(new Color(255, 255, 255));  btnNewButton\_2.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  textField.setText(textField.getText()+"%");  }  });  btnNewButton\_2.setBounds(225, 86, 85, 21);  contentPane.add(btnNewButton\_2);    JButton btnNewButton\_3 = new JButton("CE");  btnNewButton\_3.setBackground(new Color(128, 128, 128));  btnNewButton\_3.setForeground(new Color(255, 255, 255));  btnNewButton\_3.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  textField.setText("");  }  });  btnNewButton\_3.setBounds(320, 86, 85, 21);  contentPane.add(btnNewButton\_3);    JButton btnNewButton\_4 = new JButton("7");  btnNewButton\_4.setForeground(new Color(255, 255, 255));  btnNewButton\_4.setFont(new Font("Tahoma", Font.BOLD, 15));  btnNewButton\_4.setBackground(new Color(60, 60, 60));  btnNewButton\_4.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  if(textField.getText().contains("=")) {  textField.setText("");  }  textField.setText(textField.getText()+"7");  }  });  btnNewButton\_4.setBounds(35, 117, 85, 21);  contentPane.add(btnNewButton\_4);    JButton btnNewButton\_5 = new JButton("4");  btnNewButton\_5.setForeground(new Color(255, 255, 255));  btnNewButton\_5.setFont(new Font("Tahoma", Font.BOLD, 15));  btnNewButton\_5.setBackground(new Color(60, 60, 60));  btnNewButton\_5.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  if(textField.getText().contains("=")) {  textField.setText("");  }  textField.setText(textField.getText()+"4");  }  });  btnNewButton\_5.setBounds(35, 148, 85, 21);  contentPane.add(btnNewButton\_5);    JButton btnNewButton\_6 = new JButton("1");  btnNewButton\_6.setForeground(new Color(255, 255, 255));  btnNewButton\_6.setFont(new Font("Tahoma", Font.BOLD, 15));  btnNewButton\_6.setBackground(new Color(60, 60, 60));  btnNewButton\_6.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  if(textField.getText().contains("=")) {  textField.setText("");  }  textField.setText(textField.getText()+"1");  }  });  btnNewButton\_6.setBounds(35, 179, 85, 21);  contentPane.add(btnNewButton\_6);    JButton btnNewButton\_7 = new JButton("0");  btnNewButton\_7.setForeground(new Color(255, 255, 255));  btnNewButton\_7.setFont(new Font("Tahoma", Font.BOLD, 15));  btnNewButton\_7.setBackground(new Color(60, 60, 60));  btnNewButton\_7.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  if(textField.getText().contains("=")) {  textField.setText("");  }  textField.setText(textField.getText() + "0");  }  });  btnNewButton\_7.setBounds(35, 208, 85, 21);  contentPane.add(btnNewButton\_7);    JButton btnNewButton\_8 = new JButton("8");  btnNewButton\_8.setForeground(new Color(255, 255, 255));  btnNewButton\_8.setFont(new Font("Tahoma", Font.BOLD, 15));  btnNewButton\_8.setBackground(new Color(60, 60, 60));  btnNewButton\_8.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  if(textField.getText().contains("=")) {  textField.setText("");  }  textField.setText(textField.getText()+"8");  }  });  btnNewButton\_8.setBounds(130, 117, 85, 21);  contentPane.add(btnNewButton\_8);    JButton btnNewButton\_9 = new JButton("5");  btnNewButton\_9.setForeground(new Color(255, 255, 255));  btnNewButton\_9.setFont(new Font("Tahoma", Font.BOLD, 15));  btnNewButton\_9.setBackground(new Color(60, 60, 60));  btnNewButton\_9.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  if(textField.getText().contains("=")) {  textField.setText("");  }  textField.setText(textField.getText()+"5");  }  });  btnNewButton\_9.setBounds(130, 148, 85, 21);  contentPane.add(btnNewButton\_9);    JButton btnNewButton\_10 = new JButton("2");  btnNewButton\_10.setForeground(new Color(255, 255, 255));  btnNewButton\_10.setFont(new Font("Tahoma", Font.BOLD, 15));  btnNewButton\_10.setBackground(new Color(60, 60, 60));  btnNewButton\_10.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  if(textField.getText().contains("=")) {  textField.setText("");  }  textField.setText(textField.getText()+"2");  }  });  btnNewButton\_10.setBounds(130, 179, 85, 21);  contentPane.add(btnNewButton\_10);    JButton btnNewButton\_11 = new JButton(".");  btnNewButton\_11.setBackground(new Color(60, 60, 60));  btnNewButton\_11.setForeground(new Color(255, 255, 255));  btnNewButton\_11.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  textField.setText(textField.getText()+".");  }  });  btnNewButton\_11.setBounds(130, 208, 85, 21);  contentPane.add(btnNewButton\_11);    JButton btnNewButton\_12 = new JButton("9");  btnNewButton\_12.setForeground(new Color(255, 255, 255));  btnNewButton\_12.setFont(new Font("Tahoma", Font.BOLD, 15));  btnNewButton\_12.setBackground(new Color(60, 60, 60));  btnNewButton\_12.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  if(textField.getText().contains("=")) {  textField.setText("");  }  textField.setText(textField.getText()+"9");  }  });  btnNewButton\_12.setBounds(225, 117, 85, 21);  contentPane.add(btnNewButton\_12);    JButton btnNewButton\_13 = new JButton("6");  btnNewButton\_13.setForeground(new Color(255, 255, 255));  btnNewButton\_13.setFont(new Font("Tahoma", Font.BOLD, 15));  btnNewButton\_13.setBackground(new Color(60, 60, 60));  btnNewButton\_13.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  if(textField.getText().contains("=")) {  textField.setText("");  }  textField.setText(textField.getText()+"6");  }  });  btnNewButton\_13.setBounds(225, 148, 85, 21);  contentPane.add(btnNewButton\_13);    JButton btnNewButton\_14 = new JButton("3");  btnNewButton\_14.setForeground(new Color(255, 255, 255));  btnNewButton\_14.setFont(new Font("Tahoma", Font.BOLD, 15));  btnNewButton\_14.setBackground(new Color(60, 60, 60));  btnNewButton\_14.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  if(textField.getText().contains("=")) {  textField.setText("");  }  textField.setText(textField.getText()+"3");  }  });  btnNewButton\_14.setBounds(225, 179, 85, 21);  contentPane.add(btnNewButton\_14);    JButton btnNewButton\_15 = new JButton("=");  btnNewButton\_15.setBackground(new Color(129, 149, 203));  btnNewButton\_15.setForeground(new Color(0, 0, 0));  btnNewButton\_15.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  sendMessage(textField.getText());  textField.setText("");  }  });  btnNewButton\_15.setBounds(225, 208, 85, 21);  contentPane.add(btnNewButton\_15);    JButton btnNewButton\_16 = new JButton(":");  btnNewButton\_16.setBackground(new Color(128, 128, 128));  btnNewButton\_16.setForeground(new Color(255, 255, 255));  btnNewButton\_16.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  if(textField.getText().contains("=")) {  textField.setText(getValueAfterEqualSign(textField.getText()) + " : ");  }else {  textField.setText(textField.getText()+" : ");  }  }  });  btnNewButton\_16.setBounds(320, 117, 85, 21);  contentPane.add(btnNewButton\_16);    JButton btnNewButton\_17 = new JButton("x");  btnNewButton\_17.setBackground(new Color(128, 128, 128));  btnNewButton\_17.setForeground(new Color(255, 255, 255));  btnNewButton\_17.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  if(textField.getText().contains("=")) {  textField.setText(getValueAfterEqualSign(textField.getText()) + " x ");  }else {  textField.setText(textField.getText()+" x ");  }  }  });  btnNewButton\_17.setBounds(320, 148, 85, 21);  contentPane.add(btnNewButton\_17);    JButton btnNewButton\_18 = new JButton("-");  btnNewButton\_18.setBackground(new Color(128, 128, 128));  btnNewButton\_18.setForeground(new Color(255, 255, 255));  btnNewButton\_18.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  if(textField.getText().contains("=")) {  textField.setText(getValueAfterEqualSign(textField.getText()) + " - ");  }else {  textField.setText(textField.getText()+" - ");  }  }  });  btnNewButton\_18.setBounds(320, 179, 85, 21);  contentPane.add(btnNewButton\_18);    JButton btnNewButton\_19 = new JButton("+");  btnNewButton\_19.setBackground(new Color(128, 128, 128));  btnNewButton\_19.setForeground(new Color(255, 255, 255));  btnNewButton\_19.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  if(textField.getText().contains("=")) {  textField.setText(getValueAfterEqualSign(textField.getText()) + " + ");  }else {  textField.setText(textField.getText()+" + ");  }  }  });  btnNewButton\_19.setBounds(320, 208, 85, 21);  contentPane.add(btnNewButton\_19);    JButton btnNewButton\_20 = new JButton("Del");  btnNewButton\_20.setForeground(new Color(255, 255, 255));  btnNewButton\_20.setBackground(new Color(192, 192, 192));  btnNewButton\_20.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  if(textField.getText().contains("=")) {  textField.setText("");  }else {  String result = removeOuterCharacter(textField.getText());  textField.setText(result);  }  }  });    btnNewButton\_20.setBounds(341, 51, 64, 21);  contentPane.add(btnNewButton\_20);    JButton btnNewButton\_21 = new JButton("Lịch sử");  btnNewButton\_21.setForeground(new Color(255, 255, 255));  btnNewButton\_21.setBackground(new Color(192, 192, 192));  btnNewButton\_21.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  showHistoryWindow();  }  });  btnNewButton\_21.setBounds(35, 10, 85, 21);  contentPane.add(btnNewButton\_21);    JButton btnNewButton\_22 = new JButton("Thoát");  btnNewButton\_22.setForeground(new Color(198, 54, 0));  btnNewButton\_22.setBackground(new Color(255, 255, 255));  btnNewButton\_22.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  dispose();  }  });  btnNewButton\_22.setBounds(341, 239, 85, 21);  contentPane.add(btnNewButton\_22);  }  public void connectToServer() {  try {  socket = new Socket("localhost", 12345);  out = new PrintWriter(socket.getOutputStream(), true);  in = new BufferedReader(new InputStreamReader(socket.getInputStream()));    new Thread(new Runnable() {  @Override  public void run() {  String serverMessage;  try {  while ((serverMessage = in.readLine()) != null) {  updateTextField(serverMessage);  addToHistory(serverMessage);  }  } catch (IOException e) {  e.printStackTrace();  }  }  }).start();  } catch (IOException e) {  e.printStackTrace();  }  }  private void showHistoryWindow() {  JFrame historyFrame = new JFrame("Lịch sử");  JPanel panel = new JPanel();  JTextArea textArea = new JTextArea(20, 30);  JScrollPane scrollPane = new JScrollPane(textArea);  textArea.setEditable(false);  textArea.setFont(new Font("Arial", Font.PLAIN, 14));  // Thêm lịch sử vào textArea  for (String item : historyList) {  textArea.append(item + "\n");  }  panel.add(scrollPane);  historyFrame.getContentPane().add(panel);  historyFrame.pack();  historyFrame.setVisible(true);  }  private void updateTextField(final String message) {  EventQueue.invokeLater(new Runnable() {  @Override  public void run() {  // Thêm tin nhắn mới vào cuối textField mà không ghi đè lên tin nhắn cũ  textField.setText(textField.getText() + message + "\n");  }  });  }  public void sendMessage(String message) {  if (out != null) {  out.println(message);  }  }  public static String getValueAfterEqualSign(String expression) {  // Check if the expression contains the equal sign  if (expression.contains("=")) {  // Split the string into parts separated by the equal sign  String[] parts = expression.split("=");  // Check if there's a part after the equal sign  if (parts.length > 1) {  // Return the part after the equal sign, trimmed of any leading/trailing whitespace  return parts[1].trim();  } else {  throw new IllegalArgumentException("Invalid expression: No value found after the equal sign.");  }  } else {  throw new IllegalArgumentException("Invalid expression: No equal sign found.");  }  }  public static String removeOuterCharacter(String input) {  // Kiểm tra xem chuỗi có đủ dài để có ký tự ngoài cùng hay không  if (input.length() >= 2) {  // Trả về phần của chuỗi từ vị trí 1 đến vị trí trước ký tự cuối cùng  return input.substring(0, input.length() - 1);  } else {  // Nếu chuỗi không đủ dài, không cần xóa ký tự ngoài cùng, trả về chuỗi gốc  return input;  }  }  } |

Mô tả: Ứng dụng máy tính này cho phép người dùng nhập các phép tính và gửi đến máy chủ để tính toán. Kết quả tính toán được trả về và hiển thị trong cửa sổ, cùng với lịch sử các phép tính đã thực hiện.

Giải thích chi tiết:

1. Phương thức Main:

Tạo một đối tượng của lớp MayTinh và hiển thị cửa sổ.

Kết nối với máy chủ thông qua phương thức connectToServer().

1. Constructor:

Thiết lập cấu trúc giao diện của cửa sổ JFrame.

Thêm các thành phần giao diện như ô nhập liệu, các nút số và các nút toán tử.

Gắn các sự kiện vào các nút để thêm chức năng cho ứng dụng.

Kết nối với máy chủ thông qua phương thức connectToServer().

1. Phương thức connectToServer():

Kết nối với máy chủ thông qua một socket.

Tạo một luồng để nhận tin nhắn từ máy chủ và cập nhật giao diện khi có tin nhắn mới.

1. Phương thức sendMessage():

Gửi tin nhắn đến máy chủ thông qua luồng ghi.

1. Phương thức getValueAfterEqualSign():

Cập nhật giao diện với tin nhắn từ máy chủ. Đảm bảo rằng tin nhắn mới được thêm vào cuối textField mà không ghi đè lên tin nhắn cũ.

1. Phương thức getValueAfterEqualSign():

Trích xuất giá trị sau dấu bằng trong biểu thức tính toán.

1. Phương thức removeOuterCharacter():

Loại bỏ ký tự ngoài cùng khỏi chuỗi.

1. Phương thức showHistoryWindow():

Hiển thị cửa sổ lịch sử khi người dùng nhấn vào nút "Lịch sử". Đây là nơi người dùng có thể xem lại các phép tính đã thực hiện.

Client\_TCP.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103 | package Client;  import javax.swing.\*;import java.awt.\*;import java.awt.event.ActionEvent;import java.awt.event.ActionListener;import java.io.BufferedReader;import java.io.IOException;import java.io.InputStreamReader;import java.io.PrintWriter;import java.net.Socket;  public class Client\_TCP extends JFrame {  private Socket client;  private BufferedReader in;  private PrintWriter out;  private boolean done;  private JTextField messageField;  private JTextArea chatArea;  public Client\_TCP() {  setTitle("Client");  setSize(400, 300);  setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  JPanel panel = new JPanel();  panel.setLayout(null);  chatArea = new JTextArea();  chatArea.setEditable(false);  JScrollPane scrollPane = new JScrollPane(chatArea);  scrollPane.setBounds(0, 0, 386, 242);  panel.add(scrollPane);  JPanel bottomPanel = new JPanel(new BorderLayout());  bottomPanel.setBounds(0, 242, 386, 21);  messageField = new JTextField();  bottomPanel.add(messageField, BorderLayout.CENTER);  JButton sendButton = new JButton("Send");  sendButton.setBackground(new Color(255, 255, 255));  sendButton.addActionListener(new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {  sendMessage();  }  });  bottomPanel.add(sendButton, BorderLayout.EAST);  panel.add(bottomPanel);    JButton btnNewButton = new JButton("Thoát");  btnNewButton.addActionListener(new ActionListener() {  public void actionPerformed(ActionEvent e) {  sendDisconect();  dispose();  }  });  btnNewButton.setBackground(new Color(255, 255, 255));  btnNewButton.setForeground(new Color(198, 54, 0));  bottomPanel.add(btnNewButton, BorderLayout.WEST);  getContentPane().add(panel);  try {  client = new Socket("127.0.0.1", 9999);  out = new PrintWriter(client.getOutputStream(), true);  in = new BufferedReader(new InputStreamReader(client.getInputStream()));  new Thread(new InputHandler()).start();  } catch (IOException e) {  e.printStackTrace();  }  }  private void sendMessage() {  String message = messageField.getText();  out.println(message);  chatArea.append("You: "+message+"\n");  messageField.setText("");  }  private void sendDisconect() {  String message = "Ngắt kết nối!!";  out.println(message);  }  private class InputHandler implements Runnable {  @Override  public void run() {  try {  String message;  while ((message = in.readLine()) != null) {  chatArea.append(message + "\n");  }  } catch (IOException e) {  e.printStackTrace();  }  }  }  public static void main(String[] args) {  SwingUtilities.invokeLater(new Runnable() {  @Override  public void run() {  new Client\_TCP().setVisible(true);  }  });  }  } |

Mô tả: Ứng dụng này cho phép người dùng nhập tin nhắn trong JTextField và gửi đến máy chủ. Các tin nhắn từ máy chủ sẽ được hiển thị trong JTextArea, giúp người dùng có thể theo dõi cuộc trò chuyện. Nút "Thoát" cho phép người dùng đóng ứng dụng và gửi thông báo ngắt kết nối tới máy chủ.

Giải thích chi tiết:

1. Constructor:

Thiết lập cấu trúc giao diện của cửa sổ JFrame.

Thêm một JTextArea để hiển thị các tin nhắn trong cửa sổ chat và một JTextField để nhập tin nhắn mới.

Thêm một nút "Send" để gửi tin nhắn.

Kết nối với máy chủ thông qua socket TCP.

Tạo một luồng mới để xử lý dữ liệu đầu vào từ máy chủ.

1. Phương thức sendMessage():

Lấy tin nhắn từ JTextField.

Gửi tin nhắn đến máy chủ thông qua luồng ghi (PrintWriter).

Hiển thị tin nhắn đã gửi trong JTextArea.

1. Phương thức sendDisconect():

Gửi tin nhắn "Ngắt kết nối!!" để thông báo cho máy chủ về việc ngắt kết nối.

1. Phương thức InputHandler:

Được triển khai từ Runnable, xử lý dữ liệu đầu vào từ máy chủ.

Mỗi khi có tin nhắn mới từ máy chủ, tin nhắn đó sẽ được hiển thị trong JTextArea.

1. Phương thức main():

Tạo một đối tượng của lớp Client\_TCP và hiển thị cửa sổ.

3.1.3 Test Demo

3.1.3.1 Hướng dẫn cách chạy chương trình.

Yêu cầu:

MySQL Workbench 8.0 CE

EclipseIDE

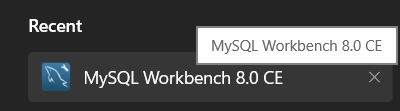
JavaSE-1.8

mysql-connector-j-8.3.0.jar (Project Final trong thư mục Demo đã cài đặt sẵn)

qlsv.sql (file chứa trong thư mục Demo)

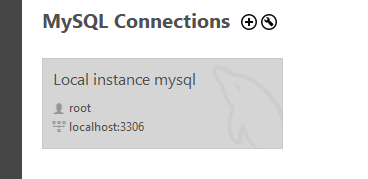
Cài đặt hệ cơ sở dữ liệu:

Bước 1: Mở MySQL Workbench 8.0 CE



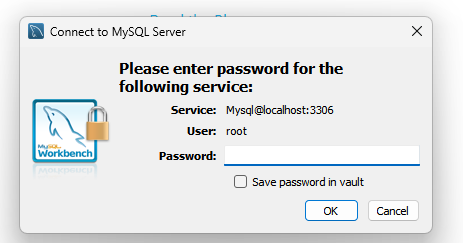
Hình 33: Mở MySQL Workbench.

Bước 2: Kết nối với MySQL Server.



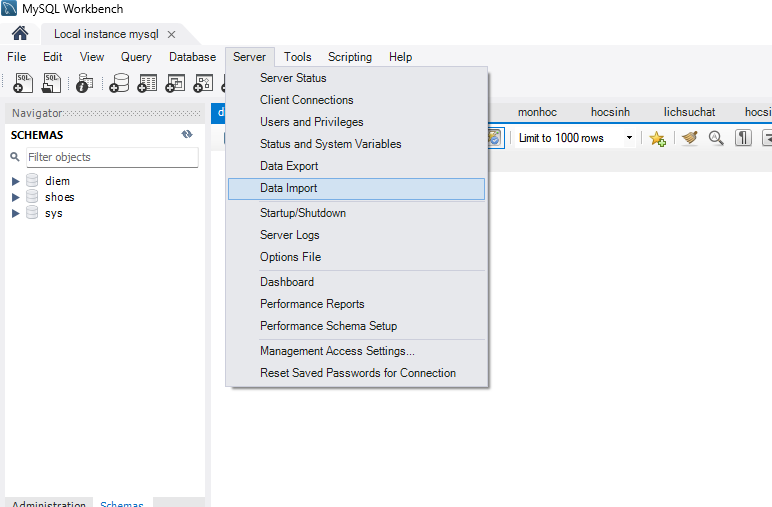
Hình 34: Kết nối với MySQL server.

Bước 3: Đăng nhập.



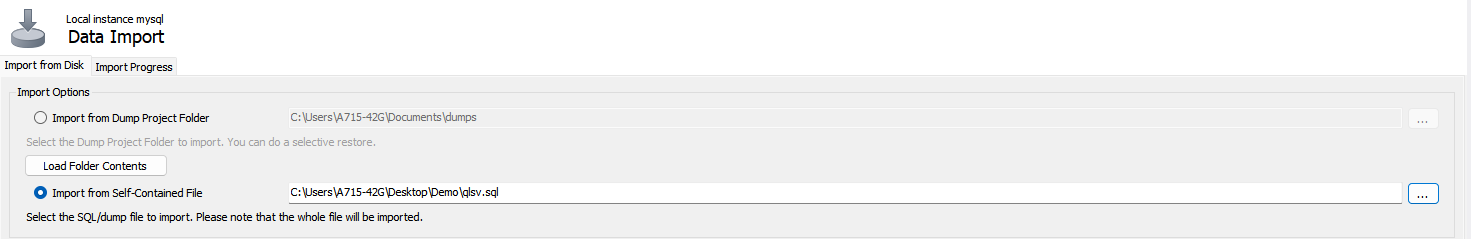
Hình 35: Đăng nhập.

Bước 4: Trên thanh công cụ chọn Server, tiếp tục chọn Data Import.



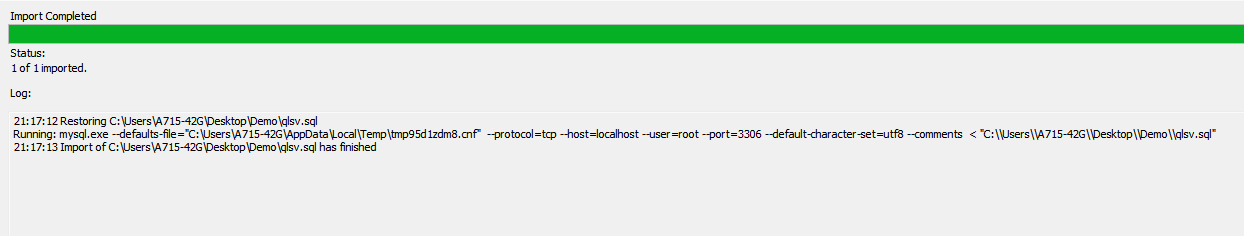
Hình 36: Chọn Server > chọn DataImport.

Bước 5: Chọn Import from Seft-Contained File. Dẫn đường link đến file qlsv.sql trong thư mục Demo. Sau đó bấm Start Import.



Hình 37: Import qlsv.sql.

Bước 6: Thông báo Import kết thúc.

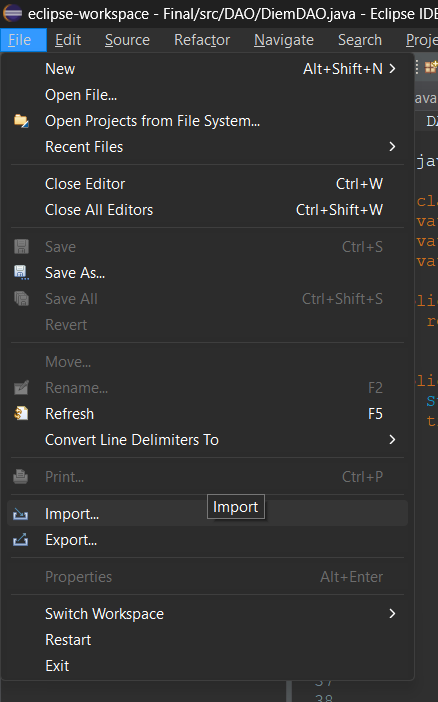


Hình 38: Import thành công.

Cài đặt dự án Java

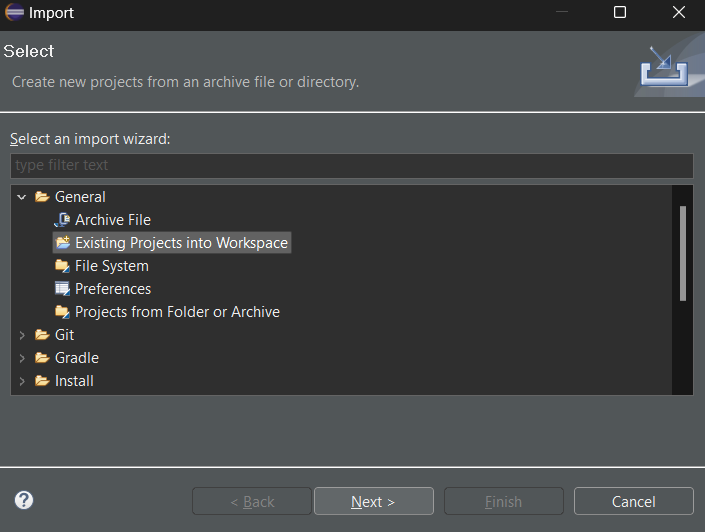
Bước 1: Mở Eclipse IDE.

Bước 2: Chọn File trên thanh công cụ, sau đó chọn Import.



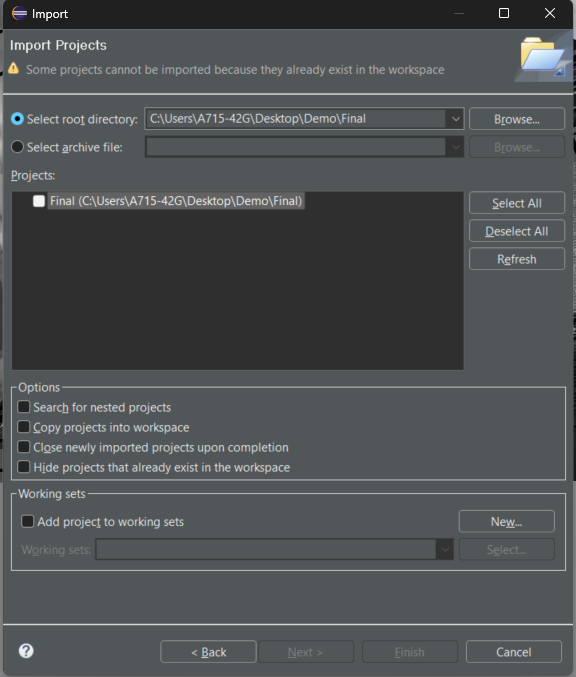
Hình 39: Import dự án java.

Bước 3: Chọn Existing Projects into Workspace trong thư mục General, sau đó nhấn Next.



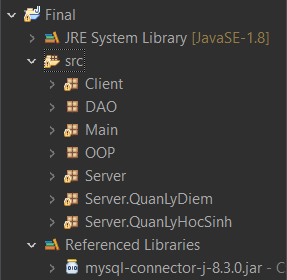
Hình 40: Chọn Existing Projects into Workspace.

Bước 4: Dẫn đường link đến thư mục Final trong thư mục Demo, sau đó nhấn Finish (Do em đã có dự án này rồi nên nút Finish không nhấn được).



Hình 41: Dẫn đường link và nhấn Finish.

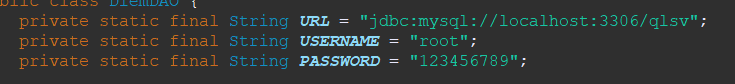
Bước 5: Thêm thành công dự án Final.



Hình 42: Dự án Final đã được thêm thành công.

Cài đặt 3 file DAO (Bỏ qua nếu username và password giống)

-Chỉnh sửa lại 3 file DAO nếu không giống với tài khoản kết nối với MySQL.

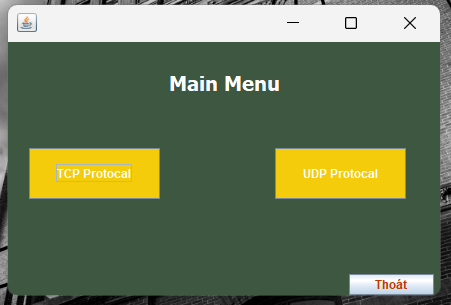


Hình 43: Cấu hình tài khoản kết nối.

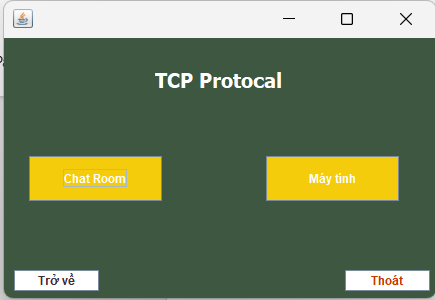
Sau khi đã thực hiện các bước trên chúng ta đã có thể chạy chương trình, file MainMenu.java nằm ở package Main. Hãy mở file và Run thử.

Chúng ta cũng có thể Run file MainMenu.jar ở thư mục Demo để kiểm tra.

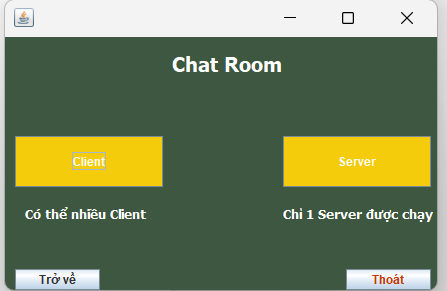
3.1.3.2 Kết quả test chương trình.



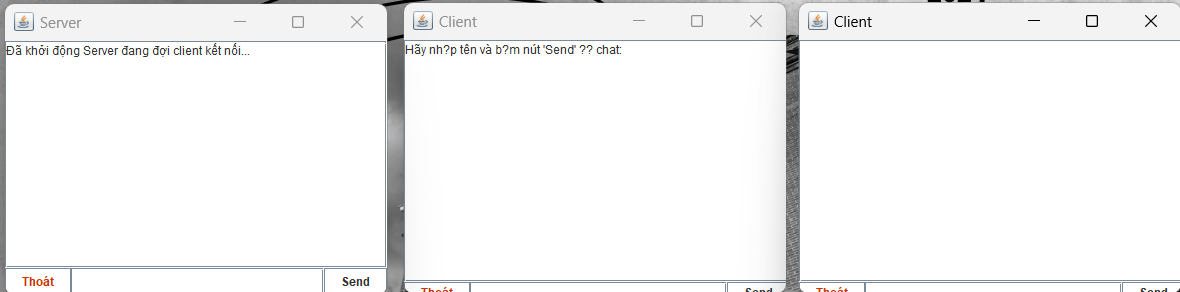
Hình 44: Mở MainMenu.jar



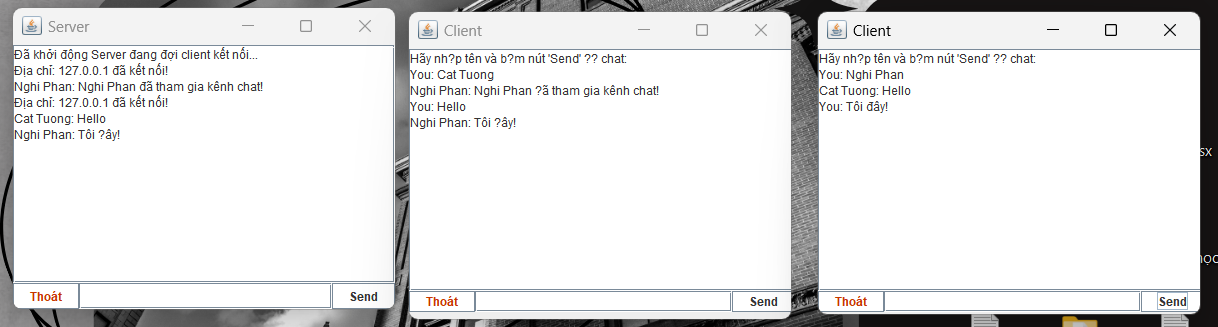
Hình 45: Mở frame TCP Protocal.



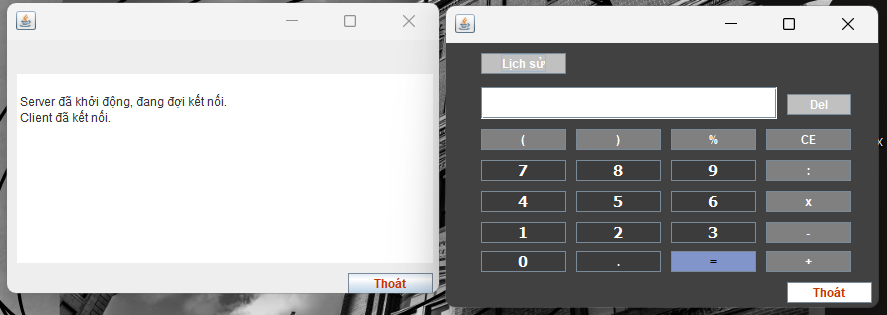
Hình 46: Mở frame Chat Room.



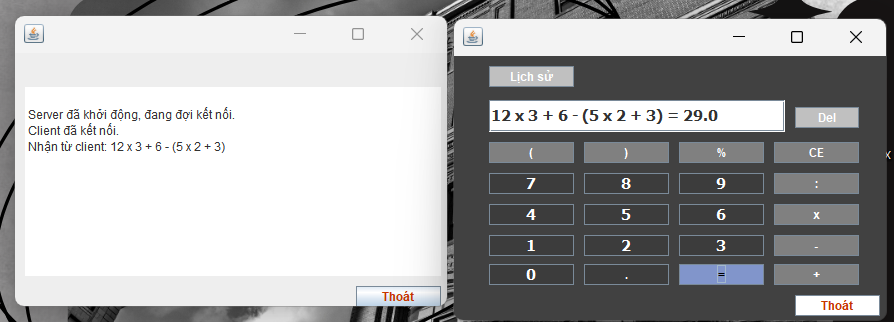
Hình 47: Mở Server và 2 Client.



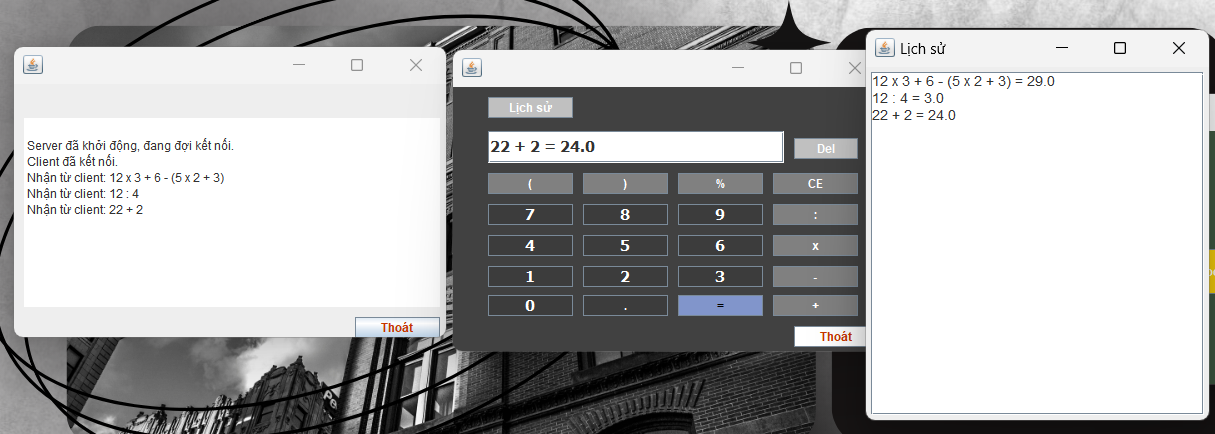
Hình 48: Thử giao tiếp với nhau.



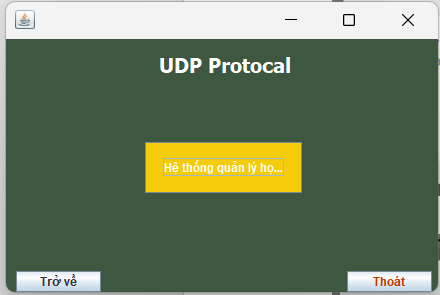
Hình 49: Bấm nút Máy tính để mở 2 frame MayTinh và Server.



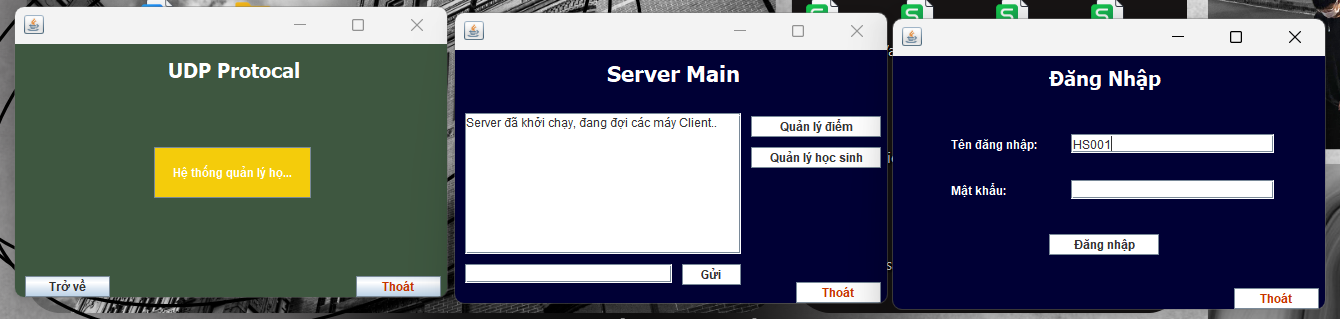
Hình 50: Thử tính toán.



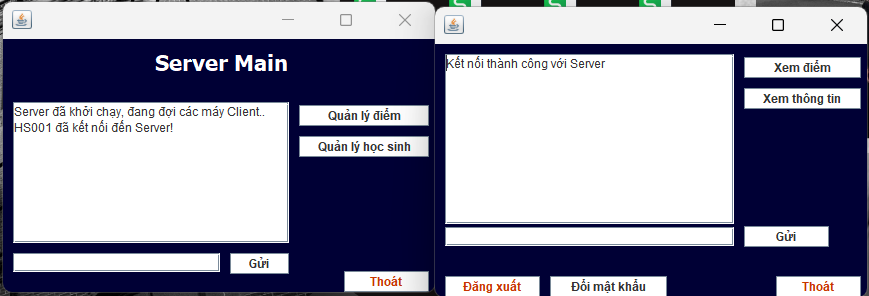
Hình 51: Chức năng xem lịch sử.



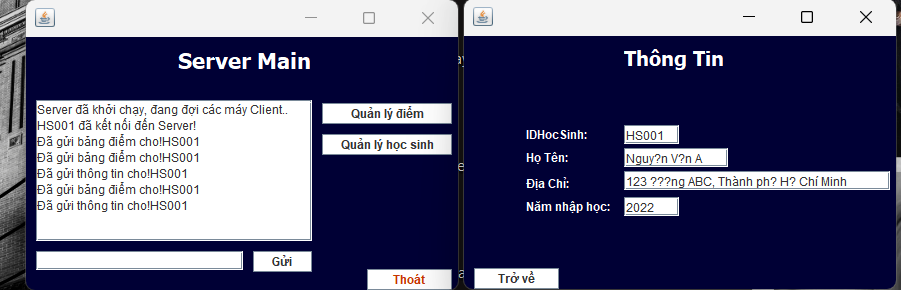
Hình 52: UDP Protocal frame.



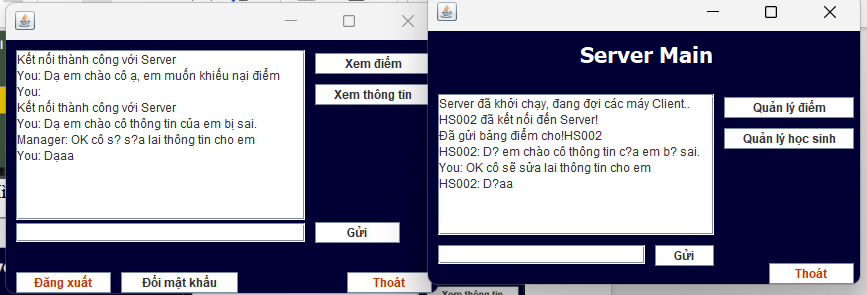
Hình 53: Bấm vào nút Hệ thống quản lý học sinh.



Hình 54: Đăng nhập với mật khẩu là "123".



Hình 55: Xem thông tin ở Client.



Hình 56: Chat giữa Client và Server



Hình 57: Quản lý điểm.



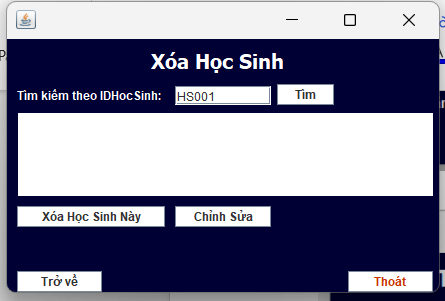
Hình 58: Chỉnh sửa điểm.



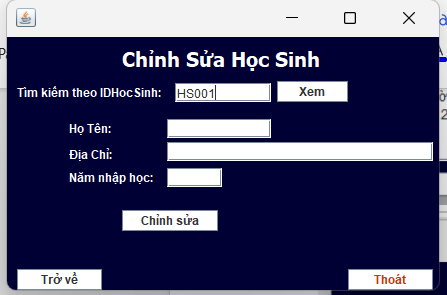
Hình 59: Quản lý học sinh.



Hình 60: Thêm học sinh.



Hình 61: Xóa học sinh.



Hình 62: Chỉnh sửa học sinh.

CHƯƠNG 4: KẾT LUẬN

* 1. Những kết quả dạt được.

- Xây dựng triển khai thành công ứng dụng Client - Server.

+ Ứng dụng giao diện đồ họa (GUI) cho phép người dùng gửi và nhận tin nhắn thông qua kết nối TCP và UDP.

+ Giao diện người dùng thân thiện với khung chat và các nút chức năng rõ ràng.

+ Chức năng gửi tin nhắn và hiển thị tin nhắn được thực hiện thành công.

+ Chức năng ngắt kết nối và đóng ứng dụng được triển khai.

* 1. Những hạn chế.

- Giao diện người dùng

+ Giao diện hiện tại chỉ cung cấp các chức năng cơ bản mà chưa có nhiều tính năng nâng cao như thông báo tin nhắn mới, lưu trữ lịch sử chat, hoặc hỗ trợ nhiều người dùng.

- Xử lý lỗi và thông báo lỗi:

+ Ứng dụng chưa có hệ thống xử lý và thông báo lỗi rõ ràng khi xảy ra sự cố kết nối hoặc các lỗi khác trong quá trình hoạt động.

- Bảo mật:

+ Hiện tại ứng dụng chưa có các biện pháp bảo mật cần thiết như mã hóa dữ liệu truyền tải hay xác thực người dùng.

* 1. Hướng phát triển.

- Nâng cấp giao diện người dùng:

+ Cải thiện giao diện người dùng với các tính năng như thông báo tin nhắn mới, hỗ trợ gửi tệp tin, lưu trữ và tìm kiếm lịch sử chat.

- Tăng cường bảo mật:

+ Triển khai các biện pháp bảo mật như mã hóa SSL/TLS cho kết nối TCP, xác thực người dùng trước khi cho phép truy cập vào hệ thống.

-Cải thiện khả năng xử lý lỗi:

-Xây dựng hệ thống xử lý lỗi và thông báo lỗi rõ ràng để giúp người dùng và nhà phát triển dễ dàng nhận biết và khắc phục các sự cố.

TÀI LIỆU THAM KHẢO

**Tiếng Việt**

[1] Giáo Trình Lập Trình Mạng - Hà Mạnh Đào https://www.studocu.com/vn/document/truong-dai-hoc-bach-khoa-dai-hoc-da-nang/ki-thuat-lap-trinh/giao-trinh-lap-trinh-mang-ha-manh-dao-1009170/35847155

**Tiếng Anh**