TRƯỜNG ĐẠI HỌC GIAO THÔNG VÂN TẢI

KHOA CÔNG NGHỆ THÔNG TIN

---------------o0o---------------



**BÀI TẬP LỚN MÔN HỌC** **CÔNG NGHỆ JAVA**

Đề tài*: Thiết kế app quản lí tài chính cá nhân*

**Giảng viên hướng dẫn:** TS. Vũ Huấn

**Sinh viên thực hiện:** Đinh Ngọc Trung

**Lớp & Mã sinh viên:** CNTT 6- K62- 211203908

**Hà Nội, tháng 11 năm 2022**

Mục lục

[**LỜI NÓI ĐẦU** 3](#_Toc134043808)

[**I. Tổng quan và mô tả chương trình:** 4](#_Toc134043809)

[1. Giới thiệu chung về công nghệ sử dụng 4](#_Toc134043810)

[2. Tổng quan: 4](#_Toc134043811)

[3. Mô tả bài toán: 4](#_Toc134043812)

[**II. Thiết kế chương trình:** 6](#_Toc134043813)

[**III. Thực thi và cài đặt:** 7](#_Toc134043814)

[**1.** **Controller:** 7](#_Toc134043815)

[**2.** **Lib:** 8](#_Toc134043816)

[**3.** **Model:** 8](#_Toc134043817)

[**4.** **View:** 8](#_Toc134043818)

[**III. Một số chức năng chính:** 10](#_Toc134043819)

[**1.** **Phân chia chi tiêu:** 10](#_Toc134043820)

[**2.** **Thiết lập giới hạn chi tiêu:** 11](#_Toc134043821)

[**3.** **Sử dụng cơ sở dữ liệu để lưu trữ thông tin về các khoản chi tiêu của người dùng:** 12](#_Toc134043822)

[**IV. Kết luận:** 16](#_Toc134043823)

[1. Kết quả đạt được 16](#_Toc134043824)

[2. Hạn chế 16](#_Toc134043825)

[3. Hướng phát triển 16](#_Toc134043826)

[**V. Tài liệu tham khảo:** 17](#_Toc134043827)

# **LỜI NÓI ĐẦU**

Quản lý tài chính cá nhân là một kỹ năng quan trọng mà mỗi người cần phải có để có một cuộc sống vững vàng và độc lập tài chính. Việc quản lý tài chính cá nhân giúp chúng ta tiết kiệm tiền, tránh khỏi nợ nần, tạo dựng được sự độc lập tài chính và xây dựng một cuộc sống ổn định. Đầu tiên, quản lý tài chính cá nhân giúp chúng ta tiết kiệm tiền bằng cách phát hiện ra những khoản chi không cần thiết và từ đó cắt giảm chi phí. Thứ hai, việc quản lý tài chính cá nhân giúp chúng ta tránh khỏi nợ nần bằng cách kiểm soát và quản lý các khoản nợ một cách hiệu quả. Thứ ba, việc quản lý tài chính cá nhân giúp chúng ta đạt được sự độc lập tài chính, tức là có thể tự mình kiếm tiền và sử dụng tiền một cách thông minh.

Cuối cùng, việc quản lý tài chính cá nhân giúp chúng t xây dựng được một cuộc sống vững vàng và ổn định bằng cách tạo ra một kế hoạch tài chính dài hạn và đầu tư vào các khoản đầu tư có lợi. Mặc dù việc quản lý tài chính cá nhân có thể khó khăn đôi khi, nhưng đó là một kỹ năng có thể học được. Bằng cách học và thực hành, chúng ta có thể quản lý tài chính cá nhân một cách hiệu quả và đạt được sự thành công trong cuộc sống.

Báo cáo này sẽ trình bày vế quá trình phát triển ứng dụng quản lí tài chính, bao gồm các bước từ thiết kế đến triển khai và kiểm thử. Em cũng sẽ trình bày về các tính năng cơ bản của ứng dụng, cách thức hoạt động, các thuật toán và cấu trúc dữ liệu được sử dụng, và các tính năng bổ sung khác.

Công nghệ nền tảng mà em sử dụng là Java. Lập trình game bằng Java đòi hỏi kiến thức về lập trình hướng đối tượng và các thư viện đồ họa như AWT, Swing để tạo ra giao diện đồ họa. Sử dụng Java để phát triển ứng dụng cũng có lợi thế về tính di động và đa nền tảng, vì Java được hỗ trợ trên nhiều hệ điều hành khác nhau.

# **I. Tổng quan và mô tả chương trình:**

## Giới thiệu chung về công nghệ sử dụng

Java là một trong những ngôn ngữ lập trình hướng đối tượng. Nó được sử dụng trong phát triển phần mềm, trang web, game hay ứng dụng trên các thiết bị di động.

Java được khởi đầu bởi James Gosling và bạn đồng nghiệp ở Sun MicroSystem năm 1991. Ban đầu Java được tạo ra nhằm mục đích viết phần mềm cho các sản phẩm gia dụng, và có tên là Oak.

Java được phát hành năm 1994, đến năm 2010 được Oracle mua lại từ Sun MicroSystem.

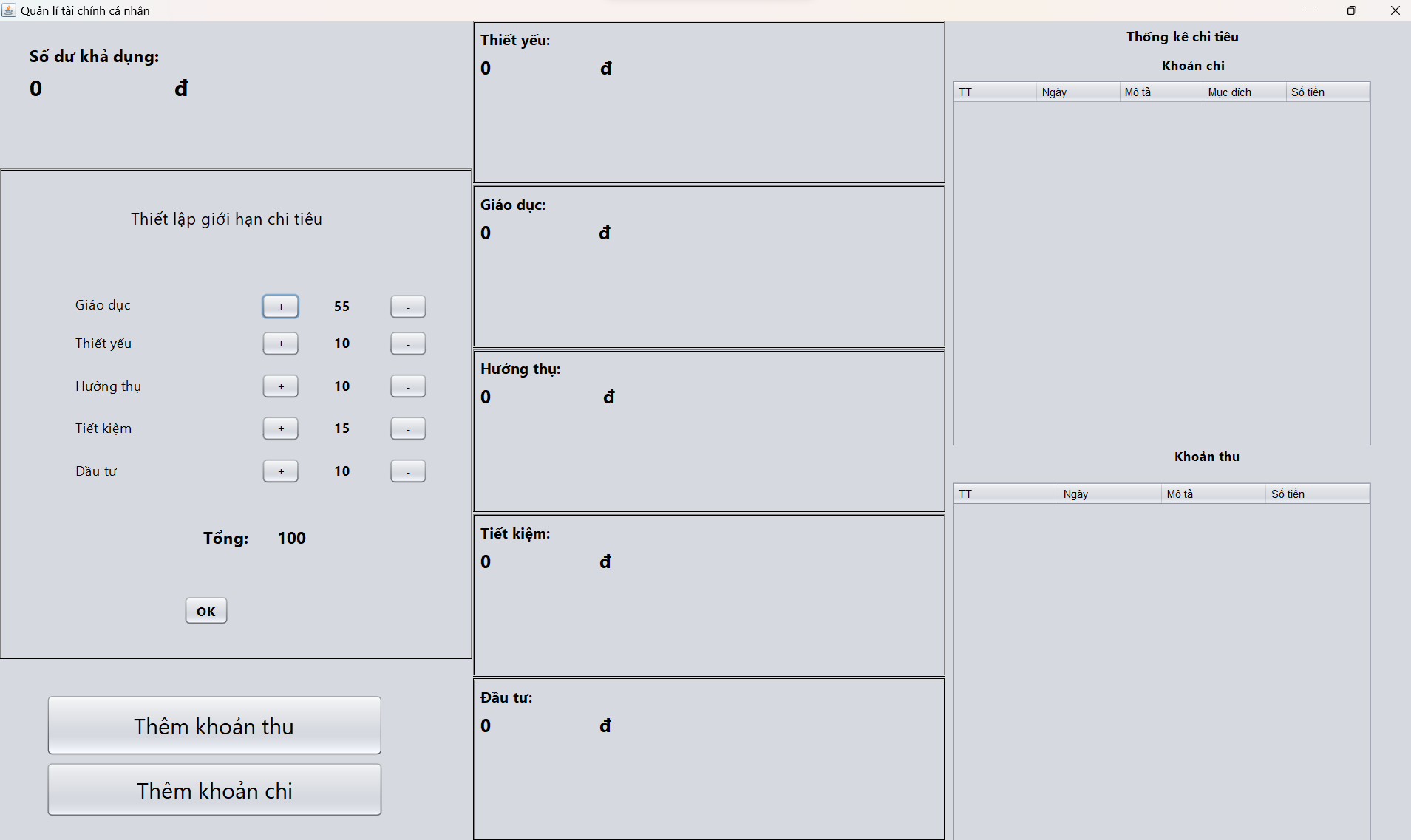
Java được tạo ra với tiêu chí “Viết (code) một lần, thực thi khắp nơi” (Write Once, Run Anywhere – WORA). Chương trình phần mềm viết bằng Java có thể chạy trên mọi nền tảng (platform) khác nhau thông qua một môi trường thực thi với điều kiện có môi trường thực thi thích hợp hỗ trợ nền tảng đó.

## Tổng quan:

Thiết kế app quản lí tài chính cá nhân được thưc hiện theo quy tắc 5 chiếc hũ chia theo tỉ lệ mặc định:

* 55% cho nhu cầu thiết yếu (tiền điện, tiền nhà, ăn uống…)
* 15% cho các hoạt động giáo dục (tiền học, mua sách…)
* 10% cho nhu cầu hưởng thụ (đi chơi, xem phim, mua sắm…)
* 10% dành cho hoạt động đầu tư (mua đất, chứng khoán, mua vàng…)
* 10% để tiết kiệm (mua xe, đi du lịch, hoặc dùng cho các trường hợp phát sinh chi phí ngoài ý muốn: chữa bệnh, đám cưới…)

## Mô tả bài toán:

* Khởi tạo app: một frame hiển thị thông tin về số dư trong tài khoản, tiền trong từng phần (thiết yếu, giáo dục, hưởng thụ, đầu tư, tiết kiệm), phần thiết lập giới hạn chi tiêu, thống kê chi tiêu.
* User: sử dụng chuột để thao tác.
* Số dư khả dụng: cho người dùng biết số tiền hiện có của mình, nếu chi tiêu vượt quá số dư thì sẽ hiện khoản tiền âm và được liên kết với cơ sở dữ liệu
* Thiết lập giới hạn chi tiêu: cho phép người thiết lập lại từng mục chi tiêu để phù hợp với mức thu nhập và chi tiêu của mình. Nếu tổng vượt quá 100 thì sẽ có thông báo lỗi.
* Thêm khoản thu & thêm khoản chi: cho phép người dùng thêm khoản thu hoặc khoản chi trong ngày của mình kèm với mô tả cho khoản thu và khoản chi đó.
* Phần chính giữa: hiển thị số tiền trong từng mục chi tiêu, có thể hiện phần âm.
* Thống kê thu chi: cho người dùng biết những khoản thu chi của mình gồm số thứ tự, ngày, mô tả, mục đích, số tiền. Được liên kết với cơ sở dữ liệu để lưu lại thông tin thu chi.

# **II. Thiết kế chương trình:**

**Bước 1: Xác định yêu cầu**

Bước đầu tiên là xác định các yêu cầu chức năng và phi chức năng của chương trình. Điều này bao gồm sự tìm hiểu về mục đích của chương trình, các tính năng cần thiết, dữ liệu đầu vào và đầu ra cũng như các ràng buộc về kỹ thuật và chức năng.

**Bước 2: Phác thảo chương trình**

Sau bước xác định các yêu cầu của chương trình là quá trình lập kế hoạch thiết kế. Kế hoạch này bao gồm việc phân tích và thiết kế kiến trúc chương trình, chỉ định chức năng, tạo sơ dữ liệu.

**Bước 3: Thiết kế giao diện**

Thiết kế giao diện bao gồm tạo bố cục, chọn màu, định vị các nút và bố cục giao diện.

**Bước 4: Lập trình**

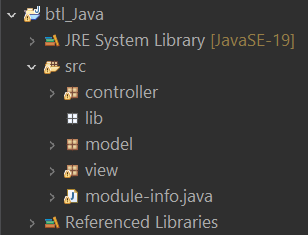
Sử dụng Java để có thể lập trình các đối tượng, tương tác, chức năng, cũng như điều chỉnh theo bố cục đã có.

**Bước 5: Sửa lỗi**

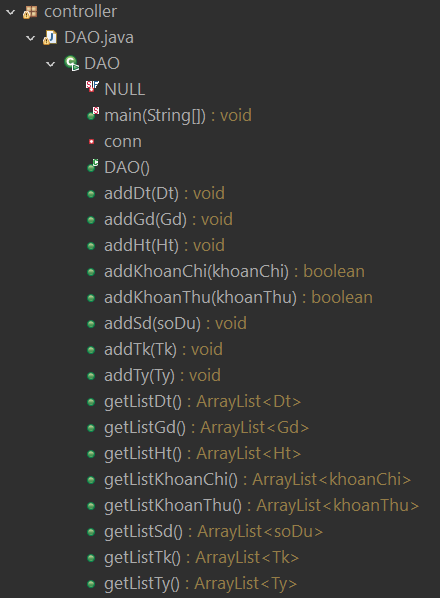
Sau khi viết xong mã, bước tiếp theo là kiểm tra chương trình để đảm bảo nó hoạt động đúng và không có lỗi.

# **III. Thực thi và cài đặt:**

Để xây dựng em đã chia chương trình thành 4 package:



1. **Controller:**



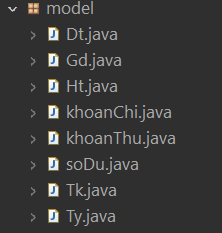
* Lớp DAO: tạo liên kết với cơ sở dữ liệu.

1. **Lib:**

nơi add 2 thư viện cần sử dụng để kết nối với cơ sở dữ liệu



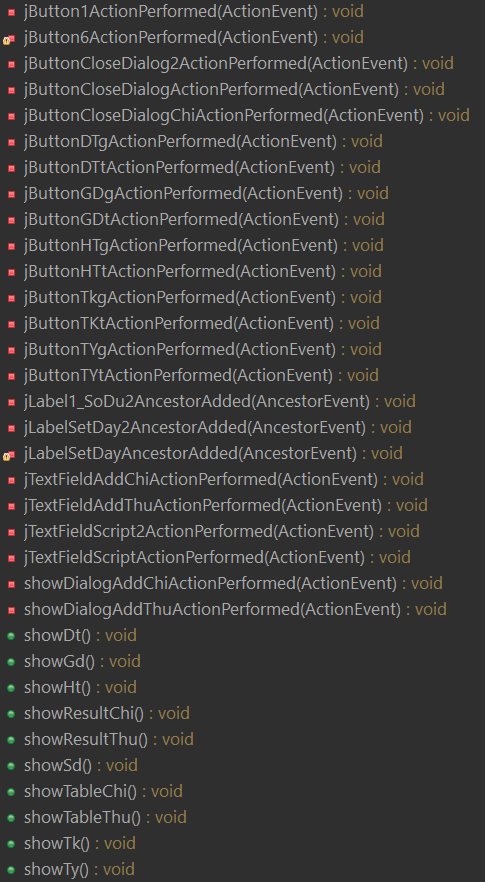
1. **Model:**



Nơi chứa thuật toán những đối tượng chính của chương trình: số dư, khoản thu, khoản chi, thiết yếu, giáo dục, hưởng thụ, tiết kiệm, đầu tư.

1. **View:**





Chứa giao diện của app và xử lí sự kiện của người dùng.

# **III. Một số chức năng chính:**

1. **Phân chia chi tiêu:**

Sau khi nhập khoản thu thì số tiền sẽ lập tức được chia đều ra các hũ theo tỉ lệ mặc định 55-15-10-10-10.

private void jButtonCloseDialog2ActionPerformed(java.awt.event.ActionEvent evt) {

jDialogAddThu.setVisible(false);

int x=Integer.*parseInt*(jTextFieldAddThu.getText());

int y=Integer.*parseInt*(jLabel1\_SoDu2.getText());

jLabel1\_SoDu2.setText(x+y+"");

int a= Integer.*parseInt*(jLabelSetTY.getText());

int b= Integer.*parseInt*(jLabelSetGD.getText());

int c= Integer.*parseInt*(jLabelSetHT.getText());

int d= Integer.*parseInt*(jLabelSetTK.getText());

int e= Integer.*parseInt*(jLabelSetDT.getText());

int f=Integer.*parseInt*(jLabel1\_SoDu2.getText());

if(a==0) {

jLabelTY.setText(jLabelTY.getText());

} else jLabelTY.setText(f\*a/100+"");

if(b==0) {

jLabelGD.setText(jLabelGD.getText());

} else jLabelGD.setText(f\*b/100+"");

if(c==0) {

jLabelHT.setText(jLabelHT.getText());

} else jLabelHT.setText(f\*c/100+"");

if(d==0) {

jLabelTK.setText(jLabelTK.getText());

} else jLabelTK.setText(f\*d/100+"");

if(e==0) {

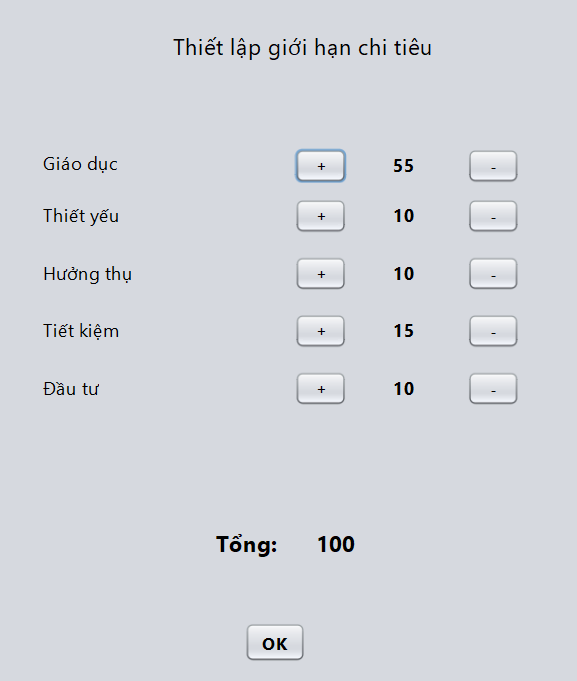
jLabelDT.setText(jLabelDT.getText());

} else jLabelDT.setText(f\*e/100+"");

}

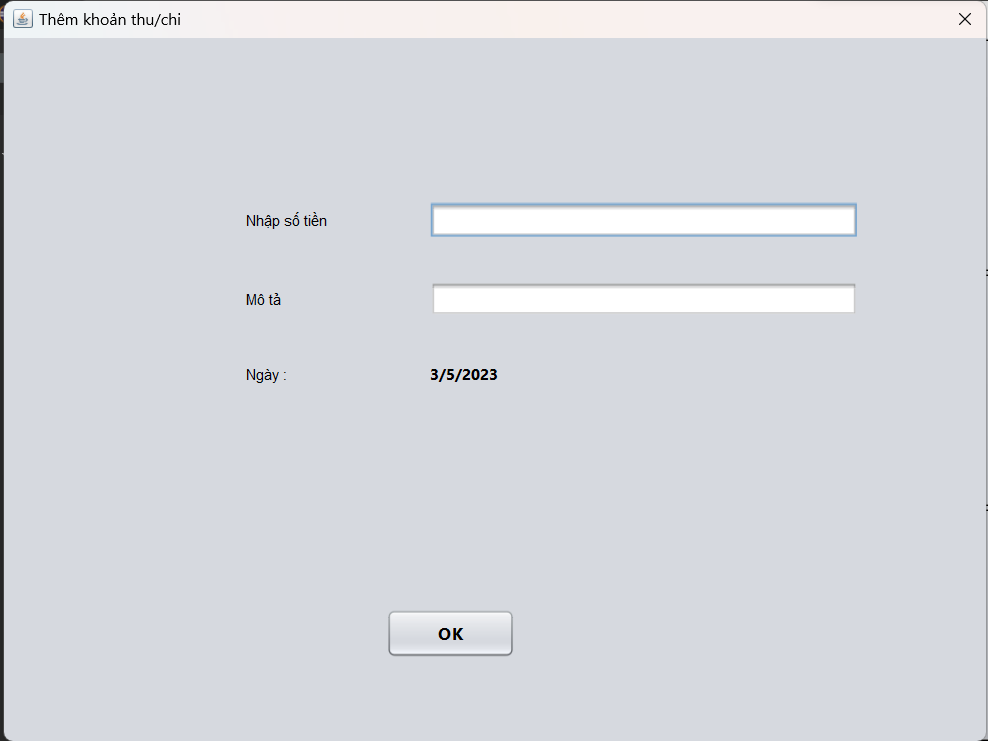
1. **Thiết lập giới hạn chi tiêu:**

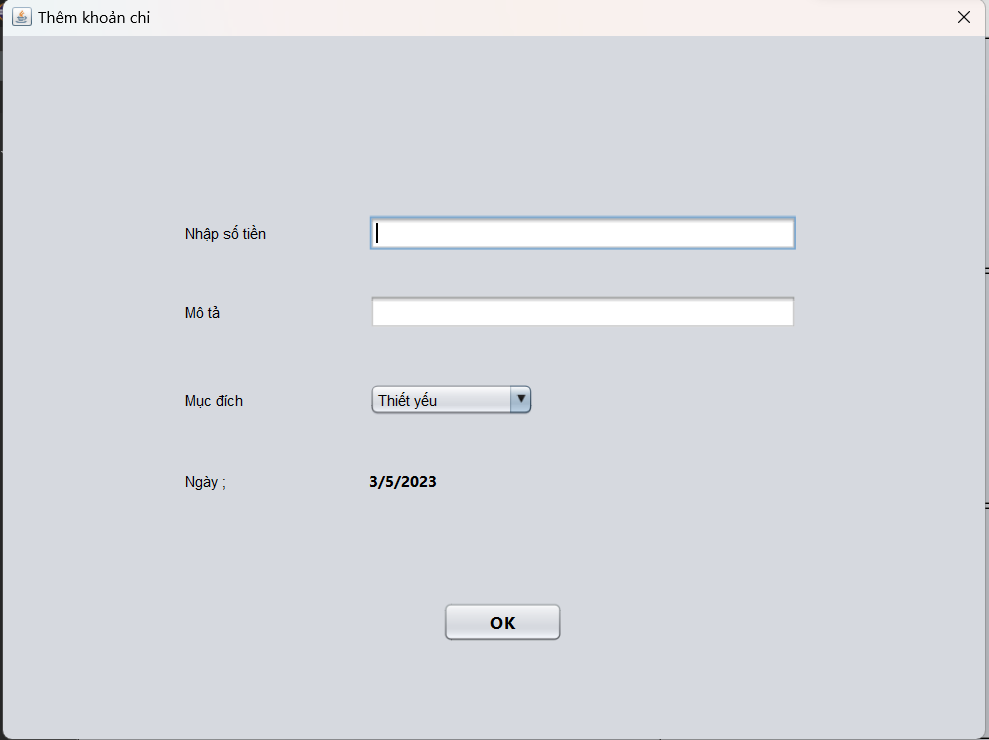
Người dùng có thể tùy chỉnh % số tiền được phân bổ vào từng hũ để phù hợp với mức thu nhập và chi tiêu của mình.



Nếu tổng >100 thì sẽ xuất hiện thông báo lỗi và yêu cầu người dùng thiết lập lại.

1. **Sử dụng cơ sở dữ liệu để lưu trữ thông tin về các khoản chi tiêu của người dùng:**





public DAO(){

try {

DriverManager.*registerDriver*(new com.mysql.jdbc.Driver());

String url= "jdbc:mySQL://127.0.0.1:3306/thuChi\_day?useSSL=false";

String username= "root";

String password= "";

conn = DriverManager.*getConnection*(url, username, password);

System.***out***.println("ok");

} catch (SQLException e) {

e.printStackTrace();

}

}

public boolean addKhoanThu(khoanThu kt){

String sql = "INSERT INTO `thu`(`TT`,`Ngày`, `Mô tả`, `Số tiền`) "

+ "VALUES(?,?, ?,?);";

try {

PreparedStatement ps = conn.prepareStatement(sql);

ps.setInt(1, ***NULL***);

ps.setDate(2, new Date(kt.getDay().getTime()));

ps.setString(3, kt.getMoTa());

ps.setInt(4, kt.getSoTien());

return ps.executeUpdate() > 0;

} catch (Exception e) {

e.printStackTrace();

}

return false;

}

public ArrayList<khoanThu> getListKhoanThu(){

ArrayList<khoanThu> list = new ArrayList<>();

String sql = "SELECT \* FROM thu;";

try {

PreparedStatement ps = conn.prepareStatement(sql);

ResultSet rs = ps.executeQuery();

while(rs.next()){

khoanThu kt = new khoanThu();

kt.setDay(rs.getDate("Ngày"));

kt.setMoTa(rs.getString("Mô tả"));

kt.setSoTien(rs.getInt("Số tiền"));

list.add(kt);

}

} catch (Exception e) {

e.printStackTrace();

}

return list;

}

Trên là ví dụ thêm khoản thu vào cơ sở dữ liệu my SQL. Và ta sử dụng ArrayList để lưu thông tin về khoản thu và được hiện sẵn ra khi người dùng khởi động ứng dụng.

# **IV. Kết luận:**

## Kết quả đạt được

* Ứng dụng đã có các gần như đầy đủ chức năng để giúp người dùng quản lí tài chính cá nhân hiệu quả.
* Giao diện dễ sử dụng, dễ tiếp cận.

## Hạn chế

* Ứng dụng chưa tối ưu được 100% cho người dùng.
* Vẫn có thể xuất hiện một sô lỗi nhỏ, nhưng không đáng kể. Một số chức năng chưa hoàn thiện.
* Giao diện ứng dụng còn khá đơn giản.

## Hướng phát triển

* Tối ưu thao tác.
* Sử dụng thêm màu sắc giúp giao diện nâng cao trải nghiệm với người dùng
* Hoàn thiện chức năng thiết lập giới hạn chi tiêu.
* Thiết kế thêm phần thống kê thu nhập theo tuần, tháng, năm để nâng cao tối đa tính ứng dụng vào thực tiễn

# **V. Tài liệu tham khảo:**

Power point bài giảng môn công nghệ java.