Banking System using Core Java

This project is a simplified example of a banking system, and it uses text files to store account information. It demonstrates file I/O (input/output) (File Handling) operations and basic error handling. Users can perform banking operations in a continuous loop until they choose to exit the program.

Here is the source code :

import java.io.\*;

import java.util.Scanner;

public class BankingSystem {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

while (true) {

System.out.println("Welcome to the Banking System");

System.out.println("1. Create Account");

System.out.println("2. Deposit Money");

System.out.println("3. Withdraw Money");

System.out.println("4. Check Balance");

System.out.println("5. Exit");

System.out.print("Enter your choice: ");

int choice = scanner.nextInt();

switch (choice) {

case 1:

createAccount();

break;

case 2:

depositMoney();

break;

case 3:

withdrawMoney();

break;

case 4:

checkBalance();

break;

case 5:

System.out.println("Exiting the program.");

System.exit(0);

default:

System.out.println("Invalid choice. Please try again.");

}

}

}

public static void createAccount() {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter account number: ");

String accountNumber = scanner.nextLine();

System.out.print("Enter account holder's name: ");

String accountHolder = scanner.nextLine();

System.out.print("Enter initial balance: ");

double balance = scanner.nextDouble();

try {

FileWriter fileWriter = new FileWriter(accountNumber + ".txt");

PrintWriter printWriter = new PrintWriter(fileWriter);

printWriter.println("Account Number: " + accountNumber);

printWriter.println("Account Holder: " + accountHolder);

printWriter.println("Balance: " + balance);

printWriter.close();

System.out.println("Account created successfully.");

} catch (IOException e) {

System.out.println("An error occurred.");

e.printStackTrace();

}

}

public static void depositMoney() {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter account number: ");

String accountNumber = scanner.nextLine();

System.out.print("Enter the amount to deposit: ");

double amount = scanner.nextDouble();

try {

File file = new File(accountNumber + ".txt");

if (file.exists()) {

BufferedReader reader = new BufferedReader(new FileReader(file));

String line;

StringBuilder content = new StringBuilder();

while ((line = reader.readLine()) != null) {

content.append(line).append("\n");

}

reader.close();

// Parse the balance from the existing file content

String[] lines = content.toString().split("\n");

double currentBalance = Double.parseDouble(lines[2].substring(9));

// Update the balance

double newBalance = currentBalance + amount;

lines[2] = "Balance: " + newBalance;

// Write the updated content back to the file

FileWriter fileWriter = new FileWriter(accountNumber + ".txt");

PrintWriter printWriter = new PrintWriter(fileWriter);

for (String updatedLine : lines) {

printWriter.println(updatedLine);

}

printWriter.close();

System.out.println("Deposit successful.");

} else {

System.out.println("Account not found.");

}

} catch (IOException e) {

System.out.println("An error occurred.");

e.printStackTrace();

}

}

public static void withdrawMoney() {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter account number: ");

String accountNumber = scanner.nextLine();

System.out.print("Enter the amount to withdraw: ");

double amount = scanner.nextDouble();

try {

File file = new File(accountNumber + ".txt");

if (file.exists()) {

BufferedReader reader = new BufferedReader(new FileReader(file));

String line;

StringBuilder content = new StringBuilder();

while ((line = reader.readLine()) != null) {

content.append(line).append("\n");

}

reader.close();

// Parse the balance from the existing file content

String[] lines = content.toString().split("\n");

double currentBalance = Double.parseDouble(lines[2].substring(9));

// Check if the withdrawal amount is valid

if (amount <= currentBalance) {

// Update the balance

double newBalance = currentBalance - amount;

lines[2] = "Balance: " + newBalance;

// Write the updated content back to the file

FileWriter fileWriter = new FileWriter(accountNumber + ".txt");

PrintWriter printWriter = new PrintWriter(fileWriter);

for (String updatedLine : lines) {

printWriter.println(updatedLine);

}

printWriter.close();

System.out.println("Withdrawal successful.");

} else {

System.out.println("Insufficient balance.");

}

} else {

System.out.println("Account not found.");

}

} catch (IOException e) {

System.out.println("An error occurred.");

e.printStackTrace();

}

}

public static void checkBalance() {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter account number: ");

String accountNumber = scanner.nextLine();

try {

File file = new File(accountNumber + ".txt");

if (file.exists()) {

BufferedReader reader = new BufferedReader(new FileReader(file));

String line;

while ((line = reader.readLine()) != null) {

if (line.startsWith("Balance: ")) {

double balance = Double.parseDouble(line.substring(9));

System.out.println("Current Balance: " + balance);

break;

}

}

reader.close();

} else {

System.out.println("Account not found.");

}

} catch (IOException e) {

System.out.println("An error occurred.");

e.printStackTrace();

}

}

}