OOPS WITH JAVA

CS23333

MINI PROJECT

SIMPLE BANKING APPLICATION

DONE BY:

NAME: SAKTHI ELUMALAI MANICKAVELU
ROLL NUMBER: 231401091
CLASS: CSBS-'B'

AIM:

The aim of the project is to develop a **Simple Banking Application** in Java that allows users to perform basic banking operations such as depositing money, withdrawing money, checking their account balance, and exiting the application. This project demonstrates the use of object-oriented programming (OOP) concepts, the `Scanner` class for user input, and basic control structures like loops and conditionals.

ALGORITHM:

1. Start the Program

- Import the `Scanner` class for user input.
- Define the `SimpleBankingApplication` class with attributes: `accountHolderName`, `accountNumber`, and `balance`.

2. Initialize the Account

- Prompt the user to enter their name, account number, and initial balance.
- Create an object of the `SimpleBankingApplication` class with the provided details.

3. Display Menu Options

- Present a menu with the following options in a loop:
    1. Deposit Money
    2. Withdraw Money
    3. Check Balance
    4. Exit

4. Handle User Input

- Prompt the user to select an option.
- Based on the user's choice, perform the corresponding action:
    - **Deposit**:

        - Prompt the user to enter the deposit amount.
        - Add the amount to the balance if it's valid (greater than 0).
    - **Withdraw**:
        - Prompt the user to enter the withdrawal amount.
        - Deduct the amount from the balance if sufficient funds are available.
    - **Check Balance**:
        - Display the current account balance.
    - **Exit**:
        - Thank the user and terminate the program.

5. Handle Invalid Inputs

- If the user enters an invalid option, prompt them to re-enter a valid choice.

6. Exit the Program

- Close the `Scanner` object and terminate the program when the user selects the exit option.

PROGRAM:

```
import java.util.Scanner;

public class SimpleBankingApplication {

    private double balance;

    private String accountHolderName;

    private String accountNumber;


    public SimpleBankingApplication(String accountHolderName, String accountNumber, double initialBalance) {

    this.accountHolderName = accountHolderName;
```

```
this.accountNumber = accountNumber;

this.balance = initialBalance;

}

public void deposit(double amount) {

if (amount > 0) {

balance += amount;

System.out.println("Hi " + accountHolderName + " (Account: " +
accountNumber + "), you have successfully deposited " + amount);

} else {

System.out.println("Invalid deposit amount. Try again.");

}

}

public void withdraw(double amount) {

if (amount > 0 && amount <= balance) {

balance -= amount;

System.out.println("Hi " + accountHolderName + " (Account: " +
accountNumber + "), you have successfully withdrawn " + amount);

} else if (amount > balance) {

System.out.println("Hi " + accountHolderName + " (Account: " +
accountNumber + "), you have insufficient balance for this transaction.");

} else {

System.out.println("Invalid withdrawal amount. Try again.");
```

```
    }

    }

    public void checkBalance() {

    System.out.println("Hi " + accountHolderName + " (Account: " +
accountNumber + "), your current balance is " + balance);

    }


    public static void main(String[] args) {

    Scanner scanner = new Scanner(System.in);


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

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

    String name = scanner.nextLine();

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

    String accountNumber = scanner.nextLine();

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

    double initialBalance = scanner.nextDouble();

    scanner.nextLine(); // Clear the leftover newline character


    SimpleBankingApplication account = new
SimpleBankingApplication(name, accountNumber, initialBalance);

    int choice;
```

```
do {
System.out.println("\n--- Menu ---");
System.out.println("1. Deposit");
System.out.println("2. Withdraw");
System.out.println("3. Check Balance");
System.out.println("4. Exit");
System.out.print("Enter your choice: ");
choice = scanner.nextInt();
scanner.nextLine(); // Clear the leftover newline character

switch (choice) {
        case 1:
        System.out.print("Enter deposit amount: ");
        double depositAmount = scanner.nextDouble();
        scanner.nextLine(); // Clear the leftover newline character
        account.deposit(depositAmount);
        break;
        case 2:
        System.out.print("Enter withdrawal amount: ");
        double withdrawalAmount = scanner.nextDouble();
        scanner.nextLine(); // Clear the leftover newline character
```

```
                account.withdraw(withdrawalAmount);
                break;
                case 3:
                account.checkBalance();
                break;
                case 4:
                System.out.println("Thank you, " + name + " (Account: " +
accountNumber + "), for using the Simple Banking Application. Goodbye!");
                break;
                default:
                System.out.println("Invalid choice. Please choose a valid
option.");
        }
        } while (choice != 4);


        scanner.close();
        }
}
```

OUTPUT:

```
Welcome to the Simple Banking Application!
Enter your name: Joe
Enter your account number: 12345
Enter your initial balance: ?100000

--- Menu ---
1. Deposit
2. Withdraw
3. Check Balance
4. Exit
Enter your choice: 1
Enter deposit amount: ?50000
Hi Joe (Account: 12345), you have successfully deposited ?50000.0

--- Menu ---
1. Deposit
2. Withdraw
3. Check Balance
4. Exit
Enter your choice: 2
Enter withdrawal amount: ?25000
Hi Joe (Account: 12345), you have successfully withdrawn ?25000.0

--- Menu ---
1. Deposit
2. Withdraw
3. Check Balance
4. Exit
Enter your choice: 3
Hi Joe (Account: 12345), your current balance is ?125000.0
```

```
--- Menu ---
1. Deposit
2. Withdraw
3. Check Balance
4. Exit
Enter your choice: 4
Thank you, Joe (Account: 12345), for using the Simple Banking Application. Goodbye!
```

CONCLUSION:

The **Simple Banking Application** successfully simulates basic banking operations like deposit, withdrawal, and balance inquiry. It demonstrates the practical implementation of Java programming concepts such as:

- **Object-Oriented Programming**: Encapsulation of account details in a class.
- **Control Structures**: Use of loops and conditionals to handle user interactions.
- **Input Handling**: Use of the `Scanner` class for interactive inputs.

By implementing this project, beginners gain hands-on experience in Java programming, reinforcing their understanding of key concepts like classes, methods, and user interaction. The program can be further extended with features such as transaction history or multi-user support.