Courier Management System

Coding Task 1

Control Flow Statements

---

1. Write a program that checks whether a given order is delivered or not based on its status (e.g**., "Processing," "Delivered," "Cancelled").** Use if-else statements for this.

2. Implement a **switch-case statement** to categorize parcels based on their weight into **"Light," "Medium," or "Heavy."**

3. Implement User Authentication **1.** Create a login system for employees and customers using Java **control flow statements.**

4. Implement Courier Assignment Logic **1.** Develop a mechanism to assign couriers to shipments based on predefined criteria **(e.g., proximity, load capacity)** using loops.

Now that the SQL part is completed, we are stepping into the coding part.

The first task focuses on implementing basic control flow statements in Java.

Task 1 demonstrates the use of control flow statements in Java by implementing different functionalities such as checking order status, categorizing parcels based on weight, user authentication, and assigning couriers to shipments. It utilizes if-else conditions, switch-case statements, and loops to handle decision-making processes effectively.

These include conditional statements (if-else and switch-case) and loops to handle courier-related operations.

Below are the tasks covered:

Task 1.1: Check Order Status

Task 1.2: Categorize Parcels by Weight

Task 1.3: Implement User and Employee Authentication

Task 1.4: Courier Assignment Logic

Main Method (psvm)

The main method presents a menu-driven interface for the user to choose from four different options. It continuously runs using a while (true) loop until the user selects the exit option. Based on the user's choice, the corresponding method is invoked using a switch statement.

The switch statement is used in the main method to determine which operation the user wants to execute. It evaluates the user's choice and calls the appropriate method for handling the respective task. If the input does not match any case, it prints an error message.

```java
package main;

import java.util.Scanner;

public class Task1 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        while (true) {
            System.out.println("\nTask 1 - Choose an option:");
            System.out.println("1. Check Order Status");
            System.out.println("2. Categorize Parcel by Weight");
            System.out.println("3. User Authentication");
            System.out.println("4. Employee Authentication");
            System.out.println("5. Assign Courier to Shipment");
            System.out.println("6. Exit");
            System.out.print("Enter your choice: ");

            int choice = scanner.nextInt();
            scanner.nextLine();

            switch (choice) {
                case 1:
                    checkOrderStatus(scanner);
                    break;
                case 2:
                    categorizeParcelWeight(scanner);
                    break;
                case 3:
                    userAuthentication(scanner);
                    break;
                case 4:
                    employeeAuthentication(scanner);
                    break;
                case 5:
                    assignCourier(scanner);
                    break;
                case 6:
                    System.out.println("Exiting Task 1...");
                    scanner.close();
                    return;
                default:
                    System.out.println("Invalid choice! Please select a valid option.");
            }
        }
    }
```

Task 1.1: Check Order Status

Write a program that checks whether a given order is delivered or not based on its status (e.g., "Processing," "Delivered," "Cancelled"). Use if-else statements for this.

This function takes the order status as input and checks whether it is "Processing," "Delivered," or "Cancelled" using an if-else condition.

Code :

```java
// 1. Check Order Status
public static void checkOrderStatus(Scanner scanner) {
    System.out.print("Enter order status (Processing/Delivered/Cancelled): ");
    String status = scanner.nextLine().trim();

    if (status.equalsIgnoreCase( anotherString: "Processing")) {
        System.out.println("Your order is still being processed.");
    } else if (status.equalsIgnoreCase( anotherString: "Delivered")) {
        System.out.println("Your order has been delivered! 🎉");
    } else if (status.equalsIgnoreCase( anotherString: "Cancelled")) {
        System.out.println("Your order has been cancelled.");
    } else {
        System.out.println("Invalid status entered. Please enter a valid order status.");
    }
}
```

Output :

```
Task 1 - Choose an option:
1. Check Order Status
2. Categorize Parcel by Weight
3. User Authentication
4. Employee Authentication
5. Assign Courier to Shipment
6. Exit
Enter your choice: 1
Enter order status (Processing/Delivered/Cancelled): Processing
Your order is still being processed.
```

```
Enter your choice: 1
Enter order status (Processing/Delivered/Cancelled): Delivered
Your order has been delivered!
```

```
Enter your choice: 1
Enter order status (Processing/Delivered/Cancelled): Cancelled
Your order has been cancelled.
```

Task 1.2: Categorize Parcels by Weight

Implement a switch-case statement to categorize parcels based on their weight into "Light," "Medium," or "Heavy."

This function takes the parcel's weight as input and categorizes it into "Light," "Medium," or "Heavy" using a switch-case structure.

Code :

```
// 2. Categorize Parcel Weight
public static void categorizeParcelWeight(Scanner scanner) {
    System.out.print("Enter parcel weight in kg: ");
    double weight = scanner.nextDouble();

    String category;
    if (weight < 1) {
        category = "Light";
    } else if (weight <= 5) {
        category = "Medium";
    } else {
        category = "Heavy";
    }

    System.out.println("The parcel is categorized as: " + category);
}
```

Output :

```
Task 1 - Choose an option:
1. Check Order Status
2. Categorize Parcel by Weight
3. User Authentication
4. Employee Authentication
5. Assign Courier to Shipment
6. Exit
Enter your choice: 2
Enter parcel weight in kg: 5
The parcel is categorized as: Medium
```

```
Enter parcel weight in kg: 10
The parcel is categorized as: Heavy
```

Task 1.3: Implement User and Employee Authentication

Create a login system for employees and customers using Java control flow statements.

This function simulates a basic authentication system where users enter a username and password. It checks if the credentials match the predefined values.

Code:

UserAuthentication :

```java
// 3. User Authentication
public static void userAuthentication(Scanner scanner) {
    String correctUsername = "admin";
    String correctPassword = "password123";

    System.out.print("Enter username: ");
    String username = scanner.nextLine();
    System.out.print("Enter password: ");
    String password = scanner.nextLine();

    if (username.equals(correctUsername) && password.equals(correctPassword))
        System.out.println("Login successful!");
    } else {
        System.out.println("Invalid credentials!");
    }
}
```

EmployeeAuthentication

```java
// 4. Employee Authentication
public static void employeeAuthentication(Scanner scanner) {
    String[] employeeIDs = {"EMP001", "EMP002", "EMP003"};
    String[] employeePasswords = {"pass1", "pass2", "pass3"};

    System.out.print("Enter Employee ID: ");
    String empID = scanner.nextLine();
    System.out.print("Enter Password: ");
    String empPassword = scanner.nextLine();

    boolean authenticated = false;
    for (int i = 0; i < employeeIDs.length; i++) {
        if (employeeIDs[i].equals(empID) && employeePasswords[i].equals(empPassword)) {
            authenticated = true;
            break;
        }
    }

    if (authenticated) {
        System.out.println("Employee authentication successful!A");
    } else {
        System.out.println("Invalid Employee ID or Password.");
    }
}
```

Output :

```
Task 1 - Choose an option:
1. Check Order Status
2. Categorize Parcel by Weight
3. User Authentication
4. Employee Authentication
5. Assign Courier to Shipment
6. Exit
Enter your choice: 3
Enter username: admin
Enter password: password123
Login successful!
```

```
Enter your choice: 4
Enter Employee ID: EMP002
Enter Password: pass2
Employee authentication successful!
```

```
Enter your choice: 3
Enter username: asdsf
Enter password: dgsdg
Invalid credentials!
```

```
Enter your choice: 4
Enter Employee ID: EMP0011
Enter Password: pas12
Invalid Employee ID or Password.
```

Task 1.4: Courier Assignment Logic

Develop a mechanism to assign couriers to shipments based on predefined criteria (e.g., proximity, load capacity) using loops.

This function assigns a courier to a shipment based on predefined criteria such as proximity and load capacity using a loop.

Code :

```
public static void assignCourier(Scanner scanner) {
    String[] couriers = {"Courier A", "Courier B", "Courier C", "Courier D"};
    int[] distances = {5, 3, 8, 2};
    double[] capacities = {10.0, 15.0, 8.0, 12.0};
```

```
        System.out.print("Enter shipment weight (kg): ");
        double shipmentWeight = scanner.nextDouble();

        String assignedCourier = null;
        int minDistance = Integer.MAX_VALUE;

        for (int i = 0; i < couriers.length; i++) {
            if (capacities[i] >= shipmentWeight && distances[i] < minDistance) {
                assignedCourier = couriers[i];
                minDistance = distances[i];
            }
        }

        if (assignedCourier != null) {
            System.out.println("Assigned " + assignedCourier + " (Distance: " + minDistance + " km) ");
        } else {
            System.out.println("No available couriers for this shipment! ");
        }
    }
}
```

Output :

```
Task 1 - Choose an option:
1. Check Order Status
2. Categorize Parcel by Weight
3. User Authentication
4. Employee Authentication
5. Assign Courier to Shipment
6. Exit
Enter your choice: 5
Enter shipment weight (kg): 12
Assigned Courier D (Distance: 2 km)
Enter shipment weight (kg): 14
Assigned Courier B (Distance: 3 km)
```

Conclusion

In this task, I implemented various control flow statements through real-world applications such as order status checking, parcel categorization, user authentication, and courier assignment. By effectively utilizing if-else statements, switch-case structures, and loops, the program provides a simple yet efficient way to automate decision-making processes in a logistics-based scenario.
