Courier Management System

Coding Task 2

Loops and Iteration

---

5. Write a Java program that uses a for loop to display all the orders for a specific customer.

6. Implement a while loop to track the real-time location of a courier until it reaches its destination

---

Loops and iteration are fundamental programming constructs that allow repetitive execution of code blocks until a specified condition is met. In Java, loops such as for, while, and do-while help automate repetitive tasks efficiently. Using loops minimizes redundancy and enhances code readability. This document demonstrates the implementation of loops to manage orders and track courier locations dynamically.

Below are the tasks covered:

Task 2.1: Display All Orders for a Specific Customer

Task 2.2: Track Real-Time Location of a Courier

Task 2.1: Display All Orders for a Specific Customer

Write a Java program that uses a for loop to display all the orders for a specific customer.

This program allows users to track a courier's real-time location based on its CourierID. It retrieves tracking details from the courier_tracking table and continuously updates the user until they choose to stop.

Code :

```
package main;

import util.DBConnUtil;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Scanner;

public class OrderService {
    public void displayOrdersForCustomer() {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter Customer ID: ");
        int customerID = scanner.nextInt();

        String query = "SELECT courierID, receiverName, receiverAddress, " +
                "weight, status, " +
                "trackingNumber, deliveryDate FROM courier WHERE senderID = ?";

        try (Connection conn = DBConnUtil.getConnection();
             PreparedStatement pstmt = conn.prepareStatement(query)) {

            pstmt.setInt( parameterIndex: 1, customerID);
            ResultSet rs = pstmt.executeQuery();
```

```
boolean hasOrders = false;
System.out.println("\nOrders for Customer ID: " + customerID);
System.out.println("-------------------------------------------------");
while (rs.next()) {
    hasOrders = true;
    System.out.println("Courier ID      : " + rs.getInt( columnLabel: "courierID"));
    System.out.println("Receiver Name   : " + rs.getString( columnLabel: "receiverName"));
    System.out.println("Receiver Address: " + rs.getString( columnLabel: "receiverAddress"));
    System.out.println("Weight          : " + rs.getDouble( columnLabel: "weight") + " kg");
    System.out.println("Status          : " + rs.getString( columnLabel: "status"));
    System.out.println("Tracking Number : " + rs.getString( columnLabel: "trackingNumber"));
    System.out.println("Delivery Date   : " + rs.getString( columnLabel: "deliveryDate"));
    System.out.println("-------------------------------------------------");
}

if (!hasOrders) {
    System.out.println("No orders found for this customer.");
}
```

```
    } catch (SQLException e) {
        e.printStackTrace();
    } finally {
        scanner.close();
    }
}

public static void main(String[] args) {
    OrderService orderService = new OrderService();
    orderService.displayOrdersForCustomer();
}
```

Output :

```
Enter Customer ID: 1

Orders for Customer ID: 1
-------------------------------------------------
Courier ID      : 1
Receiver Name   : Alice Smith
Receiver Address: 456 Elm St
Weight          : 2.5 kg
Status          : In Transit
Tracking Number : TRK10001
Delivery Date   : 2024-03-20
-------------------------------------------------
Courier ID      : 53
Receiver Name   : Eva Green
Receiver Address: 654 Ave, FL
Weight          : 2.8 kg
Status          : Delivered
Tracking Number : TRK5003
Delivery Date   : 2025-03-18
-------------------------------------------------
Courier ID      : 54
Receiver Name   : Eva Green
Receiver Address: 654 Ave, FL
Weight          : 3.5 kg
Status          : Delivered
Tracking Number : TRK5004
Delivery Date   : 2025-03-18
-------------------------------------------------
```

```
Enter Customer ID: 4

Orders for Customer ID: 4
-------------------------------------------------
Courier ID      : 4
Receiver Name   : Daniel Lee
Receiver Address: 654 Cedar St
Weight          : 2.8 kg
Status          : In Transit
Tracking Number : TRK10004
Delivery Date   : 2024-03-21
-------------------------------------------------
```

```
Enter Customer ID: 101

Orders for Customer ID: 101
--------------------------------------------------
```

Work Flow :

1. The user enters a Customer ID.
2. The program queries the database for orders linked to the customer.
3. Using a for loop, the retrieved orders are displayed.

Task 2.2: Track Real-Time Location of a Courier

Implement a while loop to track the real-time location of a courier until it reaches its destination

This program allows users to track a courier's real-time location based on its CourierID. It retrieves tracking details from the courier_tracking table and continuously updates the user until they choose to stop.

Code :

```
package main;

import util.DBConnUtil;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Scanner;

public class CourierTrackingService {
    public static void trackCourier() {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter Courier ID: ");
        int courierID = scanner.nextInt();
        scanner.nextLine(); // Consume newline

        boolean tracking = true;

        while (tracking) {
            try (Connection conn = DBConnUtil.getConnection()) {
                String query = "SELECT trackingID, status, currentLocation," +
                        " lastLocation, nextLocation " +
                        "FROM courier_tracking WHERE courierID = ?";
```

```
try (PreparedStatement pstmt = conn.prepareStatement(query)) {
    pstmt.setInt( parameterIndex: 1, courierID);
    try (ResultSet rs = pstmt.executeQuery()) {
        if (rs.next()) {
            String trackingID = rs.getString( columnLabel: "trackingID");
            String status = rs.getString( columnLabel: "status");
            String currentLocation = rs.getString( columnLabel: "currentLocation");
            String lastLocation = rs.getString( columnLabel: "lastLocation");
            String nextLocation = rs.getString( columnLabel: "nextLocation");

            // Display the tracking details
            System.out.println("\nTracking ID: " + trackingID);
            System.out.println("Status: " + status);
            System.out.println("Current Location: " + currentLocation);
            System.out.println("Last Location: " + lastLocation);
            System.out.println("Next Location: " + nextLocation);

            // Stop tracking if the courier is delivered
            if ("Delivered".equalsIgnoreCase(status)) {
                System.out.println("\nCourier has been delivered.");
                break;
            }
        } else {
            System.out.println("No tracking information found for this Courier ID.");
            break;
        }
    }
```

```
            // Ask the user if they want to refresh tracking
            System.out.print("\nDo you want to refresh tracking? (yes/no): ");
            String input = scanner.nextLine();
            if (!input.equalsIgnoreCase( anotherString: "yes")) {
                tracking = false;
            }
        }
        scanner.close();
        System.out.println("\nTracking stopped.");
    }

    public static void main(String[] args) {
        trackCourier();
    }
}
```

Output :

```
Enter Courier ID: 1

Tracking ID: TRK10001
Status: In Transit
Current Location: Chicago, IL
Last Location: St. Louis, MO
Next Location: Indianapolis, IN

Do you want to refresh tracking? (yes/no): yes

Tracking ID: TRK10001
Status: In Transit
Current Location: Chicago, IL
Last Location: St. Louis, MO
Next Location: Indianapolis, IN

Do you want to refresh tracking? (yes/no): no

Tracking stopped.
```

```
Enter Courier ID: 16

Tracking ID: TRK10016
Status: Delivered
Current Location: Delivered at Destination
Last Location: Transit Hub - New York, NY
Next Location: null

Courier has been delivered.

Tracking stopped.
```

Work Flow :

1. The user enters a Courier ID.
2. The program fetches tracking details from the courier_tracking table.
3. A while loop keeps asking the user if they want to refresh the tracking details.
4. If the user types refresh, the updated location details are fetched. If the user types exit, the tracking process stops.

Conclusion

I have implemented these tasks using loops and iteration. The for loop was used to display all orders for a customer, and the while loop was implemented to track courier location in real time until it reaches the destination.
