**EX NO: 09 JSTL**

**DATE:**

Assume you have a list of Order objects, each with the following properties: orderId, customerName, and orderDate. Using JSTL core and formatting tags ,create a table that displays the order information sort based on order date in the following format:

|  |  |  |
| --- | --- | --- |
| orderId | customerName | orderdate |
| 001 | John Doe | 2022-03-10 |
| 002 | Jane smith | 2022-03-12 |
| 003 | Bob johnson | 2022-03-15 |

**AIM:**

To write a JSTL program for the given scenario.

**ALGORITHM:**

Step 1: Start

Step 2: Define a class named Order with private fields orderId, customerName, and orderDate. Also, provide a constructor to initialize these fields and getter methods to access them.

Step 3: Create a Main class with the main method.

Step 4: Instantiate an ArrayList named orders to hold Order objects.

Step 5: Create three Order objects with sample data and add them to the orders list.

Step 6: Use Collections.sort() method to sort the orders list based on the orderDate field of Order objects. This is done using a Comparator that compares Order objects based on their order date.

Step 7: terate over the sorted orders list and print the order information (OrderID, CustomerName, and OrderDate) in a tabular format using System.out.printf().

Step 8: Stop.

**PROGRAM:**

import java.util.ArrayList;

import java.util.Collections;

import java.util.Comparator;

import java.util.List;

class Order {

private String orderId;

private String customerName;

private String orderDate;

public Order(String orderId, String customerName, String orderDate) {

this.orderId = orderId;

this.customerName = customerName;

this.orderDate = orderDate;

}

public String getOrderId() {

return orderId;

}

public String getCustomerName() {

return customerName;

}

public String getOrderDate() {

return orderDate;

}

}

public class Main {

public static void main(String[] args) {

List<Order> orders = new ArrayList<>();

orders.add(new Order("001", "John Doe", "2022-03-10"));

orders.add(new Order("002", "Jane Smith", "2022-03-12"));

orders.add(new Order("003", "Bob Johnson", "2022-03-15"));

Collections.sort(orders, Comparator.comparing(Order::getOrderDate));

System.out.println("OrderID\tCustomerName\tOrderDate");

<c: forEach (Order order : orders) {

System.out.printf("%s\t%s\t%s\n", order.getOrderId(), order.getCustomerName(), order.getOrderDate());

}

</c:forEach>

}

}

**OUTPUT:**

OrderID CustomerName OrderDate

001 John Doe 2022-03-10

002 Jane Smith 2022-03-12

003 Bob Johnson 2022-03-15

|  |  |
| --- | --- |
| Rubrics | Marks |
| Code & Output  (20) |  |
| Quiz (5) |  |
| Timely Submission (5) |  |
| Total (30) |  |
| Initial |  |

**RESULT:**

Thus, the JSTL program for the given scenario has been executed and the output is verified successfully.