Data Structures and Algorithms

SuperSet ID:6412063

Exercise 3: Sorting Customer Orders

Code:

class Order {

    int orderId;

    String customerName;

    double totalPrice;

    public Order(int orderId, String customerName, double totalPrice) {

        this.orderId = orderId;

        this.customerName = customerName;

        this.totalPrice = totalPrice;

    }

    public String toString() {

        return "Order ID: " + orderId + ", Customer: " + customerName + ", Total: ₹" + totalPrice;

    }

}

public class OrderSorting {

    public static void bubbleSort(Order[] orders) {

        int n = orders.length;

        for (int i = 0; i < n - 1; i++) {

            for (int j = 0; j < n - i - 1; j++) {

                if (orders[j].totalPrice > orders[j + 1].totalPrice) {

                    Order temp = orders[j];

                    orders[j] = orders[j + 1];

                    orders[j + 1] = temp;

                }

            }

        }

    }

    public static void quickSort(Order[] orders, int low, int high) {

        if (low < high) {

            int pivotIndex = partition(orders, low, high);

            quickSort(orders, low, pivotIndex - 1);

            quickSort(orders, pivotIndex + 1, high);

        }

    }

    private static int partition(Order[] orders, int low, int high) {

        double pivot = orders[high].totalPrice;

        int i = low - 1;

        for (int j = low; j < high; j++) {

            if (orders[j].totalPrice <= pivot) {

                i++;

                Order temp = orders[i];

                orders[i] = orders[j];

                orders[j] = temp;

            }

        }

        Order temp = orders[i + 1];

        orders[i + 1] = orders[high];

        orders[high] = temp;

        return i + 1;

    }

    public static void displayOrders(Order[] orders) {

        for (Order order : orders) {

            System.out.println(order);

        }

    }

    public static void main(String[] args) {

        Order[] orders = {

            new Order(201, "Dharshini", 2500.00),

            new Order(202, "Karthik", 5200.50),

            new Order(203, "Ravi", 1500.75),

            new Order(204, "Priya", 4200.20),

            new Order(205, "Anita", 3000.00)

        };

        System.out.println("Original Orders:");

        displayOrders(orders);

        System.out.println("\nSorted by Bubble Sort:");

        bubbleSort(orders);

        displayOrders(orders);

        orders = new Order[] {

            new Order(201, "Dharshini", 2500.00),

            new Order(202, "Karthik", 5200.50),

            new Order(203, "Ravi", 1500.75),

            new Order(204, "Priya", 4200.20),

            new Order(205, "Anita", 3000.00)

        };

        System.out.println("\nSorted by Quick Sort:");

        quickSort(orders, 0, orders.length - 1);

        displayOrders(orders);

    }

}

Output:

A screenshot of a computer

Description automatically generated