**Exercise 3: Sorting Customer Orders**

**Code:**

 class CustomerOrderSorter {

    static 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 "OrderID: " + orderId + ", Customer: " + customerName + ", Total Price: " + totalPrice;

        }

    }

    // Bubble Sort

    public static void bubbleSort(Order[] orders) {

        int n = orders.length;

        boolean swapped;

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

            swapped = false;

            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;

                    swapped = true;

                }

            }

            if (!swapped) break; // Stop if already sorted

        }

    }

    // Quick Sort

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

        if (low < high) {

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

            quickSort(orders, low, pi - 1);

            quickSort(orders, pi + 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 printOrders(String message, Order[] orders) {

        System.out.println(message);

        for (Order o : orders) {

            System.out.println(o);

        }

    }

    public static void main(String[] args) {

        Order[] orders = {

            new Order(101, "Alice", 250.50),

            new Order(102, "Bob", 150.75),

            new Order(103, "Charlie", 300.20),

            new Order(104, "Diana", 180.40)

        };

        printOrders("Original Orders:", orders);

        // Bubble Sort

        Order[] bubbleSortedOrders = orders.clone();

        bubbleSort(bubbleSortedOrders);

        printOrders("\nOrders after Bubble Sort:", bubbleSortedOrders);

        // Quick Sort

        Order[] quickSortedOrders = orders.clone();

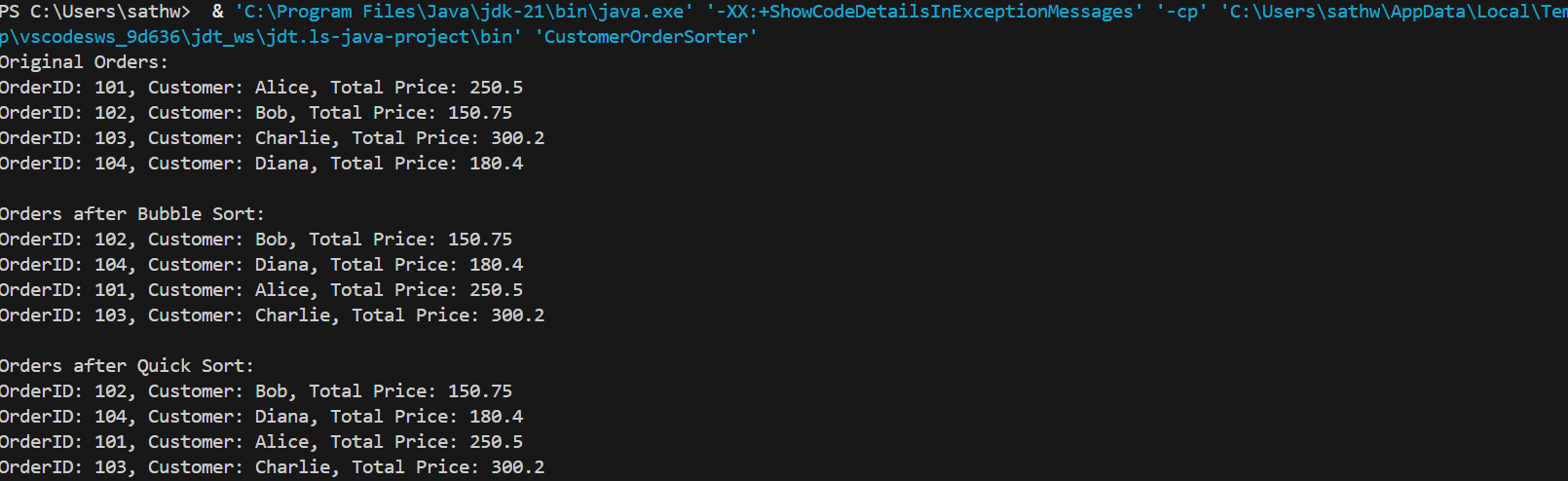
        quickSort(quickSortedOrders, 0, quickSortedOrders.length - 1);

        printOrders("\nOrders after Quick Sort:", quickSortedOrders);

    }

}

**Output:**

****