**MODULE-2**

**DATA STRUCTURE AND ALGORITHM**

SUPERSET ID:6407550

**Exercise 3: Sorting Customer Orders**

**Order.java**

package customer;

public 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;

}

@Override

public String toString() {

return orderId + " | " + customerName + " | ₹" + totalPrice;

}

}

**SortOperations.java**

package customer;

public class SortOperations {

// Bubble Sort

public static void bubbleSort(Order[] orders) {

int n = orders.length;

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

boolean 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; // Optimization

}

}

// Quick Sort

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;

}

}

**Main.java**

package customer;

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Order[] orders = {

new Order(101, "Alice", 4500.0),

new Order(102, "Bob", 1200.0),

new Order(103, "Charlie", 7500.0),

new Order(104, "Diana", 3000.0),

new Order(105, "Eve", 9000.0)

};

Scanner sc = new Scanner(System.in);

System.out.println("Choose Sorting Method:\n1. Bubble Sort\n2. Quick Sort");

int choice = sc.nextInt();

if (choice == 1) {

SortOperations.bubbleSort(orders);

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

} else {

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

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

}

for (Order o : orders) {

System.out.println(o);

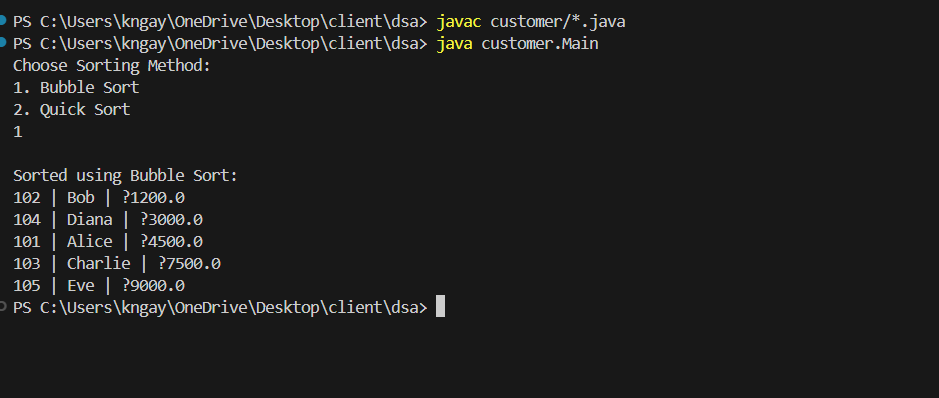
}

sc.close();

}

}

**OUTPUT:**

****