**Exercise 3: Sorting Customer Orders**

**Program:**

package Algorithms\_DataStructures.sortingCustomers;

import java.util.Arrays;

class Order{

    public int orderId;

    public String customerName;

    public int totalPrice;

    Order(int id,String name,int price){

        this.orderId = id;

        this.customerName = name.toLowerCase();

        this.totalPrice = price;

    }

    public String toString(){

         return orderId + " - " + customerName + " - " + totalPrice ;

    }

}

class Sort{

    public void bubbleSort(Order []a){

        int i,j;

        Order []orders=Arrays.copyOf(a, a.length);

        for(i=0;i<orders.length-1;i++){

            for(j=0;j<orders.length-i-1;j++){

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

                    Order temp = orders[j];

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

                    orders[j+1] =temp;

                }

            }

        }

        for(i=0;i<orders.length;i++){

            System.out.println(orders[i]);

        }

    }

    public void quickSort(Order[] a) {

    Order[] orders = Arrays.copyOf(a, a.length);

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

    for (Order order : orders) {

        System.out.println(order);

    }

}

private void quickSortHelper(Order[] orders, int low, int high) {

    if (low < high) {

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

        quickSortHelper(orders, low, pi - 1);

        quickSortHelper(orders, pi + 1, high);

    }

}

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

    int 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 class Main {

    public static void main(String[] args) {

        Order[] orders = {

            new Order(105, "Alice", 250),

            new Order(110, "Bob", 400),

            new Order(102, "Charlie", 150),

            new Order(109, "Diana", 500),

            new Order(101, "Eve", 300),

            new Order(108, "Frank", 300),

            new Order(103, "Grace", 100),

            new Order(107, "Heidi", 450),

            new Order(104, "Ivan", 200),

            new Order(106, "Judy", 275)

        };

        System.out.println("BubbleSort..........");

        Sort s = new Sort();

        s.bubbleSort(orders);

        System.out.println("quickSort..........");

        s.quickSort(orders);

    }

}

A computer screen shot of a black screen

AI-generated content may be incorrect.