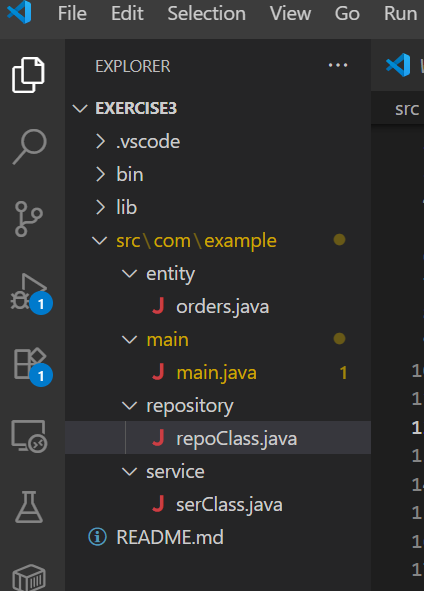
**Sorting Customer Orders**

| **Algorithm** | **Best Case** | **Average Case** | **Worst Case** |
| --- | --- | --- | --- |
| Bubble Sort | O(n) | O(n²) | O(n²) |
| Quick Sort | O(n log n) | O(n log n) | O(n²) |

**CODE:**



orders.java:

package com.example.entity;

public class orders {

private int orderId;

private String customerName;

private double totPrice;

public orders(int orderId, String customerName, double totPrice) {

this.orderId = orderId;

this.customerName = customerName;

this.totPrice = totPrice;

}

public int getOrderId() {

return orderId;

}

public void setOrderId(int orderId) {

this.orderId = orderId;

}

public String getCustomerName() {

return customerName;

}

public void setCustomerName(String customerName) {

this.customerName = customerName;

}

public double getTotPrice() {

return totPrice;

}

public void setTotPrice(double totPrice) {

this.totPrice = totPrice;

}

}

Main.java

package com.example.main;

import com.example.entity.orders;

import com.example.service.serClass;

public class main {

private static serClass ser = new serClass();

public static void main(String args[]) {

orders[] order = {

new orders(101, "Sasi", 504.35),

new orders(102, "Preethi", 635.21),

new orders(103, "Arun", 7852.26),

new orders(104, "Geetha", 526.99),

new orders(105, "Lio", 8963.206)

};

System.out.println();

System.out.println("Bubble Sort(Total Price) : ");

System.out.println();

System.out.println("Before bubble sort : ");

for (orders it : order) {

System.out.println(it.getOrderId() + "\t" + it.getCustomerName() + "\t" + it.getTotPrice());

}

System.out.println();

System.out.println("After bubble sort : ");

ser.bubbleSort(order);

for (orders it : order) {

System.out.println(it.getOrderId() + "\t" + it.getCustomerName() + "\t" + it.getTotPrice());

}

System.out.println();

orders[] order1 = {

new orders(106, "Sahu", 784.35),

new orders(107, "Riya", 665.21),

new orders(108, "Anu", 852.26),

new orders(109, "Kiya", 526.99),

new orders(110, "Ben", 963.206)

};

System.out.println("Quick Sort(Total Price) : ");

System.out.println();

System.out.println("Before quick sort : ");

for (orders it : order1) {

System.out.println(it.getOrderId() + "\t" + it.getCustomerName() + "\t" + it.getTotPrice());

}

System.out.println();

System.out.println("After quick sort : ");

ser.quickSort(order1);

for (orders it : order1) {

System.out.println(it.getOrderId() + "\t" + it.getCustomerName() + "\t" + it.getTotPrice());

}

}

}

repoClass.java:

package com.example.repository;

import com.example.entity.orders;

public class repoClass {

public void bubbleSort(orders order[]) {

int n = order.length;

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

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

if (order[j].getTotPrice() > order[j + 1].getTotPrice()) {

orders temp = order[j];

order[j] = order[j + 1];

order[j + 1] = temp;

}

}

}

}

public int quick(orders order[], int low, int high) {

double val = order[low].getTotPrice();

int i = low;

int j = high;

while (i < j) {

while (order[i].getTotPrice() <= val && i <= high - 1) {

i++;

}

while (order[j].getTotPrice() > val && j >= low + 1) {

j--;

}

if (i < j) {

orders temp = order[i];

order[i] = order[j];

order[j] = temp;

}

}

orders temp = order[low];

order[low] = order[j];

order[j] = temp;

return j;

}

public void qS(orders order[], int low, int high) {

if (low < high) {

int pivot = quick(order, low, high);

qS(order, low, pivot - 1);

qS(order, pivot + 1, high);

}

}

public void quickSort(orders order[]) {

qS(order, 0, order.length - 1);

}

}

serClass.java :

package com.example.service;

import com.example.entity.orders;

import com.example.repository.repoClass;

public class serClass {

private static repoClass repo = new repoClass();

public void bubbleSort(orders order[]) {

repo.bubbleSort(order);

}

public void quickSort(orders order[])

{

repo.quickSort(order);

}

}

## 

## 

## **Output :**

