**ATIVIDADE 4**

**ATENÇÃO:**

* **Esta Atividade deverá ser feita em GRUPO DE PELO MENOS 04 ALUNOS E DE NO MÁXIMO 08 ALUNOS embora a entrega deverá ser feita INDIVIDUALMENTE no Classroom.**
* **Atividades feitas individualmente ou entregues com atraso NÃO SERÃO CONSIDERADAS.**

**Grupo**

Rafael Rossetto Guitarrari RA : 823158602

Andrey de Freitas Souza RA : 823217536

Gabriel Farah De lima RA: 822231424

Fabrício de Barros Narbon RA:822227166

Bianca Alves Ribeiro RA: 8222240261

Luiz Gustavo França de Abreu RA: 823210075

Gabrielle Garcia Paz  RA: 823126085

Webster Diógenes Rodrigues RA:8222242764

1) Execute o código da Aula 4 e anote modifique o programa para uma fila de 5 pacientes. Escreva aqui o código e o print de saída da execução.

**RESPOSTAS**

import java.util.Comparator;

import java.util.PriorityQueue;

import java.util.Queue;

public class App {

    static class CustomerOrderComparator implements Comparator<CustomerOrder> {

        @Override

        public int compare(CustomerOrder o1, CustomerOrder o2)

        {

            return o1.getOrderAmount() < o2.getOrderAmount() ? 1 : -1;

        }

    }

    static  class CustomerOrder implements Comparable<CustomerOrder> {

        private int orderId;

        private double orderAmount;

        private String customerName;

        public CustomerOrder(int orderId, double orderAmount, String customerName) {

            this.orderId = orderId;

            this.orderAmount = orderAmount;

            this.customerName = customerName;

        }

        @Override

        public int compareTo(CustomerOrder o) {

            return o.orderId > this.orderId ? 1 : -1;

        }

        @Override

        public String toString() {

            return "orderId:" + this.orderId + ", orderAmount:" + this.orderAmount + ", customerName:" + customerName;

        }

        public double getOrderAmount() {

            return orderAmount;

        }

    }

    public static void main(String[] args) throws Exception {

        CustomerOrder c1 = **new** CustomerOrder(1, 100.0, "customer1");

        CustomerOrder c2 = **new** CustomerOrder(3, 50.0, "customer3");

        CustomerOrder c3 = **new** CustomerOrder(2, 300.0, "customer2");

        CustomerOrder c4 = **new** CustomerOrder(2, 150.0, "customer2");

        CustomerOrder c5 = **new** CustomerOrder(2, 3.0, "customer2");

        Queue<CustomerOrder> customerOrders = **new** PriorityQueue<>(**new** CustomerOrderComparator());

        customerOrders.add(c1);

        customerOrders.add(c2);

        customerOrders.add(c3);

        customerOrders.add(c4);

        customerOrders.add(c5);

        while (!customerOrders.isEmpty()) {

            System.out.println(customerOrders.poll());

        }

    }

}

Output:

Foto em preto e branco

Descrição gerada automaticamente