**TECHNOLOGICAL INSTITUTE OF THE PHILIPPINES**

**QUEZON CITY**

**COLLEGE OF INFORMATION TECHNOLOGY EDUCATION (CITE)**

**CS 201 - Data Structures and Algorithms**

|  |  |
| --- | --- |
| **Name: Aristotle Buenaventura** | **Date: September 20, 2021** |
| **Program/Section: IT21S1** | **Instructor: Ms. Rosmina Joy M. Cabauatan** |
| **Assessment Task: Exercise 4 - Queue** | |

**Code**

**RunnerQ** **Class**

package exercise4;

import javax.swing.JOptionPane;

public class RunnerQ {

public static void main(String[] args) {

// boolean values used for the condition of while loop

boolean isExit = true;

boolean isExitTransaction = true;

while(isExit) {

// size of the queue

byte size = (byte) Integer.parseInt(JOptionPane.showInputDialog("Enter the number of customers: "));

//constructor

POS queue = new POS(size);

// inserting the names of the customer inside the queue

for(int i=0;i<size;i++) {

String customer = JOptionPane.showInputDialog("Enter the name of the customer: ");

queue.add(customer);

}

// transaction processes

for(int i=1;i<size+1;i++) {

System.out.print("Customers waiting in line are: ");

queue.print();

System.out.println();

System.out.println("Now serving customer no. " + i);

System.out.println("Welcome to Coffee Shop:");

System.out.print("What is your order? " );

queue.front();

String orderName=JOptionPane.showInputDialog("What is your order? ");

System.out.println();

System.out.println(orderName);

System.out.println("How many?");

int orderNum=Integer.parseInt(JOptionPane.showInputDialog("How many? "));

System.out.println(orderNum);

System.out.print("You have ordered " + orderNum + " " + orderName + ", ");

queue.front();

System.out.println();

System.out.println("Please proceed to the waiting area for your order.");

if(i<size) {

System.out.println();

System.out.println("(Next customer)");

}

System.out.println();

queue.delete();

}

// new transaction

System.out.println("No more waiting customers, Press [T] to continue the transaction or [Q] to quit");

while (isExitTransaction) {

String transaction = JOptionPane.showInputDialog("No more waiting customers, Press [T] to continue the transaction or [Q] to quit").toUpperCase();

if(transaction.equals("T")) {

System.out.println("You have selected [T], the system will run again.");

System.out.println();

isExit=true;

isExitTransaction = false;

} else if(transaction.equals("Q")) {

System.out.println();

System.out.println("You have selected [Q], the system will now be closed.");

isExit=false;

isExitTransaction = false;

} else {

JOptionPane.showMessageDialog(null, "Wrong Input, please input the correct response.","Warning Message", JOptionPane.WARNING\_MESSAGE);

System.out.println("Wrong Input");

isExitTransaction = true;

}

} isExitTransaction = true;

}

}

} // End of the program

**POS Class**

package exercise4;

public class POS {

private static int front, rear, size;

private static String[] queue;

POS(int c)

{

front = rear = 0;

size = c;

queue = new String[size];

}

// function to insert an element

// at the rear of the queue

static void add(String data)

{

// check queue is full or not

if (size == rear) {

System.out.printf("\nQueue is full\n");

return;

}

// insert element at the rear

else {

queue[rear]= data;

rear++;

}

return;

}

// function to delete an element

// from the front of the queue

static void delete()

{

// if queue is empty

if (front == rear) {

System.out.printf("\nQueue is empty\n");

return;

}

// shift all the elements from index 2 till rear

// to the right by one

else {

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

queue[i] = queue[i + 1];

}

// store 0 at rear indicating there's no element

if (rear < size)

queue[rear] = null;

// decrement rear

rear--;

}

return;

}

// print queue elements

static void print()

{

int i;

if (front == rear) {

System.out.print("\nQueue is Empty\n");

return;

}

// traverse front to rear and print elements

for (i = front; i < rear; i++) {

System.out.print(queue[i]);

if(i<rear-1) {

System.out.print(" <- ");

}

}

return;

}

// print front of queue

static void front()

{

if (front == rear) {

System.out.printf("\nQueue is Empty\n");

return;

}

System.out.printf(queue[front]);

return;

}

}

**Output**

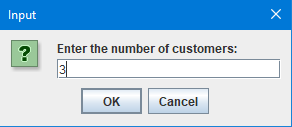


Figure 1. Enter the Number of Customers

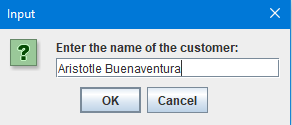


Figure 2. Enter the Name of Customer

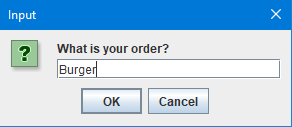


Figure 3. Enter the Customer’s Order

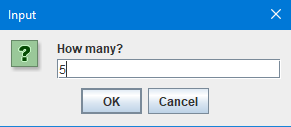


Figure 4. Enter the Quantity

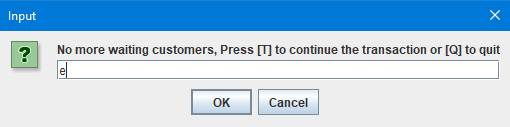


Figure 5. New Transaction

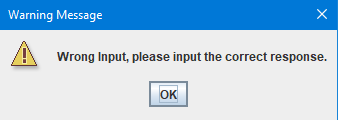


Figure 6. Wrong Input

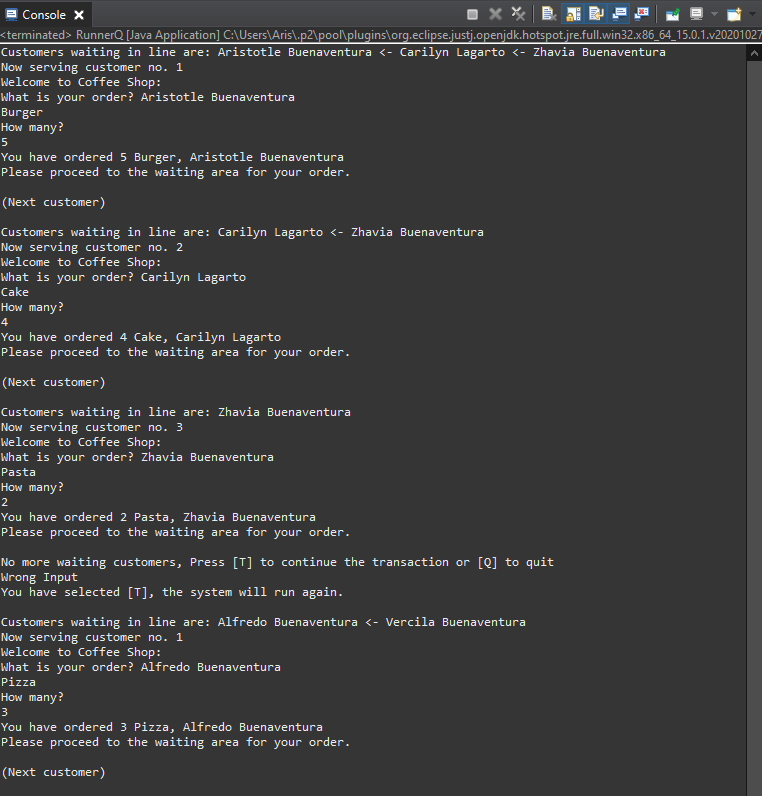
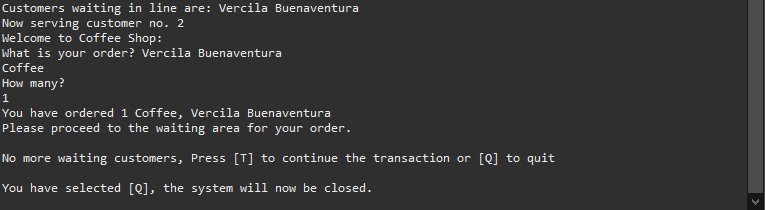


Figure 7. The Final Output