Simulation: Two Servers Code

Name \ Abdulrahman salah anwer

Customer Class

import java.util.Random;  
  
public class Customer {  
 int id ;  
 int IAT ;  
 int ST ;  
 int AT ;  
 int WT ;  
 int timeSpend ;  
 static int *totalWaitingTime* = 0;  
 static int *numOfCustomers* = 0;  
  
 String servingServer ;  
  
  
 public Customer() {  
 *numOfCustomers*++;  
 }  
  
 public int IATGenerator() {  
 Random random = new Random();  
 int randDegit = random.nextInt(100);  
 int returnedIAT = 0;  
 if (randDegit >= 0 && randDegit <= 24) {  
 returnedIAT = 1;  
 } else if (randDegit >= 25 && randDegit <= 64) {  
 returnedIAT = 2;  
 } else if (randDegit >= 65 && randDegit <= 84) {  
 returnedIAT = 3;  
 } else if (randDegit >= 85 && randDegit <= 99) {  
 returnedIAT = 4;  
 }  
 return returnedIAT;  
 }  
  
 public String toString() {  
 return this.id +"\t"+ this.AT +"\t"+ this.IAT +"\t"+ this.ST +"\t"+ this.WT +"\t"+ this.servingServer +"\t\t\t\t"+ this.timeSpend ;  
 }  
  
}

Baker Class

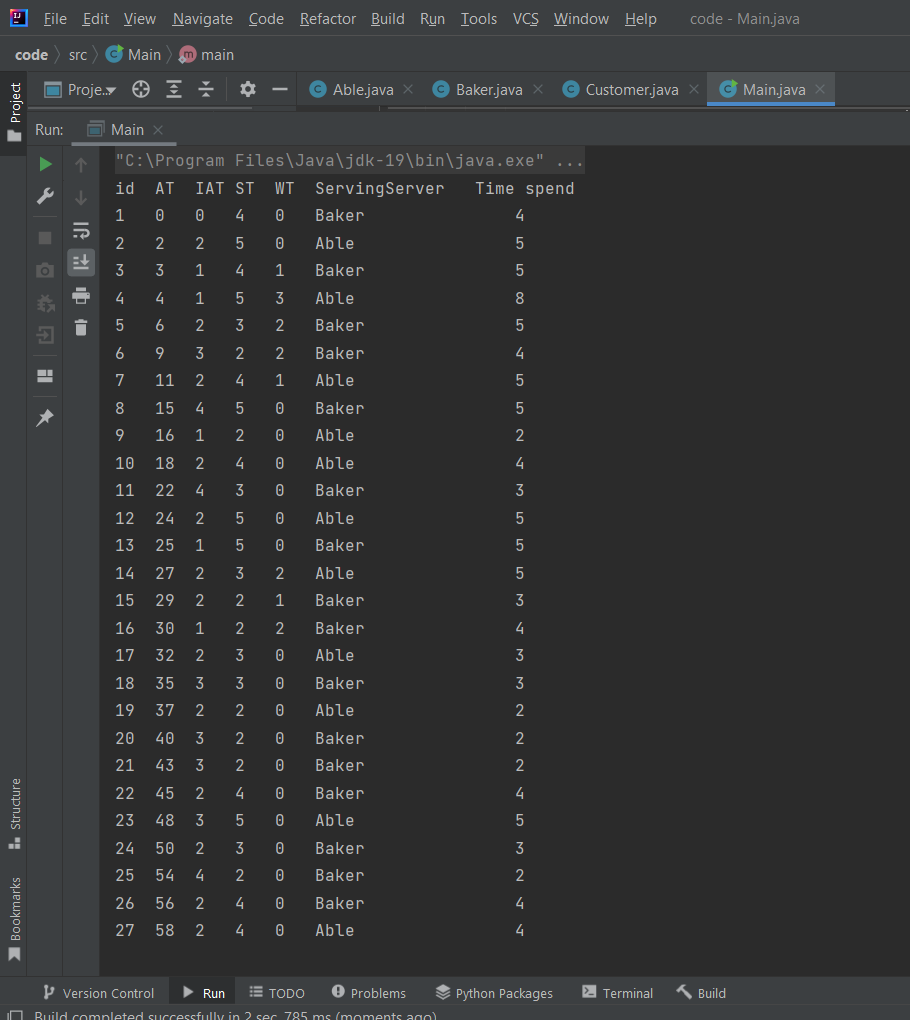
import java.util.Random;  
  
public class Baker {  
 int BakerST;  
 int state;  
 int currentST;  
 int totalServiceTime;  
 public int BakerSTGenerator() {  
 Random random = new Random();  
 int randDegit = random.nextInt(100);  
 if (randDegit >= 0 && randDegit <= 34) {  
 BakerST = 2;  
 } else if (randDegit >= 35 && randDegit <= 59) {  
 BakerST = 3;  
 } else if (randDegit >= 60 && randDegit <= 79) {  
 BakerST = 4;  
 } else if (randDegit >= 80 && randDegit <= 99) {  
 BakerST = 5;  
 }  
 return BakerST;  
 }  
  
}

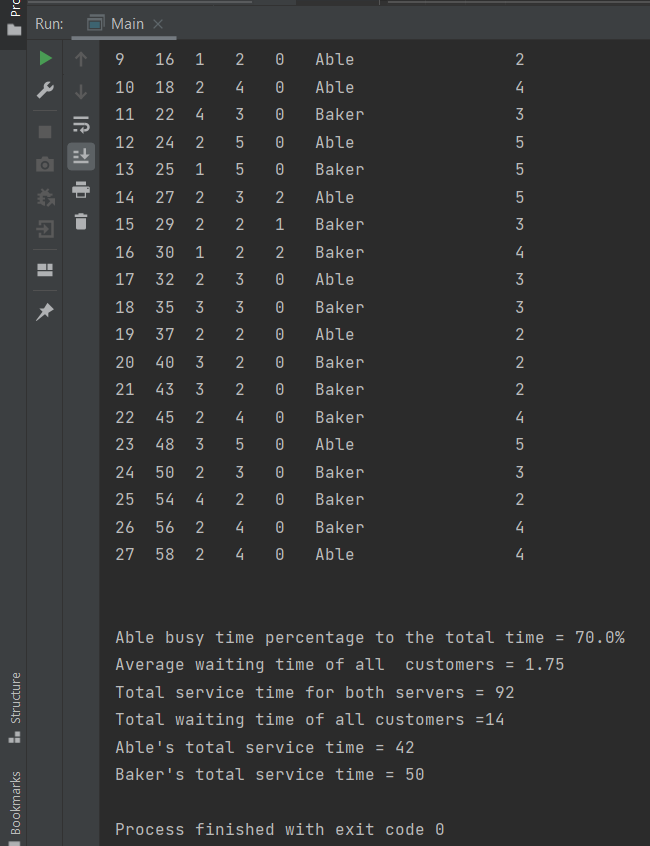
Able class

import java.util.Random;  
  
public class Able {  
 int AbleST;  
 int state = 0;  
 int currentST;  
 int totalServiceTime;  
  
 public int AbleSTGenerator() {  
 Random random = new Random();  
 int randDegit = random.nextInt(100);  
 if (randDegit >= 0 && randDegit <= 29) {  
 AbleST = 2;  
 } else if (randDegit >= 30 && randDegit <= 57) {  
 AbleST = 3;  
 } else if (randDegit >= 58 && randDegit <= 82) {  
 AbleST = 4;  
 } else if (randDegit >= 83 && randDegit <= 99) {  
 AbleST = 5;  
 }  
 return AbleST;  
 }  
}

Main Class

import java.util.ArrayList;  
import java.util.LinkedList;  
import java.util.Queue;  
  
public class Main {  
 public static void main(String[] args) {  
 Able Able = new Able();  
 Baker Baker = new Baker();  
 Queue<Customer> customersQ = new LinkedList<Customer>();  
 ArrayList<Customer> customersInfo = new ArrayList<Customer>();  
 int AtNext = 0;  
 int IDNext = 1;  
 int IATNext = 0;  
 int WTNext = 0 ;  
 int TotalServiceTimeForTwoServers = 0 ;  
 int TheTotalWT = 0 ;  
 int numOfCustomersWaited = 0;  
 for (int min = 0; min <= 59; min++) {  
 if (AtNext == min) {  
 Customer newCustomer = new Customer();  
 newCustomer.id = IDNext;  
 newCustomer.AT = AtNext;  
 newCustomer.IAT = IATNext;  
 customersQ.add(newCustomer);  
 IATNext = newCustomer.IATGenerator();  
 AtNext = (IATNext) + AtNext;  
 IDNext++;  
 }  
 if (Baker.state == 0) {  
 if (customersQ.peek() != null) {  
  
 Customer removedFromQ = new Customer();  
 Baker.state = 1;  
 removedFromQ = customersQ.poll();  
 removedFromQ.ST = Baker.BakerSTGenerator();  
 removedFromQ.WT = min - removedFromQ.AT ;  
 removedFromQ.timeSpend = removedFromQ.ST + removedFromQ.WT ;  
 removedFromQ.servingServer = "Baker";  
 Baker.currentST = removedFromQ.ST;  
 Baker.totalServiceTime = Baker.totalServiceTime + removedFromQ.ST;  
 customersInfo.add(removedFromQ);  
 }  
 } else {  
 Baker.currentST--;  
 if (Baker.currentST == 1) {  
 Baker.state = 0;  
 }  
 }  
 if (Able.state == 0) {  
 if (customersQ.peek() != null) {  
 Customer removedFromQ = new Customer();  
 Able.state = 1;  
 removedFromQ = customersQ.poll();  
 removedFromQ.ST = Able.AbleSTGenerator();  
 removedFromQ.WT = min - removedFromQ.AT ;  
 removedFromQ.timeSpend = removedFromQ.ST + removedFromQ.WT ;  
 removedFromQ.servingServer = "Able";  
 Able.currentST = removedFromQ.ST;  
 Able.totalServiceTime = Able.totalServiceTime + removedFromQ.ST;  
 customersInfo.add(removedFromQ);  
 }  
 } else {  
 Able.currentST--;  
 if (Able.currentST == 1) {  
 Able.state = 0;  
 }  
 }  
 }  
 TotalServiceTimeForTwoServers = Able.totalServiceTime + Baker.totalServiceTime ;  
 for (int i = 0 ; i < customersInfo.size(); i++){  
 TheTotalWT = TheTotalWT + customersInfo.get(i).WT ;  
 }  
 for (int i = 0 ; i < customersInfo.size(); i++){  
 if (customersInfo.get(i).WT != 0){  
 numOfCustomersWaited = numOfCustomersWaited + 1 ;  
 }  
 }  
 System.*out*.println("id" + "\t" + "AT" + "\t" + "IAT" + "\t" + "ST" + "\t" + "WT" + "\t" + "ServingServer" + "\t" + "Time spend");  
 for (int i = 0; i < customersInfo.size(); i++) {  
 System.*out*.println(customersInfo.get(i));  
 }  
 System.*out*.println("\n");  
 System.*out*.println("Able busy time percentage to the total time = " + ((double) Able.totalServiceTime / 60) \* 100 + "%");  
 System.*out*.println("Average waiting time of all customers = " + ((double) TheTotalWT) / numOfCustomersWaited);  
 System.*out*.println("Total service time for both servers = " + TotalServiceTimeForTwoServers );  
 System.*out*.println("Total waiting time of all customers ="+ TheTotalWT);  
 System.*out*.println("Able's total service time = " + Able.totalServiceTime);  
 System.*out*.println("Baker's total service time = " + Baker.totalServiceTime);  
 }  
  
}

The Output 



Done ..