// (2/5)Arrival Class

// Nick Pinero and Jake Frommer

// 4/23/15

// Creates a queue that will hold

// the data for several customers.

// this data can be read into the

// program from a file or could be

// manually done. It also removes

// and returns the customer form the

// top of the queue

//imports packages

import java.io.File;

import java.io.FileNotFoundException;

import java.util.Scanner;

//class name

public class Arrival{

//Instance Variables

//creates two nodes called

//front and back. A customer

//is a node

private Customer front;

private Customer back;

//creates an int variable

//to keep track of the number

//of items in the queue

private int numItems;

//Methods

//constructors

public Arrival(){

front = null;

back = null;

numItems = 0;

}

//is true if numItems is 0

//shows us if the queue is

//empty or not

public boolean isEmpty(){

return numItems == 0;

}

//creates and adds a customer

//to the queue

public void addCustomer(int arrivalTime, int serviceTime, String

customerName){

//creates a new customer

//called nick which is the

//item being added to the queue

Customer nick = new Customer(arrivalTime,serviceTime,customerName);

//if front is pointing

//to null

if(front == null){

//set front to nick

//or the new item

front = nick;

//sets back to front

back = front;

//increases num items

numItems ++;

//otherwise

} else {

//set the back pointer's next

//customer to the new customer

//and then set back to the next

//item which is the new customer.

back.setNextCustomer(nick);

back = back.getNextCustomer();

//increase numItems

numItems ++;

//front.display();

//front.getNextCustomer().display();

}

}

//method that will calculate the total service

//amongst all members and return that total

//time

public int totalService(){

//creates a current pointer

//to move through the queue

Customer cur = front;

//creates a current service int

//variable which will store the

//current service time of the customer

int curService = cur.getService();

//creates a total int variabel which

//will store the total service time

//amongst all customers.

int total = curService;

//while the cur pointer isn't equal

//to null...

while(cur != null){

//set cur service to the current

//customer's service time

curService = cur.getService();

//add the cur service time to the

//total service time

total = total + curService;

//advance the current pointer to the

//next customer

cur = cur.getNextCustomer();

}

//return the total service time after the

//while loop is finished.

return total;

}

//removes a customer from the list.

public Customer removeCustomer(){

//creates a new customer called

//item which is used as basically

//a place holder for the item

//being removed and what will be

//returned in the end.

Customer item = new Customer();

//if numItems is 1

if(numItems == 1){

//decrease numItems

numItems --;

//set item to front

item = front;

//set front and back to null

//because now the queue is empty

front = null;

back = null;

//otherwise

} else {

//decrease numItems

numItems --;

//set item to front so it can

//be put out to the user

item = front;

//set front to the next customer

front = front.getNextCustomer();

}

return item;

//item.display();

//front.display();

}

//gets the next customer in

//the list/queue and returns

//the next customer

public Customer nextCustomer(){

return front.getNextCustomer();

}

//returns the customer at the

//top of the arrival queue

public Customer peek(){

return front;

}

//displays the customers in the

//list/queue

public void displayArrivalList(){

//if the queue/list is

//not empty

if(!isEmpty()){

//creates a new variable to

//move through the list/queue

Customer cur = front;

//while cur is not null

while(cur != null){

//display the item at

//cur

cur.display();

//set cur to the next item in

//the list/queue so it can

//be printed out next.

cur = cur.getNextCustomer();

}

//otherwise

} else {

//prints out the list is empty

System.out.println("\tThe List is Empty");

}

}

//reads in a file to determine

//arrival times of customers

public void readArrival(String fileName){

//creates a new customer

//Customer cust = new Customer();

//creates a new scanner to

//read in the file.

Scanner fileInput;

//makes a variable to store

//the file being read in

File inFile = new File(fileName);

try{

//allows the file to be read by

//the scanner

fileInput = new Scanner(inFile);

//while the file has something

//next and isn't empty.

while(fileInput.hasNext()){

//adds a new customer to the list.

addCustomer(fileInput.nextInt(),fileInput.nextInt(),fileInput.next());

//advances the file to the next line.

//fileInput.nextLine();

}

//closes the input file.

fileInput.close();

}

catch (FileNotFoundException e) {

System.out.println(e);

System.exit(1); //IO error; exit program

}//exit catch

}

}