4–Parallel programming exercise

Submission date:17.0 6.20 20at 23:59

general description

The goal of this exercise is to experiment with "simultaneous" programming and synchronization betweenthreads, using different tools. This is the final exercise of the course, throughout the exercise note that you are adhering to all the principles of object-oriented software development that have been learned so farin the course. It is highly recommended to read all the instructions from beginning to end before starting work, to understand the system diagram well **(attached below – it is recommended to look at the diagram while understanding the story),** and to work according to the instructions foundat the end of the document.

The story of the deed

In light of the success of the system for selling tickets to cultural events (Work 3), it was decided in the pizza delivery chain "The **Modular Pizza"** to computerize the work process from the ordering stage to the delivery stage to the customers. The chaininitially created a system for one branch forthe purpose of examining the move.

Description of the process in general

Calls to order pizza arrive at the branch call queue. There are 3 dispatchers who receive the calls. Each conversation includes information about the desired order (how many pizzas, shipping address, etc.). The dispatcher who receives the call creates an order in the system that includes the price to be paid and the details received in the call. The dispatchers put the orders in the order of arrival. There are 2 slots whose job it is to pull the orders out of the queue. After completing the order, the tile converts the address to a driving distance from the branch in kilometers, and enters the order into the branch's information system. There are 3 kitchen workers who pull out the orders from the information system, when they will always pull out orders closer first (and among them those who arrived at the hotline first in order of arrival). Each kitchen worker who has pulled out an order prepares the delivery of the pizzas, and puts them in a ready-made delivery queue. There are 2 couriers at the branch who pull the shipments out of this queue. Each courier will wait until he has collected several shipments (depending on his capacity) and then sets off.

When a call arrives for an order for a quantity of pizzas greater than 10 (defined as an invitation to an event), the dispatcher transfers the call to a call queue before the shift manager, for handling and closing a special price. The manager takes the details of the order, closes a price with the customer, and enters the order himself into the information system.

The objects in the system

You must realize all the objects listed in the instructions, without exception. The **following sections list 13 objects (includingGUIand queues). All of them must be redeemed with the same name as the name that appears here in English. You can add additional objects (including inheritance and interfaces) as you see fit.**

Objects in the story

1. שיחה (Call)

Each call from a customer has the following characteristics:

1. How many pizzas the customer wants to order
2. Shipping address (string)
3. Customer credit number for payment

In addition, each call has a duration (in seconds) that will take the dispatcher to handle the call, and a time to get to the call center (from the beginning of the day, in seconds – will be recommended bySleep). The time to get to the hotline is actually how many seconds the call has to pass before she decides to enter the queue.

All conversations will be created at the start of the run from an input file (see details below). Each call will wait until its arrival time at the hotline. When it's time for arrival, the call will go into the call queue of the dispatchers.

After the dispatcher handles the call and creates an invitation (see details in the next section), the call will finish its function. Please note, after entering the call queue, the calls themselves are waiting. When the dispatcher finishes handling them, the call is informed of this and ends its function. Please note that if an dispatcher forwarded a call to the manager (and did not create the invitation himself – details below), the call will end its function only when the manager finishes handling it. **Below is a flowchart describing the workflow of a call and its relationship to the dispatcher.**

2. מוקדן (Clerk)

The branch has 3 dispatchers. All dispatchers have a common call queue. Each dispatcher has a name, and a salary of 2 NIS for each call he received. When there are no calls in the queue, the dispatchers wait. When a call arrives, all thedispatchers try to answer the call (pull it out of the queue). A dispatcher who succeeded - checks the amount of pizzas in the order. If there are 10 (not included) pizzas or less, the dispatcher creates an order object from the conversation (detailed in the next section). The dispatcher creates the price field at the order according to a base price of 25 NIS per pizza. After the order is created, the dispatcher puts the invitation in a reservation queue before the slots. The duration of the call center's work on a call is according to the duration of the call, and will be recommended bySleep. After the call is completed, the dispatcher notifies the call that he has finished handling it.

If a call arrives at the dispatcherwith an order quantity greater than 10 (inclusive), the dispatcher forwards it to the call queuebefore the manager from whom the shift manager pulls out (a process that takes half a second, and will be recommended bySleep). Once the dispatcher has transferred the call to the manager's queue, he**does not inform** the call that the handling has ended. Only when the manager finishes handling the call will the manager notify the call that it has ended.

The dispatchers know in advance how many calls are supposed to arrive. They finish their work when the amount of calls that have arrived is equal to the amount of calls they knew about in advance. Please note that when one dispatcher handles the last call, the others will be waiting for calls. After he finishes handling it, he must inform the other two dispatchers that there are no more calls and they must also finish.

3. הזמנה (Order)

The orders are created by the dispatchers and transferred from them to the slots (or by the manager and entered directly into the information system).

Each order has the following characteristics:

1. Serial number of the same order (number running from 1 to the order count)
2. Quantity of pizzas, shipping address, credit number (same as those on the call)
3. Total price for payment after discount
4. Call arrival time

After an operator creates an order, she enters a reservation queue and is pulled out by slots. After an administrator creates an order, he inserts it into the information system.

4. משבץ (Scheduler)

In the branch there are 2 slots. Both slots have a common reservation queue. Each slot has a name, and a salary of 4 NIS for every second of work. When there are no reservations in the queue, the slots wait. When an order arrives, both slots try to pull it out of the queue. A tile that manages to retrieve an order converts the address listed in the booking to a distance, and adds to the order the information about the distance in kilometers. Conversion is carried out according to the following method:

*Each word in the address counts as 1 km addition to the distance (2 words – 2 km)

*In addition, if the first character in the address is a-h, an extra 0.5 km is added to the distance, if the first character is i-p, an extra 2 km is added to the distance, if the first character is q-z, added to the distance 7 km. If the first character in the address is a number, the number is the addition to the distance in km.

The total distance in kilometers will be the total addition from the amount of words and from the first character. For example,ABCRoad 44would be 3 (words) + 0.5 (a)–a total of 3.5 km.

The working time of the tile on an order is a quarter of the distance calculated (to be recommended bySleep). After completing the work on an order, the tile inserts it into the information system where it is stored according to thecorresponding ket guria. The information system stores the information but also enters the orders into the network's database (see details below).

After entering the information system, the tile prints to the screen:New Order Arrivedin addition to the serial number ofthe order entered (as a reminder, each order has a serial number). The slots finish their work after receiving notice from the manager that the day has ended (if they were waiting for invitations, the manager must notify them to stop waiting).

5. KitchenWorker (KitchenWorker)

There are 3 kitchen workers. All kitchen workers work with the information system of the branch (jointly). Each kitchen worker has a name, and a salary of 2 NIS for each pizza he made. When there are no orders in the information system, employees wait. When an order arrives, all employees try to pull it out andprepare it. When an employee pulls out an order, the information cult center will always give him orders closer to the first one (see details in the information system) if any. Of the upcoming bookings, reservations will be retrieved in order of arrival. The duration of a kitchen worker's work on a pizza will be given as input from the GUI (see below), when of course if an order includes several pizzas then the continuation will be multiplied by the number of pizzas (recommended bySleep). The kitchen worker createsthepizzasand delivery For the same order, and puts the shipment in a ready-made delivery queue (is a blocked queue) from which the couriers pull out. If there is no place to put the shipment in the queue, the kitchen workers wait. The kitchen workers finish their work after receiving notice from the manager that the day has ended (if they were waiting for orders the manager must inform them to stop waiting).

6. Pizza Delivery(PizzaDelivery)

The deliveries are created by the kitchen workers. Each delivery includes the amount of pizzas as per order. Each shipment has the following characteristics:

1. The original shipping address
2. Distance in km as calculated by the slot from the order
3. A collection of the pizzas included in the delivery (of your choice – you can save a number or define an object).

The shipments are pulled out by the couriers from the ready-made delivery queue (the blocked queue).

7. Pizza Delivery (PizzaGuy)

There are several pizza delivery people in the branch (the number will be given as input from the GUI, see details below). Couriers have a common delivery queue. Eachsalh has a name and a salary that depends on the amount of shipments transported and the distance traveled (see details below). To this salary is added a tip from customers. In addition, each courier has capacity for the amount of shipments that can be transported in one trip (regardless of whether the shipment is small or large) - an integer and random number between 2 and 4 (inclusive), to be determined with the creation of the messenger once. This number is fixed for the courier and does not change. As long as there are no deliveries ready, couriers wait. Once there is a shipment ready, both couriers try to take it. Each courier collects shipments until it has reached its capacity. Once the capacity has been reached, the courier goes on a journey. The couriers travel at a speed of 1 km per second – the travel time will be indicated bySleep. The couriers travel the distance reserved for each shipment. Once at the destination, the courier drops off the shipment, collects a tip from the customer (a random integer between 0 and 15) – the duration is 1 second and will be recommended bySleep. The courier then continues the journey to the next destination until he has finished dropping off all the shipments, and then returns to pick up more shipments from the branch (or wait if there are none).

Calculation of the salary of couriers per day according to the formula:

$$Payment = 3 * NumOfDeliveries + 4 * DistanceDrove + Tips$$

Each time a courier returns from a shipment, he informs the shift manager how many shipments he has transferred (see details in the next section). When there are 10 or fewer shipments left, the manager informs couriers to change their capacity to one shipment per trip only (instead of the initial capacity of 2-4). The couriers finish their work after receiving notice from the manager that the day has ended (if they were waiting for the delivery, the manager must inform them to stop waiting).

8. Shift Manager(Manager)

The shift manager is responsible for the functioning of the branch and handling large orders. During the day, the manager waits to queuecalls for large ordersin front of him. When a call (from the dispatchers) arrives, hepulls it out and creates an invitation object. The price will be set at a base price of 15 NIS per pizza, and a 10% discount on the total order if it contains more than 20 pizzas. The manager updates all relevant order details (by invitation object). In addition, the manager enters the distance in kilometers according to the same rules defined by the tiles (converting the address to the distance), and tries to enter the invitation into the information system. The manager's total working time for a calltakes2secondsand will be recommended bySleep. As soon as a manager has finished handling the call, he notifies the call that it has ended and the call can end its function.

The shift manager knows in advance according to the forecast how many calls will arrive at the branch during the day (but does not know how many calls will include large orders). In addition, he knows how many orders have been sent to customers so far in practice. Every time a courier returns from a trip, he informs the manager how many shipments he has delivered. When the manager is informed that all the shipments have been transferred (shipments have been transferred as the amount of callsthat were supposed to arrive), The manager knows that the working day is over. Please note that if the manager has so far been waiting for calls himself, when a courier delivers a shipment, he must make the manager stop waiting and check whether the day has ended (i.e. all shipments have been delivered). In addition, when the manager checks how many shipments are left to be delivered, if there are 10 or fewer shipments left (inclusive) – he informs the couriers that from now on only one shipment will be collected at a time (and will not wait each time to collect according to their initial capacity).

At the end of the working day, the manager informs the slots, kitchen workersand couriersthat the day has ended. After that, the manager prints to the screen the total salary of all employees, the amount of shipments delivered, and the total revenue of the branch (calculate how to transfer to him or save the information required for the calculations). It is not the printing dawn that the manager finishes his work, and the whole day ends.

Queues, Information System,GUI

9. Unblocked queue

A queue that is not blocked, can be entered without space limitation. **Calculate how it would be best to implement this appointment so that it fits the system modularly and according to what is taught in the course.** This queue instances in the system as described above:

1. Call queue 1 before the dispatchers, to which the calls insert themselves and the dispatcherspull out.
2. Call queue 1 before the manager, to which the dispatchers put calls into large orders and the manager pulls out.
3. Order queue 1 before the slots, to which the dispatchers insert orders and the slots pull out.

10. תור חסום (Bounded Queue)

A blocked queue can be inserted as long as there is no deviation from 12objects in the queue. **Calculate how it would be best to implement this appointment so that it fits the system modularly and according to what is taught in the course.** There is a single instance of this queue in the system, between the kitchen workers and the couriers. The kitchen workers put ready-made deliveries into it, and the couriers pull out of it. **When this queue cannot be entered, the kitchen workers wait**.

11. מערכת המידע (Information System)

Is not a queue in the full sense of the word. The information system also containsall the orders that still need to be made. In the information system, reservations are stored according to distance distribution – reservations of up to 3 km (including 3), reservations of 3-8 km (excluding 3 and including 8), and reservations of 8 km or more (not including 8). **Think about how to save the information. No points will be dropped on less elegant ways of solution, but you are free to use existing data structures on which information can also be found on the Internet.**

Uses of the system – it is recommended for each use to give a separate method:

1. Slotsand the manager enter into the reservation information system (the system slots the order in the appropriate category of distance). After the tile or managerenters an order, he prints the details (prints to the screen:New Order Arrivedin addition to the serial number of the order he entered (as a reminder, each order has a serial number).
2. Kitchen workers pull out of the information system. The employee asks the information system for an order (no input), and the system automatically gives him an order from a closer distance category (if any), and from that category the order that arrived earlier, and the order leaves the system (but not from the database!). If there are no orders, the information system does not return anything (returns null).
3. In the process of retrieving the order from the information system, it is necessary to print the name of the cook who pulled out the order and his current salary after the order is retrieved (2 NIS more). Then print out the details of the order: pizza number, price, distance, address, credit number.

Please note: this is an old information system, and only one person can access it at the same time **– you need to prevent a situation where two objects: a slot, a manager, a kitchen worker access the information system at the same time.**

12. GUI – Graphical User Interface

The GUI is the first thing that is displayed to the user when the program starts running. It will include**<u>at least</u>**a title, two input fields as listed below (with matching labels), and two buttons as detailed below. You can add text/functionality to the GUIthat doesn't contradict the instructions as you see fit (e.g. a window title or an option to close when you click the red X).

Inputs that the GUI will allow to enter:

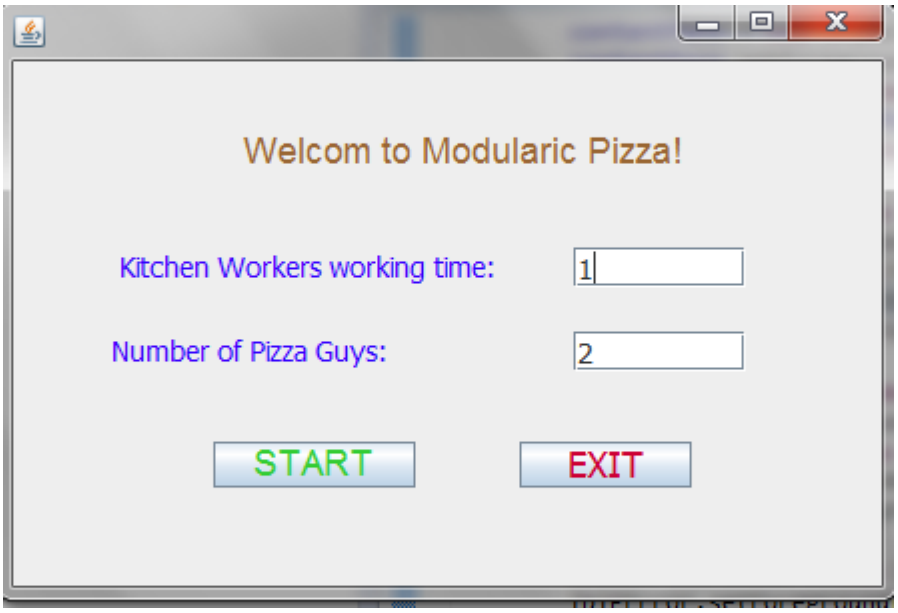
1. The duration of work of a kitchen worker on pizza. **A depolar value that will appear in** the **GUIwhen the program runs will be**1. The field must allow entering integers.
2. The amount ofcouriers there are in the branch. **A depolar value that will appear in** the **GUIwhen you run the program will** be 2. The field must allow integers to be entered.

Note: There is no need toperform input checks on the fields in the GUIafter the user enters them (for example, that they did not accidentally enter a negative number).

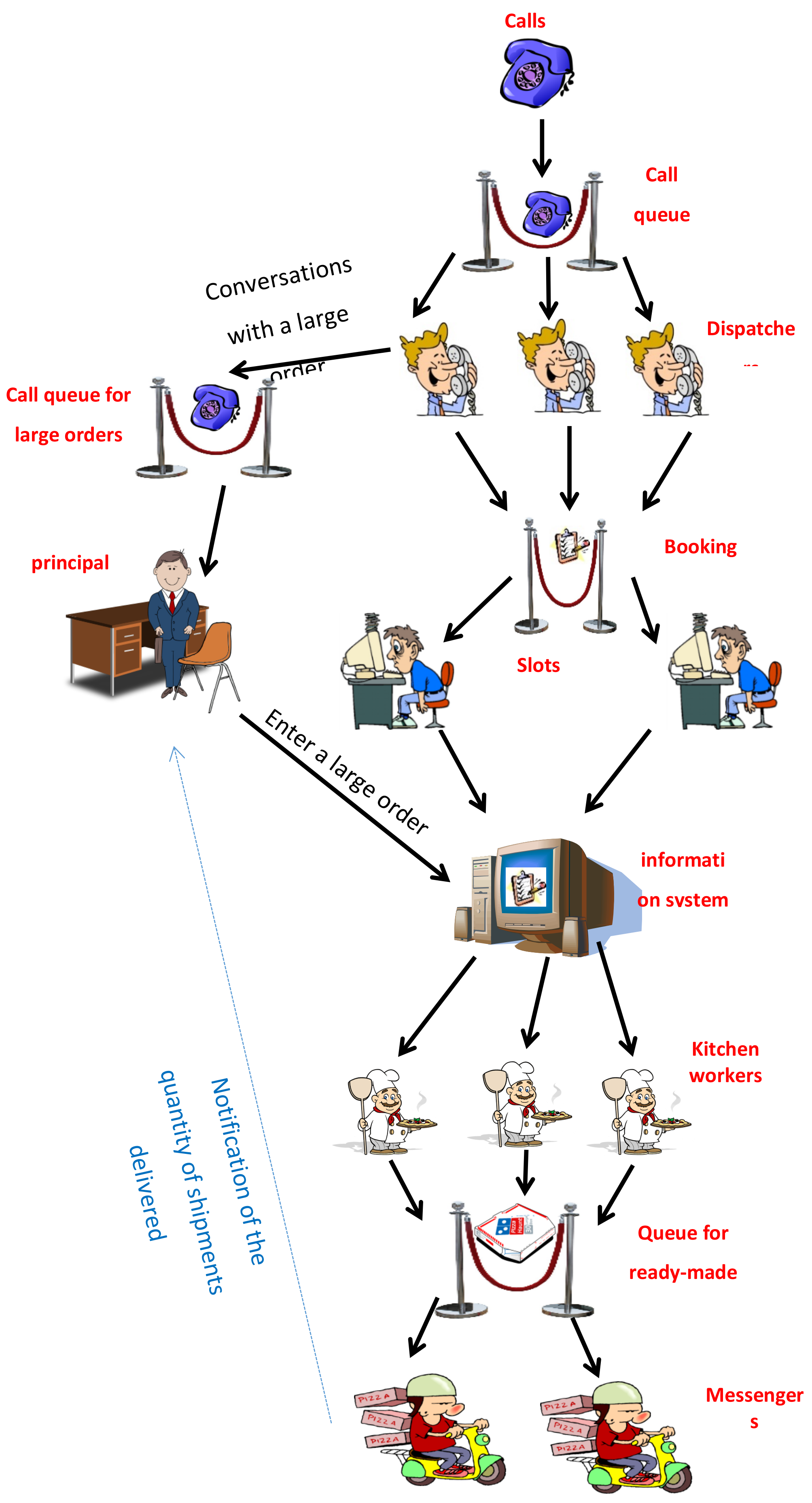
- When pressingSTART, variables will be initialized, the data from the input file will be read, then all objects will be created, their work will begin, and the calls will begin to reach the system. All objects in the system will begin their "work" at the same time.
- Rerun must be enabled by pressingSTART again. In the additional run, the user can change the values of the variables.
- Next to theSTART button will be placed an EXIT button, pressing which will exit the program. It can be assumed that this button will not be pressed until the working day is completely over.

<u>Note</u>: Please note that the GUI automatically creates a main function. This function will serve as your main function.For self-testing, it's a good idea to createa standalone mainduring the writing process before you've created the GUI. **<u>It is highly recommended to leave this partand</u>** the **<u>GUI</u>** integration **<u>towards the end of the work.</u>**

Attached is an example of the GUI. Note that the text and formatting are not binding – try to be original.

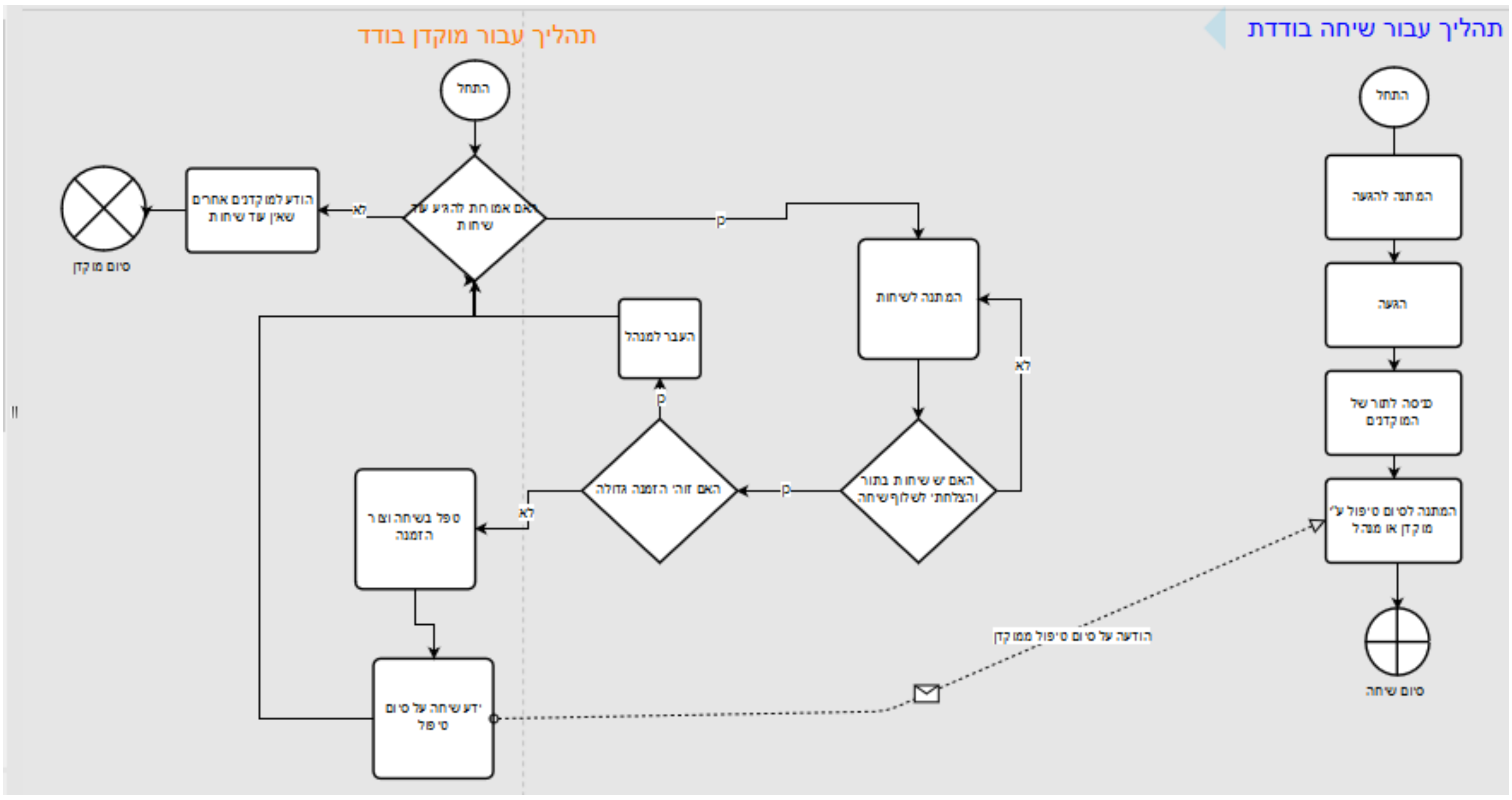


System schematic diagram



Process diagram for call center and dispatcher

Attached is an example flowchart to clarify the process and the relationship between calland dispatchers. Note that this is a simple chart that simultaneously shows the process of a call and a dispatcher and the relationship between them, rather than the manager's process. Of course, if the dispatcher forwarded a call to the manager she



would continue to wait until the manager informs her that she can finish.

Additional highlights and guidelines

The input file

- Attached is a **sample**text file with the call data: credit number, amount of pizzas on order (and delivery), time that passes from the beginning of the run until the arrival of the call (actual sleep will be performedwhen the start of each call), time it takes to handle the call. The work will be tested with a text file **of** the same format (the fields can accept values according to the data in the text file) but not necessarily with the same data. Adaptthe reading from your file to this format. Note that each column is **separated by a single "Tab" from another column and that the first row is a header row. When reading, you ignorethe title bar. Pay attention to what data (and what type) there is in each column.** At the beginning of the program run, you need to read the data from the input file and enter the data relevant to the various objects, as defined for each object. There is nosubdata for an object that has not been defined for it (for example, giving the slots information about how many calls are due to arrive, or giving the manager information about how many large calls are supposed to arrive, etc.). **Points will be deducted for giving information that is not necessarily necessary for the operation of the object according to the instructions.**
- **You must realize all the objects listed in the instructions, without exception. You can add more objects as you see fit.**
- Note that the working day ends only when all the objects running at the same time have finished their work and have "died". **Points will be dropped if at the end of the day there are threads that do not wake up and continue to wait.**

- **Most important**: This work will examine the principles of object-oriented software development that were learned throughout the course. Pay attention to the principles ofparallel programming (synchronization), closing, TOP DOWN programming, and when inclusion is required and when only mutual pointers are required (not containment). In addition, the information must pass through the system without giving unnecessary pointers between the objects – each object gains access to other objects only if it is required to do so for its work. In addition, Try to think about the soft taxin reality and how it would have been conducted. **Solutions that do not meet this will lose points.**
- You can add additional departments as needed, as long as the principles taught in the course are maintained.
- **If there are extreme cases that are not specified in the work, you can assume what will be convenient for you, provided that there is no contradiction to the instructions given.**

Recommended work order

1. Read the instructions.
2. Read the instructions again.
3. From i=1 to i=10,follow sections 1-2 of the "Recommended work order" (i++).
4. Think about what isin thesystem is Thread and what is notThread.
5. Plan your solution process in detail. During programming, remember to programTop Down. It is highly recommended not to finish programming everything to the end but during the Top Down programming leave functions and methods that you have not fully realized in order to first complete the general structure of the process.
6. Implementthe queue classes in a basic way – a blocked queue and an unblocked queue in accordance with the principles learned in the course. Exercise thinking in this section. Please note that there are queues for calls and queues for reservations. You may need to add features to your queues later on.
7. Start by building the skeleton of the work - completely realize the order object and the delivery object. Continue to the creation of all the following objects in

a basic way - constructors, access methods if required - in the following order: caller, dispatcher, slot, kitchen worker, courier, manager, information system.

8. Start with a deeper realization of the following objects: conversationalist, dispatcher, slot, kitchen worker, courier, manager. Leave complex functionality like end-of-day service, database entry, large order handling.
9. Fully implement the information system.
10. Finish realizing complex functions of all objects.
11. Build a main to check the work so far.
12. Take care of the prints that take place during the run – the calculations and the form of printing. Pay attention to synchronization in print (that no mixed printing will come out when different objects try to print to the screen).
13. Pay attention to the transfer of information in the system - how each person notifies someone else that they have finished, and take care of the process of ending the day.
14. Build the object that handles the database.
15. Build the GUI.

    Note: It is strongly recommendedthat you leave the GUIto the end. First check that the job works without them (build an independent main function, and findsync issues).

Submission Instructions and General Information

- Serving in pairs. For individuals – with the approval of the course supervisor, Roy Zivan only.
- **You must add code(note)** documentation **throughout the work. Code without comments will not be checked.**
- The exercise willbe submitted as a zip file to the submission box in the model.
- The entire project folder (with **all the departments you are required to redeem) must be submitted**, with the folder name being the social security numbers. For example:11111111_22222222. This folder must be inserted into the zip.
- Deferrals for work will be givenand by the course supervisor Roy Zivan only.
- Responsible practitioner: Ben Rahmut

- Checkingout exercises – MartynAbadi.
- Every day a delay leads to a reduction of 5 points.
- Questions should be asked in the forum only, and **only regarding the guidelines and** **<u>not how to implement the work.</u>**