Project 2: ExcelFed

CmpE 250, Data Structures and Algorithms, Fall 2021

Instructor: H. B. Yılmaz
TA: Özlem Şimşek
SA: Leyla Yayladere, Berk Atıl, Hasan Baki Küçükçakıroğlu

Due: **24.11.2021 Wednesday, 23.55**

1 Introduction

Heraclitus once said "The only constant in life is change". Heraclitus was a wise man, indeed, but he missed an exception: if you take CMPE 250, you solve a Discrete Event Simulation (DES) project!

Well, but what is DES?

Enter, Wikipedia: "A discrete-event simulation (DES) models the operation of a system as a (discrete) sequence of events in time. Each event occurs at a particular instant in time and marks a change of state in the system. Between consecutive events, no change in the system is assumed to occur; thus the simulation time can directly jump to the occurrence time of the next event". In other words, DES is the representation of a certain process by simulating certain events which occur at certain times. Hence, time is not continuous in DES and progress according to the events. As a demonstrative example, a process of taking a coffee can be simulated as follows:

- TIME: 09:00 Go into the checkout queue.

- TIME: 09:05 Order your coffee and make your payment.

- TIME: 09:10 Go into the serving queue.

- TIME: 09:15 Wait for the preparation of your coffee and enjoy your free time.

- TIME: 09:20 Grab your coffee and enjoy!

Note that in DES the only things that count are events and everything else is ignored. In the case of the customer, until 09:20 only 5 things happened. Additionally, the time progressed in a discrete fashion (i.e. jumped from one event to the next one). [1]
In this project, you are expected to simulate training procedure of ExcelFed, which is a tennis club founded by his excellency

---

[1]For further reading about DES you can visit here or Wikipedia

2   Details of the ExcelFed

Do you know his excellency? This is not even a question, everyone knows him. Yes, I am talking about Roger Federer[2] who is the best tennis player ever! He has decided to retire since he is not young enough to continue his career. However, he wants to transfer his great experience to the new generation. That's why, he has founded ExcelFed.

Roger is very experienced so he knows almost everything that plays a huge role in the development of the tennis player. He has already hired great nutritionists and chefs. Additionally, he has a consulting team consists of Rafael Nadal and Novak Djokovic. Furthermore, he has rent nice tennis courts. His aim is to train children in very good conditions so that there are better tennis players.

There is one "little" problem though. He needs great training coaches, masseurs, and physiotherapists but he does not know how many he needs. He may hire many staffs but this would be a waste of money and there are not many outstanding staffs. **Therefore, he decided to simulate the training procedure of ExcelFed with fictional players and staffs to find how many of them would be enough**. However, his relationship with simulation tools and coding is terrible so he needs your help! Let Roger provide more detail.

- There might be multiple training coaches, masseurs and physiotherapists depending on the simulation configuration. In this case, every physiotherapist has his/her own service time and training, and massage duration depending on the player's need.

- There are exactly three queues in the system: one for training, one for massage and one for physiotherapy. So, each coach shares **a common queue** and similar for the others.

- The training queue works in **first-come-first-served** fashion. Thus, the first player to enter the queue is served before the others. If two players arrive at the same time, the one with the lower ID is served first. When the training of the player is **finished**, he/she enters the physiotherapy queue, **immediately**.

- Since more training time requires more urgent rehabilitation, physiotherapy queue works in a prioritized manner. Therefore, in this queue, **the more the training time, the higher the priority**. If there are more than one player that have the same training time, then the one that arrived earlier is served before. If they arrived at the same time as well, then the one with the lower ID is served first. Note that, one player may come to the training more than one time, for the prioritization, you should consider only the current training time not the cumulative.

- Because massage is an advanced service, players should deserve them by their skills. Hence, in the massage queue, **the higher the skill level, the earlier the service**. If there are more than one player that are in the same skill level, then the one that arrived earlier is served before. If they arrived at the same time as well, then the one with the lower ID is served first

- For all of the services, players visit the first available staff for the service (the available staff with the smallest ID) when they leave the queue.

---

[2]A video to enjoy the talent of Roger Federer.

- Each player is allowed to take at most **3** massage service. Hence, whenever a player attempts to enter the massage queue 4th time, this is called an "invalid attempt".

- Since it is hard for the players to estimate how much time they spend in the queues, there could be some cases such that players may try to come to the training or massage when they are already in the training, physiotherapy or massage process. These attempts should be "canceled" so they are called "canceled attempts".

- One can enter the physiotherapy queue only after the training, no direct entrance is possible.

Below, is a schematic description of the training procedure of his excellency.
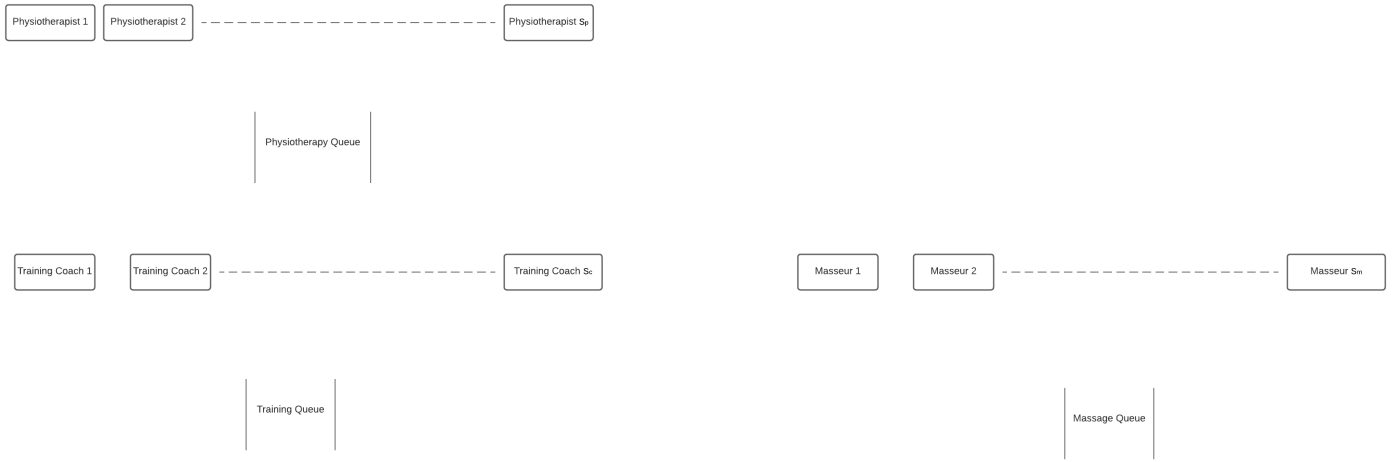

Figure 1: A schema of the queues

Therefore, for our purposes, there are two types of events triggered by the players: coming to the training and massage request. Keep also in mind that internal events, such as leaving the training, should also be triggered by the simulation. Roger Federer expects you to simulate the training procedure of ExcelFed using these external and internal discrete events and collect some statistics.

3   Input & Output

3.1   Input

Roger provides all simulation configuration files in the following format:

- The first line contains an integer N that denotes the total number of players in ExcelFed.

- Each of the next N line contains two integer: the ID of the player and his/her skill level. These players will be given in **sorted** order by ID.

- The next line is the line of an integer A that denotes the total number of arrival to the training and massage.

- Each of the next A lines contains a character A_t denoting the type of arrival (either "m"(massage) or "t"(training)), an integer denoting the ID of the player, the second T that denotes the arrival time, and d duration of the process. These events will **not** be given in **sorted** order.

- The next line comprises an integer S_p that denotes the number of physiotherapists and a list of floats of size S_p. The ith element of the list denotes the service time of the ith physiotherapist.

- The last line contains two integers: S_c that denotes the number of training coaches and S_m that denotes the number of masseurs.

Table 1 demonstrates an example input provided by Rafael Nadal, the best consultant in the consultancy team.

Sample Input File	Explanations
3	There are 3 players.
0 2	The player with ID 0 has a skill level 2.
1 3	The player with ID 1 has a skill level 3.
2 1	The player with ID 2 has a skill level 1.
9	There 9 arrivals in total.
t 0 1 1.2	The player with ID 0 arrives for the training at second 1 and the training will take 1.2 seconds.
t 1 1.2 2	
m 0 6 3	The player with ID 0 arrives for the massage at second 6 and the massage will take 3 seconds.
m 1 6.3 1.8	
m 0 10 1.2	
t 2 10.1 4	
m 0 15 1.2	
m 2 16 1.5	
m 0 20 3	
1 2	There is 1 physiotherapist and her service time is 2 seconds.
1 1	There are 1 training coach and 1 masseur.

Table 1: Sample Input with Explanations

3.2 Output

Roger needs you to collect the following statistics to evaluate the configuration and output them in separate lines, in the order they are given. Please round the statistics to **exactly 3 decimal points**. In case you cannot report any of the following 15 statistics, **print -2** for each of them in order to adhere the output format.

1. Maximum length of the training queue.

2. Maximum length of the physiotherapy queue.

3. Maximum length of the massage queue.

4. Average waiting time in the training queue.

5. Average waiting time in the physiotherapy queue.

6. Average waiting time in the massage queue.

7. Average training time.

8. Average physiotherapy time.

9. Average massage time.

10. Average turnaround time (Turnaround time: Total time passed from the training queue entrance until leaving the physiotherapy service.) To compute, sum all turnaround times and divide it by the number of turnarounds, which is also equal to the number of total training arrivals.

11. ID of the player who spent the most time in the physiotherapy queue and the waiting time of that player in seconds. If more than one player spent the same amount of time, choose the one with the smallest ID.

12. ID of the player who spent the least time in the massage queue and the waiting time of that player in seconds, among the ones who took three massage services. If more than one player spent the same amount of time, choose the one with the smallest ID. If there is no player that took three massage services, print -1 for both.

13. Total number of invalid attempts to get a message service.

14. Total number of canceled attempts including both training and massage attempts.

15. Total seconds passed during the whole simulation.

Nadal also showed the courtesy to share the expected output for the input in Table 2 with explanations. He also showed the progress of queues and services in Table 3 where the numbers in the parenthesis in the "service" columns represent the number of remaining seconds to finish the process.

3.3 Java Project Outline

Your java project will be named **Project2**. Your entry class for the project will be named **project2main**. All your .java files will be under folder **Project2/src**. Your project should be compatible with Java 16. Your program will be compiled with below command:

javac Project2/src/*.java -d Project2/bin –release 16

The input and output files can be at any folder. Design your code in order to accept full path for file arguments. Your program will be run with below command:

java project2main <inputfile> <outputfile>

Make sure that your final submission compiles and runs with these commands.

Output File	Explanation
1	Between seconds 1.2 and 2.2 Player 1 waits for the training.
0	No one waits for the physiotherapist.
1	Player 1 and Player 0 waits for the massage but not together so the length of the queue is at most 1.
0.333	In total, there are 3 training events and only Player 1 waits for 1 second so the answer is 1/3.
0	No waiting for the physiotherapy queue.
0.875	Player 1 waits for 2.7 seconds and Player 0 waits for 0.8 seconds so in total 3.5 seconds waiting. Also there are 4 valid massage attempts so 3.5/4.
2.4	Total training time is 1.2+2+4 so 7.2 and the answer is 7.2/3.
2	Since there is 1 physiotherapist and her service time is 2 seconds average is also 2 seconds.
1.8	Total massage time is 3+1.2+1.2+1.8 so 7.2 and there are 4 massage services, the answer is 1.8.
4.733	Player 0 arrives at 1 and leaves the physiotherapy service at 4.2. Player 1 arrives at 1.2 and leaves the physiotherapy service at 6.2. Player 2 arrives at 10.1 and leaves the physiotherapy service at 16.1.
0 0	No waiting in physiotherapy queue so the smallest ID is chosen.
0 0.8	Only Player 0 takes 3 massage services.
1	Player 1 attempts to take 4 massage services 1 of which is invalid.
1	Massage attempt by Player 2 at 16 is canceled.
20	It is the end of the last event which is the invalid massage attempt by Player 0.

Table 2: Expected Output File and Explanation of each Statistic

Second	Training Queue	Training Service	Physiotherapy Queue	Physiotherapy Service	Massage Queue	Massage Service
1	-	Player 0 (2.2)	-	-	-	-
1.2	Player 1	Player 0 (2.2)	-	-	-	-
2.2	-	Player 1 (4.2)	-	Player 0 (4.2)	-	-
4.2	-	-	-	Player 1 (6.2)	-	-
6	-	-	-	Player 1 (6.2)	-	Player 0 (9)
6.2	-	-	-	-	-	Player 0 (9)
6.3	-	-	-	-	Player 1	-
9	-	-	-	-	-	Player 1 (10.8)
10	-	-	-	-	Player 0	Player 1 (10.8)
10.1	-	Player 2 (14.1)	-	-	Player 0	Player 1 (10.8)
10.8	-	Player 2 (14.1)	-	-	-	Player 0 (12)
12	-	Player 2 (14.1)	-	-	-	-
14.1	-	-	-	Player 2 (16.1)	-	-
15	-	-	-	-	-	Player 0 (16.2)
16.1	-	-	-	-	-	Player 0 (16.2)
16.2	-	-	-	-	-	-

Table 3: Step by step progress of the queues and services. Numbers in the parentheses denote the exact second to leave the service.

4   Grading

Grading of this project is based on the automatic compilation and run and the success of your code in test cases. If your code compiles and runs with no error, then you will get **10/100** of the project grade. The rest of your grade will be the sum of collected points from each test case. Each test case will have equal weight. However, each statistic may **not** contribute to the

overall point for the test case **equally**. We haven't decided about their weights, as soon as we decide, we will share with you. Maximum project grade is 100. If your submission is not auto-runnable, then **you will receive 0 at first and then have to issue an objection**.

5 Warnings

1. This is an individual project.

2. All source codes are checked automatically for similarity with other submissions and exercises from previous years. Make sure you write and submit your own code. Any sign of cheating will be penalized and you will get **-50** points for the project and you will get **F** grade in case of recurrence in any other project.

3. There will be time limit on test cases, so it is important to write your code in an efficient manner.

4. You can add as many files as you can as long as they are in the "src" folder and your project can be compiled as above. But the entry point of your program should be "project2main.java".

5. Make sure you document your code with necessary inline comments and use meaningful variable names. Do not over-comment, or make your variable names unnecessarily long. This is very important for partial grading.

6 Submission Details

You will zip up your project folder and submit on Moodle as a single .zip file. The name of the zip file is **Cmpe250_Project2_<studentid>.zip** No other type of submission will be accepted.

Some tips regarding the project

1. First, ensure to understand the project with all definitions and concepts before implementing. **DO NOT START CODING RIGHT AWAY!**. When you are comfortable with project in every aspect, it is highly recommended to simulate some test cases by hand and think of edge cases. Then, you will be ready to start coding. Otherwise, you might end up like in Figure 2

2. We do think that this project is one of the most important ones in this course. Therefore, we suggest starting this project early(not on the last day!).

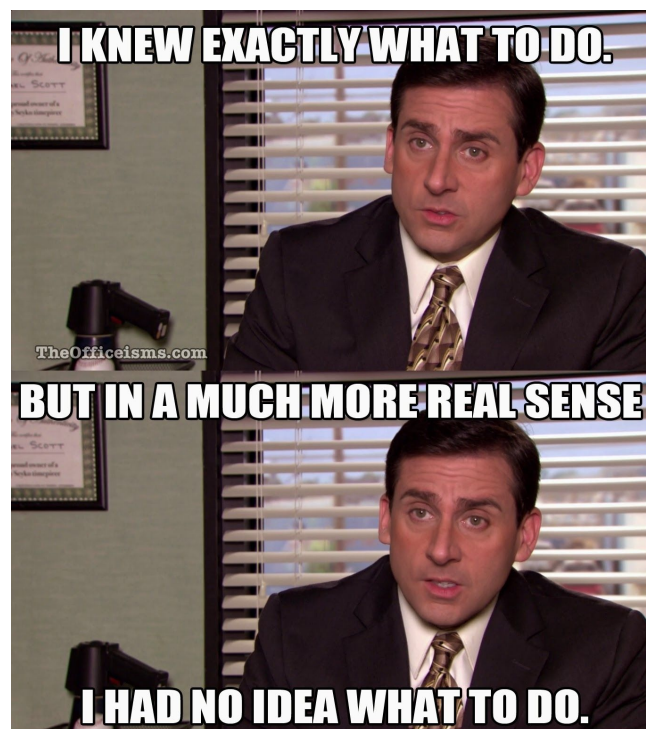3. For a more clean and understandable code, try to code object-oriented. Think about the classes that you can use. An object-oriented design will make the code easier to develop and to debug.

Figure 2: Starting to Code Right Away