Documentație P2

Creați o aplicație pentru gestiunea concurenților de la un concurs de programare. Programul înregistrează scorul obținut de fiecare concurent la 10 probe diferite, fiecare probă este notat cu un scor de la 1 la 10. Fiecare participant este identificat printr-un număr de concurs, scorul este ținut într-o listă unde concurentul 3 are scorul pe poziția 3 în listă . Programul trebuie sa ofere următoarele funcționalități:

1. Adaugă un scor la un participant.

a) Adaugă scor pentru un nou participant (ultimul participant)

b) Inserare scor pentru un participant.

2. Modificare scor.

a) Șterge scorul pentru un participant dat.

b) Șterge scorul pentru un interval de participanți.

c) Înlocuiește scorul de la un participant.

3. Tipărește lista de participanți.

a) Tipărește participanții care au scor mai mic decât un scor dat.

b) Tipărește participanții ordonat după scor

c) Tipărește participanții cu scor mai mare decât un scor dat, ordonat

după scor.

4. Operații pe un subset de participanți.

a) Calculează media scorurilor pentru un interval dat (ex. Se da 1 și

5 se tipărește media scorurilor participanților 1,2,3,4 și 5

b) Calculează scorul minim pentru un interval de participanți dat.

c) Tipărește participanții dintr-un interval dat care au scorul multiplu de 10.

5. Filtrare.

a) Filtrare participanți care au scorul multiplu unui număr dat. Ex. Se da

numărul 10, se elimină scorul de la toți participanții care nu au scorul

multiplu de 10.

b) Filtrare participanți care au scorul mai mic decât un scor dat.

6. Undo

• Reface ultima operație (lista de scoruri revine la numerele ce existau

înainte de ultima operație care a modificat lista)

Plan de iterații

**Iteration 1 (Week 4):**

- create the main elements of the problem and write basic

implementations for data manipulation features

• target functionalities: 1ab, 2abc, 4abc, 5ab

• target modules: participant.py

- data visualization feature

• target functionalities: 3abc

• target modules: participant.py, table\_printer.py

- UI mockup

• target modules: ui.py

**Iteration 2 (Week 5):**

- undo feature implementation

• target functionalities: 6

• target modules: ...

- UI polishing

• target modules: ui.py

**Iteration 3 (Week 6):**

- refactoring, bug fixes and optimization

• target functionalities: all

• target modules: all

Scenariu de rulare

|  |  |  |
| --- | --- | --- |
| User input | Program | Description |
|  | Enter your option: | Requests option |
| 1 |  |  |
|  | Enter yout option: | Requests suboption |
| a |  | 1a = add new participant |
|  | The 10 scores separated by spaces : |  |
| 9 9 8 10 9 9 10 10 10 9 |  |  |
|  | Added participant with code 1 at index 1 |  |
| *(add some more participants)* | | |
|  | Enter your option: |  |
| 3 |  |  |
|  | Enter your option: |  |
| b |  | 3b = print ordered by score |
|  | ║ Crt. ║ Code ║ P1 ║ P2 ║ P3 ║ P4 ║ P5 ║ P6 ║ P7 ║ P8 ║ P8 ║ P10 ║ Score ║ Score ║  ╠══════╬══════╬════╬════╬════╬════╬════╬════╬════╬════╬════╬═════╬═══════╬═══════╣ ║ 2 ║ 2 ║ 10 ║ 10 ║ 10 ║ 10 ║ 10 ║ 10 ║ 10 ║ 10 ║ 10 ║ 10 ║ 100 ║ 100 ║  ╠══════╬══════╬════╬════╬════╬════╬════╬════╬════╬════╬════╬═════╬═══════╬═══════╣ ║ 1 ║ 1 ║ 9 ║ 9 ║ 8 ║ 10 ║ 9 ║ 9 ║ 10 ║ 10 ║ 10 ║ 9 ║ 93 ║ 93 ║  ╠══════╬══════╬════╬════╬════╬════╬════╬════╬════╬════╬════╬═════╬═══════╬═══════╣ ║ 6 ║ 6 ║ 9 ║ 8 ║ 7 ║ 5 ║ 3 ║ 10 ║ 1 ║ 7 ║ 6 ║ 10 ║ 66 ║ 66 ║  ╠══════╬══════╬════╬════╬════╬════╬════╬════╬════╬════╬════╬═════╬═══════╬═══════╣ ║ 3 ║ 3 ║ 6 ║ 7 ║ 1 ║ 2 ║ 3 ║ 5 ║ 6 ║ 8 ║ 6 ║ 10 ║ 54 ║ 54 ║ ╠══════╬══════╬════╬════╬════╬════╬════╬════╬════╬════╬════╬═════╬═══════╬═══════╣ ║ 5 ║ 5 ║ 1 ║ 7 ║ 4 ║ 5 ║ 9 ║ 3 ║ 6 ║ 4 ║ 10 ║ 2 ║ 51 ║ 51 ║ ╠══════╬══════╬════╬════╬════╬════╬════╬════╬════╬════╬════╬═════╬═══════╬═══════╣ ║ 4 ║ 4 ║ 10 ║ 10 ║ 1 ║ 1 ║ 4 ║ 5 ║ 4 ║ 3 ║ 2 ║ 8 ║ 48 ║ 48 ║ ╠══════╬══════╬════╬════╬════╬════╬════╬════╬════╬════╬════╬═════╬═══════╬═══════╣ ║ 7 ║ 7 ║ 1 ║ 1 ║ 2 ║ 3 ║ 4 ║ 1 ║ 5 ║ 6 ║ 10 ║ 1 ║ 34 ║ 34 ║ ╚══════╩══════╩════╩════╩════╩════╩════╩════╩════╩════╩════╩═════╩═══════╩═══════╝ | Prints list of participants ordered by score |
|  | Enter your option: |  |
| 2 |  |  |
|  | Enter your option: |  |
| b |  | 2b = delete range of parts. |
|  | Give the interval bounds separated by space : |  |
| 2 4 |  | Deletes parts. Index 2,3,4 |
|  | Enter your option: |  |
| 4 |  |  |
|  | Enter your option: |  |
| a |  | 4a = Find average score for an interval of paricipants by index (default: entire list) |
|  | Give the interval bounds separated by space : |  |
| \n |  | (select entire list) |
|  | The average score for interval [1,4] is 61.0 |  |
|  | Enter your option: |  |
| 5 |  |  |
|  | Enter your option: |  |
| a |  | 5a = filter parts. with score multiple of a number |
|  | Give factor: |  |
| 2 |  |  |
|  | Filtered 2 participants out of 4. |  |
|  | Enter your option: |  |
| 6 |  | Undo |
|  | Enter your option: |  |
| 3, then a, then 100 |  | Print list of participants with score less than 100 |
|  | Prints list of 4 participants. |  |
| 7 |  | Exit |
|  | (Execution ended) |  |

Lista de activități

**T1. Create a new Participant based on given code and scores.**

- target functionalities: 1a, 1b

- points of interest: Participant

**T2. Option to change the participant's score for a given problem.**

- target functionalities: 2c-1

- points of interest: Participant.change\_score

**T3. Option to change the participant's score for all problems.**

- target functionalities: 2c-2

- points of interest: Participant.change\_scores

**T4. Add a new participant to the participants list by their scores.**

- target functionalities: 1a

- points of interest: ParticipantsList, ParticipantsList.add\_participant\_by\_score

**T5. Insert a new participant to the participants list by their code & scores.**

- target functionalities: 1b

- points of interest: ParticipantsList.insert\_participant

**T6. Query a subset interval of the participants list**

- target functionalities: 4a, 4b

- points of interest: ParticipantsList.average, ParticipantsList.min

**T7. Filter participants based on a given condition.**

- target functionalities: 5a, 5b

- points of interest: ParticipantsList.filter

**T8. Print list of participants satisfying a given condition and in a given order**

- target functionalities: 3a, 3b, 3c, 4c

- points of interest: ParticipantsList.print, TablePrinter

**T9. UI implementation**

- points of interest: TablePrinter, Operation, UserInterface