Laborator 4 – P2. CONCURS

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**

a. Reface ultima operație (lista de scoruri revine la numerele ce existau înainte de ultima operație care a modificat lista)

Planificare Iteratii

**Iteratia 1**

Crearea unei console user-friendly cu toate task-urile posibile

Citirea unui cod si trimiterea catre functiile care rezolva acel cod

Functionalitatile 1 + 2 + 3, User-Interface + Codul din spate

* Validarea datelor
* Manipularea facila a bazei de date
* Operatiile de baza ( adaugare, stergere, ordonare)
* Cateva operatii de tipul GET, necesare functionalitatilor
* Teste

Scenarii de rulare iteratia 1 – Functionalitatea 1:

User UI Functionalitati

|  |  |  |
| --- | --- | --- |
|  | Meniu | [] |
| jdsasdas |  |  |
|  | Introduceti o comanda valida |  |
| 1a |  |  |
|  | Introduceti nr de concurs |  |
| 25 |  |  |
|  | Introduceti scorul |  |
| 250 |  |  |
|  |  | [[25, 250, [250]] |
|  | Operatia a fost realizata cu succes |  |
| 25 |  |  |
|  | Input gresit – ID Deja utilizat |  |
| 30 |  |  |
|  | Introduceti nr de concurs |  |
| 350 |  |  |
|  |  | [[25, 250, [250]], [30, 350, 350]] |
|  | Operatia a fost realizata cu succes |  |
| 1b |  |  |
|  | Introdu nr de concurs |  |
| 35 |  |  |
|  | Input gresit - ID-ul nu exista |  |
| 25 |  |  |
|  | Introdu scorul |  |
| 200 |  |  |
|  |  | [[25, 225, [250, 200]], [30, 350, [350]] |
|  | Operatia a fost realizata cu succes |  |
| 2a |  |  |
|  | Input gresit – ID-ul nu exista |  |
| 25 |  |  |
|  |  | [[30, 350, 350]] |
|  | Operatia a fost realizata cu succes |  |
| 1a |  |  |
|  | Introduceti numar de concurs |  |
| 1 |  |  |
|  | Introduceti scor |  |
| 1 |  |  |
|  |  | [[1,1,[1]][30, 350,[ 350]] |
|  | Operatia a fost realizata cu succes |  |

**Iteratia 2**

Citirea unui cod si trimiterea catre functiile care rezolva acel cod

Functionalitatile 4+5, User-Interface + Codul din spate

* Validarea datelor
* Manipularea facila a bazei de date
* Operatiile superioare ( tiparire, selectie )
* Impartirea pe module cu coeziune maxima
* Teste

**Iteratia 3**

Citirea unui cod si trimiterea catre functiile care rezolva acel cod

Functionalitea 6, User-Interface + Codul din spate

* Crearea unei liste care va retine lista dupa fiecare modificare
* Manipularea facila a acelei liste
* Operatiia de UNDO
* Impartirea pe module cu coeziune maxima
* Teste