Documentatie Lab 7-9

Mutu Vlad

grupa 214/2

**Cerinta:**

**P5. Organizare evenimente**

Scrieți o aplicație pentru organizarea de evenimente Aplicația stochează:

persoana <ID>,<nume>,<adresa>

eveniment <ID>,<data>,<durata>,<descriere>

Creați o aplicație care permite:

• gestiunea listei de persoane și evenimente.

• adaugă, șterge, modifică, lista de persoane, lista de evenimente

• căutare persoane, căutare evenimente

• Înscriere persoană la eveniment.

• Rapoarte:

• Lista de evenimente la care participă o persoană ordonat alfabetic după descriere, după dată

• Persoane participante la cele mai multe evenimente

• Primele 20% evenimente cu cei mai mulți participanți (descriere, număr participanți)

|  |  |
| --- | --- |
|  | **Lista de functionalitati:** |
| **F1** | Generarea, stergerea si modificarea unei liste de persoane si evenimente |
| **F2** | Adaugarea de persoane/evenimente in lista de persoane/evenimente |
| **F3** | Afisarea listei de persoane/evenimente |
| **F4** | Cautarea de persoane si evenimente |
| **F5** | Inscrierea unei persoane la un eveniment |
| **F6** | Afisarea listei de evenimente la care participa o persoana ordonata dupa descriere/dupa data |
| **F7** | Afisarea persoanelor care participa la cele mai multe evenimente |
| **F8** | Afisarea primelor 20% evenimente cu cei mai multi participanti, afisandu-se descrierea  evenimentului si numarul de participanti |

**Planul de iteratii:**

**Iteratia 1:**

Generarea, stergerea si modificarea unei liste de persoane si evenimente

Adaugarea de persoane/evenimente in lista de persoane/evenimente

Afisarea listei de persoane/evenimente

**Iteratia 2:**

Cautarea de persoane si evenimente

persoane dupa nume

evenimente dupa descriere

**Iteratia 3:**

Inscrierea unei persoane la un eveniment

Afisarea listei de evenimente la care participa o persoana ordonata dupa descriere/dupa

Afisarea persoanelor care participa la cele mai multe evenimente

Afisarea primelor 20% evenimente cu cei mai multi participanti, afisandu-se descrierea

evenimentului si numarul de participanti

**Scenarii de rulare:**

**Functionalitatile 1,2,3:**

|  |  |  |
| --- | --- | --- |
| **user** | **program** | **descriere** |
| - | >>> | programul asteapta comanda |
| adauga\_persoana 1,Andrei,Mestecenilor | persoana adaugata cu succes! | programul a adaugat persoana cu:  ID = 1  Nume = Andrei  Adresa = Mestecenilor |
| adauga\_persoana,2,Stefan,Nufarului | persoana adaugata cu succes! | programul a adaugat persoana cu:  ID = 2  Nume = Stefan  Adresa = Nufarului |
| sterge\_persoana,2 | persoana stearsa cu succes! | programul elimina din lista persoana cu ID = 2 |
| adauga\_eveniment,1,15/11/2023,6,Bal | eveniment adaugat cu succes! | programul adauga evenimentul cu:  ID = 1  Data:15/11/2023  Durata:6 ore  Descrierea:Bal |
| sterge\_eveniment,1 | eveniment sters cu succes | programul sterge evenimentul cu ID = 1 |
| afiseaza\_lista\_de\_persoane | [1] Nume: Andrei, Adresa: Mestecenilor | programul afiseaza intreaga lista de persoane |
| afiseaza\_lista\_de\_evenimente | [1] Data: 15/11/2023, Durata: 6 ore, Descriere: Bal | programul afiseaza intreaga lista de evenimente |

**Functionalitatile 4,5,6:**

|  |  |  |
| --- | --- | --- |
| **user** | **program** | **descriere** |
| - | >>> | programul asteapta comanda |
| adauga\_persoana 1,Andrei,Mestecenilor | persoana adaugata cu succes! | programul a adaugat persoana cu:  ID = 1  Nume = Andrei  Adresa = Mestecenilor |
| adauga\_persoana,2,Stefan,Nufarului | persoana adaugata cu succes! | programul a adaugat persoana cu:  ID = 2  Nume = Stefan  Adresa = Nufarului |
| cauta\_persoana,Stefan | [2] Nume: Stefan,Adresa: Nufarului | programul cauta si afiseaza persoana cu Nume=Stefan |
| adaugare\_eveniment,1,13/11/2023,2,Concert | eveniment adaugat! | programul adauga evenimentul cu:  ID=1  Data=13/11/2023  Durata=2 ore  Descriere=Concert |
| inscriere 2,1 | persoana inscrisa! | programul inscrie persoana cu ID=2 la evenimentul cu ID=1 |
| adaugare\_eveniment,1,15/11/2023,6,  Bal | eveniment adaugat! | programul adauga evenimentul cu:  ID=1  Data=15/11/2023  Durata=6 ore  Descriere=Bal |
| afisare\_evenimente\_sortate\_dupa\_data | [1] Data: 13/11/2023, Durata: 6 ore, Descriere: Concert  [2] Data: 15/11/2023, Durata: 2 ore,  Descriere: Bal | programul afiseaza lista de evenimente sortata dupa data |
| afisare\_evenimente\_sortate\_dupa\_descriere | [2] Data: 15/11/2023, Durata: 2 ore,  Descriere: Bal  [1] Data: 13/11/2023, Durata: 6 ore,  Descriere: Concert | programul afiseaza lista de evenimente sortata dupa descriere |

**Functionalitatile 7,8:**

|  |  |  |
| --- | --- | --- |
| **user** | **program** | **descriere** |
| - | >>> | programul asteapta comanda |
| adauga\_persoana 1,Andrei,Mestecenilor | persoana adaugata cu succes! | programul a adaugat persoana cu:  ID = 1  Nume = Andrei  Adresa = Mestecenilor |
| adauga\_persoana,2,Stefan,Nufarului | persoana adaugata cu succes! | programul a adaugat persoana cu:  ID = 2  Nume = Stefan  Adresa = Nufarului |
| adaugare\_eveniment,1,13/11/2023,2,Concert | eveniment adaugat! | programul adauga evenimentul cu:  ID=1  Data=13/11/2023  Durata=6 ore  Descriere=Concert |
| adaugare\_eveniment,2,15/11/2023,2,Bal | eveniment adaugat! | programul adauga evenimentul cu:  ID=2  Data=15/11/2023  Durata=2 ore  Descriere=Bal |
| inscriere 1,1 | persoana inscrisa! | programul inscrie persoana cu ID=1 la evenimentul cu ID=1 |
| inscriere 1,2 | persoana inscrisa! | programul inscrie persoana cu ID=1 la evenimentul cu ID=2 |
| afisare\_persoane\_inscrise\_la\_cele\_mai\_  multe\_evenimente | [1] Nume: Andrei, Adresa: Mestecenilor | programul afiseaza persoana inscrisa la cele mai multe evenimente(in cazul acesta persoana cu ID=1 care e inscrisa la 2 evenimente |
| adaugare\_eveniment,3,17/11/2023,8,Nunta | eveniment adaugat! | programul adauga evenimentul cu:  ID=3  Data=17/11/2023  Durata=8 ore  Descriere=Nunta |
| adaugare\_eveniment,4,18/11/2023,2,Botez | eveniment adaugat! | programul adauga evenimentul cu:  ID=4  Data=18/11/2023  Durata=2 ore  Descriere=Botez |
| adaugare\_eveniment,5,19/11/2023,6,Majorat | eveniment adaugat! | programul adauga evenimentul cu:  ID=5  Data=19/11/2023  Durata=6 ore  Descriere=Majorat |
| inscriere 2,1 | persoana inscrisa! | programul inscrie persoana cu ID=2 la evenimentul cu ID=1 |
| afisare\_evenimente\_cu\_cei\_mai\_multi  \_participanti | [1] Data:13/11/2023, Durata: 2 ora  Descriere:Concert | programul afiseaza primele 20% evenimente cu cei mai multi participanti |
| afisare\_evenimente\_sortate\_dupa\_data,1 | 1,1/1/1,6,Bal  3,15/11/2022,7,Concert  2,15/10/2023,8 Nunta | programul afiseaza lista de evenimente la care este inscrisa persoana cu ID = 1 sortata dupa data |
| afisare\_evenimente\_sortate\_dupa\_descriere,1 | 1,1/1/1,6,Bal  3,15/11/2022,7,Concert  2,15/10/2023,8 Nunta | programul afiseaza lista de evenimente la care este inscrisa persoana cu ID = 1 sortata dupa descriere |

**Lista de taskuri:**

|  |  |
| --- | --- |
|  | **Task** |
| **T1** | **creare persoana**  creaza person\_id(input)  creaza person\_name(input)  creaza person\_adress |
| **T2** | **creare eveniment**  creaza event\_id(input)  creaza event\_date(input)  creaza event\_length(input)  creaza event\_description(input) |
| **T3** | **validare persoana**  valideaza person\_id  valideaza person\_name  valideaza person\_adress  cu PersonValidator |
| **T4** | **validare eveniment**  valideaza event\_id  valideaza event\_date  valideaza event\_length  valideaza event\_description  cu EventValidator |
| **T5** | **store persoana**  (create repository for persoana)  valideaza persoana  adauga persoana |
| **T6** | **store eveniment**  (create repository for eveniment)  valideaza eveniment  adauga eveniment |
| **T7** | **add persoana**  (create controller for persoana) |
| **T8** | **add eveniment**  (create controller for eveniment) |
| **T9** | **delete persoana**  input id persoana  sterge din repository |
| **T10** | **delete eveniment**  input id eveniment  sterge din repository |
| **T11** | **update persoana**  input nume persoana  input adresa persoana  validare  adaugare |
| **T12** | **update eveniment**  input data eveniment  input durata eveniment  input descriere eveniment  validare  adaugare |
| **T13** | **print persoane**  print \_\_personlist |
| **T14** | **print evenimente**  print \_\_eventlist |
| **T15** | **cauta persoana**  input nume  cautare id  search in \_\_personlist |
| **T16** | **cauta eveniment**  input descriere  cautare id  search in \_\_eventlist |
| **T17** | **inscriere persoana**  genereaza repo inscriere  genereaza controller inscriere  validare inscriere  adaugare in repo |
| **T18** | **print inscrieri**  print \_\_registerlist |

**Analiza complexitatii functiei get\_most\_registered(self, k):**

|  |  |
| --- | --- |
| **Functia** | **Analiza complexitatii functiei** |
| def get\_most\_registered(self, k):  *"""  afiseaza k persoane participante la cele mai multe evenimente  :param k: int  :return:  """* registers = self.\_\_repo\_register.get\_all\_registers()  nr\_participari = {}  for atendance in registers:  person\_id = registers[atendance].get\_registered\_person().get\_person\_ID()  if person\_id not in nr\_participari:  nr\_participari[person\_id] = 0  nr\_participari[person\_id] += 1  rez = []  for person\_id in nr\_participari:  rez.append(self.\_\_repo\_person.searchPersonID(person\_id))  if k > len(rez):  raise Exception("persoane inscrise insuficiente!")  else:  return rez[:k] | **n = lungimea listei de inscrieri**  **Caz favorabil:**  **(pentru cazul in care o singura persoana a fost inscrisa la vreun eveniment)**  Tetha(n)=n+n+1+n-k ≈ 3\*n  -crearea listei de inscrieri => n  -parcurgerea intregii liste de inscrieri cu fiecare operatie avand costul 1 => n  -parcurgerea liste de participari cu fiecare operatie avand costul 1, len(nr\_participari) = 1 => 1  **-**slice-ul listei = > n-k  **Caz defavorabil:**  **(fiecare persoana este inscrisa la un singur eveniment)**  Tetha(n)=n+3\*n+n+n-k ≈ 6\*n  -crearea listei de inscrieri => n  -parcurgerea intregii liste de inscrieri, fiecare operatie efectuandu-se cel putin o data => 3\*n  -parcurgerea listei nr\_participari avand lungimea n cu fiecare operatie avand costul 1=> n  -slice-ul listei =>n-k  **Cazul mediu:**  **(fiecare persoana participa la cel putin 2 evenimente)**  Tetha(n)=n+n/2+n+n/2+n-k ≈ 5\*n  -crearea intregii liste de inscrieri => n  -parcurgerea intregii liste de inscrieri cu ≈ n/2 executari ale instructiunii: nr\_participari[person\_id]=0  (cost = 1) =>n/2  n executari ale instructiunii:  nr\_participari[person\_id] += 1 (cost = 1) => n  -parcurgerea listei nr\_participari cu lungimea ≈ n/2 =>n/2  -slice-ul listei => n-k |