**Documentatie – P2 Gestionare Studenti&Probleme**

Luca David,214/2

**Enunt**

Scrieți o aplicație pentru gestiunea notelor și a problemelor de laborator pentru o disciplină.

Aplicația gestionează:

• studenți: <studentID>,<nume>,<grup>

• problemă laborator: <număr laborator\_număr problemă>,<descriere>, <deadline>

Creați o aplicație care permite:

• gestiunea listei de studenți și probleme de laborator.

• adaugă, șterge, modifică, lista de studenți, listă de probleme

• căutare student, căutare problemă

• Asignare laborator/Notare laborator

• Creare statistici:

• lista de studenți și notele lor la o problema de laborator dat, ordonat: alfabetic după nume,

după notă.

• Toți studenții cu media notelor de laborator mai mic decât 5. (nume student și notă)

**Functionalitati**

**Adăugare de studenti/probleme**

Adauga\_student/problema()  
• adaugă student(adauga\_student,id\_ul,numele,grupa) in lista ca o instanta a clasei Student()

• adaugă problema(adauga\_ problema,id\_ul,descrierea,deadline-ul) in lista ca o instanta a clasei Problema()

**Stergeri studenti/probleme**

Sterge\_student/problema()

• sterge un student(sterge un student dupa id ul lui)

• sterge o problema(sterge o problema dupa id ul ei)

**Căutări studenti/probleme**

**cautari()**

•Cauta un student dupa id si afiseaza pozitia sa din lista si caracteristicile acestuia(in caz ca acesta exista in lista,in caz contrar se afiseaza un mesaj corespunzator)

•Cauta o problema dupa id si afiseaza pozitia sa din lista si caracteristicile acesteia(in caz ca aceasta exista in lista,in caz contrar se afiseaza un mesaj corespunzator)

**Plan de iteratii**

Iteratia 1 :

• afisare studenti/probleme

• adaugă, șterge, modifică, lista de studenți, listă de probleme

• căutare student, căutare problemă

Iteratia 2:

• Asignare laborator/Notare laborator

Iteratie 3:

• Creare de statistici:

• lista de studenți și notele lor la o disciplină dată, ordonat: alfabetic după nume, după notă.

• Primi 20% din studenți ordonat dupa media notelor la toate disciplinele (nume și notă)

**Scenarii de rulare**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Scenarii pentru functionalitatea adauga\_student/problema()** | | | | |
|  | Utilizator | Program | Descriere | |
| 1 |  |  | Consideram studentii ( id, nume, grup) :  1, ’Andrei’,170  2, ’Adi’,173 | |
| 2 | **adaugare\_student()** | Introduce un nou student in lista | Asteapta introducerea unui id,nume si grupa pentru student | |
| 3 | 3,’Matei’,543 | 1, ’Andrei’,170  2, ’Adi’,173  3,’Matei’,543 | | Programul adauga studentul la lista de studenti |
|  | „date eronate” | Mesaj de avertizare | | Daca sunt introduse date eronate programul trimite mesaje corespunzatoare pentru a alerta user-ul |

|  |  |  |  |
| --- | --- | --- | --- |
| **Scenarii pentru functionalitatea modifica\_student/problema()** | | | |
|  | Utilizator | Program | Descriere |
| 1 |  |  | Consideram studentii ( id, nume, grup) :  1, ’Andrei’,170  2, ’Adi’,173 |
| 2 | Modificare\_cheltuiala | Modifica datele unui student | Programul un id si noile date ale studentului.Cauta studentul cu id ul respectiv si il actualizeaza |
| 3 | intrare | 1,”Matei”,24 | Functia primeste ca parametrii id\_ul,noul nume si noua grupa |
| 4 | iesire | 1,”Matei”,24  2, ’Adi’,173 | Programul actualizeaza studentul |
| 5 | intrare | 3,”David”,3 | Programul nu gaseste stundetul cu id ul 3 deoarece acesta nu se afota in lista |
| 6 | iesire | „student inexistent” | Programul afiseaza un mesaj corespunzator |

|  |  |  |  |
| --- | --- | --- | --- |
| **Scenarii pentru functionalitatea print\_student/problema()** | | | |
|  | Utilizator | Program | Descriere |
| 1 |  |  | Consideram studentii ( id, nume, grup) :  1, ’Andrei’,170  2, ’Adi’,173 |
| 2 | Print\_stundeti | 1, ’Andrei’,170  2, ’Adi’,173 | Afiseaza toti studentii existenti in lista |
|  | Print\_studenti(listsa goala) | „nu exista stundeti in lista” | Daca lista este goala se va afisa un mesaj corespunzator |

|  |  |  |  |
| --- | --- | --- | --- |
| **Scenarii pentru functionalitatea sterge\_student/problema()** | | | |
|  | Utilizator | Program | Descriere |
| 1 |  |  | Consideram studentii ( id, nume, grup) :  1, ’Andrei’,170  2, ’Adi’,173 |
| 2 | Sterge\_student | Sterge un student din lista | Asteapta introducerea unui id si va sterge studentul cu id ul respectiv din lista. |
| 3 | Intrare : 1 | Sterge studentul cu id ul 1 | Cauta studentul cu id ul egal cu cel introdus de user si il elimina din lista |
| 4 | iesire | 2, ’Adi’,173 | Lista noua va fi fara studentul sters |
| 5 | Intrare:5 | Cauta stundetul cu id ul 5 | Programul primeste un id inexistent |
| 6 | iesire | „student inexistent” | Cum nu poate gasi studentul,afiseaza un measj inexistent |

|  |  |  |  |
| --- | --- | --- | --- |
| **Scenarii pentru functionalitatea asigneaza\_problema()** | | | |
| Utilizator | Program | Descriere |
|  |  | Consideram studentii ( id, nume, grup) :  1, ’Andrei’,170  2, ’Adi’,173  Consideram problemele:  1,”da”,”1/1/2020 |
| Asigneaza\_problema()(fara probleme sau studeti) | Asigneaza unui student o problema | Asteapta introducerea unui id si a unei probleme pentru a asigna studentului respectiv acea problema |
| Asigneaza\_problema(1,1) | Nu exista studenti in lista  Nu exista probleme in listaa | Cauta studentul cu id ul introdus si problema introdusa |
| Asigneaza\_problema(1,2) | Problema inxistenta | Cauta datele introduse de user |
| Asigneaza\_problema(3,1) | Student inexistent | Cauta datele introduse de user |
| Asigneaza\_problema(1,1) | Andrei are de facut problema 1 | Asigneaza stundetului 1 problema 1 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Scenarii pentru functionalitatea asigneaza\_nota()** | | | |
| Utilizator | Program | Descriere |
|  |  | Consideram studentii ( id, nume, grup) :  1, ’Andrei’,170  2, ’Adi’,173  Consideram problemele:  1,”da”,”1/1/2020 |
| Asigneaza\_nota() | Nu exista asignari de probleme | Cauta asignarile de probleme existente |
| Asigneaza\_problema(1,1)  Asignare\_nota(1,2,3) | Problema inexistenta | Nu gaseste problema introdusa |
| Asigneaza\_problema(1,2)  Asignare\_nota(3,1,3) | Student Inexistetn | Nu gaseste stundetul introdus |
| Asigneaza\_problema(3,1)  Asignare\_nota(2,1,3) | Stundetul nu are asignara aceasta problema | Verifica daca datele introduse de user sunt corecte,in caz contrar afiseaza un mesaj corespunzator |
| Asigneaza\_problema(1,1)  Asignare\_nota(1,1,7) | Andrei a obtinut la problema 1 nota 7 | Verifica daca datele introduse de user sunt corecte,in caz contrar afiseaza un mesaj corespunzator |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Scenarii pentru functionalitatea ordonare\_nume()** | | | | |
|  | Utilizator | Program | Descriere | |
| 1 |  |  | Consideram studentii ( nume,nota) :  „Andrei”,7  „Matei”,5 | |
| 2 | **Ordonare\_nume()**  **\*nu sunt asignari** | „lista de asignari este goala” | Ordoneaza studentii in functie de nume alfabetic | |
| 3 | **Ordonare\_nota()**  **\*cu studentii** | „Andrei”,7  „Matei”,5 | | Programul stocheaza intr-o lista numele si nota iar apoi lista este ordonata in functie de primul parametru,acesta fiind numele |
|  | „date eronate” | Mesaj de avertizare | | Daca sunt introduse date eronate programul trimite mesaje corespunzatoare pentru a alerta user-ul |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Scenarii pentru functionalitatea ordonare\_nota()** | | | | |
|  | Utilizator | Program | Descriere | |
| 1 |  |  | Consideram studentii ( nume,nota) :  „Andrei”,7  „Matei”,5 | |
| 2 | **Ordonare\_nota()**  **\*nu sunt asignari** | „lista de asignari este goala” | Ordoneaza studentii in functie de nota | |
| 3 | **Ordonare\_nota()**  **\*cu studentii** | „Matei”,5  „Andrei”,7 | | Programul stocheaza intr-o lista numele si nota iar apoi lista este ordonata in functie de al doilea parametru,acesta fiind nota |
|  | „date eronate” | Mesaj de avertizare | | Daca sunt introduse date eronate programul trimite mesaje corespunzatoare pentru a alerta user-ul |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Scenarii pentru functionalitatea medie\_studenti()** | | | | |
| Utilizator | Program | Descriere | |
|  |  | Consideram studentii ( nume,nota) :  „Andrei”,7,5  „Matei”,5,2 | |
| **Medie\_studenti()**  **\*nu sunt asignari** | „lista de asignari este goala” | Afiseaza studentii si media lor(daca media este sub 5) | |
| **Medie\_studenti()**  **\*cu studentii** | „Andrei”,3.5 | | Programul stocheaza intr-o lista numele si media iar apoi verifica daca media este sub 5.In caz contrar nu afiseaza elevul respectiv |
| **Medie\_studenti()**  **\***  „Andrei”,7  „Matei”,5  **\*** | [] | | Niciunul dintre studenti nu are media sub 5 |
| „date eronate” | Mesaj de avertizare | | Daca sunt introduse date eronate programul trimite mesaje corespunzatoare pentru a alerta user-ul |

***Lista de taskuri***

* ***Adaugare***

|  |  |
| --- | --- |
| **T1** | ***Introducerea studentului /problemei dorit/e (cu ajutorul functiei adauga\_student/problema())*** |
| ***T2*** | ***Introducerea id\_ului,numelui,grupului*** |
| ***T3*** | ***Validarea datelor introduse de utilizator*** |

* ***Stergere***

|  |  |
| --- | --- |
| **T1** | ***Sterge un student/problema din lista dupa id\_ul acestuia/acesteia(cu ajutorul functiei sterge\_student/problema())*** |
| ***T2*** | ***Verficarea existentei id\_ului introdus de user*** |
| ***T3*** | ***In caz ca id\_ul nu este corect se afiseaza un mesaj corespunzator.Altfel,programul loop-eaza prin lista de obiecte eliminand pe cel cu id\_ul identic cu cel introdus de user*** |