Plan/ Documentatie laboratoare 7-9

# Cerinta

O imagine care conține text

Descriere generată automat

# Functionalitate necesara

1. **F1.** Adaugare, stergere, modificare a listelor de student si de probleme
2. **F2.** Cautare student si problema
3. **F3.** Asociere laborator/ nota corespunzatoare
4. **F4.** Creare de statistici:
   1. Lista de studenti si notele lor la o problema de laborator, ordonat alfabetic dupa nume sau dupa nota
   2. Toti studentii cu media notelor de laborator mai mic decat 5 (Nume student + nota)

# Arhitectura/ Principii urmarite

Urmarim implementarea proiectului sub forma OOP, prin utilizarea claselor si a obiectelor. Principiul de baza: “totul este un obiect”

Arhitectura este una stratificata, codul este impartit in:

* UI
* Service
* Domain
* Infrastructure/ Repository
* Testing
* Validation
* Errors

Fiecare strat are propriile seturi de responsabilitati

Urmarim principiile GRASP (General Responsibility Assignment Software Patterns/Principles):

* High Cohesion
* Low Coupling
* Information Expert
* Controller
* Protected Variations
* Creator
* Pure Fabrication

# Scenarii de rulare

**F1**

|  |  |  |
| --- | --- | --- |
| Intentie utilizator | Rezultat program | Descriere limbaj natural |
| Deschidere program | UI | Afisare mesaje corespunzatoare si meniu principal |
| add\_student | Creare obiect student | Este creat obiectul student cu acele atribute |
| modify\_student | Modificare obiect student | Obiectul de referinta este modificat |
| add\_problem | Adaugare obiect problema laborator | Obiectul problema laborator este create cu atributele corespunzatoare |
| remove\_student | Stergere obiect student corespunzator | Este sters obiectul de referinta din clasa student |
| add\_student | Creare obiect student | Este creat un nou obiect student cu attribute specificate |
| exit | Iesire din program | Incheiere program (cu salvarile de rigoare in fisiere) |

**F2**

|  |  |  |
| --- | --- | --- |
| Intentie utilizator | Rezultat program | Descriere limbaj natural |
| Deschidere program | UI | Afisare mesaje corespunzatoare si meniu principal |
| search\_student | Cautare obiect student dupa ID | Este cautat obiectul student corespunzator ID-ului introdus |
| search\_problem | Cautare obiect problema laborator dupa ID | Este cautat obiectul problema laborator dupa ID |
| exit | Iesire din program | Incheiere program |

**F3**

|  |  |  |
| --- | --- | --- |
| Intentie utilizator | Rezultat program | Descriere limbaj natural |
| Add\_assignment | Creare si stocare obiect assignment | Va fi creat un obiect assignment cu datele de intrare specificate, care va fi validat si in cazul corectitudinii va fi stocat in mediul de repozitoare |
| Grade\_assignment | Modificarea notei in obiectul cu ID specificat | Obiectul gasit dupa cautarea ID-ului va avea proprietatea de nota modificata, de la nota anterioara(eventual inexistenta) la cea noua |
| Remove\_assignment | Eliminarea din mediul de stocare al assignmentului | Assighnmentul cu ID specificat va fi eliminat din mediul de stocare, toate informatiile relevante vor fi sterse |
| Exit | Parasirea programului | Programul salveaza datele daca este cazul (file repo) si inchide instanta de rulare |

**F4**

|  |  |  |
| --- | --- | --- |
| Intentie utilizator | Rezultat program | Descriere limbaj natural |
| Stats\_order [name/grade] | Adunarea si afisarea statisticilor cerute | Programul aduna datele necesare si afiseaza o lista ordonata de studenti dupa nume sau nota |
| Stats\_mean | Modificarea notei in obiectul cu ID specificat | Programul aduna datele si afiseaza o lista cu toti studentii ale caror medie de note este sub 5 |

# Cazuri de testare

|  |  |  |
| --- | --- | --- |
| Date introduse | rezultat add\_student | Rezultat in limbaj natural |
| 1, Voinopol, 217 | obj\_student(1,”Voinopol”,217) | Este creat un obiect al clasei student cu atributele specificate |
| 1, Tot Voinopol, 217 | Exception: ID\_already\_exists | Eroare, ID existent |

|  |  |  |
| --- | --- | --- |
| Date introduse | rezultat modify | Rezultat in limbaj natural |
| 1, Voinopol, 217 | obj\_student(1,”Voinopol”,217) | Modificarea obiectului existent la ID-ul specificat cu noile atribute |
| 2, Tot Voinopol, 217 | Obj\_student(2,”Tot Voinopol”, 217) | Modificarea obiectului student cu ID=2, noile attribute sunt aplicate |

|  |  |  |
| --- | --- | --- |
| Date introduse | rezultat remove\_student | Rezultat in limbaj natural |
| 1 | Success message | Este sters obiectul din memorie si este afisat mesajul de success (in UI) |
| 2 | Success message | Este sters obiectul din memorie si este afisat mesajul de success (in UI) |

* Cazuri de testare analoage pentru obiectele de laborator

# Planificare

|  |  |
| --- | --- |
| **Iteratie** | **Functionalitati planificate** |
| I1 | F1 |
| I2 | F2, F3 |
| I3 | F4 |

**Iteratia 1:** 3.11.2021 – 10.11.2021

**7.11.2021:**

Stabilirea fundatiei codului

Gandirea functiilor de testare

Implementarea F1

**Iteratia 2:** 10.11.2021 – 17.11.2021

**13/14.11.2021:**

Implementarea F2 + testare

Implementarea F3 + testare

**Iteratia 3:** 17.11.2021 – 24.11.2021

**18/19.11.2021:**

Trecerea la repozitoare cu fisiere

**22.11.2021:**

Implementarea F4 + testare