

Metode Avansate de Programare, Seminar 11

Cerinte de proiectare:

1. Interfata **IRepository**<ID, T extends Entity<ID>> expune o singura metoda publica, **IEnumerable**<T> **findAll()**, care returneaza o colectie formata din toate entitatile de tip T.
2. Clasa **InMemoryRepository**<ID, E> implementeaza **IRepository** si foloseste un dictionar pentru stocarea datelor.
3. Clasa abstracta **InFileRepository**<ID, E> extinde **InMemoryRepository**.
4. Clasa **Angajat** are urmatoarele attribute:
 - a. *angajatId:String*
 - b. *nume:String*
 - c. *venitPeOra:double*
 - d. *nivel:Enum {Junior, Medium, Senior}*
5. Clasa **Sarcina** are urmatoarele attribute:
 - a. *sarcinaId:String*
 - b. *deficultate:Enum {Usoara, Medie, Grea}*
 - c. *nrOreEstimate:int*
6. Clasa **Pontaj** are urmatoarele attribute:
 - a. *angajat:Angajat*
 - b. *sarcina:Sarcina*
 - c. *data:DateTime*
7. Fisierele *angajati.txt*, *sarcini.txt*, *pontaje.txt* contin informatiile despre angajati, sarcini si pontaje, fiecare inregistrare pe o linie noua, attributele despărțite prin “,”. In fisierul *pontaje.txt* o linie va fi formata din:
idAngajat, idSarcina, data.
8. La nivel de service se va folosi interfața **IRepository** de mai sus.
9. Interacțiunea cu utilizatorul va avea loc numai in UI (Consola)
10. Interfața **IRepository** nu va expune alte metode decat cele specificate mai sus.
Clasele repository pot avea metode ajutatoare (*loadFromFile* etc)
11. La nivel de repository se pot folosi oricâte clase pentru implementarea interfeței **IRepository** si pentru interacțiunea cu fișierele;

Cerinte functionale:

1. Implementati **InMemoryRepository**.
2. Implementati **InFileRepository**.
3. Implementati **InFileRepository** pentru toate cele 3 entitati. Meniul pentru utilizator va avea cate o optiune pentru afisarea listelor celor 3 entitati.