

EXAMEN FINAL

JAVA 2

Cerinte minime de realizare a aplicatiei:

- 10p Construirea unei interfete grafice formata din trei ferestre/dialoguri pentru aplicatia de client
- 20p Realizarea unui sistem distribuit folosind RMI, care sa expuna doua service-uri
- 20p Definirea a trei entitati si a doua relatii intre ele de orice tip (1:1, 1:M, N:M) folosind JPA
- 10p Folosirea design pattern-ului arhitectural MVC
- 5p Implementarea a doua operatii CRUD (Create Read Update Delete) pentru fiecare dintre entitati
- 5p Modularizarea proiectului (module separate pentru aplicatia de server, de client si pentru elementele comune celor doua)
- 20p Scrierea unui cod curat:
 - respectarea principiilor SOLID
 - urmarirea standardelor de scriere a codului (identificatori sugestivi, metode scurte, evitarea pe cat se poate a folosirii valorii *null*, etc.)
 - IMPORTANT:** comentariile fac exceptie. Pot fi folosite oricate comentarii
 - evitarea codului duplicat
 - aplicarea design pattern-urilor acolo unde este nevoie

Bonusuri:

- +5p Optimizare de fetch-uri. Vor fi aduse din baza de date doar informatiile despre entitate care sunt de interes.
Exemplu: avand o entitate “departament” care are ca atribut o colectie de “angajati”, cand voi vrea sa afisez toate departamentele, nu voi aduce din baza de date si angajatii din fiecare departament, pentru ca nu sunt necesari la afisare
- +5p Doua folosiri ale unora dintre urmatoarele: *ElementCollection*, *cheie primara compusa*, *mosteniri intre entitati* (acelasi element poate fi folosit si de doua ori pentru a primi punctele bonus)
- +15p **1M\$ Feature:** realizarea unei functionalitati deosebite, care necesita implementarea unui mecanism mai complex
Exemplu 1: pentru o aplicatie ce gestioneaza livrarile care trebuie facute de o firma de curierat, un 1M\$ Feature ar fi ordonarea adreselor de livrare pentru o anumita zi astfel incat costul total de livrare sa fie minim.
IMPORTANT: va fi acordat un punctaj cel putin partial daca functionalitatea merge pe cazurile obisnuite. Punctajul total este dat daca implementarea merge inclusiv pe cazurile limita.
Exemple de cazuri limita pentru contextul de mai sus:
 - nu exista destinatii de livrare pentru ziua respectiva
 - indiferent de ordinea adreselor de livrare, costul minim este acelasi, etc.
Exemplu 2: o aplicatie distribuita de tip chat in care un client sa afle ca a primit un mesaj fara sa fie necesare interogari periodice catre server