



PROGRAMARE AVANSATĂ PE OBIECTE

Proiect (244)

Maria Cristina Chițu

Maria-Cristina.Chitu@endava.com

Fiecare student va lucra la un **proiect individual**. Proiectul este structurat în mai multe etape. Condiția de punctare a proiectelor este aceea ca acestea să nu prezinte erori de compilare, sa ruleze și să implementeze cerințele date.

Termene de predare:

- **Etapa I: 2 aprilie 2020**

- **Etapa II: 22 aprilie 2020**

- **Etapa III: 12 mai 2020**

Pentru cei de anul 3 (restanta), termenul pentru toate etapele: 30 aprilie 2020.

Codul scris trebuie să îndeplinească anumite standarde de claritate, concizie si lizibilitate.

Recomandăm coding style-ul:

- Google: <https://google.github.io/styleguide/javaguide.html> sau
- Oracle: <https://www.oracle.com/technetwork/java/codeconvtoc-136057.html>

Păstrați consistența coding style-ului pentru fiecare etapa (de exemplu nu folosiți într-o etapă două stiluri diferite de a pune parantezele). Vom aplica mici depunțări legate de calitatea codului.

Pentru informații suplimentare: Maria-Cristina.Chitu@endava.com

Etapa I:

1. Definirea sistemului:

- sa se creeze o lista pe baza temei alese cu cel puțin 10 actiuni/interogari care se pot face în cadrul sistemului si o lista cu cel puțin 8 tipuri de obiecte.

2. Implementare: sa se implementeze în limbajul Java o aplicatie pe baza celor definite la primul punct. Aplicatia va contine:

- clase simple cu attribute private / protected si metode de acces
- cel puțin doua colectii diferite capabile sa gestioneze obiectele definite anterior (List, Set, Map, etc.) dintre care cel puțin una sa fie sortata – se vor folosi array-uri uni- /bidimensionale in cazul in care nu se parcurg colectiile pana la data checkpoint-ului.
- utilizare mostenire pentru crearea de clase aditionale si utilizarea lor în cadrul colectiilor;
- cel puțin o clasa serviciu care sa expuna operatiile cu entitatile definite
- clasa main din care sunt facute apeluri catre servicii

Etapa II:

1. Extindeti proiectul din prima etapa prin realizarea persistentei utilizând fisiere.

Se vor realiza fisiere de tip csv (comma separated values => Fiecare coloana din fisier este separata de virgula. Exemplu: nume,prenume,varsta) pentru cel puțin 4 dintre clasele definite in prima etapa

- se vor realiza servicii singleton generice pentru scrierea si citirea din fisiere
- la pornirea programului se vor încarca datele din fisiere utilizând serviciile create

2. Realizarea unui serviciu de audit

- se va realiza un serviciu care sa scrie într-un fisier de tip CSV de fiecare data când este executata una dintre actiunile descrise in prima etapa. Structura fisierului: nume_actiune, timestamp_actiune

Etapa III:

1. Înlocuiti serviciile realizate în etapa a II-a cu servicii care sa asigure persistenta utilizând baza de date folosind JDBC.

- sa se realizeze servicii care sa expuna operatii de tip create, read, update, delete (CRUD) pentru cel puțin 4 dintre clasele definite
- se va implementa o functionalitate multithreading;

2. Sa se realizeze o interfata grafica în care sa fie expuse cel puțin 5 dintre actiunile definite initial. Interfata va avea cel puțin 2 ecrane diferite care sa permita navigarea între ele.

Se va utiliza Swing sau JSP pentru realizarea interfetei grafice. Daca doriti sa utilizati alte framework-uri, este necesara discutia în prealabil cu asistentul responsabil.

3. Se va adăuga în fișierul exportat de serviciul de audit încă o coloana: thread_name, reprezentand numele thread-ului care a apelat metoda.

Teme sugerate:

1. structura unei organizatii (angajati, relatii ierarhice, salarii)
2. o agenda personala (categorii, întâlniri, sarcini)
3. activitatea unei companii de transport (orase, legaturi, masini, rute)
4. credite (client, credit, rate)
5. cabinet medical (pacienti, medici, retete)
6. admitere (candidat, facultate, examen)
7. vanzare de bilete online(client, eveniment, locatie)
8. software casa de marcat(metoda de plata, client, produs)
9. rezervare loc în sala de spectacol (spectacol, loc, client)
10. activitatea unei case de schimb valutar (valute, istoric curs, clienti, tranzactii)
11. sistem de gestiune fisiere (utilizatori, tip fisier, grup, tip utilizator)
12. oricare alta sugestie este binevenita, cu mentiunea de a discuta cu asistentul in prealabil