

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 și să implementeze cerințele date.

Terme de predare:

- Etapa I: 29 martie
- Etapa II: 16 aprilie
- Etapa III: 10 mai

Pentru cei de la anul 3 (restanta), termenul pentru toate etapele: 16 aprilie.

Pentru orice informație suplimentară sau neclarități: adrian.buturuga@endava.com

Etapa I:

1. Definirea sistemului:
 - a. sa se creeze o lista pe baza temei alese cu cel puțin 10 acțiuni/interogări care se pot face în cadrul sistemului și o lista cu cel puțin 8 tipuri de obiecte.
2. Implementare: sa se implementeze în limbajul Java o aplicație pe baza celor definite la punctul. Aplicația va conține:
 - . clase simple cu attribute private / protected și metode de acces
 - a. cel puțin 2 colecții diferite capabile să 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.
 - b. utilizare moștenire pentru crearea de clase adiționale și utilizarea lor în cadrul colecțiilor;
 - c. cel puțin o clasa serviciu care sa expună operațiile
 - d. o clasa main din care sunt făcute apeluri către servicii

Etapa II:

Extindeți proiectul din prima etapa prin realizarea persistenței utilizând fișiere. Se vor realiza fișiere de tip csv ¹(comma separated values) pentru cel puțin 4 dintre clasele definite in prima etapa.

- se vor realiza servicii singleton generice pentru scrierea și citirea din fișiere
 - la pornirea programului se vor încărca datele din fișiere utilizând serviciile create
2. Realizarea unui serviciu de audit
 - se va realiza un serviciu care sa scrie într-un fișier de tip CSV de fiecare data când este executata una dintre acțiunile descrise in prima etapa. Structura fișierului:
nume_actiune, timestamp

Etapa III:

Înlocuiți 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 expună operații de tip create, read, update, delete pentru cel puțin 4 dintre clasele definite
- Sa se realizeze o interfață grafica în care sa fie expuse cel puțin 5 dintre acțiunile definite inițial. Interfața va avea cel puțin 2 ecrane diferite care sa permită navigarea între ele. Se va utiliza Swing sau JSP pentru realizarea interfeței grafice. Se pot utiliza si alte framework-uri, dar ar trebui discutate inainte, la laborator.
- Se va adăuga în fișierul exportat de serviciul de audit încă o coloana: thread_name, reprezentand numele thread-ului care a apelat metoda.

¹ Fiecare coloana din fisier este separata de virgula. Exemplu: nume,prenume,varsta

Teme sugerate:

1. structura unei organizații (angajați, relații ierarhice, salarii)
2. o agendă personală (categorii, întâlniri, sarcini)
3. activitatea unei companii de transport (orașe, legături, mașini, rute)
4. credite (client, credit, rate)
5. cabinet medical (pacienți, medici, rețete)
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, clienți, tranzacții)
11. sistem de gestiune fișiere (utilizatori, tip fisier, grup, tip utilizator)
12. _____