

Tema la Programarea Orientata pe Obiecte, anul 2, semestrul 1

Tema reprezinta implementarea unui Catalog electronic cu functionalitati minimale, in care se imbină mai multe sabloane de proiectare si abordari pentru interfata grafica pentru o functionalitate armonioasa si complexa.

Timp de lucru acordat: 26 dec 2022 – 13 ian 2023, cu maxim 3 zile pauza in preajma Anului Nou.

Nivel de dificultate: mediu spre mare. Personal consider ca partea cea mai grea pentru aceasta tema a fost intelegerea corecta a cerintei si parcurgerea cu atentie a enuntului, iar legat de partea de implementare, nu s-au folosit concepte de care sa nu fi auzit la laborator sau curs, cu cateva exceptii (de exemplu: Memento).

In continuare voi detalia cateva aspecte pentru o usoara intelegere a codului si a gandirii din spatele implementarii mele:

- **Interfata TupleInterface** -> folosita pentru a accesa datele din clasa interna, private Tuple, clasa care implementeaza aceasta metoda. In continuare, atat in clasa *ScoreVisitor* cat si in clasa *Test*, am folosit aceasta interfata ca argument.
- **ScoreVisitor** -> am folosit in plus 4 metode, 2 pentru introducerea in dictionarele specifice notelor din examen si din partial, si 2 pentru obtinerea acestora (metodele get).
- **Interfata Strategy** -> am introdus intentul unei metode comune, pe care o voi folosi in clasele *BestExamScore*, *BestPartialScore*, *BestTotalScore*, iar aceasta interfata este implementata de clasa *Course* si prin intermediul metodei getBestStudent, afiseaza, in functie de strategia aleasa de profesorul titular, studentul dorit.
- **Interfata Subject** -> implementata de clasa *Catalog* care ii mosteneste si, mai departe, ii implementeaza metodele, intrucat catalogul reprezinta punctul de interes pentru observatori.
- **Interfata Observer** -> implementata de clasa *Parent* care ii mosteneste si, mai departe, implementeaza metodele, deoarece parintii sunt aceia care urmaresc parcursul academic al copiilor lor (studentii).
- **StudentPage, TeacherPage, ParentPage** - > paginile pentru student, profesor/asistent si, respective, parinti, care respecta (din punctul meu de vedere) cerintele enuntului si au avut un grad de dificultate mediu spre scazut, sunt implementate aceleasi metode invatate si exersate la laborator.
- **LoginPage** -> exercitiul bonus, pe care l-am gandit astfel:
 - Un *Student*, va folosi numele sau (prenume si nume) pentru Username, si “student” pe post de parola. Astfel, programul va citi din campul Password tipul fiecarui utilizator (student/teacher/parent/asistent), va prelua numele si va afisa pagina specifica fiecaruia.
 - Analog pentru profesori, asistenti si parinti.
- **Studentlist** (*TreeSet*), **parents** (*HashSet*), **assistantslist** (*Arraylist*), **teacherlist** (*Arraylist*) -> folosite pentru a stoca lista cu toti studentii, parintii, asistentii etc, fara duplicate, pentru a fi mai usor accesibile obiectele de tipurile respective.

Restul claselor, sunt, din punctul meu de vedere, conforme cerintei temei, iar prin intermediul comentariilor pe cod, parcurgerea lor ar trebui sa fie usurata.

FeedBack: Prima tema de pana acum care sa aiba o functionalitate practica, careia sa-i inteleg atat scopul, cat si conceptele pe care le subliniaza.