

README

POO - TEMA 1

Nu am cronometrat exact cat timp mi-a luat tema, dar am inceput tema pe 27 decembrie si am lucrat la ea pana pe 4 ianuarie (cand am terminat tot inafara de partea de testare), deci in jur de 6 zile. Partea de testare mi-a luat cam 2 ore sa fac parser-ul.

Ca dificultate nu mi s-a parut foarte grea (cred ca in general nu mi se pare greu sa scriu cod in Java, fiind un limbaj mai prietenos decat C), dar am avut probleme in a intelege cerinta deoarece unele subpuncte nu aveau sens intuitiv in contextul unui catalog. Sablonul de proiectare Visitor din cadrul temei mi-a cauzat cele mai multe probleme, chiar si acum mi se pare destul de ciudat gandit. De asemenea, Swing-ul m-a doborat de cateva ori, am pierdut 2 ore incercand sa faca o lista scrollable sa functioneze doar ca aflu ca metoda add a clasei JScrollPane nu face o componenta scrollable, ci setViewportView :(

Pentru implementare o sa incerc sa discut doar partile mai speciale si nu cele unde doar am facut ce a zis cerinta fara sa fie nevoie sa gandesc prea mult. In general, am incercat sa respect principiile programarii orientate pe obiecte, cum ar fi incapsularea.

Am inceput cu o implementare Singleton cu lazy instatiation a clasei Catalog, clasele de baza ale programului si metodele simple.

Am creat o clasa ce urmeaza sablonul de proiectare Factory pentru a facilita crearea de noi useri, aceasta primeste tipul user-ului si numele pentru a returna obiectul User cerut. Pentru cele 2 tipuri de cursuri, am folosit un builder care permite folosirea unor parametrii optionali in constructia de cursuri (asistenti, grupuri, strategie, note) si obliga selectarea unui nume, profesor si numar de credite.

In implementarea mea de Observer, parintii sunt observeri si catalogul este subiectul. Deci de fiecare data cand o nota este adaugata in catalog, parintii copilului caruia ii apartine nota primesc o notificare daca sunt abonati la catalog. Aceste notificari sunt pastrate intr-o colectie pentru a fi afisate in interfata grafica.

Pentru strategiile de calcul a celui mai bun student, am adaugat un parametru la clasa Course de tipul BestStudentStrategy unde se va pune un obiect ce reprezinta strategia aleasa. De asemenea, am extins sablonul de builder pentru a oferi si optiunea de strategie.

Am implementat sablonul Visitor alegand clasele Teacher si Assistant ca elemente vizitabile. Catalogul are acum un parametru ScoreVisitor ce poate fi folosit pentru a exporta note usor de catre asistenti si profesori. Un teacher/assistant trebuie doar sa apeleze metoda accept pentru a muta toate notele din vizitator in catalog.

Pentru mecanismul de Memento, toate notele sunt clonate si salvate intr-o colectie ce poate fi accesata mai tarziu pentru a reface statusul precedent al notelor. Acest backup devine astfel un parametru al clasei Course.

Pentru testare am folosit fisierul JSON oferit si l-am parsat uzitand biblioteca json.simple. Aceasta parsare, alaturi de interfata grafica, verifica majoritatea functiilor catalogului, restul am testat manual (memento).

Interfata grafica implementata este una simplista, pentru a vizualiza pagina unui utilizator trebuie scris numele in TextField-ul din partea de sus a interfetei si apasat unul dintre butoanele oferite bazat pe tipul user-ului (Student/Teacher/Assistant/Parent).

Pentru un student, va aparea o lista scrollable cu cursurile sale. La selectarea unui curs vor aparea informatii despre curs (Profesor, asistent, note si lista de asistenti).

Pentru un profesor sau asistent, vor aparea 2 liste, una cu toate cursurile la care acesta predă si alta cu toate notele ce nu sunt inca validate, si un buton ce valideaza toate notele nevalidate.

Pentru un parinte va aparea doar o lista cu notificari (daca acesta este abonat, in caz contrar el nu va fi gasit).

Daca numele dat nu apare in catalog, se va afisa un mesaj ce informeaza utilizatorul de acest fapt.

Consider ca sunt eligibil pentru bonusul de sistem de autentificare, avand o interfata ce permite afisarea unui continut personalizat in functie de tipul de utilizator.

Observatie: am folosit IDE-ul JetBrains IntelliJIDEA pentru dezvoltarea integrala a temei

Cam atat in legatura cu implementarea. Sper ca tema a iesit ok :)