

TEMA_POO README

Petcu Andrei 322CC

Grad de dificultate: 6/10, tema nu mi s-a părut neapărat dificilă, doar migăloasă și mult de implementat

Timpul alocat rezolvării: aproximativ 20h

Obs: Design pattern-urile Visitor și Observer nu au fost implementate, prin urmare nici funcționalitățile asociate din interfața grafică.

Implementarea temei a fost realizată urmând pașii și indicațiile din enunțul temei. Astfel, voi detalia în principal funcționalitățile și aportul adus de mine:

- Clasele User, Student, Parent, Assistant, Teacher sunt create folosind o clasă UserFactory printr-o metodă ce primește pe lângă nume, prenume, și tipul user-ului (un String ex: „Student”, „Teacher” etc.).
- Comparațiile între obiecte Student au fost făcute folosind toString (dacă 2 elevi au nume identic, consideram că ne referim la același student).
- Clasa Group am ales să moștenească un TreeSet de Student (colecție ordonată).
- Pentru clasa Course am respectat design pattern-ul Builder, Strategy și Memento. Pentru Builder câmpul de nume a fost considerat obligatoriu, iar restul opțional. Pe lângă metodele impuse, am implementat funcții getter după caz și getStudentAssistant care caută un student în grupele asociate cursului și întoarce asistentul corespunzător grupei studentului.

Interfața grafică implementată este compusă din mai multe pagini:

- Prima pagină este cea de LogIn, unde trebuie introdus rolul user-ului (Student sau Teacher) și numele complet al acestuia. Am parcurt lista de studenți sau profesorii după caz și am deschis o nouă pagină cu informații sau cu un mesaj de Fail în cazul în care user-ul nu există.
- Pagina pentru student prezintă cursurile la care este înscris user-ul și oferă notele, dar și informații despre curs: profesor, asistenți, cel mai bun student (respectând strategia cursului).
- Pagina pentru Profesor afișează cursurile la care profesorul predă și 2 liste cu studenți. Una completă, iar cea de a doua doar cu studenții promovați (respectând tipul cursului Full/PartialCourse).

Citirea tuturor datelor se realizează din fișierul JSON pus la dispoziție pe Moodle, prin parsarea fiecărui tip de „obiect” (course, student, examScore etc.). Datele sunt apoi introduse în Catalog (unic – Singleton Design Pattern) și afișate în GUI. În plus, în clasa Test se află o testare a funcționalității backup-ului.