Băluță Teodora-Valentina

322CC

Grad de dificultate: mediu

Timp alocat: 1 săptămană

Clasa User este o clasă abstractă ce conține câmpurile firstName și lastName, cât și o metodă toString care returnează numele întreg.

Clasa Parent moștenește clasa User și conține câmpuri referitoare la mama, respectiv tatăl studentului.

Clasa Student moștenește clasa User și conține câmpuri ce stochează notele la parțial, examen și nota finală. De asemenea aici sunt setați părinții studentului.

Clasa Assistant moștenește clasa User și reține numele asistenților.

Clasa Teacher moștenește clasa User și reține numele profesorilor.

Clasa UserFactory este o clasă care are metoda getUser() și returnează un obiect de tip User.

Clasa Grade este o clasă ce implementează interfețele Comparable și Cloneable. Aici sunt setate notele la parțial, examen, studentul, cursul.

Funcția compareTo compară nota totală (parțial + examen) dintre studentul curent și alt student. Funcția getTotal calculează nota totală (parțial + examen).

Clasa Group este o clasă ce modelează o colecție de obiecte de tipul Student. Fiecare grup are un asistent asociat și un ID setați. Clasa conține și două funcții de adăugare (addStudent) și ștergere (removeStudent) a studenților.

Clasa Course setează un nume, profesor, numărul de puncte credite, grupa, asistenții. Pentru adăugarea studenților și a asistenților am procedat similar, mai întâi am verificat dacă lista e goală sau nu și apoi am adăugat. La fel și pentru note (addGrade). De asemenea, am implementat și o clasă internă abstractă CourseBuilder ce a fost extinsă și în următoarele două clase.

Clasa PartialCourse moștentește clasa Course. Am calculat nota pentru fiecare student. Dacă aceasta era mai mare sau egală cu 5, atunci aceștia erau promovați.

Clasa FullCourse moștenește clasa Course. Aplicând o metodă asemănătoare cu cea de mai sus, am verificat ca pentru a fi promovați studenții trebuiau să aibă nota în parțial mai mare sau egală cu 3, iar cea din examen mai mare sau egală cu 2.

Clasa Catalog conține obiecte de tip Course. Procedând similar ca mai sus, aici adăugăm obiecte și cursuri în catalog.

Șablonul de proiectare Observer

Am creat două interfețe, **Observer** și **Subject**, cu ajutorul cărora vom anunța părinții unui student dacă acesta a luat o notă. Pentru a face asta, am creat o clasă **Notification,** care primește o

notă. În clasa Catalog sunt adăugate metodele de adăugare și ștergere ale unui observer, adică părinte. Funcția notifyObserver setează pentru fiecare părinte să primească notificare cu nota. În clasa Parent găsim metoda update care are scopul de a adăuga o notificare.

Șablonul de proiectare Strategy

Pentru ca la finalul semestrului să fie ales cel mai bun student, vom implementa trei clase, **BestPartialScore, BestExamScore și BestTotalScore** și o interfață **BestStudentStrategy.** Clasa BestPartialScore folosește o metodă numită getBestStudent ce primește ca parametru notele. Contorul bestScore stochează cea mai mare notă, iar bestStudent reține studentul respectiv, până la parcurgerea tuturor notelor. Se returnează studentul cu cea mai mare notă. Clasele BestExamScore și BestTotalScore procedează la fel. Clasa Course implementează interfața BestStudentStrategy.

Pentru testare am folosit clasa Test. În această clasă am creat trei cursuri la care participă patru studenți din aceeași grupă. Am testat BestStudentStrategy la toate cursurile pentru a afla care sunt cei mai buni studenți. Am testat și notificare pentru un singur părinte.