

TEMA POO: CATALOG

1. Clasa Catalog:

In clasa catalog avem un arraylist de courses, iar in catalogul este desginul singleton pentru a putea fi initializatb doar o data cu metoda get instance , in rest doar fiind returnat. Am 2 metode : addCourse si removeCourse care fac ce zic metodele.

De asemenea Catalog implementeaza interfata Subject (care este folosita in pt design paternul observer) si are de implementat metodele addObserver (care baga in arraylistul de observers un nou observer), remove care face inversul si notifyObservers care da update la unn nou grade folosind metoda din interfata observers care este implementata in parent pt a o afisa.

2. Clasele User, UserFactory,Student,Parent,Teacher,Assistant:

Toate clasele Student,Parent,Teacher,Assistant mostenesc User si au constructor pentru first name si last name. Ele se initializeaza de fapt folosind user Factory, iar din UserFactory metoda getUser, care are un parametru tip , iar in el se pune tipul de clasa (Student etc..) si ii apeleaza constructorul.

De asemenea, student mai are si 2 campuri de tip parinte father si mother, pe care le poate seta folosind settere.

3. Clasa Grade:

Are campurile partialScore, examScore de tip double si student de tip Student si un course de tip string care e numele cursului. S-au creat settere si gettere pt toate campurile si la final o metoda getTotal care returneaza suma dintre examScore si partialScore.

De asemenea clasa grade implementeaza 2 interfate, Clonable si Comparable, iar la comparable am scris functia de compare to(care compara in functie de cea mai mare nota) , iar la clonable cea de clone.

4. Clasa Group:

Are un assistant un id si un comparator de tip student. Are 2 constructori, unul pt toate 3 campurile, unul doar pt primele doua campuri, iar la constructorul cu comparator se scrie metoda de compare intre 2 studenti in functie de numele de familie prima dat apoi fata de prenume.

Altfel clasa are doar settere si gettere.

5. Clasa Course:

Clasa Course are un nume de tip string o strategie de tip string un Teacher de tip Teacher, un Hashset de tip assistants, un arraylist de grade-uri,un HashMap cu cheia un string si valoarea un obiect de tip group. Pentru constructor se va folosi design paternul BUILDER, iar pentru asta o sa am o clasa interna abstracta CourseBuilder, care va initializa toate campurile din Course ca unele de tip CourseBuilder, iar la final am functia de build tot abstracta, doarece build-ul se va face in functie de tipul cursului, "FullCourse" sau "PartialCourse". Dev aceea avem aceste 2 clase care extend Course si

au o clasa interna PartialCourseBuilder/FullCourseBuilder care extinde CourseBuilder si face metoda de build.

De asemenea aceste 2 clase implementeaza metoda abstracta din Course getGraduatedStudents, in functie de ce tip sunt. La PartialCourse trebuia sa aiba totalul gradeului mai mare decat 5, iar la FullCourse trebuie sa aiba partialScore-ul mai mare decat 3 si examScore-ul mai mare decat 2.

Clasa Course are toti setter si getteri posibili si cateva metode.

-addAssistant seteaza asistentul în grupa cu ID-ul indicat. Daca nu exista deja, adauga asistentul si în multimea asistentilor

-addStudent Aadauga studentul în grupa cu ID-ul indicat

-addGroup Aadauga grupa

-addGroup Instantiaza o grupa si o adauga

-getGrade: Returneaza nota unui student sau null

-addGrade: Aadauga o nota

-getAllStudents : Returneaza o lista cu toti studentii

-getAllStudentGrades : Returneaza un dictionar cu situatia studentilor

6. Clasa Test:

In prima parte clasa test ia din fisierul catalog.json si baga in catalog. In a doua parte ia din fisierul json si adauga in hashmap-uri aferente clasei ScoreVisitor si apoi in acea clasa se baga din hashmap in catalog, iar la final am implementat niste testuri pentru a trece prin toate metodele facute.

7. StudentPage1:

In aceasta clasa am facut interfata grafica pentru prima pagina, cea de student.

Cand apelez clasa in Test o fac cu un constructor StudentPage1 in care este si un string cu numele studentului la care vreau sa vad la ce cursuri e inscris, etc..

Am luat stringul si l-am impartit in doua dupa spatiu, acum am firstName si lastName. Implementez un Jlist pentru cursuri si o data ce apas pe curs o sa implementez metoda valueChanged in care o sa mi afiseze numele profului de curs, numele asistentilor asignati cursului, ce asistent are studentul, si partialScore si examScore-ul studentului.

