Dumitru Radu Andrei
Grupa 264
Grupa 4 EAP

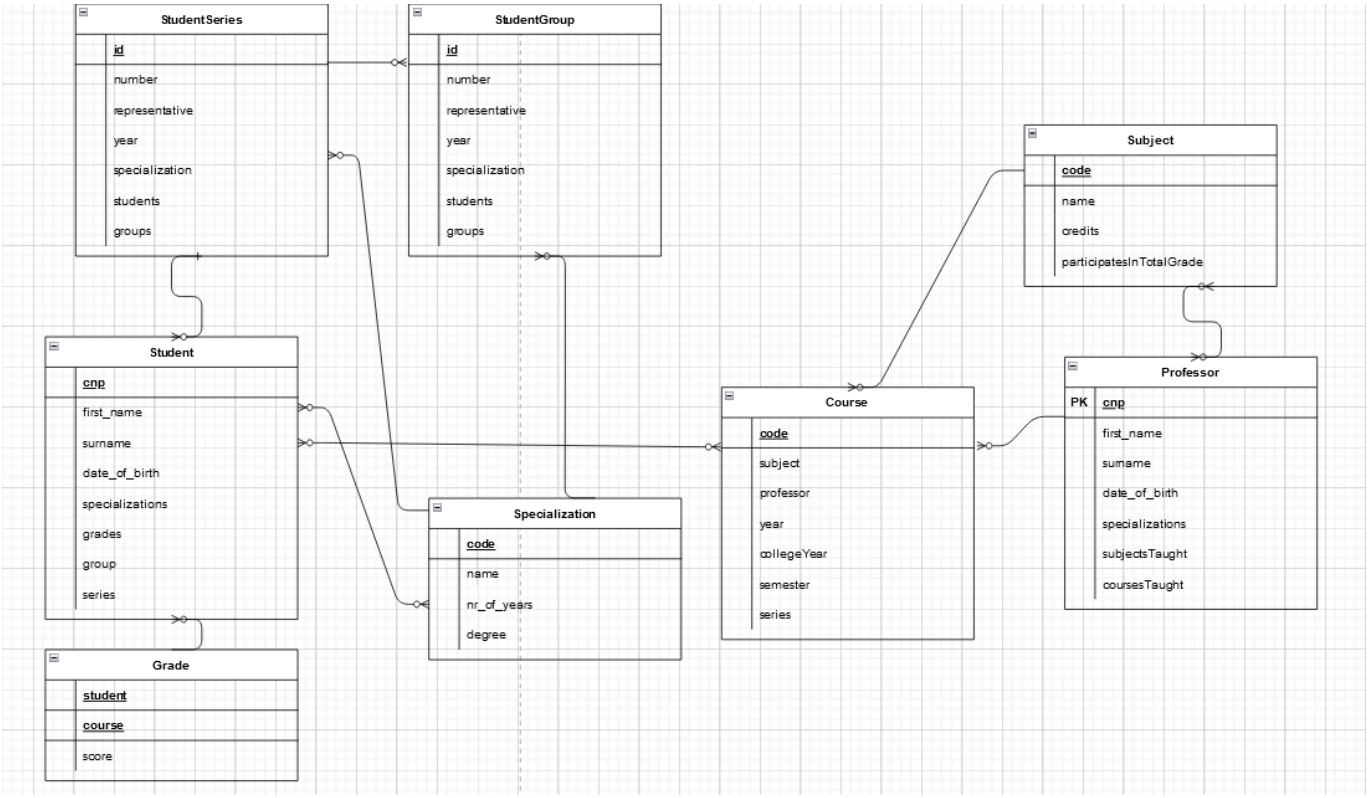
Proiect EAP
Catalog Online

Acest proiect reprezinta un catalog online al unei facultati, cu scopul de a gestiona informatii referitoare la personalul unei facultati, atat studenti cat si profesori, activitatile acestora, si notele detinute de studenti. Datele aferente sunt extrase dintr-un set de fisiere CSV, fiind apoi gestionate mai departe prin consola, fiecare actiune fiind salvata intr-un fisier CSV de audit.

Proiectul functioneaza pe baza unui model format din 8 clase principale, situate in folder-ul src/model:

- Specialization: reprezinta un profil de specializare ce poate fi urmat de studenti, precum Matematica sau Informatica
- StudentSeries: serie de studenti, impartita in mai multe grupe
- StudentGroup: grupa de studenti, ce face parte dintr-o serie
- Student: student inscris la facultate
- Professor: profesor al facultatii, care tine anumite cursuri
- Subject: Materie predata, ce reprezinta obiectul unui curs (de exemplu, Elemente Avansate de Programare)
- Course: Reprezinta un curs predat de un profesor, pentru o anumita materie, unde studentii pot avea o nota
- Grade: reprezinta o nota unui student la un anumit curs

Dumitru Radu Andrei
Grupa 264
Grupa 4 EAP



Deoarece modelul a fost gandit astfel incat un student sa poate avea o singura nota la un anume curs (doar nota finala), am definit si o clasa secundara Attendance, ce reprezinta participarea unui student la un anume curs: instantele Attendance vor fi folosite ca si chei intr-un TreeMap ce va stoca notele ca valori, pentru a reda faptul ca un student la un curs poate avea o singura nota.

Pentru a putea folosi aceste clase in colectii TreeMap, am implementat unde a fost cazul interfata Comparable, redefinind metode compareTo(), si am redefinit metodele equals() si hashcode(), pentru a se putea efectua sortarea.

Pe langa aceste clase principale din model, am utilizat in implementarea acestora si doua clase abstracte, care sunt extinse:

- Person: este extinsa de Student si Professor, pentru a retine date personale de baza precum nume si CNP
- StudentUnit, extinsa de StudentGroup si StudentSeries, punand in comun date precum numele, colectia de studenti si specializarea

In folderul src/service am creat mai multe clase ce reprezinta anumite servicii:

- CSV_Reader si CSV_Writer, clase singleton ce asigura citirea datelor din fisier, respectiv adaugarea datelor de audit in fisierul data/data.csv.
- Auditer: are rolul de a prelucra o actiune auditata pentru a fi scrisa de CSV_Writer.
- Service: clasa utilitara ce expune cele 10 servicii principale ale proiectului
- AuxService: clasa utilitara ce cuprinde mai multe metode auxiliare statice, care vor fi apelate de serviciile principale.

Dumitru Radu Andrei
Grupa 264
Grupa 4 EAP

Printre serviciile auxiliare se afla metode de adaugare pentru fiecare din clasele modelului, existand polimorfism prin supraincarcarea acestor metode: aceste metode de adaugare accepta fie niciun parametru (pentru citirea datelor obiectului de la consola), fie mai multe stringuri (pentru datele obiectului trimise sub forma de stringuri), fie datele obiectului sub tipul lor actual (de exemplu pentru situatiile cand sunt preluate din alte obiecte), fie un obiect initializat. Cand una dintre aceste metode este apelata, va apela si metodele de dupa aceasta in ordinea descrisa.

In clasele folosite am implementat un serviciu de audit, ce va scrie in fisierul data/audit.csv fiecare actiune efectuata. In implementarea acestui serviciu am folosit doua interfete, cu locatia in fisierul data/interfaces:

- Auditable: interfata implementata de toate clasele auditate, in afara de cele utilitare. Contine metoda audit() ce este redefinita in fiecare clasa pentru a cuprinde actiunile posibile efectuate de acea clasa, ce pot fi auditate. Metoda va trimite una dintre aceste actiuni pentru a fi scrisa in fisierul de audit, impreuna cu un timestamp. Metoda are un comportament default, in cazul in care nu este redefinita metoda de audit intr-o clasa ce o implementeaza, si anume de a transmite mesajul ca serviciul de audit nu este implementat pentru acea clasa.
- AuditableUtil: interfata implementata in clasele AuxService si Service, deoarece contin metode statice care nu pot apela metoda audit() din Auditable: astfel, va contine o varianta statica a metodei audit(), auditStatic()

Pentru stocarea datelor am utilizat mai multe implementari de colectii Map, respectiv Set:

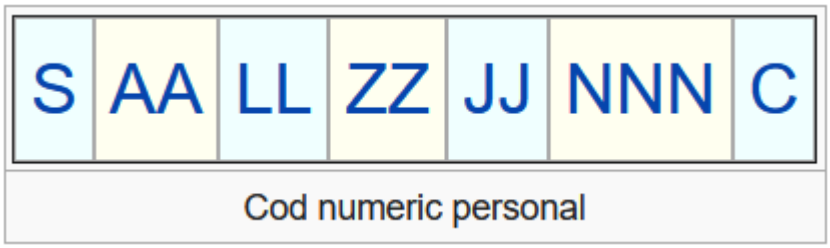
- HashMap: pentru a stoca toti elevii, respectiv toti profesorii, in clasa AuxService, folosind CNP-urile acestora ca si chei. Am folosit CNP-ul ca si cheie, deoarece nu pot exista mai multe persoane cu acelasi CNP. Am stocat persoanele intr-un HashMap si nu intr-un TreeMap, deoarece intr-un TreeMap ar fi ordonate dupa CNP, ceea ce nu ar fi relevant in acest caz, asadar ordinea stocarii poate fi aleatorie.
- TreeMap: pentru a stoca celelate obiecte incarcate din fisiere, obiectele fiecarei clase fiind stocate in cate un TreeMap corespunzator in clasa AuxService. Fiecare dintre aceste clase are cate o data membra ce o identifica un mod unic, astfel putand fi folosita ca si cheie. In acest caz am folosit cate un TreeMap, deoarece in cazul unei parcurgeri a colectiei, obiectele nu vor mai trebui sortate, fiind deja gata sortate.
- TreeSet: pentru a stoca studentii unei serie sau grupe. In acest caz am folosit o implementare de Set deoarece nu pot exista studenti duplicati, alegand TreeSet in mod similar pentru a nu mai sorta setul la momentul parcurgerii colectiei.
- LinkedHashSet: pentru a stoca specializarile elevilor, respectiv materiile si cursurile predate de profesori. In acest caz am ales sa le stochez intr-un LinkedHashSet din motive similare, pentru a nu exista duplicate, dar si pentru ca obiectele vor fi stocate in ordinea inserarii.

Am folosit implementari map in situatiile in care obiectele puteau fi identificate unic dupa o anumita data membra: astfel, daca se adauga un alt obiect cu acelasi identificator, primul va fi suprascris, asadar aceasta adaugare s-ar rezuma la o editare a valorii cheii.

Dumitru Radu Andrei
Grupa 264
Grupa 4 EAP

Pentru a acoperi situatii neprevazute, am definit mai multe exceptii in folder-ul src/exceptions:

- CNP_Exception, va fi aruncata in cazul in care un CNP nu respecta structura legala a unui CNP: https://ro.wikipedia.org/wiki/Cod_numeric_personal_(Rom%C3%A2nia



Cifra S trebuie situata intre 0 si 8, cifrele LL trebuie situate intre 1-12, reprezentand o luna, cifrele ZZ trebuie sa reprezinte o zi continuta in luna respectiva, iar cifrele JJ, ce denota un judet, trebuie sa se afle intre 1 si 52, excluzand 49 si 50.

- InvalidGradeException: va fi aruncata atunci cand scorul unei note nu se afla intre 1 si 10

Aceste exceptii aruncate sunt tratate in constructori, astfel incat obiectul respectiv sa nu mai fie initializat.

Pentru a ajuta in implementarea proiectului si pentru consistenta, am folosit si mai multe enum-uri, definite in folder-ul src/enums:

- AuditOption: contine toate actiunile ce pot fi auditate, impreuna cu cate o descriere
- AuditUtilClass: contine numele claselor ce pot fi auditate de metoda auditUtil()
- Degree: contine datele despre formele de invatamant ce pot corespunde unei specializari: licenta, master, doctorat
- ObjectRead: contine fiecare clasa pentru care pot fi citite obiecte din fisiere, impreuna cu filepath-ul corespunzator, pentru parcurgerea in metoda de citire a clasei CSV_Reader
- Semester: semestrele unui an, semestrul 1 sau 2
- Year: anii de studiu in functie de forma de invatamant: B1-4 pentru licenta, M1-2 pentru master, P1-2 pentru doctorat

Serviciile definite, ce vor fi apelate in clasa main, sunt urmatoarele:

- Populare date: specializari, materii, studenti, profesori, materii, cursuri, note
- Adaugare student
- Listare informatii despre un student

Dumitru Radu Andrei
Grupa 264
Grupa 4 EAP

```
Adding student:
First name:
Andrei
Surname:
Daniel
Date of birth (dd/mm/yyyy):
19/08/2002
CNP:
5010518390072
Group:
2020-264
Listing data for student:
Enter first name:
Andrei
Enter surname:
Daniel

Student:
    first_name='Andrei'
    surname='Daniel'
    date_of_birth=Mon Aug 19 00:00:00 EEST 2002
    CNP=5010518390072
    specializations={CTI}
    grades=
    group=2020-264
    series=2020-26
```

- Setare reprezentant de grupa/serie

```
Set representative:
Series/Group code:
2020-26
Enter first name:
Andrei
Enter surname:
Daniel
Representative Daniel Andrei set successfully to 2020-26
```

- Listare studenti din grupa/serie

Dumitru Radu Andrei
Grupa 264
Grupa 4 EAP

```
Listing students in group/series:
Series/Group code:
2020-26
Listing students:
    Matache Ioana, 261
    Popa Ion, 261
    Daniel Andrei, 262
    Ionescu Andrei, 262
Representative: Daniel Andrei
```

- Adaugare curs predat unui profesor

```
Adding course taught to professor:
Professor:
Enter first name:
Silviu
Enter surname:
Traian
Course Code:
EAP-2020-1
Course EAP-2020-1 added succesfully to Traian Silviu
```

- Adaugare/editare nota pentru un student la un curs

```
Adding grade to student:
Student:
Enter first name:
Andrei
Enter surname:
Daniel
Course code:
Course Code:
EAP-2020-1
Grade:
10
Grade added succesfully!
```

Dumitru Radu Andrei
Grupa 264
Grupa 4 EAP

- Afisare medie student pe un an de studiu (se calculeaza media ponderata in functie de numarul de credite al fiecarei materii)

```
Average grade of student:
Student:
Enter first name:
Ion
Enter surname:
Popa
College year to calculate average for:
College year (B1-4,M1-2,P1-2):
B2
Average for student Popa Ion, year B2: 8.05
```

- Afisare medie pentru o anumita serie (media dintre mediile fiecarui student cu minim o nota)

```
Average grade of series:
Series:
2020-26
Average grade of series 2020-26 : 9.02
```

- Listare cursuri unde a fost predata o anumita materie

```
Listing courses of subject:
Subject code:
EAP
Courses for Elemente Avansate de Programare (EAP):
     EAP-2020-1 Year 2020 College year B2 Semester II
```

- Inscriere student la o specializare noua

Dumitru Radu Andrei
Grupa 264
Grupa 4 EAP

```
Adding specialization to student:
Enter first name:
Andrei
Enter surname:
Daniel
Specialization code:
INFO
Specialization added succesfully!
```

In final am apelat din nou in clasa Main metoda de afisare a datelor unui student, pentru a verifica rezultatele:

```
Enter first name:
Andrei
Enter surname:
Daniel

Student:
    first_name='Andrei'
    surname='Daniel'
    date_of_birth=Mon Aug 19 00:00:00 EEST 2002
    CNP=5010518390072
    specializations={CTI INFO}
    grades=
        EAP-2020-1 Elemente Avansate de Programare 10.0
    group=2020-262
    series=2020-26

Representative of series 2020-26
```