

Să se scrie o aplicație pentru gestiunea fondului de carte al unei biblioteci. Vor fi definite următoarele clase, interfețe, enumerări:

Domeniu - enumerare pentru domenii. Constante: ISTORIE(1), MATEMATICA(2), IT(3), GEOGRAFIE(4) ... Fiecărui domeniu *i* se va asocia un cod de domeniu: `int codDomeniu`. Câmpul `codDomeniu` va avea metode de acces.

Sala - enumerare a sălilor de lectură cu locațiile acestora. Constante: VICTOR_SLAVESCU("Biblioteca Centrala"), GRIGORE_MOISIL("Facultatea de Matematica"), ... Fiecărei săli *i* se va asocia unitatea bibliotecii de care aparține: `String unitate`. Câmpul `unitate` va avea metode de acces.

Publicatie - clasă abstractă pentru modelarea conceptului de publicație. Câmpuri: `String titlu`; `String cota`, `String editura`; `int anAparitie`; `double valoareInventar`, `Domeniu[] domenii`. Clasa va avea constructor fără parametri și constructor de inițializare pentru câmpurile menționate (în ordinea în care au fost menționate!), metode de acces pentru câmpurile private, suprascriere pentru `toString()`, implementare egalitate după `cota` și comparator după `anAparitie`.

Constructorul de inițializare și metoda `get` pentru câmpul `anAparitie` vor arunca excepții pentru valori mai mari decât 2025.

Operatiuni - Interfață pentru implementarea operațiunilor de împrumut. Metode abstracte:

`void imprumut(long nrZile);` // Va fi stabilită data returnării ca data curentă + `nrZile`

`void rezervareSala(Sala numeSala);` // Se specifică sala la care este rezervată cartea

`void returnare();` //Sunt puse pe `null` campurile cu data returnării sau sala la care a fost rezervată

Carte - clasă utilizată pentru definire cărți. Este extensie de *Publicatie* și va implementa *Operatiuni*. Câmpuri: `String[] autori`, `int nrPagini`, `Sala salaRezervare`, `Date dataReturnare`. Câmpurile `salaRezervare` și `dataReturnare` sunt utilizate în procesul de rezervare la sală și împrumut. Acestea vor fi modificate prin implementarea interfeței *Operatiuni*. Clasa va avea constructor fără parametri și constructor de inițializare pentru câmpurile menționate (în ordinea în care au fost menționate!), metode de acces pentru câmpurile private, suprascriere pentru `toString()` și clonare profundă.

Revista - clasă utilizată pentru definire de reviste. Este extensie a clasei *Publicatie*. Câmpuri: `String colectia`; `int aparitiiAn`. Clasa va avea constructor fără parametri și constructor de inițializare pentru câmpurile menționate, metode de acces pentru câmpurile private, suprascriere pentru `toString()`.

1. Să se definească clasele, enumerările și interfețele menționate
2. Să se citească din fișierul `carti.csv` obiecte *Carte* și să se memoreze într-un vector de tip `Carte[]`. Citirea se va face utilizând clasa *Scanner* prin redirectarea intrării standard.
3. Să se printeze într-un fișier text cărțile din vectorul creat la punctul 2, care nu sunt împrumutate sau rezervate la sală. Vor fi salvate cota, titlul și autorii pentru fiecare carte.
4. Să se creeze listă de cărți de tip `List<Carte>` cu elementele din vectorul creat la punctul 2.
5. Să se salveze lista de cărți în fișierul `carti.dat` (Serializarea).
6. Să se restaureze lista de cărți din fișierul `carti.dat` (Deserializarea).
7. Să se efectueze operațiuni de căutare, selecție și sortare pe lista de cărți folosind metode *List* și *Collections*.
8. Să se creeze o bibliotecă de tip `jar` cu conținutul proiectului.