

Tema 1

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`; `Date dataAparitie`; `double valoareInventar`, `Domeniu[] domenii`. Clasa va avea constructor default 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 `dataAparitie`.

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

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 default 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 default și constructor de inițializare pentru câmpurile menționate (în ordinea în care au fost menționate!), metode de acces pentru câmpurile private, suprascriere pentru `toString()`.

Tema 2

Să se scrie o aplicație pentru gestiune instrumente de economisire pentru persoane fizice clienți ai unei bănci. Vor fi definite următoarele clase, interfețe, enumerări:

Moneda - enumerare pentru valută. Constante: LEU(1), EURO(2), DOLAR(3), Fiecărei valute i se va asocia un cod: `int codValuta`. Câmpul `codValuta` va avea metode de acces.

TipDepozit - enumerare pentru tipul de depozit. Constante: O_LUNA(1,0.5), DOUA_LUNI(2,0.6), TREI_LUNI(3,0.65), UN_AN(12,1), Fiecărui tip de depozit i se asociază o durată în zile: `int durata`. Câmpul `durata` va avea metode de acces.

Adresa - clasă utilizată pentru memorarea adreselor clienților. Câmpuri: `String localitate`, `judet`, `strada`, `numar`; `int cod`. Clasa va avea constructor default și constructor de inițializare pentru

câmpurile menționate (în ordinea în care au fost menționate!), metode de acces pentru câmpurile private, suprascriere pentru *toString()*.

Titular - clasă utilizată pentru evidența titularilor de cont. Câmpuri: String *nume*, long *cnp*, Adresa *adresa*. Clasa va avea constructor default 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 *cnp*, comparator după *nume* și clonare.

Cont - clasă abstractă pentru definire cont bancar. Câmpuri: Titular *titular*, Date *dataDeschidere*, Moneda *moneda*, double *valoare*, double *dobanda*, String *sucursala*. Clasa va avea constructor default și constructor de inițializare pentru câmpurile menționate (în ordinea în care au fost menționate!), metode de acces pentru câmpurile private, suprascriere pentru *toString()*, comparator după *dataDeschidere*.

ContCurent - clasă derivată din Cont. Câmp: *codIban* (String). Clasa va avea constructor default și constructor de inițializare pentru câmpurile menționate, metode de acces pentru câmpurile private, suprascriere pentru *toString()*, implementare egalitate după *codIban*.

Depozit - clasă derivată din Cont. Câmp: TipDepozit *tipDepozit*, int *codContract*. Clasa va avea constructor default și constructor de inițializare pentru câmpurile menționate, metode de acces pentru câmpurile private, suprascriere pentru *toString()*, implementare egalitate după *codContract*.

Operatiuni - Interfață implementată de clasa ContCurent. Metodă: void *depunereNumerar*(double suma).