

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(30), DOUA_LUNI(60), TREI_LUNI(90), UN_AN(365), 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, 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, metode de acces pentru câmpurile private, suprascriere pentru *toString()*, comparator după *dataDeschidere*.

Constructorul de inițializare și metoda *set* pentru câmpul *dataDeschidere* vor arunca excepții pentru o dată ulterioară datei curente.

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).

OperatiuniDepozite - Interfață implementată de clasa Depozit. Metodă: double *calculDobanda*()

1. Să se definească clasele, enumerările și interfețele menționate.
2. Să se citească din fișierul *clienti.csv* informațiile despre clienți titulari de conturi și împrumutniciți. Informațiile sunt structurate pe linii, câte o linie pentru un obiect *Persoana*. Citirea se va face utilizând clasa *Scanner* prin redirectarea intrării standard. Obiectele vor fi memorate într-un vector de tip *Persoana[]*.
3. Să se creeze listă de depozite de tip *List<Depozit>* cu elemente de tip Depozit citite dintr-un fișier text *Depozite.csv*.
4. Să se printeze într-un fișier text depozitele citite la punctul 2, care sunt în Euro. Vor fi salvate numele titularului, codul de contract și valoarea.
5. Să se salveze lista de depozite în fișierul *depozite.dat* (Serializarea).
6. Să se restaureze lista de cărți din fișierul *depozite.dat* (Deserializarea).
7. Să se efectueze operațiuni de căutare, selecție și sortare pe lista de depozite folosind metode *List* și *Collections*.
8. Să se creeze o bibliotecă de tip jar cu conținutul proiectului.