

Să se scrie o aplicație care gestionează informațiile privind apartamentele publicate spre tranzacționare de către o agenție imobiliară. Vor fi definite următoarele clase, enumerări și interfețe:

- Enumerarea *Zona* - definește constante privind locația imobilelor. Constante: DOROBANTI(1), AVIATIEI(1), DRUMUL_TABEREI(6), BERCENI(4), RAHOVA(5), TEI(2), ... Enumerarea va avea un câmp *sector* (int).
- Clasa *Agent*. Se referă la un agent imobiliar. Câmpurile private ale clasei sunt *cnp* (long), *nume* (String) și *imobile* (int[]).
- Clasa *Imobil*. Este clasă abstractă. Câmpuri private: *id* (int), *suprafataUtila* (int), *etaje* (regimul de înălțime al clădirii - int), *nrCamere* (int), *telefonP* (telefon proprietar - String), *zona* (Zona), *pret* (double), *dataP* (dată înregistrare imobil în format 'dd.MM.yyyy' - Date).
- Clasa *Apartament*. Este derivată din clasa *Imobil*. Are câmpurile private: *etaj* (int), *dotari* (String[]).
- Clasa *Casa*. Este derivată din clasa *Imobil*. Are câmpurile private: *suprafataTeren* (int), *utilități* (String[]).
- Interfața *Operatiuni*. Interfața conține o metodă pentru înregistrarea unui apartament în portofoliul unui agent și este implementată de clasa *Agent*. În interfață va fi declarată metoda *inregistrare*. Metoda va primi un *id* de imobil, va face înregistrarea prin adăugarea *id*-ului în vectorul *imobile* și va întoarce numărul de imobile din portofoliu. Tipul metodei: *int inregistrare (int id)*.

Cerințe

- 1) Fiecare clasă va avea cel puțin doi constructori: constructor *default* și constructor de inițializare
- 2) Fiecare clasă va avea metode de acces pentru câmpurile de tip privat
- 3) Fiecare clasă va avea implementare pentru metoda *toString()*
- 4) Se va implementa testarea egalității dintre imobile după *id* și comparabilitate după preț
- 5) Se va adăuga în structura clasei *Imobil* câmpul *cnpAgent*, cu cnp-ul agentului care are imobilul în portofoliu
- 6) Se va implementa clonare profundă pentru clasele *Casa* și *Apartament*
- 7) Constructorul clasei *Apartament* și metoda de inițializare corespunzătoare vor arunca excepții pentru valori ale câmpului *etaj* mai mari decât ale câmpului *etaje*
- 8) Vor fi citite apartamentele aflate în portofoliu într-o listă de tip *List<Apartament>*. În fișierul text *apartamente.csv* sunt salvate informațiile despre apartamente pe agenți. Structura fișierului: informațiile referitoare la un agent se află pe $n + 1$ linii, unde n este numărul de apartamente aflate în portofoliu. Pe prima linie sunt cnp-ul, numele agentului și numărul de apartamente aflate în portofoliu (n). Pe următoarele n linii sunt apartamentele, astfel:

cnp,nume,n

id₁,suprafataUtila₁,etaje₁,nrCamere₁,telefonP₁,zona₁,pret₁,dataP₁,etaj₁,dotari₁

...

id_n,suprafataUtila_n,etaje_n,nrCamere_n,telefonP_n,zona_n,pret_n,dataP_n,etaj_n,dotari_n

Exemplu:

1680909234564,Pop Marian,2

1,100,4,4,076654321,AVIATIEI,200000,01.01.2022,1,centrala,loc parcare

2,80,10,4,0766567821,TEI,80000,02.02.2022,1,semidecomandat

1671010288599,Manea Mihai,3

4,50,4,4,0740256214,DRUMUL_TABEREI,70000,01.03.2022,3,loc parcare

5,60,3,3,0721547896,DOROBANTI,150000,02.05.2019,3,centrala

6,70,3,3,0722765432,AVIATIEI,150000,03.05.2019,2

9) Să se sorteze lista de apartamente după preț, data publicării și telefon proprietar

10) Să se salveze lista de agenți în fișierul text *agenti.csv*, în structura:

cnp agent,nume agent,număr apartamente în portofoliu

11) Să se afișeze pentru fiecare apartament din lista de apartamente, cnp-ul agentului

12) Să se implementeze căutare-selecție apartamente după număr telefon proprietar și data publicării

13) Să se salveze lista de apartamente în fișierul binar *apartamente.dat*

14) Să se restaureze lista de apartamente din fișierul binar *apartamente.dat*