

Se dorește implementarea unei aplicații care să permită gestionarea clienților și a proprietăților imobiliare în cadrul unei agenții imobiliare nou înființate. Agenția imobiliară închiriaza atât case, cât și apartamente. Pentru abstractizarea conceptelor din lumea reală se vor implementa clasele **Locuinta**, **Apartment** și **Casa**.

(5p)

Definiți clasa **Locuinta**, conținând următoarele date membre private: **numeClient**(șir de caractere de lungime variabilă, NULL dacă locuinta nu este închiriată), **suprafataUtile**(întreg), **discount**(float). Să se suprîncarce operator++ și operator—atât în forma prefixată, cât și postfixată pentru incrementarea și decrementarea discountului, cu asigurarea valorii în intervalul 0-10. Să se suprîncarce operatorul de cast la întreg pentru returnarea suprafeței utile.

(1p)

Din clasa **Locuinta** se vor deriva public clasele **Apartment**, având în plus câmpul **etaj**(întreg) și **Casa**, având în plus câmpurile **suprafataCurte**(întreg), **nrEtaje**(întreg) și **suprafataEtaje**(float\*). Testarea se va face folosind obiecte de tipul claselor **Apartment** și **Casa**, citite din fișiere text.

(1p)

Se va adăuga la clasa **Locuinta**, metoda virtuală pură **CalculChirie**(X,Y) cu semnificațiile:

X=valoare standard chirie/mp(întreg);

Y=cu (1) / fără (0) luarea în considerare a discountului(întreg);

Metoda adăugată va fi implementată în clasa **Apartment** după formula  $X * \text{suprafataUtile} * (1 - Y * \text{discount}/100.0)$ , respectiv în clasa **Casa** după formula  $X * (\text{suprafataUtile} + 0.2 * \text{suprafataCurte}) * (1 - Y * \text{discount}/100.0)$ . Metoda va fi testată prin parcurgerea în funcția main() a unui vector de pointeri la **Locuinta\***, încărcat cu obiecte de tip **Apartment** și **Casa**.

(1p)

Să se definească clasa **AgentieImobiliara** conținând un vector de pointeri la obiecte de tip **Locuinta** alocat dinamic și să suprîncarce operator+= și operator-= pentru inserarea, respectiv eliminarea unei locuințe din listă, indiferent de tipul acesteia. Să se suprîncarce operator<< pentru afișarea clienților pentru locuințele închiriate.

(1p)

Să se construiască clasa template **Gestiune** care să stocheze numărul de apartamente sau case, fiecare cu chiria lunară aferentă. Să se construiască o specializare pentru tipul **char\*** care să stocheze numărul de clienți, împreună cu numele acestora.

(1p)

Pentru gestionarea listei de locuințe, se va utiliza o structură de tip STL vector. Structura va fi parcursă cu ajutorul unui iterator cu scopul de a calcula suma chiriilor pentru locuințele închiriate, respectiv pentru cele neînchiriate.

Observatie:

Cerintele vor fi punctate doar daca exista o secventa de cod pentru exemplificare in main si programul ruleaza corect(nu exista erori de compilare, link-editare sau executie);

La evaluare poate fi solicitata implementarea uneia sau mai multor cerinte noi(inclusiv pentru nota 5).