

Laborator 11

Exercițiu 1 – Citiți rapid laboratorul 11 postat pe OCW (10-15 min)

Exercițiu 2

```
//clasa interfata
class Copac {
    //nu are attribute si toate metodele sunt pur virtuale
    //definiti destructorul pur virtual, cei 3 get-eri, metoda "afisare" si operatorul<<
};

class Conifer: public Copac {
    string zona;
    int greutate;
    float inaltime;
public:
    //Constructorii (3 sau 2 daca folositi parametrii impliciti) + Destructor
    //Operator=
    //restul metodelor conform interfetei + ce alte metode considerati ca aveti nevoie
};

class Foios: public Copac {
    string zona;
    int greutate;
    float inaltime;
public:
    //ca mai sus
};
```

Mai e o clasă pe următoarea pagină !!

```

class PomFructifer: public Copac {
    string zona;
    int greutate;
    float inaltime;
public:
    //ca mai sus
};

```

Cerințe:

0. În implementarea cerințelor folosiți elemente STL (string, vector, sort(), swap() etc)

1. Implementați toate metodele astfel încât programul să funcționeze corect
2. Creați în main un vector de **8** elemente neomogene (cel puțin 1 din fiecare clasă)
3. Afișați vectorul folosind **doar** operatorul << (ex: cout << v[i] << endl;)
4. Sortați vectorul (nu contează dacă e crescător sau descrescător sau după ce criteriu sortați, poate fi atât după int dar și după float) și îl mai afișați o dată
5. Eliberați memoria corespunzător

Obs1: Pentru a folosi vector(sau list, stack, queue, etc, orice container) puteți de exemplu să faceți câte un șir pentru fiecare tip de obiecte (3 container-e, în cazul de față)

Obs2: Aveți grijă în sintaxa pentru sort(), fiind obiecte custom și nu tipuri de date primitive (int, float etc) va trebui să precizați după ce get-er se va face sortarea, căutați pe internet !

Punctaj:

Exercițiul 1 – 0 puncte

Exercițiul 2 - Cerința 1 - 3 puncte

Exercițiul 2 - Cerința 2 - 2 puncte

Exercițiul 2 - Cerința 3 - 2 puncte

Exercițiul 2 - Cerința 4 - 1 puncte

Exercițiul 2 - Cerința 5 - 1 puncte

1 punct din oficiu