

Particularitățile clasei

Începând cu acest laborator vom avea alături de noi un avatar pe care va trebui să îl învățăm POO astfel încât să își îndeplinească visul de a deveni un bun programator. Noul nostru coleg este NullPointerNinja care începând de azi a luat decizia de a învăța și de a scrie programe corecte din toate punctele de vedere și ne-a ales pe noi ca și îndrumători deoarece are încredere deplină că visul lui va deveni realitate alături de noi.



Pentru astăzi NullPointerNinja are în plan să afle lucruri noi despre clase, spre exemplu: ce este constructorul de copiere și cum poate fi acesta implementat corect atunci când în clasă există cel puțin un membru de tip pointer care este alocat dinamic; cum poate să implementeze operatorul de asignare și destructorul pentru o clasă și, de ce nu, să înțeleagă când este util să folosească shallow copy și deep copy. Deși este un ninja, în spatele uniforme sale, NullPointer este un băiat care are pasiuni ca toți oamenii din ziua de azi. Și ar dori ca pe parcursul laboratorului de astăzi să poată să facă un program care să îi poată oferi informații relevante cu privire la anumite flori despre care a mai citit în copilărie prin atlasele de botanică. Așadar haideți să ne luăm rolul de mentori în serios și să ne învățăm ucenicul o lecție de POO așa cum se cuvine.

Se consideră următoarea clasa în limbajul C++:

```
class Floare
{
    float pret;
    char* culoare;
    char* denumire;
};
```

Cerințe

- 1) Să se implementeze constructorul fără parametri și constructorul cu toți parametrii pentru clasa Floare. (1p)
- 2) Să se implementeze constructorul de copiere și operatorul de asignare astfel încât aceștia să realizeze deep copy. (1p)
- 3) Să se implementeze o metodă pentru afișarea datelor despre flori în consolă cât și destructorul pentru clasa Floare care să realizeze corect eliberarea resurselor. (1p)
- 4) Să se implementeze getterii și setterii pentru câmpurile clasei Floare. (3p)
- 5) În main să se creeze 2 obiecte și să se testeze metodele implementate în cadrul cerințelor anterioare. (1p)
- 6) În funcția main să se creeze un vector alocat dinamic cu minim elemente de tip Floare. Elementele nu se vor citi de la tastatură ci se vor hardcoda direct în funcția main. (1p)
- 7) Sortați vectorul creat la punctul anterior alfabetic după denumirea florii. (1p)

Cerință bonus: Să se implementeze un mecanism de asigurare a unicității între obiectele de tip Floare prin adăugarea unui membru static și a unui membru constant în clasă (hint: OCW).

Se acordă un punct din oficiu. Succes.

Precizări

- Clasa se va implementa în fișiere .h și .cpp. Iar testarea codului se va face în funcția main care va fi într-un fișier separat.
- Neimplementarea în fișiere .h și .cpp atrage după sine pierderea unui punct.
- Erorile de compilare și memory leak-urile atrag fiecare după sine pierderea unui punct.
- Alte depunctări cu privire la modul de implementare sau la corectitudinea declarării metodelor, rămân la latitudinea asistentului care vă corectează.
- Bonusul se acordă doar dacă cerințele obligatorii au fost realizate complet și corect.
- Dacă nu s-a specificat vreun mod de implementare a cerințelor rămâne la latitudinea voastră cum vreți să procedați în rezolvare.