

Cerinte suplimentare LP2

Studentii vor rezolva, LA ALEGERE, unul din cele 2 subiecte

Timp de lucru - 1h30, Oficiu - 1p

1. (Pentru temele 1 - 17)

Se considera proiectele deja realizate si se alege o clasa din cadrul acestora. Sa se aplice urmatoarele transformari:

Sa se implementeze clasa **Hibrid** – derivata din clasa aleasa si clasa **Proiect**(obtinuta in urma ierarhiei de mai jos):

```
Obiect(int id (incrementat automat),string denumire);  
Obiect_Actualizat(string data) : public Obiect;  
Obiect_Personalizat(char* autor_proiect) : public Obiect  
Proiect : public Actualizare, public Personalizare.
```

Cerinte:

- (3p) Sa se faca validarea tuturor datelor utilizate (id-urile sa fie numere naturale, string-urile sa nu contina cifre, alocarea doar a unui numar strict pozitiv de caractere, etc.)
- (1p) Fiecare clasa derivata va contine operator = supraincarcat.
- (1p) Sa se ilustreze conceptul de RTTI
- (2p) Programul sa permita executia urmatorului cod:
Obiect a("Obiect1"); Personalizare b(a, "Ionescu"); char s[100]="Proiectul ";
strcat(s,a); strcat(s,"este actualizat de "); strcat(s,b);
cout<<s; //se va afisa: Proiectul Obiect1 este actualizat de Ionescu.
- (2p) Sa se actualizeze vectorul de pointeri catre clasa de baza din proiectul initial astfel incat sa contina si adrese de tip Hibrid. Afisarea obiectelor se va realiza in doua moduri – atat prin functie virtuala de afisare, cat si prin operatorul <<.

1. (Pentru temele 18-30)

(3p) Sa se faca validarea tuturor datelor utilizate (int-urile sa nu permita alte caractere decat 0-9, string-urile sa nu contina cifre, alocarea doar a unui numar strict pozitiv de elemente, etc.).

(2p) Sa se ilustreze conceptul de RTTI (daca se utilizeaza doar una din cele 2 metode, punctajul se injumatateste)

(1p) Sa se implementeze un operator la alegere, folosind functiile deja implementate in proiect.

Operatorul sa fie implementat atat in clasa de baza cat si in cel putin o clasa derivata, iar implementarea din clasa derivata sa faca apel la operatorul din clasa de baza.

(1p) Sa se implementeze operatorul de atribuire in fiecare clasa, astfel incat sa se permita atribuirea multipla.

(2p) Implementati o clasa de management care va retine unul sau mai multi vectori de pointeri hibridi (in functie de problema voastra). Pentru aceasta implementati operatorul << si >>. Optional aceasta clasa poate fi facuta singleton (+0.5p +0.5p pentru rezolvarea memory leak-ului).

Schita unei clase singleton:

```
SinletonClass{  
    private:  
        SinletonClass(){}  
        static SinletonClass* instance;  
    public:  
        static SinletonClass* getInstance(){  
            if(!instance) instance = new SinletonClass();  
            return instance; }  
}  
SinletonClass* SinletonClass::instance = nullptr;  
/*****/
```

Cerinte suplimentare LP2

Studentii vor rezolva, LA ALEGERE, unul din cele 2 subiecte

Timp de lucru - 1h30, Oficiu - 1p

2. Se dorește implementarea unei aplicații care să permită gestionarea activității unor farmacii aparținând proprietarului X. În acest scop se vor implementa clasele:

Farmacie_abstracta(string denumire + conține cel puțin metoda virtuală pură de **calcul_venit()**)

Farmacie_offline: public **Farmacie_abstracta** (int nr_luni, double* cifra_afaceri_luna)

Farmacie_online : public **Farmacie_abstracta** (având în plus câmpurile char * adresa_web, int nr_vizitatori)

Farmacie_dubla_specializare: public **Farmacie_offline**, public **Farmacie_online** (int nr_angajati)

Cerinte:

- (2p) Clasele derivate să conțină constructori de inițializare și parametrizați (care să apeleze constructorii din clasa/clasele de bază) și destructori.
- (2p) Toate clasele derivate vor conține constructori de copiere și operatorul = supraincarcat.
- (1p) Operatorul >> să fie supraincarcat astfel încât citirea să fie făcută din fișier.
- (1p) Clasa **Farmacie_abstracta** va conține o funcție virtuală **Afisare** care va fi rescrisă corespunzător în clasele derivate.
- (0.5p) Se va supraincarca în clasa **Farmacie_dubla_specializare** operatorul de cast la întreg pentru returnarea numărului de angajați.
- (0.5p) Metoda calcul_venit() va fi implementată în clasa **Farmacie_offline** ca suma tuturor valorilor din vectorul cifra_afaceri_luna.
- (2p) Funcția main() va conține un “Demo” cu exemplificarea tuturor conceptelor implementate și este obligatorie parcurgerea unui vector de pointeri la **Farmacie_abstracta***, încarcat cu obiecte de tip **Farmacie_offline**, **Farmacie_online** și **Farmacie_dubla_specializare**.