**Minitest 2**

**Sudentii vor rezolva, LA ALEGERE, unul din cele 2 subiecte**

**Nr. 1**

Se considera proiectele deja realizate. Sa se aplice urmatoarele transformari:

(2p) Sa se implementeze clasa **Hibrid** – derivata din **X** ***(clasa derivata aleasa din ierarhia de clase din proiectul vostru)*** si clasa**Proiect**(***obtinuta in urma ierarhiei de mai jos***):

**Obiect(***int id* *(incrementat automat), char denumire\_obiect[100]***);**

**Actualizare(***int zi, luna, an***) : Obiect;**

**Personalizare(***char\* autor\_proiect***) : Obiect**

**Proiect: Actualizare, Personalizare.**

(3p) Fiecare clasa derivata va contine constructori de inițializare, parametrizati si de copiere (care sa apeleze constructorii din clasa de baza)

(1p) Sa se ilustreze conceptele de upcasting / downcasting care sa includa noile clase introduse.

(1p) In clasa Hibrid sa se evidentieze:

- operatorul >>

- functie de **Afisare** care sa faca apel la functii similare din clasele din care provine;

(1p) Programul sa permita executia urmatorului cod:

***Obiect a(1,“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.***

(1p) Sa se actualizeze vectorul de pointeri catre clasa de baza din proiectul initial astfel incat sa contina si adrese de tip **Hibrid**.

Of. 1p

**Nr. 2**

Se doreste implementarea unei aplicatii care sa permita gestionarea activitatii bibliotecii X. In acest scop se vor implementa clasele:

***Autor*** (nume – char\*);

***Biblioteca*** (denumire – char[50] – camp static);

***Carte:Biblioteca*** ( aparitie - int, titlu – char[50]);

***InregistrareCarte:Biblioteca*** (isbn - int, nrpagini – int);

***FisaBiblioteca:Carte , InregistrareCarte*** (id *–* int (incrementare autormata), Aut [ ] - vector de autori, nrAutori – intreg).

***GestiuneCarti***(int nr*, Biblioteca \*\*b);*

**Cerinte:**

(3p) Implementati ierarhia de clase. Fiecare clasa derivata va contine constructori de inițializare, parametrizati si de copiere (care sa apeleze constructorii din clasa de baza).

(1p) Fiecare clasa va contine operator = supraincarcat.

(1p) Clasa **Biblioteca** va contine functii virtuale de **Citire** si **Afisare** care vor fi rescrise corespunzator in clasele derivate.

(4p) Functia main() va contine un “Demo” cu exemplificarea tuturor conceptelor implementate, iar citirea unui obiect de tip GestiuneCarti sa implice popularea vectorului b cu obiecte de tip Carte si de tip FisaBiblioteca. Afisarea unui obiect de tip GestiuneCarti va realiza afisarea obiectelor stocate in vectorul b in doua moduri – atat prin functia **Afisare**, cat si prin operatorul <<.

“Demo”-ul va mai contine si codul umator:

***Biblioteca b(“Eminescu”); Carte c(“Humanitas”, 1998, “Ion”);***

***FisaBiblioteca fb(”Carturesti”, 1880, “Schite si Nuvele”, 45879, 356, 1,”Caragiale”);***

***char s[100]=””;***

***strcat(s,b); strcat(s,c); strcat(s,fb);***

***cout<<s; //se va afisa: EminescuHumanitasCarturesti.***

**Obs.** 1. Oficiu – 1p

2. Timp de lucru – 1h20min