

Specificații proiect POO 2020-2021

Proiectul din acest an își propune realizarea unei aplicații de emitere și gestionare a biletelor de film (cinema) în C++, ce utilizează fișiere binare și dispune de o interfață în linia de comandă.

Cerințe:

Faza 1

- Definirea a minim 5 clase ce au legătură cu tema proiectului (ex: Film, Bilet, etc)
- În clase trebuie să existe minim un câmp de tip șir de caractere alocat dinamic (char*) și minim un vector de valori numerice alocat dinamic
- În clase trebuie să existe minim un câmp static și un câmp constant
- Toate atributele vor fi definite pe zona privată a clasei, cu definirea de metode de acces acolo unde acestea sunt necesare
- Funcțiile de tip set vor conține validări pentru a asigura corectitudinea datelor
- Fiecare clasa conține minim un constructor cu parametri, iar pentru cele ce conțin câmpuri alocate dinamic se vor defini obligatoriu constructor de copiere, destructor și forma supraîncărcată a operatorului =
- Pentru fiecare clasă se vor încărca următorii operatori:
 - >> și <<
 - operatorul pentru indexare []
 - minim un operator matematic (+, -, *, sau /)
 - ++ sau -- (cu cele 2 forme)
 - operatorul cast (catre orice tip) explicit sau implicit
 - operatorul pentru negatie !
 - un operator conditional (<.,=,<,>=)
 - operatorul pentru testarea egalitatii dintre 2 obiecte ==

Faza 2

- Se implementează modulul prin care aplicația poate primi următoarele comenzi prin intermediul unui meniu:
 - Emitere bilet (cu selectare film, sala, loc, etc) => biletul este salvat într-un fișier text pentru a putea fi ulterior printat
 - Situație locuri libere (afișare în consolă sau fișier text)
 - Situație filme (ce filme rulează, la ce ore, în ce săli)
 - Operații de tip CRUD (Create Read Update Delete) pentru toate entitățile: Adăugare bilet, modificare bilet, ștergere bilet, adăugare sală, modificare sală, ștergere sală, adăugare filme, modificare film, ștergere film, etc.
- Toate datele sunt salvate pe disc în fișiere binare astfel încât să se asigure persistența acestora
- Toate comenzile vor fi tratate cu atenție astfel încât să nu existe erori de execuție

Faza 3

- Utilizarea conceptelor de derivare și compunere a claselor
- Se implementează minim o clasă abstractă (cu sau fără attribute) ce trebuie să conțină minim 2 metode virtuale pure
- Se implementează minim 2 metode virtuale (altele decât cele virtuale pure)
- Se va folosi conceptul de clase șablon (template) pentru gestionarea unei anumite entități
- Se va modifica (refactoring) codul existent pentru a utiliza minim 3 containere STL în locul vectorilor/matricelor clasice

Observații:

- Implementarea proiectului în minim 2 faze (evidențiate prin commit-uri distincte în repository-ul utilizat) condiționează prezentarea acestuia
- Copierea codului sursă aferent oricărei faze, duce la anularea întregului punctaj de seminar și implicit imposibilitatea susținerii examenului în sesiunea normală.
- Prezentarea proiectului este obligatorie pentru a putea fi punctat