

Proiect #2 - Programare Orientată pe Obiecte

Grupa 141 - semigrupele 1 și 2

Cerințe

1. Aplicația voastră trebuie să conțină **minim 5 clase** care vor avea implementate cel puțin o funcționalitate (dacă am o clasă de tip Agendă atunci aceasta ar putea afișa datele disponibile dintr-o anumită lună).
2. Fiecare clasă trebuie să implementeze următoarele:
 - a. Constructorul fără parametri
 - b. Constructorul cu toți parametri
 - c. Cel puțin 2 constructori cu parametri
 - d. Constructorul de copiere
 - e. Destructorul
 - f. Forma supraîncărcată a operatorului =
 - g. Supraîncărcarea operatorilor pentru stream << și >>
3. Trebuie să exemplifice conceptul de virtualizare.
4. Trebuie să se folosească de conceptul de moștenire (minim 3 clase în ierarhie).
5. Trebuie să se folosească de conceptul de polimorfism.
6. Trebuie să avem minim o clasă abstractă / interfață.
7. **Trebuie să fie prezent în cod, pentru nota 10, moștenirea în diamant (moștenirea multiplă) corect rezolvată pentru cazul vostru. Dacă problema moștenirii în diamant nu este prezentă proiectul vostru va putea lua maxim nota 8.**
8. Proiectul trebuie să vină cu un meniu interactiv sau o formă de meniu prin care să se poată exemplifica funcționalitățile implementate. (Ar trebui pentru sistemele de gestiune să putem efectua CRUD-uri - Create Read Update Delete)
 - a. Mențiune: Dacă am o clasă Eveniment atunci aş putea să creez un Eveniment citind informațiile despre acesta precum: dată început, dată sfârșit, număr maxim de participanți, locație, etc. Pot să afișez toate evenimentele create folosind o variabilă de tip static Eveniment* pentru a reține evenimentele într-o listă (aveți grijă la memory leaks). Aș putea să selectez un eveniment dintr-o listă și să îi actualizez attributele precum

locația sau / și datele de început final este problema voastră cum vreți să gestionați actualizarea (citiți despre **HTTP PATCH vs. PUT**). Aș putea elimina un eveniment din lista de evenimente pe baza unui cod (ID) sau a numelui.