

Proiect #1 - Programare Orientată pe Obiecte

Grupa 141 - semigrupele 1 și 2

Cerințe

1. Aplicația voastră trebuie să conțină **minim 4 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. **Fiecare clasă trebuie să respecte principiul încapsulării datelor, așadar trebuie construiți setterii și geterii necesari pentru respectarea acestui principiu.**
4. Fiecare clasă trebuie să aibă **cel puțin 3 atribute (proprietăți)**, iar în tot proiectul să regăsim variabile de următoarele tipuri:
 - a. **Int / Long Long**
 - b. **Bool**
 - c. **Char***
 - d. **Float**
 - e. **Char**
 - f. **Int* / Float*** (doar una dintre ele nu ambele neapărat)
 - g. **Static**
 - h. **Const**
5. Fiecare clasă trebuie să implementeze supraîncărcarea pentru următorii operatori:
 - a. **Operatorul de indexare - []** (care să spunem dacă avem o clasă de tip Eveniment iar în clasa Eveniment avem Participant* putem folosi operatorul de indexare pentru a accesa direct informațiile despre participant de la poziția i)

- b. Operatorul ++ sau -- (postfixat sau prefixat, pentru postfixat aveți nevoie de un parametru dummy)
 - c. Minim 2 operatori matematici (+, *, / sau -) - (să se respecte comutativitatea)
 - d. Operatorul de verificare a egalității (==) (de exemplu avem două obiecte de tip Elev și putem folosi supraîncărcarea operatorului == pentru a verifica dacă Elevii sunt din aceeași clasă sau nu)
 - e. Un operator condițional dintre (<, <=, >, >=)
6. 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 aș 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.