

Proiect 3 Programare Orientata pe Obiect:

- Proiectul 3 consta in alegerea a minim 4 clase care sa exemplifice o aplicatie, exemple: magazine de jocuri video, Netflix, etc.
- Pentru toate clasele din proiecte, trebuie sa se respecte conceptul de incapsulare a datelor, altfel proiectul se noteaza cu 0.
- Proiectele care nu ruleaza sunt notate cu 0.
- Proiectul se trimite dupa prezentarea acestuia la laborator.
Cine nu este prezent la prezentare nu beneficiază de corectatul proiectului si este notat cu 0. Cine nu este prezent la prezentare nu beneficiază de corectatul proiectului si este notat cu 0.
- Proiectul trebuie sa contina comentarii.
- Proiectele sunt trimise pe adresa de mail **poolaborator@gmail.com** si se trimite doar main.cpp redenumit astfel:
Grupa_Nume_Prenume_Proiect2.cpp, orice alta denumire sau alt fisier nu este luat in considerare pentru corectare (nu se accepta header files).
- Pentru orice tentativa de fraudă proiectul este notat cu 0.
- Proiectul trebuie sa se realizeze in limbajul de programare C++.
- Fiecare tema aleasa pentru proiect trebuie sa fie unica la nivel de semigura.
- Trebuie sa va alegeti tema si sa o notati in excel-ul aferent (link excel: https://docs.google.com/spreadsheets/d/1wNW-fX_PcuPHwJQt5A5XB-sMFtp3DbJZo7jY72kyeso/edit?usp=sharing).

Cerinte:

- Să se implementeze din librăria STL cel puțin un vector, un set, o listă și un map pentru a gestiona datele aplicației.
- Scrierea și citirea din fișier
- Aplicația va conține un meniu (în consolă) prin care utilizatorul poate alege diverse opțiuni (import/export date, citire obiect nou de la tastatură, diverse funcționalități ale aplicației) de tip Singleton
- Folosirea a cel puțin o clasă Template
- Folosirea a cel puțin 2 metode Template
- Proiectul trebuie să trateze toate excepțiile posibile care pot apărea la executarea codului (codul trebuie să arunce cel puțin 6 excepții dintre care minim 3 diferite).
- Exemplificarea conceptelor de upcasting și downcasting (trebuie să folosească și `dynamic_cast`)
- Exemplificarea conceptului de mostenire până la nepot.
(Parint->Copil->Nepot)
- Fiecare clasă să conțină:
 - Constructorul fără parametri
 - Constructorul cu toți parametri
 - Copy Constructorul
 - Operator egal
 - Destructor
 - Operator de citire
 - Operator de afișare