

POO - LAB 6

ACTIVITATE LABORATOR

1. Instalare și activare Core Guideline Checker (curs 5). Se selectează reguli cu C++ Core...
Rezolvare erori/warning-uri.

2. Implementare constructor de copiere pentru clasa din domeniu.

- a) În corpul constructorului: `cout<<"Copy constructor called"`.
- b) Eliminați copierile care nu sunt necesare (folosind `&`, `const&`, `const auto&` etc).
 - i. Afișare elemente: nu ar trebui să se facă nicio copie.
 - ii. Filtrare: dacă după filtru rezultă un obiect, ar trebui să fie o singură copie.
- c) Se ignoră warning-ul de la core checker care cere implementarea tuturor metodelor speciale (all constructors, assignment, destructor) - apare după implementare copy constructor.

Vă puteți anunța pentru predare **până la fără 20**.

! Cerințe lab 6 (iterația 1): no pointer, STL vector, STL string

CERINȚE GENERALE

- cel puțin 99% code coverage peste tot în afară de ui
- specificații complete
- fără memory leaks
- fără warning-uri (setat Warning Level 4, Treat warnings as errors)

Reminder: dacă predați laboratoare din urmă, trebuie să aveți și activitățile de laborator aferente fiecăruia și se mai poate primi o cerință scurtă, pe loc, pentru verificarea înțelegerii codului scris.

LAB 7

Cerințe: http://www.cs.ubbcluj.ro/~istvanc/oop/lab/lab6_7.pdf

Iterația 2:

- Se înlocuiește clasa `std::vector` cu o clasă proprie parametrizată (template) care implementează o listă folosind structura de date *listă înlănțuită* sau *vector dinamic* (la alegere)
- Validare date/Afișare mesaj eroare/Excepții
 - Introducere valori de intrare greșite
 - Adăugare entitate existentă deja
 - Ștergere/modificare elemente inexistente
- Diagramă UML pentru arhitectura aplicației