

POO - LAB 7

ACTIVITATE LABORATOR

<http://www.cs.ubbcluj.ro/~istvanc/ooop/exemple/VectorDinamicCPP.zip>

Se descarcă proiectul și se adaugă lista proprie în el. Se rulează testele existente pe această listă. Cerința este ca aceste teste să treacă, și să nu existe memory leak-uri.

Dacă nu aveți deja, implementați iterator pentru listă și alte metode care lipsesc.

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

CERINȚE LAB 7 (REMINDER)

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

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)
- CPP Core Guideline Checker activat: soluționare warnings referitoare la *const*, pointeri (la pointeri dacă sunt afara clasei pentru listă)
- No raw pointer (în afară de partea de implementare a listei)

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 8

Cerințe: http://www.cs.ubbcluj.ro/~istvanc/ooop/lab/lab8_9.pdf

[2 săptămâni] Se lucrează pe aplicația voastră în continuare, doar adăugați funcționalitățile menționate pentru problema aferentă.

Iterația 1 (săpt. 8)

Modificați aplicația:

- folosiți clasa vector din STL
- înlocuiți toate for-urile din aplicație cu algoritmi din STL și varianta c++11 de for (range for). Exemplu de algoritmi: *copy_if* (la filtrare), *find_if* (căutare), *sort*
- Folosiți excepții pentru a raporta erori (creați clase proprii pentru excepții).
- Adăugați funcționalitatea nouă (descrie la fiecare problema).