# Tehnici avansate de programare
# POO C++

Conf. univ. dr. ing. Lefkovits Szidónia
E-mail: szidonia.lefkovits@umfst.ro
http://sites.umfst.ro/lefkovits-szidonia
utilizator: TAP
parola: CursTAP2021

# Laborator 10

STL Iteratori

# Probleme propuse/ Teme
# Problema 1

- Fie trei containere. Primul container de tip deque, conține numerele divizibile cu 3 de la 1 la 10. Al doilea container, de tip vector, numerele divizibile cu 2, de la 1 la 10, iar al treilea container o lista care conține numerele negative de la -10 la -1.
- Să se obțină următoarele șiruri, folosind algoritmul copy.
- a) 3 6 9 2 4 6 8 10 după el. deque apar el. vectorului
- b) 2 4 6 8 10 3 6 9 înainte de el. deque apar el. vectorului
- c) 2 4 6 8 -1 -2 -3 -4 -5 10 3 6 -6 -7 9 -8 -9 -10 9 se inserează jumătatea listei înainte de ultimul el. al vect. și cealaltă jum. înainte de ult. el. al deque

# Probleme propuse/ Teme
# Problema 2

- Să se scrie o funcție șablon *înlocuire* aplicabil pentru fiecare container.
- Parametri funcției sunt: 2 iteratori (primul element și ultimul element din container sau porțiunea de container considerată), valoarea de înlocuit și noua valoare.

# Probleme / Teme 3 Facultativ

- Să se creeze o lista lineară dublu înlănțuită generală, cu șablon, care poate fi folosită în algoritmii STL.
- Pe lângă clasa ListaD se va defini o clasă Nod și o clasă iterator_fw, fiind subclasa *interatorului în avans*. și iterator_rw, fiind subclasa *iteratorului invers*.
- Clasa Nod va defini constructor implicit+ constructor cu parametru. Va clase prietene ListD, iterator_fw și iterator_rw.
- Clasa ListaD vor defini constructor, constructor de copiere, destructor, operator=, operator==, operatorul<, metodele empty, front, back, push_back, push_front, pop_back, pop_front, insert(insereaza), erase(sterge), sort, merge(concatenarea a 2 liste), begin(), end(), rbegin(), rend()

# Probleme propuse/ Teme

- Iteratorul de avans trebuie să definească componentele iterator_traits și trebuie să definească constructor implicit, de copiere, să suprascrie operatorii =,*,->,++ (pre/postincrementare),==,!=.
- Va avea o clasă prietenă ListaD.
- Iteratorul invers inversează comportarea operatorilor **++** și **--** și face posibilă definirea în ListaD a metodelor rbegin() și rend().

# Probleme propuse/ Teme

- În main: să se creeze o lista dublă. În care se inserează câteva elemnete prin push_back, push_front, insert. Se șterg elemente prin erase, pop_back, pop_front.
- Se afișează lista de la stânga la dreapta folosind iterator în avans.
- Se afișează lista de la dreapta la stânga folosind iterator invers.
- Se copiază lista ordonată într-o altă listă.
- Se concatenează lista inițială și lista ordonată. Se afișează noua listă cu metoda copy iterator de ieșire.