



# Tehnici avansate de programare POO C++

Conf. univ. dr. ing. Lefkovits Szidónia

E-mail: [szidonia.lefkovits@umfst.ro](mailto:szidonia.lefkovits@umfst.ro)

<http://sites.umfst.ro/lefkovits-szidonia>

utilizator: TAP

parola: CursTAP2021



# Laborator 3

Membri statici  
Funcții inline, funcții constante  
Funcții și clase prietene



# Probleme propuse/ Temă

## Problema 1

1. Fie clasa **Dreaptă** cu două proprietăți  $m$  și  $n$ , reprezentând panta dreptei și ordonata la origine, adică  $y=mx+n$

Să se scrie 2 constructori:

- un constructor cu doi parametri și cu valori inițiale de  $m=1$ ,  $n=0$
- constructor de copiere
- destructor

- În *main* să se creeze o Dreaptă



# Probleme propuse/Temă

## Problema 1

- Fie clasa **Punct** care va avea 2 proprietăți  $x$  și  $y$  abscisa și ordonata.
- Și în acest caz se vor defini 2 constructori și 1 destructor
- În *main* să se creeze  $n$  instanțe ale clasei Punct, într-un vector de puncte.  $n$  se citește de la tastatură/fișier.
- În clasa Punct se va defini o funcție *prietenă*, numită *distanță*, care va calcula distanța între două puncte date.



# Probleme propuse/ Temă

## Problema 1

- Clasa Dreaptă este o clasă *prietenă* clasei Punct.
- În clasa Punct se definește o proprietate statică de tip Dreapta, care va stabili dreapta.
- În clasa Punct să se declare 3 variabile statice care vor contoriza numărul de puncte în semniplanul pozitiv, semniplanul negativ și pe dreaptă. Contorizarea se face în constructor și în destructor.
- Fie punctul  $A(ax, ay)$
- Dacă  $m \cdot ax + n = ay \Rightarrow$  contor puncte pe dreaptă
- Dacă  $m \cdot ax + n - ay > 0 \Rightarrow$  semniplan pozitiv
- Altfel semniplanul negativ



# Probleme propuse/ Teme

## Problema 1

- În metoda main să se declare o metodă `centruDeGreutate` care calculează centrul de greutate a tuturor celor  $n$  puncte.



# Probleme propuse/ Teme

## Problema 2

2. Să se creeze o listă lineară simplu înlănțuită (sau dublu înlănțuită) astfel încât clasa Nod să aibă o clasă prietenă LSI/LDI.



# Temă

3. Creați o clasă **Carte** care are ca atribute: autor, titlu, ISBN (ID) și preț. Creați un vector de cărți și afișați cărțile care au autorul egal cu un autor citit de la tastatură. Folosiți funcții inline și funcții const. Afișați numărul de instanțe ale clasei Carte.





## Temă (cont.)

4. Definiți clasele **Film** și **Actor**, care conțin cel puțin 3 attribute fiecare. Clasa Film conține o listă de Actori și poate adăuga/scoate actori din listă.
- a. Numărați obiectele de tip actor create de aplicație
  - b. O funcție (prietenă) care sortează un șir de filme în funcție de buget și numărul de actori (pentru comparare folosiți o funcție prietenă)
  - c. O funcție (prietenă) care primește ca parametri două obiecte de tip film și află actorii comuni din cele două filme.

Scrieți un program care să exemplifice folosirea cerințelor de mai sus.



## Temă (cont.)

- 5. Fie clasa **Complex** cu 2 attribute re și im.
- Să se implementeze toate metodele (constructori, destructori, operatori) existente în clasa Frație (vezi Laboratorul 2).
- În loc de metodele simplifică, reciproc și cmmdc să se implementeze funcțiile prietene modul și conjugat.