



# 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 6

Moștenirea claselor2



# Probleme propuse/ Teme

## Problema 1

1. Să se implementeze o clasă abstractă **Persoana** cu attributele CNP, nume o metoda abstractă detalii fără parametri și care returnează toate datele obiectului.
  - Să se creeze o clasă **Student** care moștenește clasa Persoana. Un obiect de tip Student trebuie să stocheze următoarele informații: email și nota. Nota va fi accesibilă doar prin intermediul funcției getNota.
  - Implementarea metodei detalii returnează un string de genul "Studentul [nume] are nota [nota].", unde [numele] și [nota] reprezintă valorile câmpurilor. Să se instanțieze un obiect din clasa Student și să se folosească funcția detalii.
  -



# Probleme propuse/ Teme

## Problema 1

- Să se creeze o clasa **Profesor** care moștenește clasa Persoana. Un obiect de tip Profesor trebuie sa stocheze următoarele informații: email și materie. Materia va fi accesibila doar prin intermediul funcției getMaterie. Implementarea metodei detalii returnează un string de
- genul "Profesorul [nume] predă [materia].", unde [numele] și [materia]
- reprezintă valorile câmpurilor. Să se instanțieze un obiect din clasa Profesor și sa se folosească funcția detalii.



# Probleme propuse/ Teme

## Problema 1

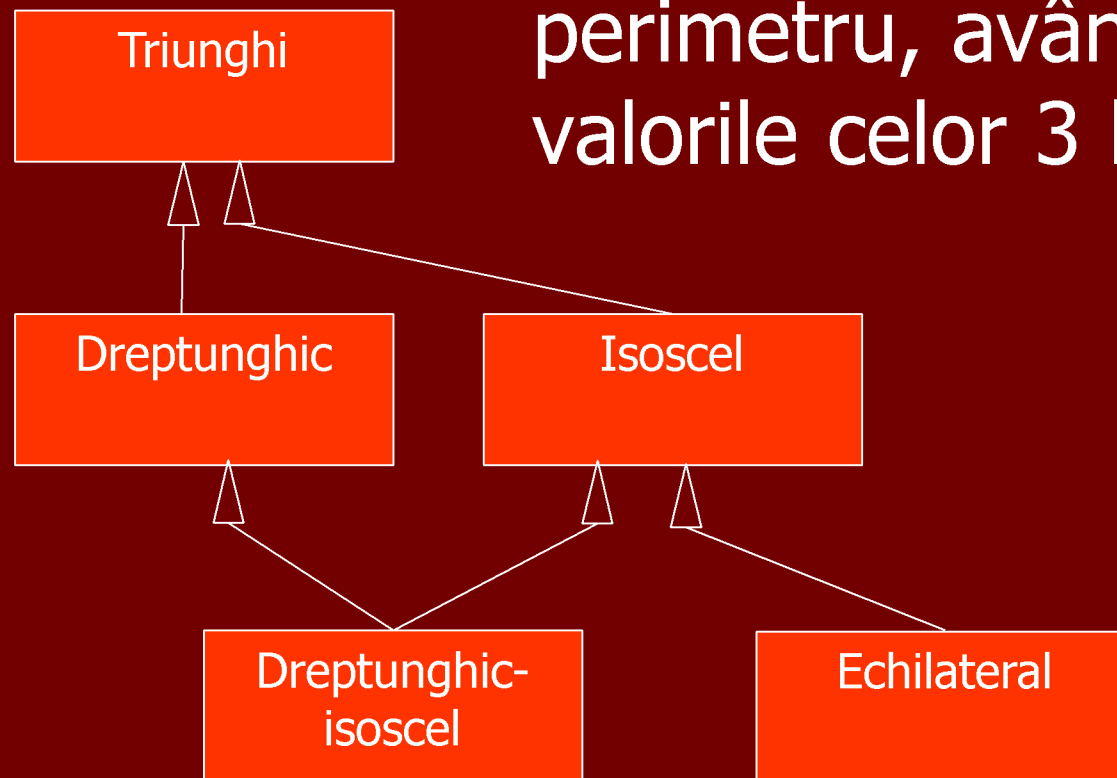
- Să se creeze o clasa **Angajat** care moștenește Persoana. Un obiect de tip Angajat trebuie să stocheze următoarele informații: email și departament. Materia va fi accesibilă doar prin intermediul
- funcției getDepartament. Implementarea funcției detalii returnează un string de genul "Angajatul [nume] face parte din departamentul
- [departament].", unde [numele] și [departament] reprezintă valorile câmpurilor. Să se instanțieze un obiect din clasa Angajat și să se folosească funcția detalii.



# Probleme propuse/ Teme

## Problema 2

2. Să se implementeze următoarea ierarhie de clase. Să se implementeze funcțiile: arie, perimetru, având la dispoziție valorile celor 3 laturi, ca atribute.





# Probleme propuse/ Teme

## Problema 3

- 3. Să se creeze o ierarhie de clase care conține o clasă abstractă **Animal** cu proprietățile *specie*, *vârstă*. cu metoda abstractă *heterotrof* și metode neabstracte get-set pentru attribute. Din această clasă să se extindă două clase **Mamifer** și **Pasăre** cu câte o metoda proprie *naște* și *zboară* și câte un atribut propriu *lungime* respectiv *nrPené*. Get-Set pentru aceste attribute.
- Apoi să se creeze clasa **Ornitorinc** care moștenește atât clasa Reptilă cât și clasa Pasăre, cu o metodă proprie *amfibiu*.
- În metoda principală să se creeze un individ din această specie, apoi să se apeleze toate metodele obiectului definit.



# Probleme propuse/ Teme

## Problema 4

- 4. Să se creeze o ierarhie de clase asemănătoare exemplului Multifuncțională din curs, care să demonstreze moștenirea multiplă în ierarhie de diamant.