



# 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 2

Clase



# Probleme propuse/ Temă

- 1. Fie clasa *Fracție* care implementează tratarea numerelor reale sub formă de fracție.
- Proprietățile clasei vor fi două numere întregi: numărătorul și numitorul fracției.
- Se vor implementa constructori astfel încât el să fie apelabil cu
  - 0 parametri și va inițializa fracția cu 0/1.



# Probleme propuse/Temă

- un singur parametru, reprezentând numărătorul, iar în acest caz numitorul se inițializează cu 1.
- doi 2 parametri, adică numărătorul și numitorul fracției.

Să se folosească valorile implicite ale parametrilor și listele de inițializare

- Să se scrie un alt constructor va fi așa numitul *copy-constructor* care va inițializa fracția cu o altă fracție dată.



# Probleme propuse/ Temă

- Metodele clasei vor fi:
- cele 4 operații de bază: adunarea, scăderea, înmulțirea, împărțirea.
- `cmmdc`, care calculează cel mai mare divizor comun a două numere întregi. Această metodă va fi **statică**. Nu are nicio legătură cu obiectele instanțiate din clasa `Fracție`, ci calculează `cmmdc` a două numere.



# Probleme propuse/Temă

- `simplifica()` care va simplifica fracția și va întoarce rezultatul.
- `reciproc()` care calculează și returnează reciprocul fracției.
- `egal(const Fractie&)` care compară obiectul curent cu un parametru de tip `Fractie` și returnează adevărat, dacă cele două fracții sunt egale, altfel returnează fals.
- `print()` care va afișa fracția sub forma numărător/numitor.



# Temă

- 2. Să se modeleze activitatea unui ghișeu bancar. Sistemul este format din următoarele clase și proprietăți corespunzătoare.
- Banca:
  - codBanca: string
  - nrclienti: unsigned
  - clienti: tablou de elemente de tip Client
- ContBancar:
  - numarCont: string
  - suma: float
  - moneda: string ("RON", "EUR")
- Client:
  - nume: string
  - prenume: string
  - adresă: string
  - nrconturi: unsigned
  - conturi: tablou de tip ContBancar



## Temă (cont.)

- Pentru conturile în RON dobânda se calculează astfel: 0.3% pe sumele sub 500 RON și 0.8%. Conturile în EUR au o dobândă 0.1%.
- Toate conturile moștenesc o **clasă abstractă Operatiuni** cu următoarele metode:





# Temă (cont.)

- public float getSumaTotala()
- (la suma existentă în cont se adaugă dobânda aferentă)
- public float getDobanda()
- public void depunere(float suma)
- public void extragere (float suma)
- Doar între conturi în RON se pot efectua transferuri.
- Metoda transfer se adaugă **doar** în ContBancar.
  - public void transfer(ContBancar& contDest, float suma)



## Temă (cont.)

- Toate clasele vor avea cel puțin 2 constructori (dintre care un constructor de copiere) și metode get și set.
- În clasa principală se vor crea 1 obiect de tip Bancă, 2 clienți. Primul client va avea 1 cont în euro și 1 cont în lei, al doilea 2 conturi în lei. În aceste conturi se vor face toate operațiunile posibile.



## Temă (cont.)

- Fiecare clasă va avea o metodă care afișează datele obiectului pentru care este apelată.
- `afisaredateBanca();` -> va apela `afisaredateClient`
- `afisaredateClient();` -> va apela `afisaredateCont`
- `afisaredateCont();`
- În main se va apela doar `afisaredateBanca`