

Laborator 13 – The Final Lab

Tema: Cont bancar 

Autor: ing. Eduard Donea

Poveste

O banca doreste sa simuleze foarte simplu gestionarea conturilor clientilor. Fiecare cont are un identificator, un titular si un sold. Clientii pot depune bani, pot retrage bani (daca exista fonduri suficiente) si pot vedea istoricul ultimelor operatii. In acest laborator, studentii vor implementa un mini-sistem de tip **cont bancar**, folosind clase C++ si concepte de baza POO.

Scopul laboratorului

- definirea si utilizarea mai multor clase in C++;
 - utilizarea compozitiei (un cont contine operatii);
 - alocare dinamica pentru un tablou simplu de operatii;
 - metode publice pentru depunere si retragere;
 - testare rapida in `main.cpp`.
-

Structura proiectului

- **Operatie** – o operatie bancara simpla (tip + suma);
 - **ContBancar** – contul unui client (titular, sold, operatii);
 - **main.cpp** – fisier de testare.
-

Clasa Operatie

Clasa `Operatie` descrie o operatie simpla din cont si contine:

- un tip (ex: "DEPUNERE" sau "RETRAGERE");
- o suma (valoare pozitiva).

Se cere:

- constructor cu parametri (tip, suma);
- metode de acces (getters) sau o metoda de afisare.

Clasa ContBancar

Clasa `ContBancar` va contine:

- un ID (int);
- numele titularului (sir de caractere / `std::string`);
- sold (float/double);
- un tablou dinamic de operatii (de exemplu, maxim 10 operatii).

Se cere:

- constructor cu parametri (id, titular, sold initial);
- destructor (daca s-a folosit memorie alocata dinamic);
- metoda `depunere(suma)` (actualizeaza soldul si salveaza operatia);
- metoda `retragere(suma)` (verifica fonduri; daca nu sunt suficiente, nu modifica soldul);
- metoda `printExtras()` care afiseaza contul si lista operatiilor salvate.

Nota: Pentru simplitate, se poate salva doar un numar fix de operatii (ex: 10). Daca se depaseste limita, se pot ignora operatiile noi sau se pot suprascrie cele mai vechi.

Fișierul main.cpp

Fișierul `main.cpp` este folosit pentru testarea functionalitatii.

Model minim de test:

- creati un obiect `ContBancar`;
- efectuati 2 depuneri si 2 retrageri (una dintre retrageri poate fi invalida);
- afisati extrasul de cont folosind `printExtras()`.

Cerințe (10p)

1. (3p) Implementarea clasei `Operatie`.
2. (4p) Implementarea clasei `ContBancar` (date + metode).
3. (2p) Gestionarea corecta a soldului si validarea retragerilor.
4. (1p) Demonstratie in `main.cpp`.

Compilare și rulare

```
g++ -std=c++17 *.cpp -o lab13
./lab13
```

Bonus GitHub Classroom

Pentru obtinerea punctajului bonus:

1. Acceptati tema de pe GitHub Classroom: <https://classroom.github.com/a/pAubWfK>
2. Implementati rezolvarea completa.
3. Incarcati proiectul in repository-ul generat.

Atenție: Repository-urile sunt verificate automat. Codul copiat intre studenti duce la anularea punctajului bonus.