

Recapitulare finală

Se consideră o aplicație software de tip Banking, care are rolul de a gestiona conturi bancare pentru clienți. Un cont bancar este caracterizat prin: soldul (număr real), valuta (enumerație) și titularul (șir de caractere alocat dinamic).

Aplicația permite operații uzuale precum: depunere, retragere și afișarea informațiilor contului, precum și calcularea unor valori financiare asociate contului, în funcție de regulile specifice fiecărui tip de cont.

În cadrul aplicației, anumite tipuri de conturi pot avea comportamente diferite față de un cont standard (de exemplu, pot calcula dobândă sau pot impune reguli suplimentare), iar aceste diferențe trebuie modelate folosind concepte de Programare Orientată Obiect.

Cerințe

- 1) Să se definească și să se implementeze minimul necesar pentru a putea crea obiecte de tip ContBancar. (1p)
- 2) Implementați metodele adaugareCash și respectiv retragereNumerar. Ambele metode primesc ca parametru suma care trebuie introdusă sau retrasă. Atenție ca retragerile să aibă sens logic, adică să nu se poată retrage mai mult decât soldul curent ș.a.m.d. (1p)
- 3) Să se implementeze operatorul de afișare << pentru clasa ContBancar. (1p)
- 4) Să se implementeze operatorii == și != care verifică dacă două conturi sunt identice respectiv diferite. Două conturi se consideră a fi identice doar dacă au același sold și respectiv aceeași valută (1p)
- 5) Specializați conturile bancare astfel încât să poată fi benefice și pentru clienți atunci când vine vorba de economii. Un cont de economii are ca și atribut specific rata dobânzii care este un număr real cuprins între 0 și 1. (2p)
- 6) Calculați dobânda implementând o metodă denumită privată calculDobanda care va fi folosită în calcularea sold-ului unui cont de economii. Atunci când returnați soldul unui cont asigurați-vă că decizia se ia la momentul execuției. Astfel pentru un ContBancar se returnează direct soldul iar pentru un ContEconomii se returnează soldul + rezultatul întors de metoda calculDobanda. (1p)
- 7) În funcția main testați clasa ContBancar prin construirea a două obiecte și apelați metodele și operatorii implementați. (1p)
- 8) Tot în funcția main construiți un vector neomogen de obiecte de tip ContBancar și populați-l cu minimum 5 itemi. Sortați acest vector după soldul contului (crescător sau descrescător). (1p)

Se acordă un punct din oficiu. Succes

Precizări

- Clasele se vor implementa în fișiere .h și .cpp. Iar testarea codului se va face în funcția main care va fi într-un fișier separat.
- Neimplementarea în fișiere .h și .cpp atrage după sine pierderea unui punct.
- Nu este permisă marcarea câmpurilor claselor părinte cu protected.
- Memory leak-urile atrag fiecare după sine pierderea unui punct.
- Alte depunctări cu privire la modul de implementare sau la corectitudinea declarării metodelor, rămân la latitudinea asistentului care vă corectează.
- Erorile de compilare pot aduce maximum nota 5 din punctajul laboratorului curent.

Dacă nu s-a specificat vreun mod de implementare a cerințelor rămâne la latitudinea voastră cum vreți să procedați în rezolvare.