

FACULTATEA CALCULATOARE, INFORMATICA SI MICROELECTRONICA
UNIVERSITATEA TEHNICA A MOLDOVEI

ANALIZA SI PRINCIPIILE PROGRAMARII ORIENTATE PE OBIECT
LUCRAREA DE LABORATOR#1

Implementare principiilor OOP

Autor:

Ana SUTREAC

lector asistent:

Mihail PECARI

lector superior:

Andrei POSTARU

Laboratory work #1

1 Scopul lucrării de laborator

În cadrul laboratorului e nevoie de realizat un software care o să conțină implementarea celor patru principii OOP (abstractizarea, mostenirea, încapsularea, polimorfism).

2 Obiective

- implementarea principiilor OOP
- abstractizare
- polimorfism
- încapsulare
- mostenire

3 Laboratory work implementation

3.1 Analiza lucrării de laborator

Link la repozitoriu - https://github.com/anashutrac/APPOO_Labs

Tema aleasa pentru laboratorul nr.1 este un cont in banca si datele asociate.

- abstractizare - este posibilitatea ca un program să ignore unele aspecte ale informației pe care o manipulează, adică posibilitatea de a se concentra asupra esențialului. Fiecare obiect în sistem are rolul unui “actor” abstract, care poate executa acțiuni, își poate modifica și comunica starea și poate comunica cu alte obiecte din sistem fără a dezvălui cum au fost implementate acele facilități. Procesele, funcțiile sau metodele pot fi de asemenea abstracte, și în acest caz sunt necesare o varietate de tehnici pentru a extinde abstractizarea

- polimorfism - numită și ascunderea de informații: Asigură faptul că obiectele nu pot schimba starea internă a altor obiecte în mod direct (ci doar prin metode puse la dispoziție de obiectul respectiv); doar metodele proprii ale obiectului pot accesa starea acestuia. Fiecare tip de obiect expune o interfață pentru celelalte obiecte care specifică modul cum acele obiecte pot interacționa cu el.

- încapsulare - este abilitatea de a procesa obiectele în mod diferit, în funcție de tipul sau de clasa lor. Mai exact, este abilitatea de a redefini metode pentru clasele derivate. De exemplu pentru o clasă Figura putem defini o metodă arie. Dacă Cerc, Dreptunghi, etc. ce vor extinde clasa Figura, acestea pot redefini metoda arie.

- moștenire - organizează și facilitează polimorfismul și încapsularea, permițând definirea și crearea unor clase specializate plecând de la clase (generale) deja definite - acestea pot împărtăși (și extinde) comportamentul lor, fără a fi nevoie de a-l redefini. Aceasta se face de obicei prin gruparea obiectelor în clase și prin definirea de clase ca extinderi ale unor clase existente. Conceptul de moștenire permite construirea unor clase noi, care păstrează caracteristicile și comportarea, deci datele și funcțiile membru, de la una sau mai multe clase definite anterior, numite clase de bază, fiind posibilă redefinirea sau adăugarea unor date și funcții noi. Se utilizează ideea: ”Anumite obiecte sunt similare, dar în același timp diferite”. O clasă moștenitoare a uneia sau mai multor clase de bază se numește clasă derivată. Esența moștenirii constă în posibilitatea refolosirii lucrurilor care funcționează.

Concluzie

Aici trebuie sa fie concluzia ta.

References

- 1 Aldebran Robotics, *official page*, www.aldebaran.com/en
- 2 Timo Ojala, *Multiresolution gray-scale and rotation invariant texture classification with local binary patterns*, 2002
- 3 Biometric, www.biometricupdate.com/201501/history-of-biometrics