Facultatea Calculatoare, Informatica si Microelectronica

Universitatea Tehnica a Moldovei

Analiza, proiectarea și programarea orientată pe obiecte

Lucrarea de laborator#1

Principiile POO

Autor:
Marina JECHIU

lector asistent:
Mihail PECARI

Laboratory work #1

1 Scopul lucrarii de laborator

Scopul lucrării de laborator este de a implementa într-un limbaj de programare o serie de clase și de a flosi în cadrul lor toate principiile POO.

2 Obiective

Obiectivele lucrării de laborator sunt:

- Crearea claselor;
- Utilizarea principiului încapsulării;
- Utilizarea principiului moștenirii;
- Utilizarea principiului abstractizării;
- Utilizarea principiului polimorfismului.;

3 Noțiuni teoretice

Programarea orientată pe obiecte presupune:

- determinarea şi descrierea claselor utilizate în program;
- crearea exemplarelor de obiecte necesare;
- determinarea interacţiunii dintre ele.

Abstractizarea reprezintă posibilitatea ca un program să ignore unele aspecte ale informaţiei pe care o manipulează, adică posibilitatea de a se concentra asupra esenţialului. **Încapsularea** reprezintă ascunderea de informații (data hiding). Obiectele nu pot schimba starea internă a altor obiecte în mod direct (ci doar prin metode puse la dispoziţie de obiectul respectiv). **Moștenirea** permite definirea şi crearea unor clase specializate plecând de la clase (generale) care sunt deja definite. Permite construirea unor clase noi, care păstrează caracteristicile şi comportarea, deci datele şi funcţiile membru, de la una sau mai multe clase definite anterior, numite clase de bază, fiind posibilă redefinirea sau adăugarea unor date şi funcţii noi. **Polimorfismul** se manifestă atunci când mai multe funcţii pot avea acelaşi nume în acelaşi domeniu de definiţie, dacă se pot diferenţia prin numărul sau tipul argumentelor de apel sau când un obiect poate lua mai multe forme.

4 Laboratory work implementation

4.1 Analiza lucrarii de laborator

https://github.com/MarinaJechiuTI154/APPO/

Pentru realizarea sarcinilor înaintate spre îndeplinire a fost utilizat limbajul Java, un limbaj specific paragmei programării orientate pe obiecte. Pentru o structurare logică mai bună, fiecare funcțional a fost integrat într-un pachet separat.

Tema aleasă pentru crearea claselor este o aplicație de evidență contabilă a stocurilor. Astfel, principalele entități ce interacționează sunt: stores (materiale, OMVSD, produse și mărfuri), persons (aici intră atât utilizatorii sistemului, cât și simpli participanți în operațiunile economice) și documents(documentele de evidență a operațiunilor economice).

Pentru a exemplifica funcționalitatea acestor obiecte, cât și relațiile dintre acestea au fost create obiecte concrete. Pentru o evidență sistematică a produselor create aceastea sunt stocate în ArrayList-uri. Avantajele folosirii acestor colecții este ușurința de stocare, prelucrare și ștergere a elementelor în cadrul lor.

Principiul încapsulării a fost utilizat în cadrul tuturor claselor, prin crearea tuturor parametrilor private. Moștenirea a fost utilizată între clasa Document și tipurile de documente, cât și între Product și tipurile de produse. Abstractizarea a fost exemplificată prin crearea clasei abstracte Person și moștenirea ei de către clasa Administration(utilizatorii care au acces]n cadrul programului) și Employee. Polimorfismul a fost utilizat în clasa main, prin crearea instanțelor unor subclase, prin declararea lor de tipul superclasei. Totodată, mai este realizată și prin intermediul funcției print, care a fost suprascrisă în subclasele clasei Document.

În figurile de mai jos sunt prezentate imagini ce exemplifică funcționalitatea aplicației.

4.2 Imagini

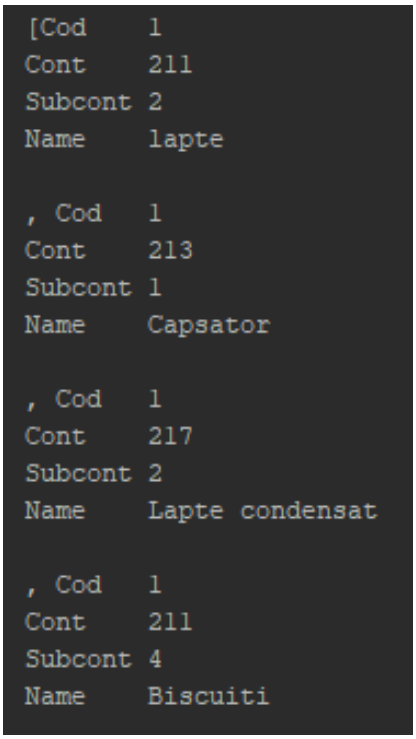

Figure 4.1– Afișarea tuturor produselor

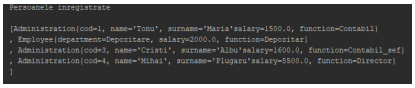

Figure 4.2– Afișarea utilizatorilor înregistrați în sistem

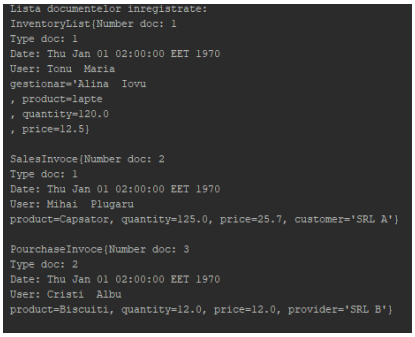

Figure 4.3– Afișarea documentelor create

Concluzie

Pentru efectuarea lucrării de laborator 1 ca limbaj de programare a fost ales limbajul Java. Acesta este un limbaj de nivel înalt, specific paradigmei POO. Tema aleasă pentru realizarea laboratorului este o aplicație de evidență contabilă a stocurilor de produse. În acest sens, au fost create clase pentru principalele obiecte care interacționează din pounct de vedere economic.

Sarcina de bază de implementare a celor 4 principii POO a fost respectată și implementată. Totodată, pentru a arăta funcționalitatea claselor au fost crate instanțe ale acestora, fiind atașate imagini ale funcționalității. Codul este încărcat pe platforma GitHub, și este disponibil pe repozitoriul APPO, indicat în link-ul din capitolul 4.