

Academia de Studii Economice din București

Facultatea de Cibernetică, Statistică și Informatică Economică

Academia de Studii Economice din București

Facultatea de Cibernetică, Statistică și Informatică Economică

Proiect PPOO

Gestiunea unui magazin online

**Student:**

Comorașu Claudiu-Adrian

**Profesor coordonator:**

Prof. Univ. Dr. Cristian Ciurea

București

2022

Cuprins

[Introducere 3](#_Toc119101590)

[Implementare 3](#_Toc119101591)

[Lista de figuri 7](#_Toc119101592)

[Bibliografie 7](#_Toc119101593)

# Introducere

Această aplicație dorește să simuleze gestiunea logică și eficientă a unui magazine online. Modul de gandire a fluxului a acestei aplicații a fost creat din perspectiva administratorului care poate să gestioneze stocurile, produsele, dar si să creeze rapoarte ca să țină evidența produselor în funcție de categorie si producator.

Această aplicație, după cum am specificat și mai sus, este destinată administratorilor pentru gestiune eficientă și persistentă a produselor.

Lucrarea de față este alcătuită dintr-un singur capitol în care sunt descriese tehnologiile folosite, cât și modul de funcționare și implementare a acestora.

# Implementare

Modul în care aplicația a fost gândită a fost ca aceasta să fie accesată prin consolă, de aceea am utilizat limbajul de programare Java. Acesta este un limbaj de programare de nivel înalt, bazat pe clasă, orientat pe obiecte, care este proiectat să aibă cât mai puține dependențe de implementare. Este un limbaj de programare de uz general destinat să permită programatorilor să scrie o dată, să ruleze oriunde, ceea ce înseamnă că, codul Java, compilat poate rula pe toate platformele care acceptă Java fără a fi nevoie să recompileze.

Persistența datelor utilizate se realizează prin intermediul fișierul binar **listaProduse.dat.** Datele sunt deserializate cand aplicația este pornită înainte ca meniul să fie afișat utilizatorului și sunt serializate când utilizatorul dorește să părăsească aplicația, serializare ce se realizeaza cu modificările făcute de utilizator.

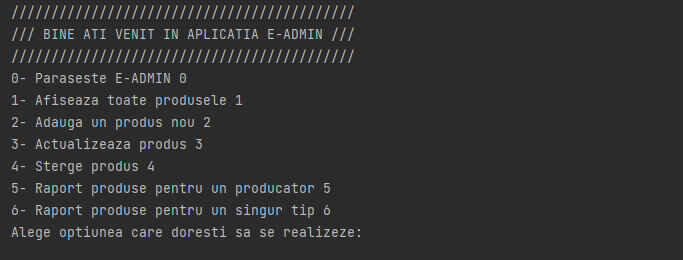


Figure 1 - Menu

Prin apăsarea tastei “1” în meniu, se vor afișa toate produsele ce se afla în “depozit”. Toate produsele se află în obiectul **ListaProduse** ce a fost creat cu ajutorul **singleton** pentru a avea o gestiune centralizată și ca lista de produse să fie comună pentru toată aplicația. Obiectul ListaProduse contine atributul **ArrayList<Produs>**, pe lângă aceasta produsele sunt adăugate și colecției **LinkedHashMap<Integer, Produs>**. De aceea, parcurgem lista de produse printr-un **for** și afișăm fiecare produs.



Figure 2 - Afișare produse

Prin apăsarea tastei “2” în meniu, se permite adaugarea unui nou produs. Datele produsului sunt introduse pe rând, într-o ordine logică, având chiar optiunea de anulare a comenzii la fiecare încercare de a completa un camp. Fiecare proprietate este validată de fiecare dată când sunt introduse în aplicație cu ajutorul claselor standard de excepții, dar și a excepției specifice **InvalidValueException.**



Figure 3 - Adaugă produs

Prin apăsarea tastei “3” în meniu, se poate actualiza un anumite produs în funcție de numărul acestuia. Produsele fiind adăugate într-o colecție de tipul **LinkedHashMap<Integer, Produs>**, putem să folosim cheia (index de tipul Integer) pentru a putea accesa produsul pe care doriim să o actualizăm. Utilizatorul trebuie să aleaga codul corespunzator în tabelul hash și poate schimba fiecare câmp în parte.



Figure 4 - Actualizare produs

Prin apăsarea tastei “4” în meniu, se poate șterge un anume produs, în funcție de numarul produsului (asemenea funcționalității de actualizare).



Figure 5 - Șterge produs

Prin apăsarea tastei “5” în meniu, se creează un raport în funcție de un producator, pentru a putea observa toate produsele pe care magazinul le deține de la producătorul introdus de către utilizator, acestea fiind păstrate în colecția **LinkedList<Produs>.** Mai mult după creearea raportului, utilizatorul poate să aleagă dacă dorește ca acesta să fie exportat înntr-un fișier în format .txt.



Figure 6 - Creeare raport producator

Prin apăsarea tastei “6” în meniu, se creeaza un raport în funcție de tipul produsului, pentru a putea observa toate produsele de un anumit tip introdus de utilizator. Datele sunt validate și păstrate într-o colecție de tipul **HashMap<String, TipProdus>** care este static, pentru a putea fi accesat prin cadrul enum-ului.



Figure 7 - Creeare raport tip produs

# Lista de figuri

[*Figure 1 - Menu* 3](#_Toc119101524)

[*Figure 2 - Afișare produse* 4](#_Toc119101525)

[*Figure 3 - Adaugă produs* 4](#_Toc119101526)

[*Figure 4 - Actualizare produs* 5](#_Toc119101527)

[*Figure 5 - Șterge produs* 5](#_Toc119101528)

[*Figure 6 - Creeare raport producator* 6](#_Toc119101529)

[*Figure 7 - Creeare raport tip produs* 6](#_Toc119101530)

# Bibliografie

1. https://en.wikipedia.org/wiki/Java\_(programming\_language)
2. https://computinglearner.com/how-to-create-a-java-console-menu-application/
3. https://www.geeksforgeeks.org/file-handling-in-java/
4. https://www.geeksforgeeks.org/java-util-hashmap-in-java-with-examples/
5. https://www.javatpoint.com/java-linkedhashmap