PROIECT

Programare Orientată pe Obiecte

GESTIUNE BIBLIOTECĂ

Student Profesor coordonator

Filip Leonard dr. Ing. Prodan Remus

Specializarea: Calculatoare

Grupa : 3121B

Anul: II

CUPRINS

[1. Formularea temei: 3](#_Toc104906692)

[2. Motivul alegerii: 3](#_Toc104906693)

[3. Elemente teoretice 4](#_Toc104906694)

[4. Implementare 5](#_Toc104906695)

[5. Manual de utilizare 5](#_Toc104906696)

[6. Bibliografie 9](#_Toc104906697)

Tema și motivația alegerii

# Formularea temei:

**Voi realiza o aplicație în limbajul de programare C + +**, ce va implica utilizarea paradigmei de programare “orientată pe obiecte” și ce **va implementa gestiune a unei** biblioteci (stocare carți în baza de date, închiriere cărți, restituire carte, etc.).

# Motivul alegerii:

Am ales această temă de proiect deoarece mi s-a părut ceva interesant de realizat, ceva nou, cu care chiar aș putea să ajut un bibliotecar să își ușureze munca și să nu mai țină evidența cărților unei biblioteci pe sute de hârtii, ci doar să stea și să utilizeze această aplicație și calculatorul să facă munca în locul lui. În loc să stea să rasfoiască mulțimea de foi și să irosească un timp îndelungat pentru căutarea unei cărți sau al unui client, acesta poate cu doar câteva taste și intr-un timp foarte scurt să gasească ceea ce dorește.

Această temă mi s-a părut interesantă din punct de vedere al realizabilității, ceva ce părea greu la început, ceva din care aveam să învăț lucruri noi, și să aprofundez **“Programarea Orientată pe Obiecte”**.

# 3. Elemente teoretice

1.1. Descrierea problemei

Această aplicație implică o serie de aspecte ce trebuiesc implementate, cum ar fi:

* introducerea, stocarea cărților și tuturor datelor despre cărți într-o bază de date locală, ce va putea fi accesată mereu, independentă de conexiunea la rețeaua de internet.
* modificarea datelor despre cărțile ce au fost introduse cu unele greșeli;
* afișarea cărților introduse într-o anumită ordine prestabilită;
* căutarea unei cărți după titlu sau după numele autorului de acare aceasta a fost scrisă;
* lista de autori(afisarea tuturor cartilor scrise de acelasi autor);
* cautarea cartii dupa tagguri;
* introducerea clienților și salvarea acestora în baza de date;
* posibilitatea de închiriere a unei sau mai multor cărți de catre același client;
* afișarea tuturor clienților existenți în baza de date;
* restituirea cărților care au fost închiriate;
* afisarea numarului de carti disponibile;
* cautare dupa un nume(oarecare);

In proiect voi folosi 3 entitati: carte, client si functii.

1.2. Abordarea teoretică a problemei

Problema realizării acestei aplicații se pune atunci când trebuie sa decidem ce tip de dată trebuie utilizat pentru a stoca informațiile despre o carte sau un client sub forma unei singure entități. Apoi trebuie ales modul în care sunt manipulate datele în timpul execuției, cum se efectuează operațiile pe care le selectează utilizatorul, cum se scrie și cum se citește din fișier și citirea de la tastatură plus afișarea pe ecran.

Pentru a putea introduce, stoca, modifica și reintroduce date cât mai eficient în program, voi implementata o bază de date locală cu ajutorul fișierelor text.

Toate informațiile vor fi puse în două fișiere text, unul în care sunt manageriate datele despre cărți și un altul pentru clienți.

1.3. Elemente specifice POO

Pentru a putea introduce noțiunea de POO (**“Programare Orientată pe Obiecte”**) am folosit conceptul de **clasă**.

**Clasa** reprezintă un tip de dată abstractă ce încapsulează atât elemente de date (date membre) pentru care s-a adoptat un anumit stil de reprezentare, cât și operații asupra datelor (metode).

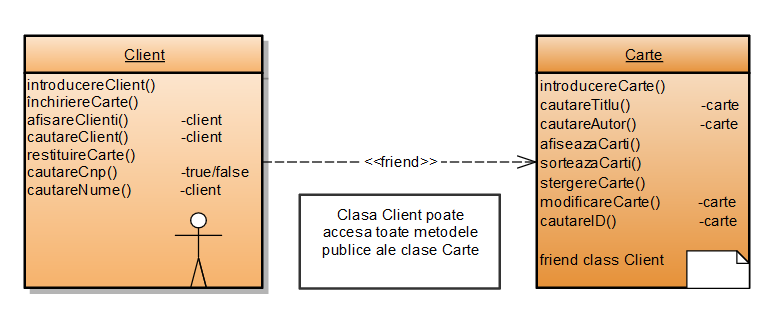
**Înacpsularea** este proprietatea datelor de a-și ascunde o parte din date și din metode. Din exteriorul obiectului sunt accesibile numai datele și metodele publice. Obiectul se comportă ca și cum ar avea două învelișuri: unul transparent care permite accesul la datele și metodele publice ale obiectului, și un al doilea înveliș opac, care cuprinde datele și metodele invizibile (neaccesibile) din exterior.

# Implementare

* 1. Tehnologii folosite

Aplicația este realizată în limbajul C++, folosind Programarea Orientată pe Obiecte, adică utilizarea conceptului de **clasă** si **obiect**, folosirea funcțiilor **friend** (prietene), supradefinirea operatorilor.

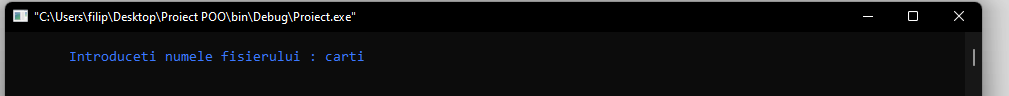
Am utilizat mediul de programare ***Code::Blocks*** ce are o interfață prietenoasă și permite folosirea multor instrumente specifice programării.



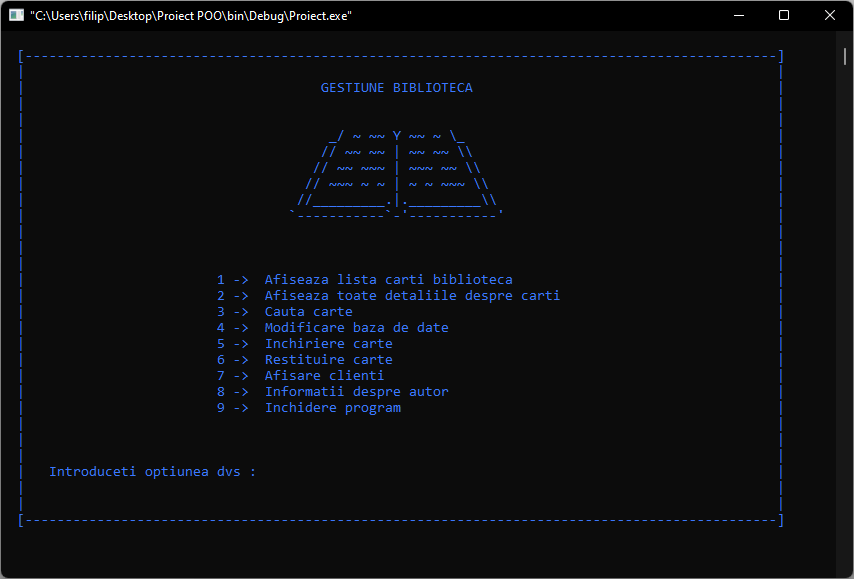
# Manual de utilizare

Rularea executabilului va duce la deschiderea unei ferestre, unde utilizatorul trebuie să introducă numele fișierului în care sunt toate cărțile. Fișierul este deja existent și se numește “carti.txt”.

***Se va introduce numele fișierului fără extensie !!! După apăsarea tastei ENTER va apărea meniul principal.***

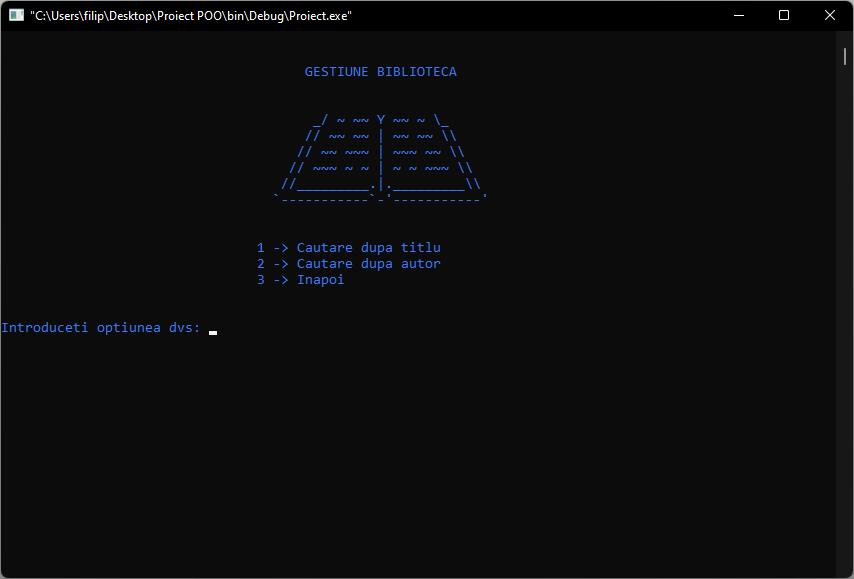


**Meniul Principal**



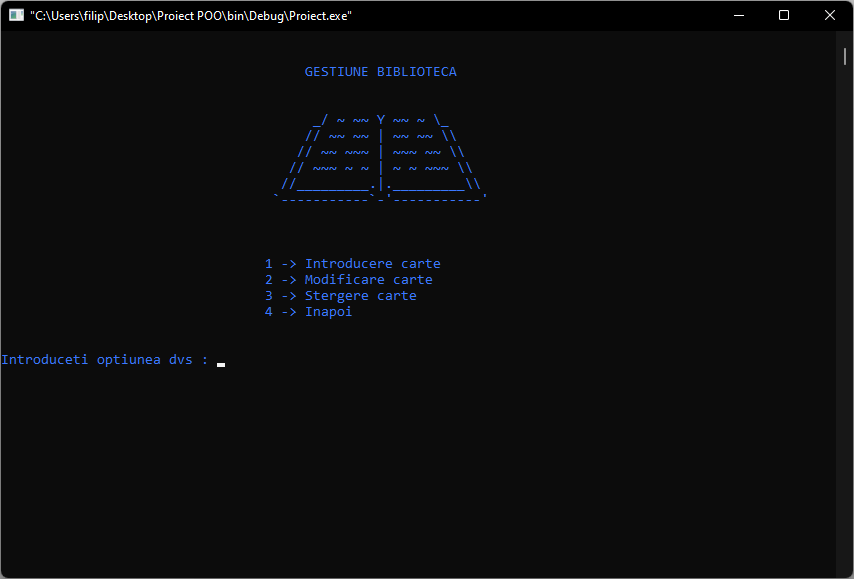
În dreptul fiecărei opțiuni este cate un număr, iar utilizatorul va introduce numărul aferent opțiunii dorite, urmat de tasta Enter.

La introducerea optiunii „**3. Cauta carte**” , tot în aceeași fereastră va aparea alt meniu în care sunt disponibile 3 optiuni: “**cautare dupa titlu**”, “**cautare dupa autor**” și „**înapoi la meniul principal**”.

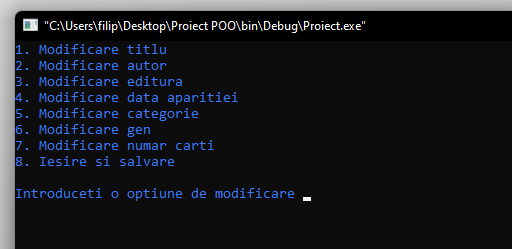


La introducerea **opțiunii 4** se va afișa următorul meniu, unde aveți posibilitatea de a introduce o carte, de a modifica o carte deja existentă în baza de date, trebuie doar să consultați detaliile despre cărți și să rețineți ISBN-ul cărții ce doriți să o modificați.

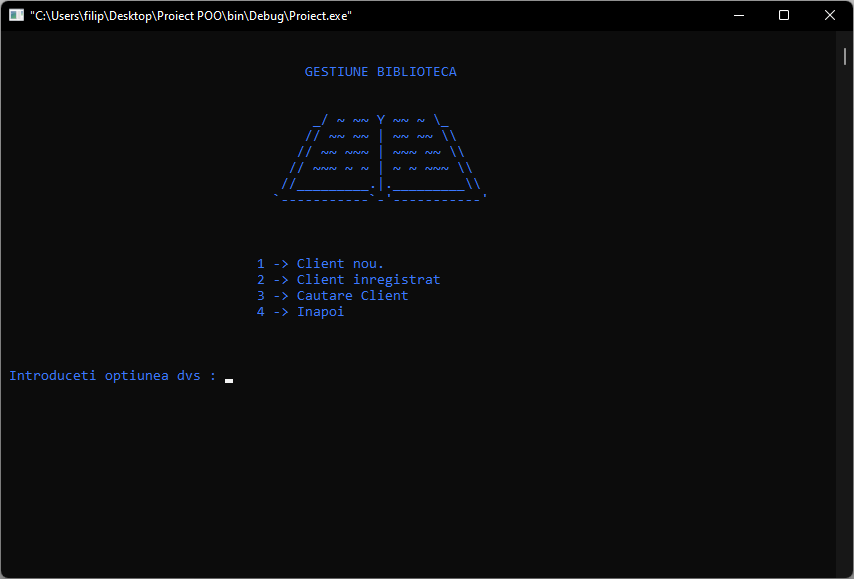
ISBN-ul este unic pentru fiecare stoc de cărți identice.



Submeniul “Modificare carte”:



Submeniul „Închiriere carte “:



**Concluzie**

Aplicaţia concepută îndeplineşte cerinţele prezentate la începutul prezentării, este usor de utilizat şi prezintă nume sugestive pentru fiecare optiune implementată.

# Bibliografie

1. <https://www.educative.io/blog/object-oriented-programming>
2. [copy - C++ Reference (cplusplus.com)](http://www.cplusplus.com/reference/algorithm/copy/)
3. [c++ - Which header file I need to include to use gotoxy() function? - Stack Overflow](https://stackoverflow.com/questions/7955145/which-header-file-i-need-to-include-to-use-gotoxy-function)