***UNIVERSITATEA “STEFAN CEL MARE”, SUCEAVA***

***FACULTATEA DE INGINERIE ELECTRICA SI STIINTA CALCULATOARELOR***

***SPECIALIZAREA CALCULATOARE***

***PROIECT DISCIPLINA POO***

***Biblioteca. Gestiunea cartilor in C++***

***Proiect realizat de: Joroveanu Alexandra***

alexandra.joroveanu@student.usv.ro

***Calculatoare anul 2***

***semigrupa 3121A***

Cuprins

[TEMA ȘI MOTIVATIA ALEGERII 3](#_Toc137649254)

[CAPITOLUL I 3](#_Toc137649255)

[I. DESCRIEREA PROBLEMEI 3](#_Toc137649256)

[II. ABORDAREA TEORETICĂ A PROBLEMEI 3](#_Toc137649257)

[III. ELEMENTELE SPECIFICE POO 3](#_Toc137649258)

[CAPITOLUL II 4](#_Toc137649259)

[TEHNOLOGII UTILIZATE 4](#_Toc137649260)

[CAPITOLUL III 4](#_Toc137649261)

[ANALIZA SOLUTIEI IMPLEMENTATE 4](#_Toc137649262)

[Concluzii 6](#_Toc137649263)

[Bibliografie 7](#_Toc137649264)

# TEMA ȘI MOTIVATIA ALEGERII

Tema proiectului este “Biblioteca. Gestiunea cărților”. Pe parcursul proiectului se va dezvolta o aplicație ce are ca scop principal gestiunea cărților dintr-o bibliotecă.

La alegerea acestei teme a contribuit faptul că în biblioteca personală se află un număr destul de mare de cărți care au fost împrumutate unor cunoștințe. Fiind destul de multe, nu mai țineam minte ce cărți am dat persoanelor și cu cât timp în urmă le dăduserăm. Așa că m-am decis că ar fi util dacă aș utiliza un sistem asemănător cu cel al bibliotecilor mai mari.

# CAPITOLUL I

ELEMENTE TEORETICE

## DESCRIEREA PROBLEMEI

Aplicația implementată este o bibliotecă personală în care se memorează detalii despre cărțile aflate în aceasta, date despre cine a împrumutat cartea respectivă și data la care a fost făcut împrumutul. De asemenea, legat de carte se cunosc titlul acesteia, autorul, editura la care a fost publicată, anul publicării și numărul de pagini. Despre persoana în posesia căreia ajunge cartea se cunosc numele, prenumele precum și vârsta.

## ABORDAREA TEORETICĂ A PROBLEMEI

Aplicația este implementată cu ajutorul programării orientate pe obiect, aceasta este dezvoltată cu ajutorul claselor, obiectelor, metodelor, moștenirii.

Conceptul de bază al programului se numeşte clasa. Clasa are un comportament asemănător structurilor de date abstracte, dar avantajul ei este ca aceste clase sunt mai uşor de implementat.

În c++ întâlnim noţiunea de derivare, care este în fapt o abstractizare a noţiunii de moştenire. O clasă care adaugă proprietăţi noi la o clasă deja existentă, vom spune ca este derivată din clasa originală. Clasa originală poarta denumirea de clasă de bază.

In cadrul acestui proiect am utilizat:

* Supradefinirea operatorilor
* Stucturi de date
* Mostenirea
* Clase

## ELEMENTELE SPECIFICE POO

Programarea orientată pe obiecte (POO), uneori denumită ca și în limba engleză, Object Oriented Programming (OOP)) este o [paradigmă de programar](https://ro.wikipedia.org/wiki/Paradigm%C4%83_de_programare)e, axată pe ideea încapsulării, adică grupării datelor și codului care operează asupra lor, într-o singură structură. Un alt concept important asociat programării orientate obiect este polimorfismul, care permite abstractizări ce permit o descriere conceptuală mai simplă a soluției. Principalul element al programării orientate pe obiect este clasa. Clasa are un comportament asemănător structurilor de date abstracte, dar avantajul ei este ca aceste clase sunt mai uşor de implementat.

Clase utilizate:

* Clasa Carte
* Clasa Cititor
* Clasa Împrumut

Un alt concept pe care îl vom utiliza este cel de ”friend”. Într-o definiție de clasă, utilizați cuvântul cheie ”friend” și numele unei funcții non-membru sau al unei alte clase pentru a acorda acces membrilor privați și protejați ai clasei.

# CAPITOLUL II

## TEHNOLOGII UTILIZATE

Pentru dezvoltarea aplicației s-a folosit ca IDE CodeBlocks, iar limbajul de programare folosit este C++.

# CAPITOLUL III

## ANALIZA SOLUTIEI IMPLEMENTATE

Pentru început vom crea cele trei clase și vom declara funcțiile care urmează să fie implementate.

Pentru clasa Carte:

O imagine care conține text, captură de ecran, Font

Descriere generată automat

Pentru clasa Cititor:

O imagine care conține text, captură de ecran, Font

Descriere generată automat

Pentru clasa Împrumut:

O imagine care conține text, captură de ecran, Font

Descriere generată automat

În cadrul proiectului vom utiliza și conceptul de friend:

Pentru clasa Carte:

O imagine care conține text, captură de ecran, Font

Descriere generată automat

Pentru clasa Cititor:

O imagine care conține text, captură de ecran, Font

Descriere generată automat

Pentru clasa Împrumut:

O imagine care conține text, captură de ecran, Font, linie

Descriere generată automat

Cum am menționat și anterior am folosit și conceptul de moştenire .

O imagine care conține text, Font, captură de ecran

Descriere generată automatDupă ce vom implementa toate funcțiile dorite, ne vom axa pe meniul principal al programului. Acesta va fi sub forma unui meniu care apela opțiunile dorite. Pentru început se va dezvolta un meniu principal:

Apoi, pentru fiecare opțiune se va crea un alt meniu.

O imagine care conține text, Font, captură de ecran, număr

Descriere generată automatO imagine care conține text, Font, captură de ecran, tipografie

Descriere generată automatO imagine care conține text, Font, captură de ecran

Descriere generată automat

Optiunea 1 Optiunea 2 Optiunea 3

# Concluzii

Am implementat in C++ aplicația “Aplicatie biblioteca” cu ajutorul programării orientate pe obiect și am aprofundat limbajul în același timp. Pe parcursul implementării am avut multe erori si bug-uri pe care am fost nevoită sa le rezolv astfel am invatat mai multe despre C++.

Elemente de programare orientata pe obiect folosite:

* Clase
* Mostenire
* Polimorfism

Aplicația poate fi îmbunătățită prin tratarea mai multor excepții sau prin ștergerea automată din fișier a informațiilor care trec de data curentă.

# Bibliografie

<http://apollo.eed.usv.ro/~remus/>

[friend (C++) | Microsoft Learn](https://learn.microsoft.com/en-us/cpp/cpp/friend-cpp?view=msvc-170)

[Curs 7 - Polimorfism.ppt [Compatibility Mode] (ucv.ro)](http://inf.ucv.ro/~mihaiug/courses/poo/slides/Curs%2007%20-%20Polimorfism.pdf)

[OOP Concepts for Beginners: What is Polymorphism (stackify.com)](https://stackify.com/oop-concept-polymorphism/)