UNIVERSITATEA ,,Ștefan cel Mare”, SUCEAVA

FACULTATEA DE INGINERIE ELECTRICĂ ȘI ȘTIINȚA CALCULATOARELOR

SPECIALIZAREA CALCULATOARE

PROIECT DISCIPLINA POO

GESTIUNE INCHIRIRERE MASINI

Autor

Studentul Varvaroi Mircea

Grupa

3133a

- SUCEAVA 2023 -

# TEMA ȘI MOTIVAȚIA ALEGERII

O aplicație de gestionare a închirierii de mașini este o platformă software specializată care permite companiilor de închirieri auto să gestioneze și să monitorizeze întregul proces al închirierilor. Această aplicație oferă funcționalități esențiale pentru administrarea eficientă a flotei de mașini disponibile pentru închiriere și pentru a oferi clienților o experiență fluidă.

Motivația din spatele alegerii implementării unei aplicații de gestionare a închirierilor de mașini poate proveni dintr-o serie de factori-cheie. În primul rând, implementarea unei astfel de aplicații poate aduce o eficiență sporită în procesul de administrare a închirierilor. Prin automatizarea și centralizarea informațiilor, se pot reduce erorile umane și timpul necesar pentru gestionarea rezervărilor, contractelor și plăților. De asemenea, o aplicație de gestionare poate oferi o imagine cuprinzătoare asupra mașinilor disponibile, rezervărilor în curs și stocului de piese de schimb, facilitând astfel planificarea și organizarea corespunzătoare a resurselor. În al doilea rând, o astfel de aplicație poate spori satisfacția clienților prin intermediul unei experiențe de închiriere simplificate și mai transparente. Prin intermediul aplicației, clienții pot efectua rezervări online, pot accesa informații despre mașinile disponibile, pot verifica detaliile contractuale și pot realiza plăți rapide și securizate. Această abordare modernă și digitală poate consolida relația cu clienții și poate crea o imagine pozitivă a companiei în fața acestora. În concluzie, implementarea unei aplicații de gestionare a închirierilor de mașini poate aduce beneficii semnificative, atât pentru eficiența operațională a companiei, cât și pentru satisfacția clienților, consolidând astfel poziția pe piață și generând o creștere a afacerii.

CUPRINS

[TEMA ȘI MOTIVAȚIA ALEGERII 2](#_Toc104887438)

[1. ELEMENTE TEORETICE 4](#_Toc104887439)

[1.1. DESCRIEREA PROBLEMEI 4](#_Toc104887440)

[1.2. ABORDAREA TEORETICĂ A PROBLEMEI 5](#_Toc104887441)

[1.3. ELEMENTE SPECIFICE POO 5](#_Toc104887442)

[2. IMPLEMENTARE 6](#_Toc104887443)

[2.1. TEHNOLOGII FOLOSITE 6](#_Toc104887444)

[2.2. DIAGRAMA DE CLASE 10](#_Toc104887445)

[3. ANALIZA SOLUTIEI IMPLEMENTATE 10](#_Toc104887446)

[3.1. FORMATUL DATELOR DE I/O 11](#_Toc104887447)

[3.2. STUDII DE CAZ 11](#_Toc104887448)

[3.3. PERFORMANȚE OBȚINUTE 13](#_Toc104887449)

[4. MANUAL DE UTILIZARE 13](#_Toc104887450)

[5. CONCLUZII 13](#_Toc104887451)

[6. BIBLIOGRAFIE 14](#_Toc104887452)

# ELEMENTE TEORETICE

În cadrul capitolului vom elabora chestiunile teoretice și obiectivele ce urmează a fi atinse în urma realizării programului. Limbajul de programare în care va fi dezvoltat programul este C++.

# DESCRIEREA PROBLEMEI

Task-ul constă în implementarea unei aplicații de gestionare a închirierilor de mașini. Aceasta implică dezvoltarea și configurarea unei platforme software care să faciliteze procesele de rezervare, planificare, monitorizare și administrare a mașinilor disponibile pentru închiriere. Scopul este de a optimiza eficiența operațională a companiei și de a oferi clienților o experiență simplificată și transparentă în procesul de închiriere a mașinilor.

# ABORDAREA TEORETICĂ A PROBLEMEI

Se va realiza o aplicație de tip meniu, în care vor exista două părți: partea de elemente constituente ale flotei de mașini, și cea a clienților care au accesat/împrumutat masini, cu următoarele opțiuni:

* MASINA:
* Adăugarea unei mașini;
* Afișarea tuturor mașinilor;
* Căutarea unei mașini;
* Ștergerea mașinilor;
* Modificarea/actualizarea mașinilor;
* Ieșirea din meniul de mașini.
* CLIENT:
* Adăugarea unui client, alături de mașina împrumutată;
* Afisarea clientului, alături de mașina împrumutată;
* Cautarea clientului;
* Modificarea/actualizarea clientului și a mașinilor împrumutate.
* Ștergerea clientului;.
* Ieșirea din meniul de client.

Datele despre fiecare mașina/client vor fi accesate cu ajutorul informațiilor stocate în fișiere..

# ELEMENTE SPECIFICE POO

Pentru a realiza aplicația, va trebui să utilizăm concepte specifice Programării Orientate pe Obiect, precum: obiecte, clase, moștenire, polimorfism, incapsulare.

O clasă reprezintă o colecție de obiecte, cu proprietăți, operații și comportamente comune. Programul va utiliza clase precum:

* Car;
* Client;

Clasele vor comunica între ele prin intermediul moștenirilor, permițând utilizarea funcțiilor ce se consideră a fi necesare. De exemplu, clasa Client poate moșteni clasa Car, astfel încât să identificăm mașinile ce au fost împrumutate de către client.

# IMPLEMENTARE

Capitolul își propune să ofere idei detaliate despre cum, conceptul de POO poate fi utilizat în cadrul realizării unei aplicatii funcționale. Elaborarea va consta în explicarea funcțiilor create și utilizate în program, alături de posibilele deficiențe pe care le posedă.

# TEHNOLOGII FOLOSITE

În sprijinul creării aplicației vom utiliza principii de bază ale Programării Orientate pe Obiect. Principala clasă pe care o va avea programul poartă denumirea de “CarRental”. Pentru clasa Car am stabilit funcții de: adăugare, ștergere, modificare, căutare și vizualizare. Clasa Client va moșteni clasa Car. Cu ajutorul fișierelor binare, ni se va permite să stocăm datele pe care le introducem de la tastatură, pentru fiecare clasă în parte.

**Pentru clasa Car:**

Funcția carMenu() ne permite accesul către celelalte facilitați ale clasei, descries mai jos. A fost creat în mod intenționat pentru a se diferenția de meniul creat pentru clasa Client.

Funcția addCar() va avea în vedere introducerea datelor despre mașina ce va fi utilizată ulterior, în relație cu celelalte funcții create. Se va seta ID-ul, marca, modelul, nr de inmatriculare și VIN-ul.

Funcția DisplayCar() ne va afișa detaliile introduse precedent, în funcția de adăugare a unei noi mașini, într-un tabel care evidențiază fiecare element introdus.

Funcția modificare va identifica id-ul mașinii introduse și ne va permite să-i modificăm: ID-ul, , marca, modelul, nr de inmatriculare și VIN-ul.

Funcția de căutare carSearchMenu() poate fi realizată pe mai multe criterii, în funcție de modul în care am dori să identificăm o mașina. Am utilizat un meniu special întreprins pentru funcționalitate. Pentru aceasta, am stabilit următoarele subfuncții:

* searchbyID();
* searchByMark();
* searchByModel();
* searchByRegNumber().

Funcția de ștergere a mașinii este realizată prin utilizarea a două fișiere binare. Detaliile despre ID-ul care a fost introdus de la tastatură vor fi șterse, celelalte existente fiind salvate într-un fișier ce poartă denumirea, inițial de “deletecar.bin”, iar ulterior va fi redenumit în fișierul binar creat prima dată: “cars.bin”.

Într-un final, toate mașinile care au fost introduse până în momentul solicitării vor putea fi vizualizate prin intermediul functiei de display

Ieșirea din program va fi furnizată prin aplicarea tastei 0.

**Pentru clasa Client:**

Funcția clientMenu() a fost creată cu scopul de a se deosebi de meniul principal pe care l-am conceput pentru consolă. Atunci când această funcție este accesată, vom fi redirecționați, din meniul principal, către cel secundar, unde dispunem de toate facilitățile pe care ni le oferă clasa Client.

Prima funcție de care suntem întâmpinați atunci când accesăm meniul pentru partea de Client este cea de addClient(). Putem adăuga un client în fișierul binar, completând următoarele detalii despre acesta:

* Nume;
* Prenume;
* Mașinia pe care a închiriat-o;
* Data la care a fost închiriata mașina;
* Data la care mașinia trebuie să fie returnată.

Funcția displayClient() ne permite să vizualizăm datele care au fost precedent introduse de la tastatură.

Funcția de căutare este pusă în funcțiune pe baza unor criterii bine stabilite, așa cum s-a prezentat anterior, la clasa Car. Funcțiile criterial create sunt:

* searchByIdClient();
* searchByFirstName();
* searchByLastName();
* searchByRentedCar();
* searchByStartDate();
* searchByReturnDate();

Funcția de modificare a detaliilor despre Client este făcută pe baza identificării persoanei cu ajutorul id-ului. Atunci când clientul va fi identificat, vom putea modifica următoarele detalii despre el: id, nume, prenume, mașina pe care a închiriat-o, data la care a fost închiriata mașina și data la care mașina trebuie să fie returnată.

Funcția de ștergere a clientului funcționează pe baza a două fișiere binare. Pe baza ID-ul clientului, se vor șterge toate detaliile despre acestea, păstrând ceilalți clienți care au împrumutat o anumită mașina. Fișierul binar pe care l-am utilizat să ștergem un client(deleteclient.bin), va fi redenumit în cel inițial (clients.bin).

Funcția de ieșire este configurată și furnizată de tasta 0.

Pentru a opera cu programul se va utiliza meniul consolă pe care ni-l furnizează mediul de dezvoltare.

Așa cum am descris anterior, am elaborat mai multe meniuri, pentru a oferi un aspect aerisit aplicației proiectate. Astfel, meniul principal are înfățișarea:

A screenshot of a computer

Description automatically generated

Atunci când accesăm meniul pentru gestionarea mașinilor, consola va adopta un nou mod de organizare:

A screenshot of a computer

Description automatically generated

Funcțiile au fost proiectate în main și introduse în case-uri. Astfel, pentru opțiunea pe care o alegem, vom putea utiliza funcția după denumirea sugestivă pe care am oferit-o în meniu.

Meniul în care am inclus funcțiile pentru clasa Client este:

A screenshot of a computer

Description automatically generated

# DIAGRAMA DE CLASE

A picture containing text, screenshot, font, diagram

Description automatically generated

# ANALIZA SOLUTIEI IMPLEMENTATE

# FORMATUL DATELOR DE I/O

Formatul datelor de intrare reprezintă modul în care sunt percepute informațiile introduse. Pentru proiectul nostru, am stabilit că toate informațiile vor fi introduse de la tastatură și stocate în fișiere. Formatul cu care am lucrat este de tip ,,binary”. Practic, fișierele text conțin informația stocată sub forma unui șir de caractere (eventual, pe mai multe linii, dar știm deja că sfârșitul de linie este și el tot un caracter). Spre deosebire de ele, fișierele binare stochează informația brut, fără prelucrări exact așa cum apare ea în memorie.

Datele de ieșire sunt informațiile stocate în fișierul binar. Astfel, când utilizăm funcția de afișare, toate datele pe care le-am introdus anterior, vor fi afișate.

# STUDII DE CAZ

Pentru această parte, vom introduce date pentru clasa Car.

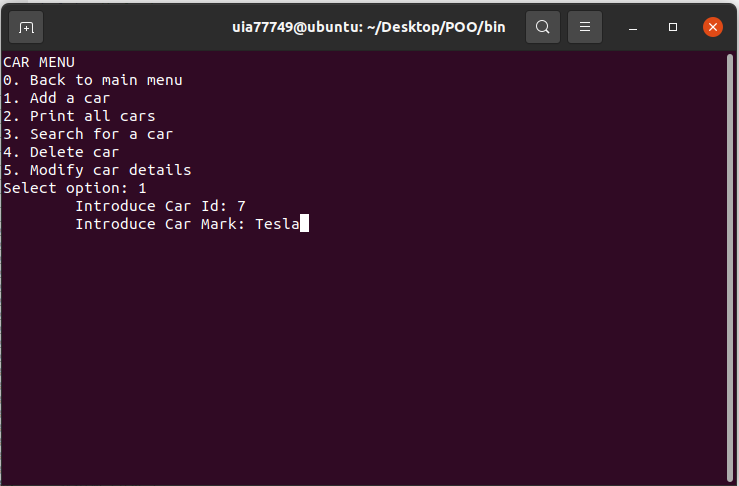
Pentru clasa Car, vom adăuga o mașina.

Ni se va solicita id-ul:

A screenshot of a computer

Description automatically generated

Următorul pas, este introducerea mărcii mașinii:



Se cere modelul mașinii:

A screenshot of a computer

Description automatically generated

Necesită introducerea unui numar de inmatriculare:

A screenshot of a computer

Description automatically generated

Totodată, ne interesează și VIN-ul:

A screenshot of a computer

Description automatically generated

Pentru a vedea detalile introduse în fișier, vom utiliza funcția de afișare:

A screenshot of a computer

Description automatically generated

Similar, se procedează și pentru datele ce trebuie introduse pentru partea de Client.

# PERFORMANȚE OBȚINUTE

Fiind o aplicație de consolă, resursele consumate sunt foarte mici. Orice fel de interacțiune a utilizatorului cu aplicația este instantă.

# MANUAL DE UTILIZARE

Programul, alături de funcțiile sale, funcționează pe baza consolei pe care am configurat-o sub formă de mai multe meniuri, în funcție de necesitate. Astfel, meniul principal este împărțit în:

* Accesarea mașina;
* Accesarea clienților.

Datele de intrare sunt introduse de la tastatură, în funcție de mesajul afișat. Mesajele proiectate sunt construite sugestiv, oferindu-i utilizatorului informația necesară pentru a introduce în mod corect datele. Toate datele pe care acesta le introduce sunt salvate în diverse fișiere binare. Acestea au rolul de a stoca tot ceea ce utilizatorul își propune să completeze. După introducerea datelor solicitate, utilizatorul va fi întâmpinat de un mesaj ce îi sugerează că operația a fost realizată.

Operațiile pe care le poate realiza utilizatorul sunt:

* Adăugarea;
* Afișarea;
* Ștergerea;
* Căutarea;
* Navigarea în alte meniuri;
* Ieșirea din program.

# CONCLUZII

Aceste funcționalități sunt concepute pentru a automatiza și a simplifica procesele de gestionare a închirierilor de mașini, sporind eficiența și oferind o experiență mai bună atât pentru companii, cât și pentru clienți.

**Avantaje:**

* Meniul intuitiv;
* Simplitatea operării cu ajutorul funcțiilor;
* Introducerea, pas cu pas, a datelor de la tastatură, fiind ghidați de text atunci când trebuie să le inserăm pe rând;
* Utilizarea fișierelor binare;
* Utilizarea a mai multor meniuri, pentru a naviga mult mai eficient și a evita încărcarea unui singur meniu cu opțiuni.

**Dezavantaje:**

* Lipsa unei baze de date avansate, precum SQL;
* Interfața minimalistă, utilizând strict consola pentru a opera.

# BIBLIOGRAFIE

* <https://www.cplusplus.com> – The C++ Resources Network;
* <https://www.w3schools.com/cpp/cpp_classes.asp> - Clase/Obiecte;
* <https://www.eecs.umich.edu/courses/eecs380/HANDOUTS/cppBinaryFileIO-2.html> - File Handling using Binary Files.