UNIVERSITATEA “STEFAN CEL MARE”, SUCEAVA

FACULTATEA DE INGINERIE ELECTRICA SI STIINTA CALCULATOARELOR

SPECIALIZAREA CALCULATOARE

**PROIECT DISCIPLINA POO**

[ Translator Român-Englez în C++ ]

Autor

Studentul GEMENIUC ALEXANDRU

**SUCEAVA . 2025**

TEMA PROIECT

# TEMA ȘI MOTIVAȚIA ALEGERII

Tema proiectului constă în realizarea unei aplicații software care funcționează ca un translator simplu din limba română în limba engleză, implementat în limbajul C++ folosind principii de programare orientată pe obiect. Aplicația oferă posibilitatea de a introduce noi termeni în dicționar, de a traduce cuvinte deja existente și de a salva aceste informații într-un fișier text, pentru a fi reutilizate ulterior.

Motivația alegerii acestei teme derivă din mai multe considerente:

* utilitatea practică: Traducerea limbilor este o activitate frecventă, fie în context educațional, fie profesional. O aplicație simplă care permite utilizatorului să-și creeze propriul dicționar personalizat poate avea un rol educativ și util în procesul de învățare a limbii engleze.
* complexitate echilibrată: Proiectul are o dificultate adecvată pentru aprofundarea conceptelor de programare orientată pe obiect (POO), permițând aplicarea noțiunilor precum clase, metode, encapsulare, gestionarea fișierelor și tratamentul excepțiilor.
* accesibilitate: Aplicația este destinată rulării în consolă, fără a necesita resurse hardware sau software avansate. Astfel, poate fi testată și utilizată pe orice sistem care suportă compilatoare C++ standard.
* scalabilitate: Arhitectura modulară a aplicației permite extinderea ușoară în viitor, spre exemplu prin adăugarea unei interfețe grafice (GUI), a unui modul de traducere inversă (engleză-română) sau a integrării cu o bază de date externă.
* consolidarea cunoștințelor de C++: Alegerea acestei teme oferă cadrul ideal pentru exersarea scrierii de cod curat, modular și robust în limbajul C++, precum și pentru utilizarea fișierelor, a STL (Standard Template Library) și a validării inputului prin excepții.

Prin urmare, acest proiect îmbină aspecte tehnice fundamentale cu aplicabilitatea practică, oferind o experiență completă de dezvoltare software în mediul C++.

CAPITOLUL I

# ELEMENTE TEORETICE

# DESCRIEREA PROBLEMEI

Proiectul urmărește dezvoltarea unei aplicații C++ care să funcționeze ca un translator simplu din limba română în limba engleză, folosind concepte de programare orientată pe obiect. Utilizatorul poate introduce cuvinte noi împreună cu traducerea acestora, iar aplicația le va stoca într-un fișier text, pentru a fi accesibile ulterior.

Scopul aplicației este de a permite traducerea rapidă a unor termeni uzuali, păstrând în același timp o structură clară și modulară a codului, precum și o interfață prietenoasă în consolă.

Aplicația se adresează în special utilizatorilor care doresc să-și construiască un vocabular personalizat sau să exerseze limba engleză, fără a depinde de conexiune la internet sau platforme externe.

# ABORDAREA TEORETICA A PROBLEMEI

Traducerea automată a cuvintelor între două limbi reprezintă o problemă clasică de asociere și căutare în informatică. Din punct de vedere teoretic, aplicația prezentată poate fi descrisă ca o structură de date care mapează o cheie (un cuvânt în limba română) la o valoare (echivalentul în limba engleză). Acest tip de relație este ușor de modelat în C++ folosind containere din STL, cum ar fi std::map.

În esență, problema poate fi împărțită în următoarele componente:

a) Reprezentarea internă a datelor

Pentru a păstra asocierea între cuvintele din cele două limbi, este necesară o structură de tip dicționar. Alegerea unei structuri de tip map<string, string> este justificată de:

* acces rapid: căutarea unui cuvânt într-un map se face în timp logaritmic (O(log n)), ceea ce asigură eficiență chiar și pentru un număr mai mare de termeni.
* evitarea duplicării cheilor: map nu permite chei duplicate, garantând unicitatea fiecărui cuvânt românesc.
* ușurință în iterație și salvare: elementele sunt stocate ordonat, ceea ce facilitează exportul în fișier și citirea ulterioară.

b) Persistența datelor

Pentru a menține consistența și continuitatea datelor introduse între sesiunile de rulare ale aplicației, a fost adoptată metoda de stocare într-un fișier text local (dictionar.txt). Fiecare linie a fișierului conține o pereche de cuvinte sub forma română-engleză, separate prin caracterul -.

Această abordare oferă:

* simplitate: fișierul poate fi ușor editat manual sau inspectat.
* portabilitate: fișierul poate fi transferat pe alt sistem fără pierderi de date.
* flexibilitate: formatul text permite adaptarea ulterioară pentru importuri/exporturi în alte aplicații.

c) Interacțiunea cu utilizatorul

Aplicația oferă o interfață în linie de comandă, structurată sub forma unui meniu interactiv. Utilizatorul poate alege să introducă o pereche nouă de traducere sau să solicite traducerea unui cuvânt deja existent.

În cazul în care cuvântul nu este regăsit în dicționar, aplicația îl solicită de la utilizator și îl adaugă automat, contribuind astfel la extinderea progresivă a bazei de date.

d) Validarea datelor și tratarea excepțiilor

Pentru a asigura corectitudinea datelor introduse, aplicația include un sistem de validare a inputului prin clase și metode dedicate. Se verifică:

* ca opțiunile din meniu să fie formate doar din cifre (evitând introducerea de caractere invalide),
* ca termenii introduși ca traduceri să conțină doar litere și eventual spații.

Eventualele erori sunt tratate prin aruncarea și prinderea de excepții, evitând astfel oprirea neașteptată a programului și oferind utilizatorului posibilitatea de a corecta inputul.

# ELEMENTE SPECIFICE POO

Aplicația este realizată folosind paradigma Programării Orientate pe Obiect (POO), oferind o structurare clară, modulară și scalabilă a codului sursă. În cadrul proiectului, această abordare s-a concretizat prin utilizarea claselor, a metodelor și a principiilor fundamentale ale POO: încapsulare, abstracție, modularitate și gestionarea excepțiilor.

a) Încapsularea datelor

Prin clasa Dictionar, datele (cuvintele și traducerile) sunt protejate și gestionate intern. Membrii variabili (ex: **map<string, string> cuvinte**) sunt declarați private, fiind accesibili doar prin metodele publice precum **adaugaCuvinte()**, **traduceCuvinte()** sau **salveazaInFisier()**. Această abordare asigură controlul asupra modului în care datele sunt modificate sau accesate, reducând riscul erorilor externe.

b) Separarea responsabilităților (modularitate)

Fiecare clasă are o responsabilitate clară:

* *Dictionar*: gestionează operațiile de bază asupra cuvintelor (adăugare, traducere, salvare/încărcare din fișier).
* *Aplicatie*: oferă logica principală a aplicației și interacțiunea cu utilizatorul.
* *TratareExceptii*: se ocupă exclusiv de validarea inputului și de gestionarea excepțiilor.

Această separare respectă principiul Single Responsibility și permite o mentenanță ușoară a codului.

c) Abstracția

Interfața oferită utilizatorului este simplificată, ascunzând detaliile tehnice (fișiere, mapări interne). Utilizatorul interacționează doar cu meniul aplicației, fără să cunoască implementarea internă a claselor, ceea ce reflectă aplicarea principiului de abstracție în proiectare.

d) Tratarea excepțiilor

Prin clasa TratareExceptii, aplicația implementează un mecanism robust de tratare a erorilor. Metodele sale aruncă excepții (invalid\_argument) în caz de input incorect, care sunt apoi capturate și gestionate la nivelul interfeței (Aplicatie::ruleaza()), prevenind astfel întreruperi bruște ale programului și oferind utilizatorului mesaje clare de eroare.

e) Scalabilitate OOP

Structura actuală, bazată pe clase bine definite și izolate, permite extinderea aplicației într-un mod controlat:

* se pot adăuga noi funcționalități (ex: traducere inversă, căutare parțială),
* se poate introduce o interfață grafică fără a modifica logica de bază,
* se pot înlocui fișierele text cu baze de date, fără a afecta restul aplicației.

# FORMATUL FIȘIERULUI DE INTRARE/IEȘIRE

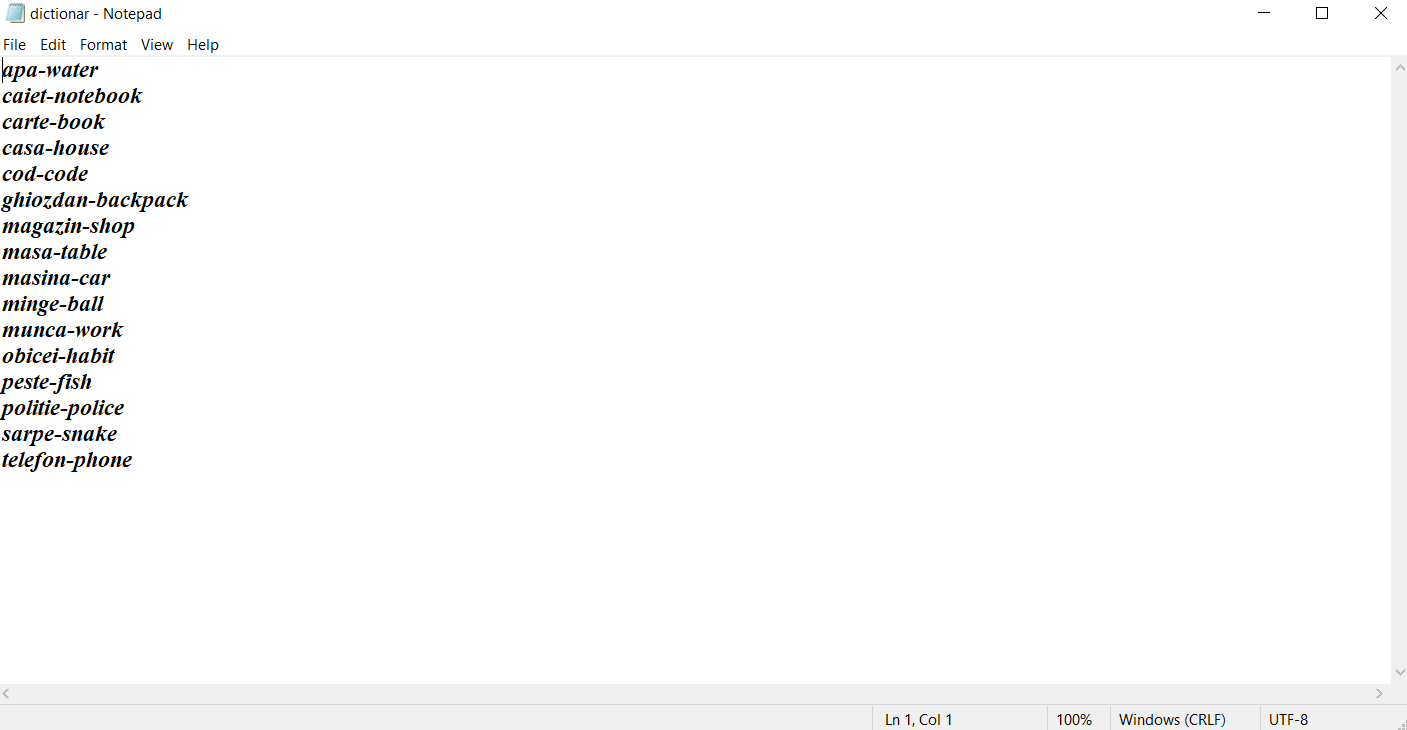
Pentru a asigura persistența datelor între sesiunile de rulare ale aplicației, aceasta utilizează un fișier text extern cu denumirea **dictionar.txt**, care stochează perechile de cuvinte în următorul format simplu și intuitiv:

*cuvant\_romana-cuvant\_engleza*

Caracteristici ale formatului:

* fiecare linie reprezintă o singură pereche de traducere.
* cuvintele din limba română și cea engleză sunt separate prin caracterul - (minus).
* nu se acceptă mai multe perechi pe aceeași linie.
* fișierul este ușor de citit și editat manual în caz de nevoie.

Exemplu de conținut al fișierului:



Avantaje ale acestui format:

* **portabilitate**: Fișierul poate fi mutat și accesat de pe orice sistem fără conversii suplimentare.
* **simplitate**: Poate fi citit sau modificat și în afara aplicației, cu un simplu editor de text.
* **compatibilitate**: Poate fi adaptat ușor pentru export sau conversie în alte formate (CSV, JSON etc.).

# STUDII DE CAZ PRIVIND INTRODUCEREA ȘI TRADUCEREA CUVINTELOR

Pentru a valida funcționalitatea aplicației și a demonstra utilizarea corectă a opțiunilor din meniu, se pot analiza următoarele studii de caz:

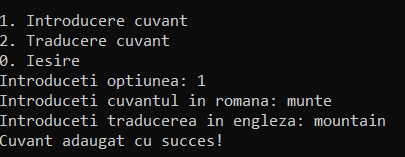
*Caz 1*: Introducerea unei noi perechi de traducere

Situație:

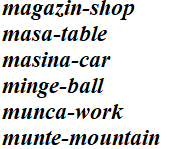
Utilizatorul dorește să adauge cuvântul „munte” și traducerea sa în limba engleză, „mountain”.

Pași:

1. Selectează opțiunea **1. Introducere cuvânt**.
2. Introduce „munte” → validat ca un cuvânt format doar din litere.
3. Introduce „mountain” → validat la fel.
4. Cuvântul este adăugat în dicționar și salvat automat în fișierul **dictionar.txt**.



Rezultatul în fișier:



*Caz 2*: Traducerea unui cuvânt existent în dicționar

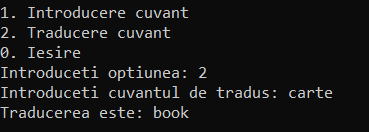
Situație:

Utilizatorul dorește să traducă cuvântul „carte”, care a fost deja introdus anterior.

Pași:

1. Selectează opțiunea **2. Traducere cuvânt**.
2. Introduce „carte”.
3. Aplicația găsește traducerea în map și afișează:

Traducerea este: book



*Caz 3*: Traducerea unui cuvânt inexistent (auto-adăugare)

Situație:

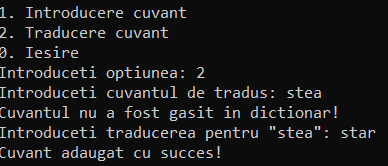
Utilizatorul introduce un cuvânt care nu se regăsește în dicționar, de exemplu „stea”.

Pași:

1. Selectează **2. Traducere cuvânt**.
2. Introduce „stea”.
3. Aplicația returnează:

*Cuvântul nu a fost găsit în dicționar!*

1. Utilizatorul este invitat să introducă traducerea pentru „stea”.
2. Introduce „star” → perechea este adăugată automat și salvată în fișier.



Rezultatul în fișier:

