



*Colegiul Național „Unirea” Focșani*

# Proiect pentru examenul de atestat

*„Bancomatul virtual”*

**Autor: Istrate Alexandru-Daniel**







# *Cuprins*

***Capitolul 1.*** De ce am realizat acest proiect? Pentru cine este destinat?

***Capitolul 2.*** Descrierea și structura proiectului

***Capitolul 3.*** Modul de realizare

***Capitolul 4.*** Resursele necesare utilizării

***Capitolul 5.*** Bibliografie



## *De ce am realizat acest proiect?*

### *Pentru cine este destinat?*

Am ales ca, în cadrul temei mele de atestat, să simulez funcționalitatea unui bancomat deoarece este unul dintre dispozitivele pe care le folosim aproape săptămânal sau lunar, fiind mereu necesar să retragem bani din conturile noastre bancare, să depunem, sau să verificăm de ce sume dispunem la un anumit moment. În era digitală, suntem dependenți de sistemul bancar, prin intermediul căruia reușim să utilizăm banii în timp real, având control asupra lor în orice moment, însă pentru a nu întâmpina probleme grave de securitate și funcționalitate, acest sistem are nevoie de o echipă numeroasă care să se ocupe de mentenanță și să rezolve în câteva momente diversele situații întâlnite. Am considerat interesant acest sistem de lucru, felul în care, constant, dispozitivul cu care interacționăm noi este conectat la o bază de date care este modificată de alte sute de utilizatori în același timp.

Acest program este destinat pentru cei care vor să aibă un mic „bancomat” chiar la ei acasă, ce conține funcțiile principale pe care noi le folosim zi de zi. Precum jocul Monopoly este făcut să învețe copiii să gestioneze bani și proprietăți, oferindu-le acces nelimitat la sumele pe care le dețin, astfel funcționează și acest program, care îți oferă libertatea de a modifica și vizualiza sumele oricărui utilizator introdus în baza de date, dar și de a adăuga nenumărați alți oameni în sistem.

Poate vrei să strângi fondul clasei și te-ai plictisit de banalul caietel și plic, sau poate faci chetă cu niște prieteni pentru un cadou. Cu acest program, poți să-ți adaugi toți prietenii sau colegii și să ții evidența banilor pe care îi împrumuți sau îi primești într-un mod mai... diferit! La bază e doar o simulare, însă cred cu tărie că ar putea deveni util pentru a ține contabilitatea unor sume de bani într-un mod mai distractiv și educativ.



## *Descrierea și structura proiectului*

După cum am prezentat, softul realizat de mine imită un bancomat cu 4 opțiuni: adăugarea unui nou client în baza de date existentă (fig. 1), retragere numerar, depunere numerar și verificare sold (fig. 2). Pentru a înlocui accesul la un cont cu un card fizic, am optat pentru opțiunea unui cod de client unic, prin intermediul căruia se identifică utilizatorul pentru a-și putea accesa „contul bancar”.

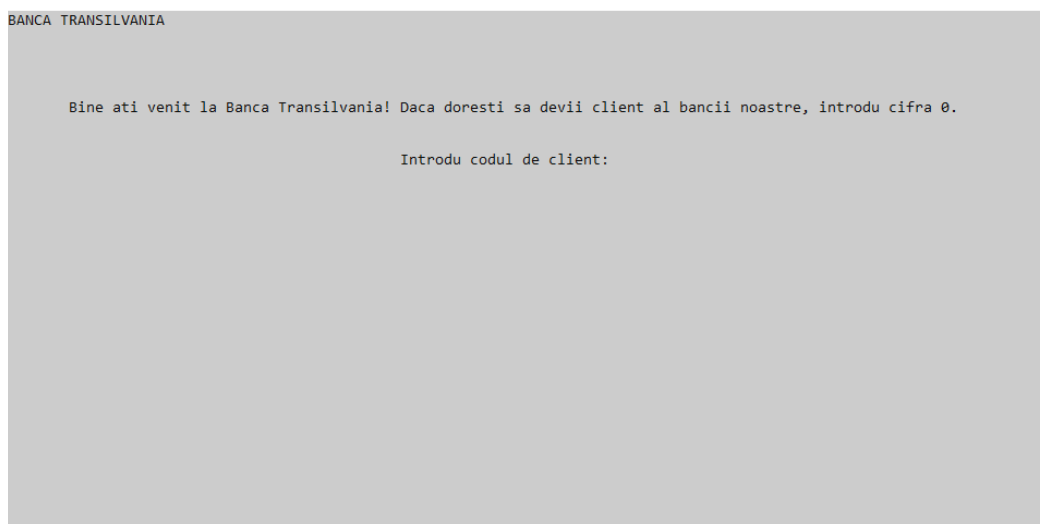


Fig. 1 – Meniul principal al „Bancomatului virtual”



Fig. 2 – Operațiunile ce se pot realiza pentru un client existent



În meniul de adăugare a unui client nou, se afișează următorul cod de client disponibil, acestea fiind generate în ordine crescătoare, și i se solicită utilizatorului să-și introducă datele: nume, prenume(fig. 3) și cod PIN(fig. 4), un cod unic pentru a nu putea accesa contul altei persoane. Inițial, contul se generează cu suma de 0 lei, iar la o nouă accesare a softului clientul își poate depune bani în contul personal.

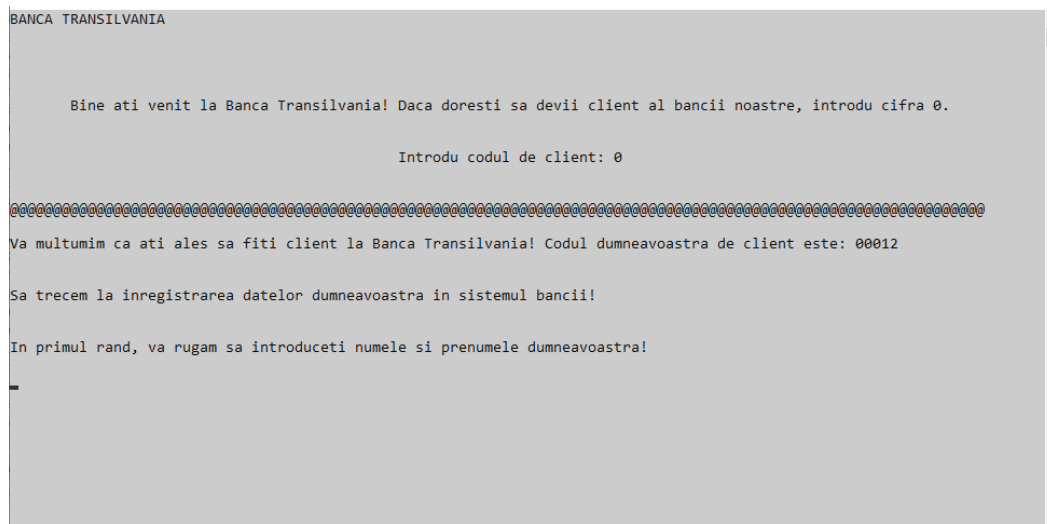


Fig. 3 – Modulul de introducere a unui nou utilizator(primul pas)

Foarte bine! Acum va rugam sa alegeti un cod PIN prin care sa va accesati contul!

Fig. 4 – Modulul de introducere a unui nou utilizator(al doilea pas)

Pentru un user existent, se introduce codul de client, iar dacă nu există, pe ecran se va afișa un mesaj prin care utilizatorul este anunțat că datele introduse de el nu aparțin bazei de date(fig. 5). Dacă datele introduse sunt corecte, se va afișa pe ecran numele persoanei care deține contul și opțiunea de a introduce codul PIN(fig.6). Pentru a mă apropia de realitate,



clientul are posibilitatea de a introduce de maxim 3 ori codul PIN, iar după a treia încercare programul se oprește, afișând mesajul prin care este anunțat că a fost depășită limita de încercări pentru codul PIN(fig. 7).

```
Codul de client introdus nu se afla in baza noastra de date!  
  
Multumim pentru utilizarea bancomatelor noastre!  
  
Va asteptam cu drag si data viitoare!  
Process returned 0 (0x0)   execution time : 4.484 s  
Press any key to continue.
```

Fig. 5 – Cod de client incorect

```
Bine ai venit, Istrate Alexandru-Daniel!  
  
Introdu codul PIN: █
```

Fig. 6 – Ecran introducere cod PIN

```
Bine ai venit, Istrate Alexandru-Daniel!  
  
Introdu codul PIN: 2222  
  
Codul pin a fost introdus gresit! Mai ai 2 incercari  
  
Introdu codul PIN: 2222  
  
Codul pin a fost introdus gresit! Mai ai 1 incercari  
  
Introdu codul PIN: 2222  
  
Ai atins limita de incercari!  
Process returned 0 (0x0)   execution time : 34.301 s  
Press any key to continue.
```

Fig. 7 – Ecran cod PIN incorect



Apoi, dacă este introdus corect codul PIN, apare meniul cu cele 3 operațiuni pe care utilizatorul le poate realiza asupra unui cont(fig. 2). Pentru a efectua una dintre operațiuni, se introduce cifra corespunzătoare acțiunii: 1 – retragere, 2 – depunere, 3 – afișare sold.

Pentru opțiunea de retragere, se introduce suma pe care utilizatorul își dorește să o scoată din contul personal(fig. 8). Dacă suma este mai mare decât cea din cont, este mai mică decât 0 sau egală cu 0, programul se oprește și afișează avertismentul cu privire la faptul că suma introdusă nu poate fi retrasă(fig. 9).

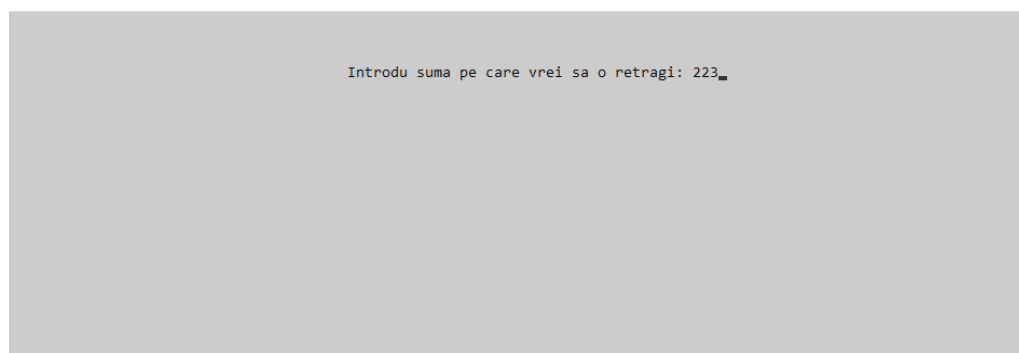


Fig. 8 – Retragere numerar

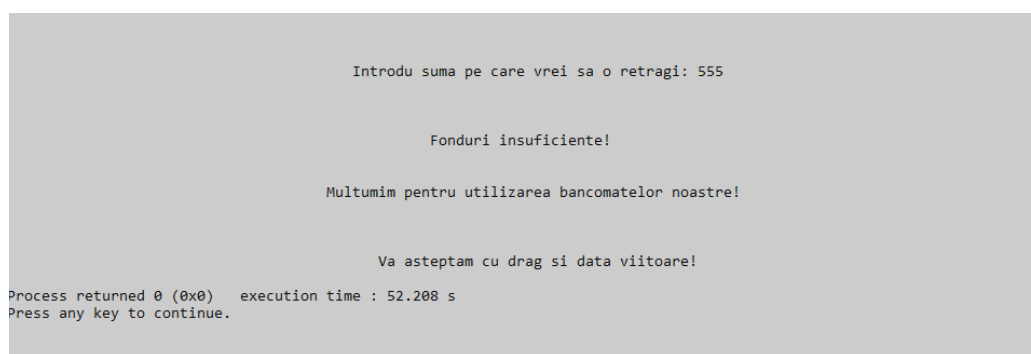


Fig. 9 – Sumă incorectă pentru retragere numerar





Pentru opțiunea de depunere, se introduce suma pe care utilizatorul își dorește să o adauge în cont(fig. 10). Această sumă nu are reglementări maxime, însă trebuie să fie mai mare decât 0, altfel programul se oprește și afișează mesajul corespunzător(fig. 11).

În final, ultima opțiune pe care o poate realiza programul, cea de afișare sold, scrie pe ecran suma disponibilă din contul personal, iar programul se oprește(fig.12).

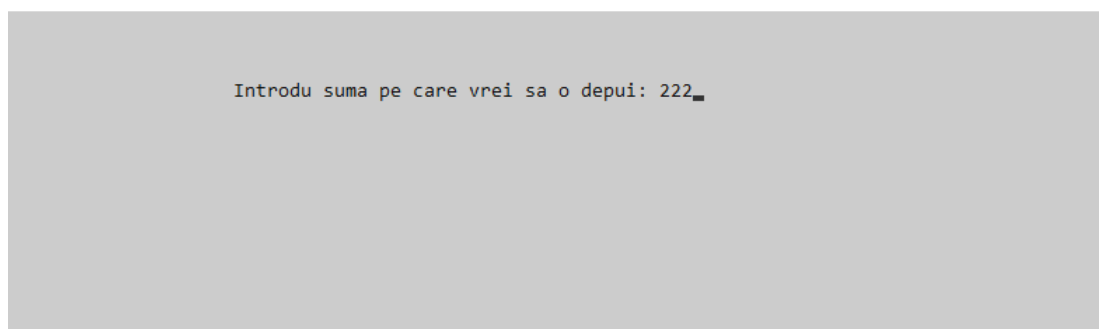


Fig. 10 – Depunere numerar

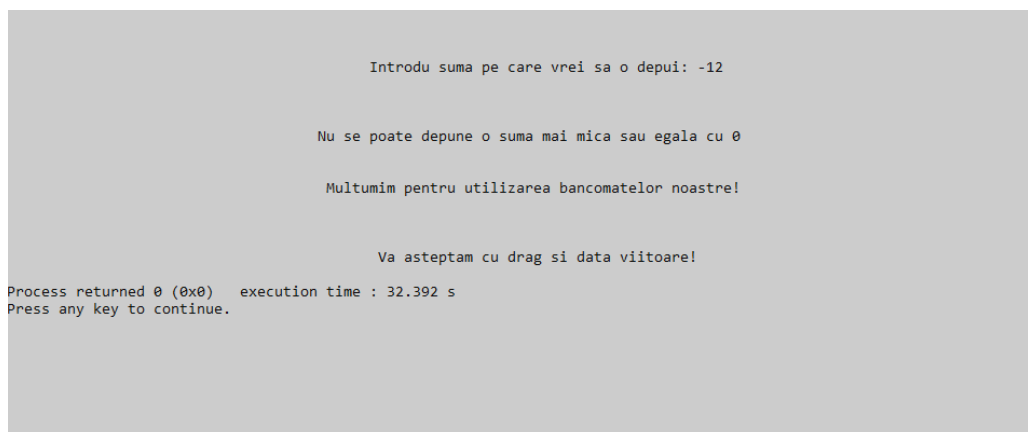


Fig. 11 – Sumă incorectă pentru depunere numerar

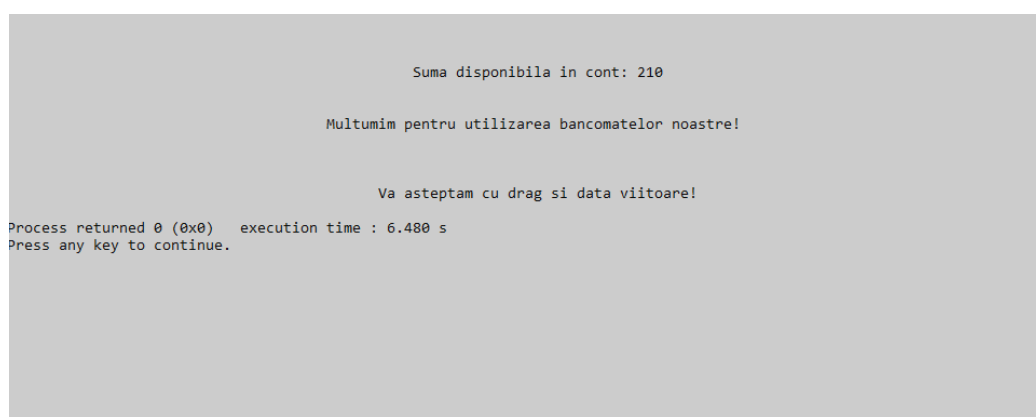


Fig. 12 – Verificare sold



## *Modul de realizare*

Programul prezentat a fost realizat în limbajul de programare C++. El este un limbaj de programare general, compilat. Este un limbaj multi-paradigmă, cu verificarea statică a tipului variabilelor ce suportă programare procedurală, abstractizare a datelor, programare orientată pe obiecte. În anii 1990, C++ a devenit unul dintre cele mai populare limbaje de programare comerciale, rămânând astfel până azi.

Pentru a putea utiliza diferite opțiuni ale programării în C++, am introdus, în primul rând, 5 librării(fig. 13) care oferă acces la diverse operațiuni:

- **Iostream:** librăria standard pentru a introduce și a afișa date prin consolă;
- **Fstream:** librăria prin care programul citește și scrie date în fișiere;
- **Cstring:** permite lucrul datelor de tip char prin intermediul unor funcții specifice;
- **Windows.h:** conține declarații pentru toate funcțiile din Windows API;
- **Iomanip:** manipulează ieșirea programelor C++;

```
#include <iostream>
#include <fstream>
#include <cstring>
#include <windows.h>
#include <iomanip>
```

Fig. 13 – librăriile folosite de program



Toate instrucțiunile utilizate se află în cadrul funcției *main*, pentru o manipulare mai ușoară a variabilelor și a comenzilor care alcătuiesc programul. Partea de început reprezintă citirea datelor atât din fișier, cât și din consolă, pentru a putea identifica dorința utilizatorului. Astfel, clientul se poate „conecta” la contul personal sau poate își poate crea un nou cont, introducând cifra 0 (fig. 14).

```
int main()
{
    system("Color 70");

    ifstream in("date.in");
    ofstream out2("date_copy.in");

    char s[200], copy[200], v[6], pi[5], sum[10], *p;
    int op=0, ok=0, cash, var=2, okl=0;

    cout<<"BANCA TRANSILVANIA"<<endl<<endl<<endl;
    cout << endl<<endl<< setw (110)<< "Bine ati venit la Banca Transilvania! Daca doresti sa devii client al bancii noastre, introdu cifra 0. ";
    cout << endl<<endl<< setw (190)<< "Introdu codul de client: ";

    cin.getline(v,6);

    if(atoi(v)==0)
    {
    else{
```

Fig. 14 – Citirea datelor

În cazul introducerii unui nou utilizator, se creează o copie a datelor din fișierul de intrare spre un fișier de ieșire, care, ulterior, își inversează rolurile pentru a putea scrie date în baza de date pe care programul o utilizează (fig. 15). După ce clientul își introduce datele personale (nume și prenume, cod PIN), acestea sunt adăugate la finalul fișierului, clienții fiind așezați în fișier crescător, în funcție de codul de client.

```
ofstream out2("copy.txt");

while(in.getline(s,200) && !in.eof())
{
    out2<<s<<endl;
    p=strtok(s," ");

}

int t=atoi(p+3)+1;
int pin;

in.close();
out2.close();

ifstream in7("copy.txt");
ofstream out5("date.in");
```

Fig. 15 – Schimbul de date între fișiere

În cazul unui client existent, acesta introduce codul unic ce i-a fost atribuit, iar programul citește fiecare linie din baza de date prin intermediul unui structuri repetitive cu test inițial (*while*) până găsește numărul furnizat, moment în care, prin funcția *break* se oprește. Cu ajutorul unei variabile ce poate lua ca valoare 0 sau 1, se verifică dacă programul a găsit numărul introdus, iar dacă acesta nu a fost regăsit, utilizatorul este avertizat asupra incorectitudinii datelor introduse (fig. 16). De asemenea, în urma introducerii



unui număr valid, codul PIN este verificat, iar cu ajutorul unui contor se verifică dacă a fost depășit numărul de introduceri ale acestui cod.

```
while(in.getline(s,200) && ok==0) //se cauta codul de client in baza de date
{
    strcpy(copy, s);
    p=strtok(s, " ");

    if(strcmp(p,v)==0) {ok=1; break;}

    else in2<<copy<<endl;

}

if(ok==0) cout << endl<<endl<< setw (210)<<"Codul de client introdus nu se afla in baza noastra de date!";
```

Fig. 16 – verificarea codului de client

Opțiunea „Retragere numerar” constă în introducerea de la tastatură a sumei pe care clientul își dorește să o scoată din cont, iar dacă aceasta nu depășește valoarea regăsită în fișier, afișează pe ecran suma rămasă în cont(fig. 17).

```
if(op==1){

    cout << endl<<endl<< setw (200)<<"Introdu suma pe care vrei sa o retragi: ";
    cin.ignore();
    cin.getline(sum,9);

    x=atoi(sum);

    if(x>j) {cout << endl<<endl<< setw (190)<<"Fonduri insuficiente!"; in2<<p<<endl;}
    else {y=j-x;

    cout << endl<<endl<< setw (193)<<"Soldul actual din cont este: "<<y;
        in2<<y<<endl;
    }
}
```

Fig. 17 – Codul pentru „Retragere numerar”

„Depunere numerar” se comportă în linii mari ca și varianta anterioară, doar că în această situație, codul trebuie să verifice dacă suma este mai mare decât 0, neavând restricție maximală(fig. 18), iar „Verificare cont” reprezintă simpla afișare a sumei care se regăsește în contul utilizatorului(fig.19).

```
if(op==2){cout << endl<<endl<< setw (200)<<"Introdu suma pe care vrei sa o depui: ";
    cin.ignore();
    cin.getline(sum,9);
    x=atoi(sum);
    if(x<=0) {cout << endl<<endl<< setw (205)<<"Nu se poate depune o suma mai mica sau egala cu 0"; in2<<p<<endl;}
    else{
        y=x+j;

    cout << endl<<endl<< setw (193)<<"Soldul actual din cont este: "<<y;
        in2<<y<<endl;
    }
}
```

Fig. 18 – Codul pentru „Depunere numerar”



```
else
    if (op==3) {cout << endl<<endl<< setw (193)<<"Suma disponibila in cont: "<<p; in2<<p<<endl;
    }
```

Fig. 19 – Codul pentru „Verificare sold”

Acest program se bazează pe funcții specifice tipului de date *char*, prin intermediul cărora programul citește, verifică și manipulează informația primită:

- `Cin.getline(s,100)`: se vor citi în șirul *s* caracterele din stream-ul de intrare (de la tastatură) până la apariția caracterului sfârșit de linie(NULL), dar nu mai mult de (n-1) caractere;
- `Strcpy(x,y)`: copiază caracterele din șirul aflat la adresa sursă, inclusiv caracterul nul, în șirul al cărui prim element se află la adresa din șirul destinație. Am folosit această funcție pentru a putea copia codurile de client și cele PIN, cât și numele clienților, din adresa \*p într-un vector care să-mi permită prelucrarea acestora;
- `Strtok(p, " ")`: extrage dintr-un sir de caractere câte un subșir (cuvânt) delimitat de caractere din șirul de separatori. Această funcție presupune lucrul cu pointeri, deoarece returnează adresa subșirului găsit;
- `Strcmp(x,y)`: Compară lexicografic cele două șiruri de caractere. Utilă pentru a compara datele din fișier cu cele introduse de la tastatură, astfel reușind să identifice clienții potriviți;
- `Atoi(p)`: realizează conversia unui șir într-un număr întreg. Prin intermediul acestei funcții, am avut posibilitatea de a prelucra mult mai ușor sumele din fișierul de intrare, care erau citite sub formă de vector de tip *char*.

În plus, relevante sunt și funcțiile care determină afișarea într-un mod special a datelor în consolă:

- `Setw`: setează lățimea câmpului sau numărul de caractere care urmează să fie afișate înaintea unui anumit câmp;
- `System("cls")`: pentru a șterge ecranul în consolă.



## *Resursele necesare utilizării*

Proiectul a fost realizat cu ajutorul programului CodeBlocks și Virtual Studio Code, iar cerințele de sistem ale programului sunt dintre cele mai comune posibile, fiind accesibil pentru o gamă largă de utilizatori.

### **Resurse minime :**

#### 1) Hardware :

- Procesor (CPU) : 1 GHz +
- Memorie (RAM): 512 MB
- Spațiu liber pe hard-disk : 50 MB
- Monitor :1024\*768

#### 2) Software :

- Sistem de operare : Windows 98/XP

### **Resurse recomandate:**

#### 1) Hardware :

- Procesor (CPU) : 2.25 GHz +
- Memorie (RAM): 1 GB
- Spațiu liber pe hard-disk : 100 MB
- Monitor : VGA 64-bit color (1024x768)

#### 2) Software :

- Sistem de operare : Windows XP



## *Bibliografie*

- **Programe:**

CodeBlocks



Visual Studio Code



- **Site-uri:**

Pbinfo.ro



Cplusplus.com



stackoverflow.com





*Programul poate fi  
accesat și prin codul  
QR de mai jos*

