Capitolul 3: Etapele realizării unui program

Instrucțiunile pentru rezolvarea unei anumite probleme sunt grupate într-un **program**, scris de un programator sau de o echipă de programatori. Cea de-a doua situație se întâlneşte în special în cazul programelor "de firma", de complexitate ridicată, care sunt livrate odată cu sistemul de calcul (sistemul de operare şi diverse programe utilitare) sau se achiziționează ulterior (jocuri, compilatoare, programe pentru rularea aplicațiilor specifice unui anumit domeniu etc.).

Pentru scrierea unui program se utilizează, după cum ştiți, un limbaj de programare**,** cu sintaxă şi semantică riguros definite. În cazul de față programele sunt scrise în limbajul Rodin un limbaj gandit pentru a preda romaneste notiunile de baza ale limbajului de programare "C".

Mediul de programare Rodin

Mediul de programare Rodin este dedicat dezvoltării programelor scrise în limbajul Rodin. Dintre componentele sale menționăm:

- Editorul de programe – TotalEdit; acesta este un editor de texte specializat, conceput astfel încât să uşureze dactilografierea şi modificarea programelor sursă, realizat de firma CoderTools. Explicatii cum sa configurati TotalEdit astfel incat sa lucreze impreuna cu rodin exista pe site, intr-un fisier .pdf de doua pagini. Vedeti si paginile urmatoare (fig.22).

Introducerea textului unui program se face caracter cu caracter. Poziția următorului caracter introdus este pusă în evidență printr-un marcaj, numit **cursor.** Pentru a termina o linie şi a trece la linia următoare se apasă tasta **ENTER.** Utilizarea unui editor - instrumental utilizat pentru scrierea programelor de calculator. Un editor este asemănător cu un procesor de text, prin aceea că oferă posibilitatea de a scrie linii de program, de a le edita, de a le muta, copia şi salva pe disc.

Din acest motiv, editoarele sunt adeseori denumite şi editoare de text. Un editor diferă de un procesor de text prin aceea că nu efectuează aranjarea automată a cuvintelor, într-un processor de text, atunci când ați ajuns la capătul rândului, procesorul mută cursorul şi porțiunea de cuvânt de la sfârşitul rândului pe rândul următor. Aranjarea automată a cuvintelor ar constitui un impediment pentru programele de calculator.

Rețineți că limbajele de programare sunt concise. Instrucțiunile de program nu pot fi executate împreună ca un fel de vorbire tipărită, în anumite limbaje de programare se pot pune mai multe instrucțiuni pe acelaşi rând, dar acest obicei nu este recomandat deoarece face citirea programului mai dificilă pentru dumneavoastră şi pentru alții. Cu cât un program este mai greu de citit, cu atât va fi ulterior mai dificil de întreținut şi de actualizat.

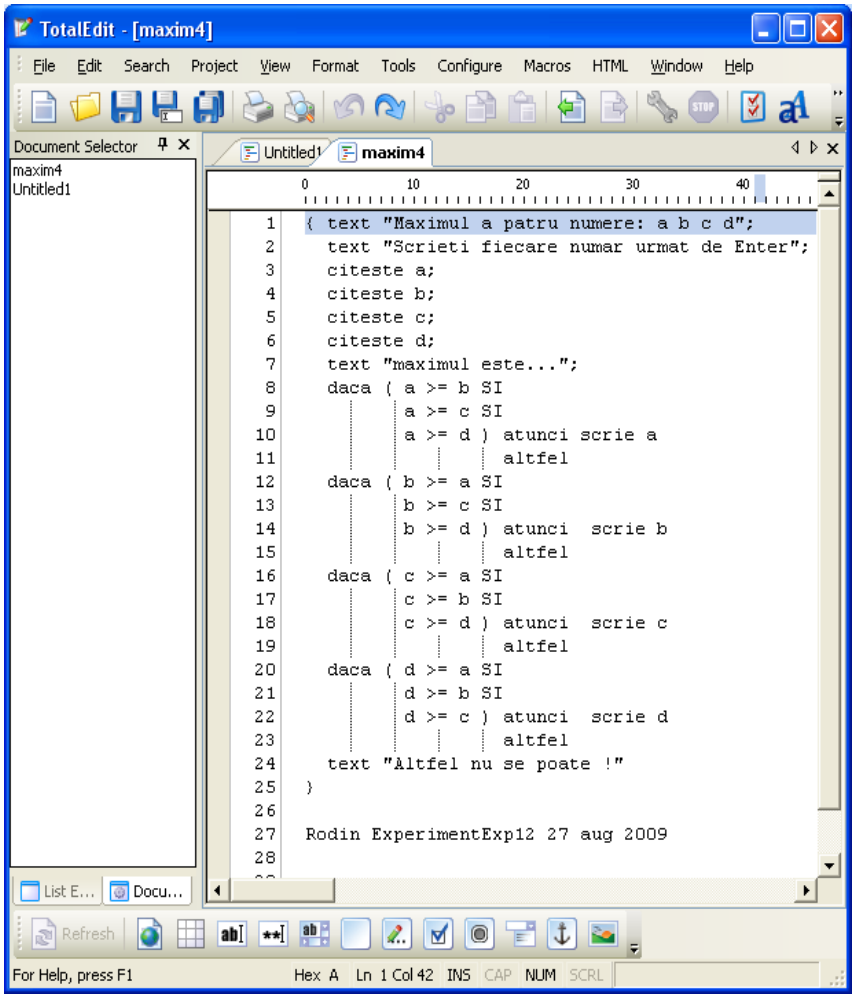

Fig.18 (editarea unui fişier text în TotalEdit)

- Componenta de gestiune a fişierelor, care asigură, în principal, salvarea în fişiere pe disc a programelor editate şi încărcarea programelor sursă de pe disc. Deşi nu este obligatoriu, recomandăm ca în cazul editării unui nou program să se fixeze numele fişierului sursă corespunzător chiar de la începutul editării. Aceasta se realizează prin intermediul comenzii de salvare (Save - Ctrl+S).

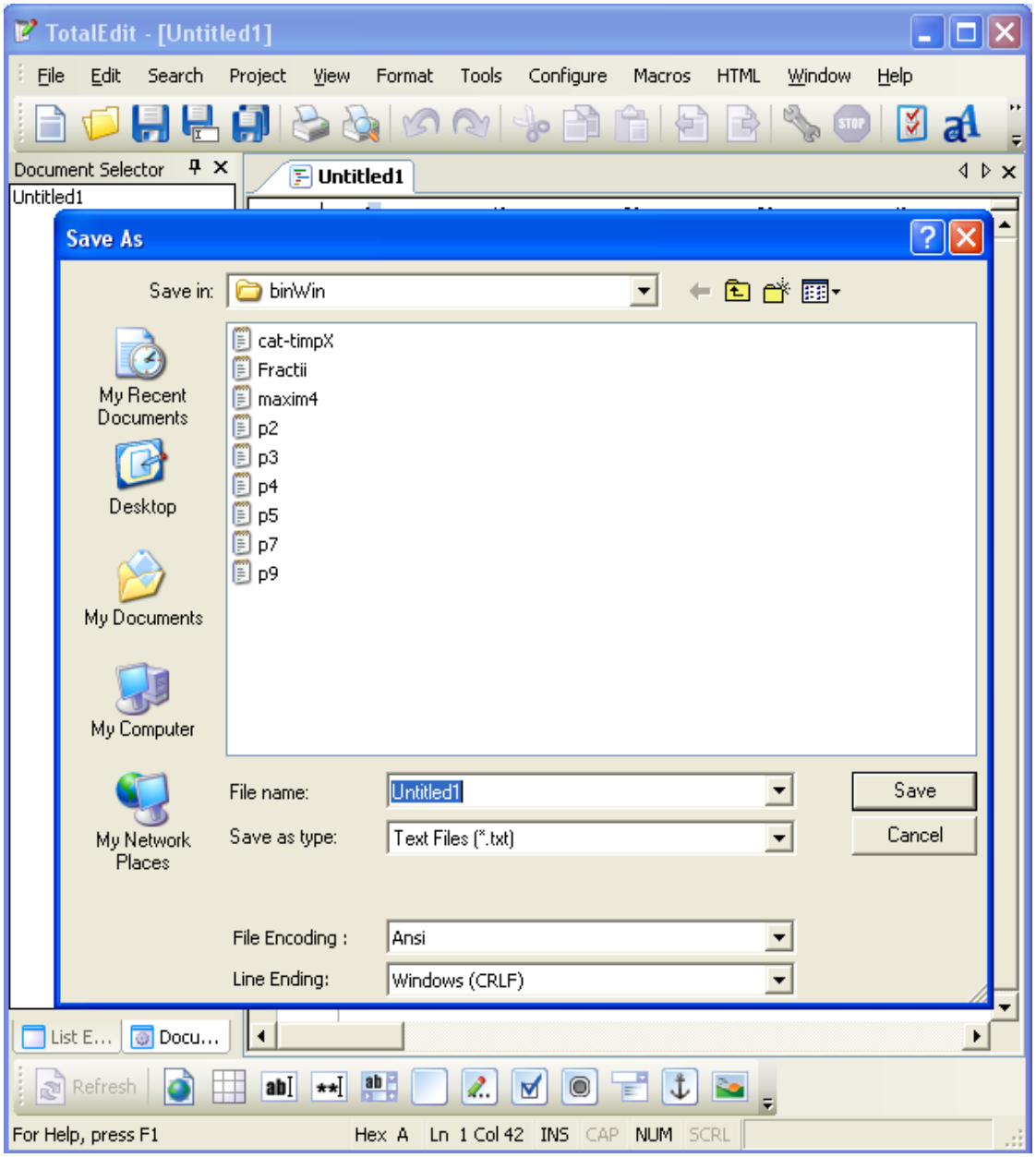

Fig.19 (salvarea fişierului text din TotalEdit)

Dacă se introduce numele **test**, mediul de programare îi adaugă automat extensia .txt, deci identificatorul complet al fişierului sursă va fi **test.txt**.

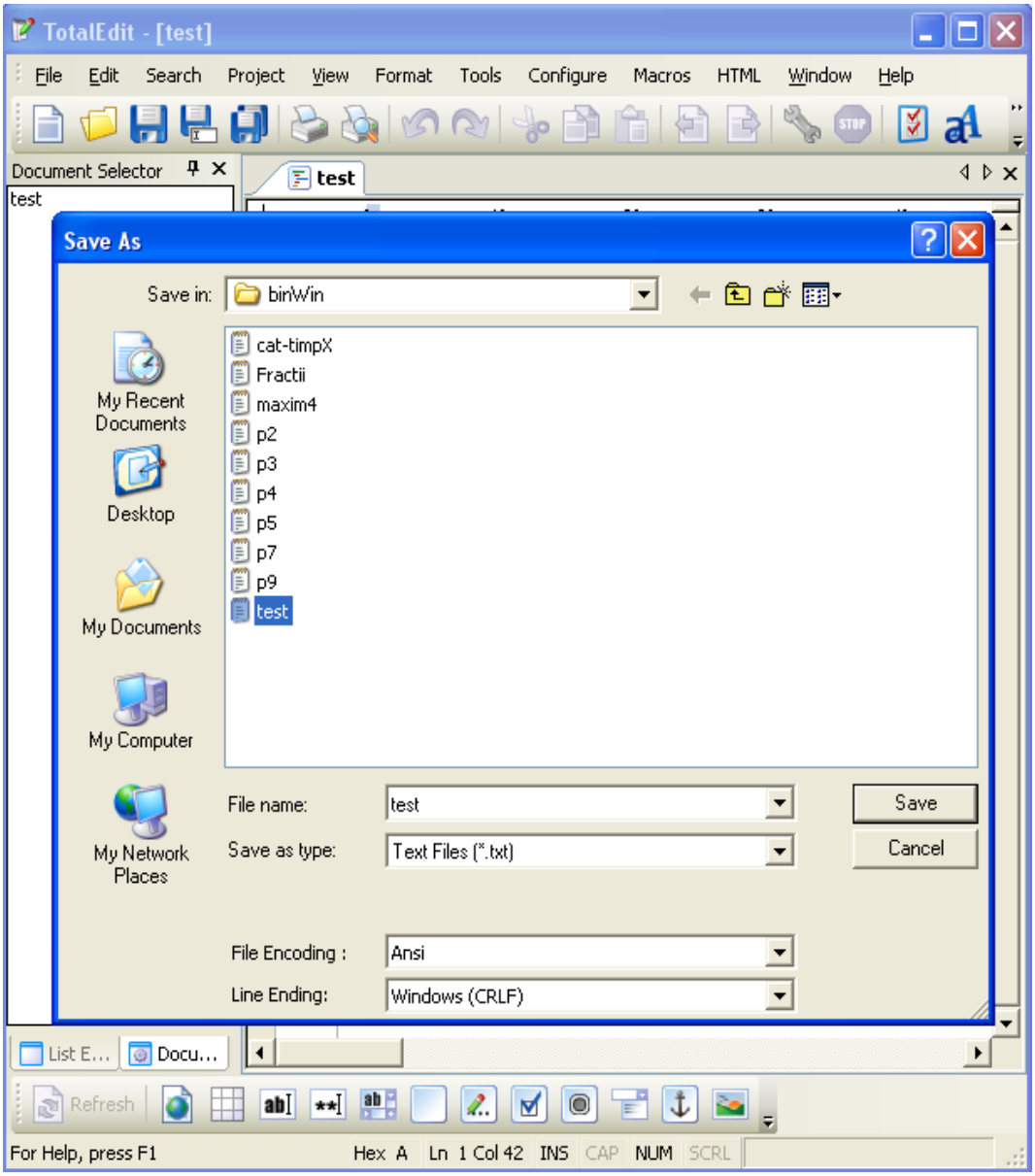

Fig.20 (idem fig.19)

Recomandăm ca, pe parcursul editării textului programului, acesta să fie salvat din când în când, pentru a evita situaţiile neplăcute, de genul "cădere de tensiune" sau "blocare sistem", în urma cărora textul editat se pierde şi este necesară reintroducerea sa.

- Dacă programul are erori, atunci compilatorul se opreşte la prima eroare întâlnită, redând controlul editorului. În acest caz în care a fost detectată o eroare se afişează un mesaj referitor la cauza probabilă a erorii, ca în figura de mai jos.

```
C:\Documents and Settings\Codrina>cd desktop

C:\Documents and Settings\Codrina\Desktop>cd experimentexp12win

C:\Documents and Settings\Codrina\Desktop\ExperimentExp12Win>cd binwin

C:\Documents and Settings\Codrina\Desktop\ExperimentExp12Win\binWin>main.exe p7.
txt
Modular Language written by Dan V Popa, Ro/Haskell Group.
29/aug/2009  -       Rodin       - Codename:ExperimentExp12
Noutate:Operatorii logici SI &&, SAU !! sunt implementati.
Puteti sa scrieti serii lungi de SAU ori de SI acum.
Am definit si o instructiune alternativa generalizata.
Cross compilat pe Linux. La adapost de virusi !
Programul:p7.txt
{ tex "Salut ai reuit sa rulezi programul"
}
main.exe: (line 1, column 3):
unexpected "t"
expecting space or Comanda corecta.
```

Fig.21 (exemplu de eroare la execuție)

- Interpretorul din fişierul main.exe (din dosarul C:\Rodin) este componenta ce controlează execuția programului. Când ea îşi încetează activitatea se revine la programul care a apelat-o, fie el cmd sau TotalEdit.

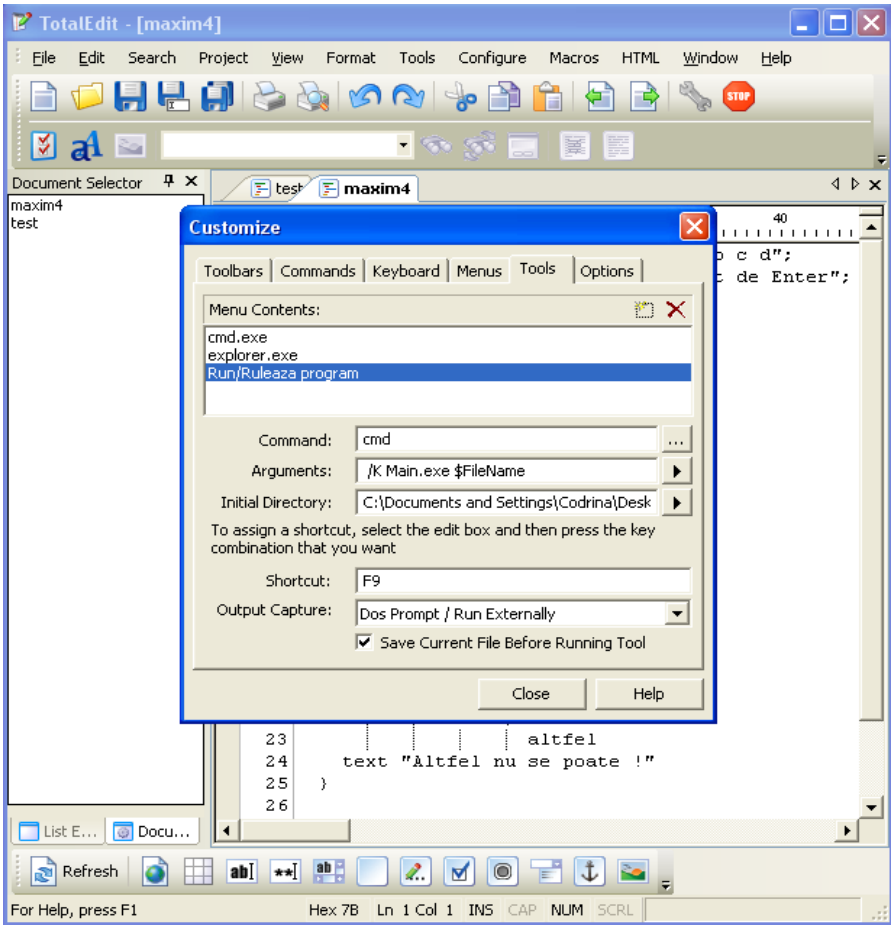

Fig.22 (setarea editorului TotalEdit pentru a putea folosi limbajul RODIN)

În imaginea de mai sus se poate vedea cum se poate seta editorul TotalEdit pentru a folosi

pseudocodul Rodin.

Un program descrie o corespondență între 2 mulțimi de valori: una este cea a **datelor inițiale sau de intrare** ale programului; cealaltă constituie **rezultatele sau datele de ieşire ale programului**. Programul nu conține doar descrierea operațiilor ce trebuie executate de calculator, ci şi o descriere a datelor cu care se prelucrează. Datele sunt fie constante fie variabile.

Datele constante nu-şi modifică valoarea pe parcursul execuției programului; celelalte o pot face însă. Mai precis, o variabilă are, la un moment dat, o singură valoare, pe care şi-o poate modifica în timpul execuției programului, ca urmare a unei atribuiri, sau de la o execuție la alta. Ilustrăm aceste noțiuni cu un exemplu foarte simplu, dar care conține multe din elementele ce alcătuiesc, în general, programele Rodin.

Exemplu in acrea par constante numerice intregi si stringuri (texte intre ghilimele):

```
{
 fie n=5;
 cat timp (n>0)
 {
  fie x=0;
  cat timp (x<5)
    {
    fie x=x+1;
    text "x=";
    scrie x
    };
  fie n=n-1;
  text "n=";
  scrie n
 }
}
```

Din exemplu se vede clar că o constantă are un *tip* şi o *valoare*. Atât tipul cât şi valoarea, sunt determinate de caraterele care intră în compunerea constantei. Valoarea unei constante nu poate fi schimbată în timpul execuției programului în care a fost utilizată.

Constante întregi şi caracter:

- **Constante întregi:**
  - Constantele întregi sunt scrise în sistemul de numerație cu baza 10. Alte limbaje admit si bazele 2, 8, 16.
  - O *constantă zecimală întreagă* este un şir de cifre zecimale care are prima cifră diferită de 0.
- **Constante caracter:**

- Prelucrarea datelor cu ajutorul calculatoarelor are în vedere, printre altele, posibilitatea lucrului pe texte formate din caractere. În acest scop, caracterele se codifică folosind coduri numerice. Cele mai utilizate coduri sunt codurile *EBCDIC (Extended Binary Coded Decimal Interchange Code)* şi *ASCII (American Standard Code for Information Interchanged).*
- La calculatoarele compatibile *IBM PC* se utilizează codul *ASCII*. Caracterele acestui cod le împartim în:
  - *Caractere negrafice* - codul lor este intervalul [0,31] la care se adaugă şi codul 127 (caracterul DEL).
  - *Spațiul* - are codul 32
  - *Caractere grafice* - codul lor este în intervalul [33,126].
- Caracterele grafice împreună cu spațiul formează setul de caractere *imprimabile.*
- O constantă caracter are ca valoare codul *ASCII* al caracterului pe care-l reprezintă. Ea are tipul *int.*

Variabile simple şi tablouri:

Într-un program utilizăm alături de date constante şi date variabile care îşi schimbă valorile în timpul execuției programului.

Dacă la o dată constantă ne putem referi folosind caracterele din compunerea ei, la o dată variabilă trebuie să ne referim altfel. Cel mai simplu mod este acela de a denumi data respectivă. Numele datei ne permite accesul la valoarea ei precum şi schimbarea valorii dacă este necesar.

În cazul în care o dată nu are legături cu alte date (de exemplu de ordine), vom spune că ea este o dată *izolată*. Numele unei date izolate se spune că reprezintă o *variabilă simplă*.

Unei date izolate îi corespunde un *tip*. În cursul execuției programului se pot schimba valorile unei date variabile dar nu şi tipul ei. Deocamdata rodin e un limbaj monotip.

Corespondența dintre numele unei date variabile şi tipul ei se defineşte in general printr-o *declarație*. Rodin fiind un limbaj monotip, cu variabile intregi, nu se folosesc declaratii.

Adesea, într-un program este util să considerăm grupe de date. Gruparea datelor se poate face în mai multe moduri.

Un mod simplu de a grupa date este acela de a considera date de *acelaşi* tip, în aşa fel încât grupa respectivă să formeze o *mulțime ordonată* de elemente la care să ne putem referi folosind *indici*. O astfel de grupă se spune ca formează un *tablou*. Unui tablou i se dă un nume. Tipul comun al elementelor unui tablou este şi tipul tabloului respectiv. De exemplu, o mulțime ordonată de întregi

reprezintă un tablou de tip întreg.

Referirea la elementele unui tablou se face printr-o *variabilă cu indici*.

O variabilă cu indici se compune din numele tabloului urmat de valorile indicilor, fiecare indice fiind reprezentat printr-o expresie inclusă între paranteze pătrate.

Exemplu: daca vectorul se numeste v si ne referim la elementul din pozitia a i-a, in limbajele de programare (n.n. Cu exceptia unor foarte vechi) se scrie v[i] nu v_i cum stiam de la matematica.

Declaţia de variabilă simplă in limbaje de tip C:

Declaraţia unei variabile simple stabileşte legătura dintre numele variabilei şi tipul valorilor pe care le poate avea variabila respectivă.

În cea mai simplă formă o declaraţie de variabilă simplă are formatul:

```
Tip lista_de_nume;
```

În calitate de *tip* putem folosi cuvintele cheie ale tipurilor predefinite.

`Lista_de_nume` se compune dintr-un nume de variabilă simplă sau mai multe separate prin virgule.

Exemple:

1. `int i,j;`

Variabilele *i* şi *j* sunt variabile simple de tip *int.* Compilatorul alocătuieşte pentru fiecare o zonă de memorie de 16 biţi.

Prin *i* ne referim la valoarea conţinută în zona de memorie alocată variabilei *i*. De asemenea, tot prin intermediu lui *i* putem atribui sau modifica valoarea din zona de memorie alocată variabilei *i.*

2. `Char c;`

Variabila *c* este o variabilă simplă de tip *char*. Ei i se alocă o zonă de memorie de un octet (8 biţi).

În această zonă putem păstra un caracter al codului *ASCII* extins, prin codul lui. Prin *c* ne putem referi la caracterul respectiv.

Notati faptul ca Rodin fiind un limbaj monotip (deocamdata), aceste declaratii nu sunt necesare, nu se folosesc.

Declaraţia de tablou:

Alte limbaje admit şi vectori multidimensionali, matrice etc.

In limbajul Rodin un tablou ca orice variabilă simplă, NU trebuie declarat înainte de a fi utilizat. Initializarea unui element printr-o atribuire aloca dinamic elementul. Tablourile pot fi rare in sensul ca elemente din ele nu sunt nici macar alocate.

Instrucţiunea *scrie*:

Instrucţiunea *scrie* poate fi folosită pentru a afişa date pe ecranul terminalului standard sub controlul unor formate. Ea poate fi apelată printr-o construcţie de forma:

```
Scrie(<exp>);
```

Unde:

<*exp*>- este o expresie. Datele gestionate prin *scrie* sunt supuse unor transformări. Aceasta din cauză că datele au un format extern şi unul intern.

Parametrul *control* al funcţiei *scrie* defineşte aceste transformări ale datelor care se afişează pe ecran, transformări care se mai numesc *conversii*. El conţine aşa numiţii *specificatori de format* care definesc conversiile datelor din format intern în format extern. În afara acestor specificatori de format, parametrul *control* al funcţiei *scrie* poate conţine succesiuni de caractere care se afişează ca atare în poziţii corespunzătoare.

În concluzie, parametrul *control* are în compunerea sa:

- Succesiuni de caractere care se afişează ca atare;
- Specificatori de format care definesc conversiile valorilor parametrilor *par1,par2,...,parn* din format intern în format extern.

Exemplu:

```
scrie ("\n pentru a termina programul actionati o tasta\n");
```