

Ministerul Educației, Culturii și Cercetării al Republicii Moldova

Anatol GREMALSCHI
Ludmila GREMALSCHI

INFORMATICĂ

Manual pentru clasa a  -a

ȘTIINȚA, 2019

CZU 004 (075.3)

G 80

Manualul este elaborat conform curriculumului disciplinar în vigoare și aprobat prin Ordinul ministrului educației (nr. 455 din 31 mai 2013). Finanțat din sursele financiare bugetare.

Comisia de evaluare: *Svetlana Golubev*, prof. șc., gr. did. superior, specialist principal la informatică, DGETS, mun. Chișinău; *Sergei Balic*, prof. șc., gr. did. al doilea, Liceul Teoretic „Vasile Alecsandri”, or. Călărași; *Valeriu Brodicico*, prof. șc., gr. did. superior, Liceul Teoretic „Boris Cazacu”, or. Nisporeni

Recenzenți: *Mircea Andronic*, șeful Catedrei de informatică economică; *Ruslan Bantea*, șeful Catedrei de informatică generală; *Vitalie Zavadschi*, director-adjunct, Colegiul de Informatică din Chișinău

Redactor: *Vasile Bahnar*

Corectori: *Mariana Belenciuc*, *Maria Cornesco*

Redactor tehnic: *Nina Duduciuc*

Machetare computerizată: *Anatol Andrițchi*

Copertă: *Vitalie Pogolșa*

Întreprinderea Editorial-Poligrafică *Știința*,

str. Academiei, nr. 3; MD-2028, Chișinău, Republica Moldova;

tel.: (+373 22) 73-96-16; fax: (+373 22) 73-96-27;

e-mail: prini_stiinta@yahoo.com; www.editurastiinta.md

Difuzare:

ÎM Societatea de Distribuție a Cărții *PRO-NOI*,

str. Alba-Iulia, 75; bloc Q; MD-2071, Chișinău, Republica Moldova;

tel.: (+373 22) 51-68-17; fax: (+373 22) 58-02-68;

e-mail: info@pronoi.md; www.pronoi.md

Toate drepturile asupra acestei ediții aparțin Întreprinderii Editorial-Poligrafice *Știința*.

Descrierea CIP a Camerei Naționale a Cărții

Gremalschi, Anatol

Informatică: Manual pentru clasa a 8-a / Anatol Gremalschi, Ludmila Gremalschi; Comisia de evaluare: Svetlana Golubev [et al.]; Min. Educației, Culturii și Cercetării al Rep. Moldova. – Ch.: Î.E.P. *Știința*, 2019 (Tipografia „BALACRON” SRL). – 116 p.

Bibliogr.: p. 115 (21 tit.): tab.

ISBN 978-9975-85-155-8

004 (075.3)

ISBN 978-9975-85-155-8

© *Anatol Gremalschi, Ludmila Gremalschi*. 2004, 2013, 2019

© Î.E.P. *Știința*. 2004, 2013, 2019

CUPRINS

Capitolul I. Păstrarea informației în foile de calcul	5
1.1. Elementele unei foi de calcul	5
1.2. Tipuri de date	9
1.3. Introducerea și editarea datelor	12
1.4. Formatarea datelor	16
1.5. Formate numerice	21
Capitolul II. Formule și calcule	26
2.1. Operatori și operanzi	26
2.2. Adresarea celulelor	30
2.3. Evaluarea formulelor	35
2.4. Funcții	39
Capitolul III. Diagrame și obiecte grafice	46
3.1. Elementele unei diagrame	46
3.2. Crearea diagramelor	50
3.3. Editarea diagramelor	55
3.4. Hărți și obiecte grafice	60
3.5. Construirea graficelor	65
Capitolul IV. Baze de date	69
4.1. Elementele unei baze de date	69
4.2. Gestionarea listelor	72
4.3. Sortarea înregistrărilor	76
4.4. Selectarea înregistrărilor	81
Capitolul V. Algoritmi	85
5.1. Algoritmi și executanți	85
5.2. Subalgoritmi	91
5.3. Algoritmi repetitivi. Ciclu cu contor	95
5.4. Algoritmi repetitivi. Ciclu cu condiție	101
5.5. Algoritmi cu ramificări	105
5.6. Algoritmul de funcționare a calculatorului	108
5.7. Generalități despre algoritmi	111
Bibliografie	115

Dragi prieteni,

Informatica ne schimbă viața, uneori într-un mod cu totul neașteptat chiar și pentru cei inițiați în domeniu. Pentru a stăpâni schimbarea, pentru a fi competitivi, pentru a beneficia de roadele acestei științe, trebuie să cunoaștem și să putem aplica întregul arsenal de metode și tehnici informaționale.

S-a stabilit că performanțele calculatoarelor moderne – viteza de operare, capacitatea memoriei interne, componența și caracteristicile unităților periferice – se dublează practic în fiecare an. Același progres spectaculos se atestă și în domeniul comunicațiilor, fapt ce permite conectarea tuturor calculatoarelor într-o rețea globală, care conține cantități enorme de informații. Aceste informații sunt produse de oameni și pentru oameni. Pentru a utiliza eficient informațiile acumulate și pentru a contribui la crearea de noi cunoștințe, fiecare om trebuie să-și formeze și să-și dezvolte cultura informațională și gândirea algoritmică. Această cultură presupune cunoașterea profundă a concepțiilor de bază ale informaticii și capacitatea fiecărei persoane de a elabora algoritmi pentru soluționarea problemelor pe care le întâmpină în activitatea cotidiană.

Dacă la începutul secolului trecut, pentru a face față provocărilor timpului, de la fiecare membru al societății se cerea știință de carte, astăzi, la început de mileniu, de la fiecare om se cere tot mai insistent și “știință de calculator”. Nu în zadar în mai multe limbi ale lumii a apărut termenul “licență (permis) de conducere a calculatorului”. Conform datelor de ultimă oră, fără acest permis, majoritatea profesiilor moderne, inclusiv cele tradiționale, devin inaccesibile.

Manualul de față are drept scop însușirea de către elevi a cunoștințelor necesare pentru prelucrarea automată a datelor cu ajutorul tabelelor de calcul și formarea gândirii algoritmice.

Fiind cele mai răspândite programe de calculator, aplicațiile de calcul tabelar sunt utilizate pentru păstrarea datelor, pentru efectuarea calculelor, pentru prezentarea informațiilor într-o formă atractivă. Ca și în cazul studierii editoarelor de texte sau a editoarelor grafice, aplicațiile de calcul tabelar reprezintă un instrument puternic, care ne ajută să cunoaștem metodele de reprezentare, păstrare și prelucrare a informației, organizarea ei în baze de date. Cunoștințele teoretice și practice din domeniul aplicațiilor de calcul tabelar vor fi de un real folos nu numai după absolvirea gimnaziului, dar și în procesul studierii altor discipline școlare, cum ar fi matematica, fizica, chimia, geografia, istoria etc.

Formarea gândirii algoritmice presupune cunoașterea și înțelegerea noțiunilor de executant, algoritm, formele de reprezentare și proprietățile algoritmilor. Materiile teoretice și practice, incluse în manual, Vă vor ajuta să faceți primii pași în lumea algoritmilor, să elaborați propriile programe de calculator.

Vă dorim succese!

Autorii

PĂSTRAREA INFORMAȚIEI ÎN FOILE DE CALCUL

1.1. ELEMENTELE UNEI FOI DE CALCUL

Termeni-cheie:

- foaie de calcul
- celulă
- referință/adresă de celulă
- valori și formule
- calcul tabelar

Cunoaștem deja că orice calculator prelucrează cele mai diverse informații: texte, imagini, secvențe audio și video etc. În calculator informația supusă prelucrării este reprezentată în formă de date, care, la rândul lor, sunt păstrate în fișiere. Fișierele sunt create și modificate cu ajutorul aplicațiilor-program care permit introducerea, prelucrarea și extragerea datelor respective. Exemple de aplicații:

Notepad (Bloc de foi pentru notițe) – un program destinat introducerii, corectării și tipăririi textelor mici;

Paint (Pictează) – un program pentru pictorii începători, destinat prelucrării unor imagini simple;

Media Player – un program pentru reproducerea secvențelor audio și video;

Word – un program care operează cu documente complexe ce conțin texte, tabele, imagini, secvențe sonore și video.

Pe parcursul dezvoltării informaticii s-a stabilit că în multe domenii ale activităților cotidiene sunt necesare aplicații care ar facilita procesele de prelucrare a datelor organizate în formă de tabele. Pentru exemplificare amintim evidența cărților într-o bibliotecă sau a mărfurilor într-un magazin, elaborarea bugetelor familiale sau de afaceri, prelucrarea datelor unui experiment științific etc. În scopul simplificării proceselor de introducere, de prelucrare și de afișare a datelor care se întâlnesc frecvent în astfel de probleme, au fost elaborate programe dedicate, denumite **aplicații de calcul tabelar**. Într-o astfel de aplicație datele supuse prelucrării sunt organizate în formă de tabele speciale, denumite **foi de calcul**. Structura unei foi de calcul este prezentată în *figura 1.1*.

O foaie de calcul este formată din următoarele elemente:

– **coloane**, identificate prin una sau două litere majuscule ale alfabetului latin: A, B, C, ..., Z, AA, AB, AC etc.;

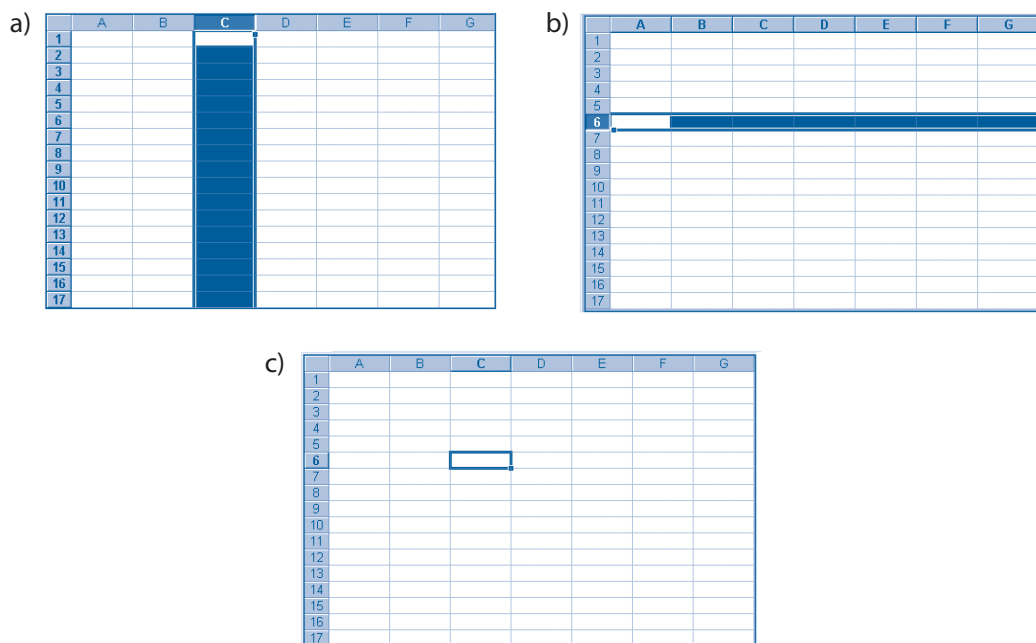


Fig. 1.1. Elementele unei foi de calcul:
a) coloană; b) rând; c) celulă

- **rânduri**, identificate prin numerele 1, 2, 3, 4 etc.;
- **celule**, identificate prin specificarea coloanei și rândului cărora le aparțin.

De exemplu, celula care se află la intersecția coloanei C și a rândului 6 (fig. 1.1c) este identificată prin combinația C6. O astfel de combinație poartă denumirea de **referință** sau **adresă de celulă**.

Numim referință sau adresă de celulă combinația formată prin juxtapunerea (alipirea) identificatorilor coloanei și rândului la intersecția cărora ea se află.

Datele supuse prelucrării și instrucțiunile referitoare la modul în care ele vor fi prelucrate se introduc în celulele foi de calcul. În general, celulele unei foi de calcul pot conține următoarele date:

1. Valori (texte, numere, date calendaristice sau mărci de timp) care reprezintă datele supuse prelucrării. Valorile sunt afișate pe ecran, tipărite la imprimantă și pot fi folosite pentru efectuarea calculului.

2. Formule care indică operațiile ce trebuie efectuate asupra valorilor din alte celule. De obicei, o formulă este alcătuită din adrese de celule și semnele matematice: + (adunarea), – (scăderea), * (înmulțirea) și / (împărțirea), care exprimă operațiile ce trebuie efectuate în procesul prelucrării informației. Fiecare formulă produce o valoare care este afișată în celula respectivă și poate fi folosită în alte formule.

Evident, în fiecare celulă luată aparte, utilizatorul poate înscrie numai un singur fel de date – o valoare sau o formulă. Imediat după modificarea conținutului oricăreia din celule, aplicația de calcul tabelar parcurge toate celulele foi de calcul și reevaluează formulele întâlnite, memorând în celulele respective și afișând pe ecran rezultatele eva-

luărilor. Întrucât capacitatea de prelucrare a calculatoarelor moderne este foarte mare, evaluarea formulilor se efectuează rapid chiar și în cazul unor foi de calcul care conțin milioane de celule.

Numim calcul tabelar prelucrarea automată a datelor din celulele unei foi de calcul. Celulele foi de calcul conțin valori ce reprezintă datele de prelucrat și formule care descriu modul de prelucrare.

În sistemul de operare Windows prelucrarea datelor din foile de calcul se realizează cu ajutorul aplicației **Microsoft Excel**, fereastra căreia este prezentată în figura 1.2.

	A	B	C	D	E	F	G
1							
2							
3		Nr. crt.	Denumire	Preț unitar	Cantitate	Cost	
4		1	Caiet	2,50	10	25,00	
5		2	Mapă	5,25	2	10,50	
6		3	Rîglă	1,60	1	1,60	
7		4	Pix	2,75	2	5,50	
8		5	Creion	1,30	4	5,20	
9		6	Radieră	0,25	2	0,50	
10			Total			48,30	
11							
12							

Fig. 1.2. Fereastra aplicației **Microsoft Excel**

Această fereastră conține următoarele elemente specifice aplicației **Microsoft Excel**:

1. Bara de titlu. Conține denumirea aplicației și numele fișierului activ, în cazul nostru, **Cumparaturi**. Amintim totodată că fișierele create de aplicația **Microsoft Excel** se numesc **agende de lucru**, iar o agendă poate conține mai multe foi de calcul.

2. Bara de meniuri. Conține meniurile **File** (Fișier), **Edit** (Editare), **View** (Aspect), **Insert** (Inserare), **Format** (Formatare), **Tools** (Instrumente), **Data** (Date), **Window** (Fereastra) și **Help** (Asistență). Fiecare dintre aceste meniuri conțin comenzi care se lansează în execuție prin metodele standard ale sistemului de operare **Windows**. De exemplu, comenzile meniului **File** permit crearea unei agende de lucru noi, deschiderea unei agende existente sau salvarea agendei curente, iar comenzile meniului **Edit** – ștergerea, copierea și inserarea datelor. Ca și în cazul altor aplicații, comenzile din meniul **Help** permit consultarea manualului de asistență, care conține informații complete despre destinația și modul de utilizare a tuturor facilităților oferite de programul de calcul tabelar.

3. Bara de instrumente standard. Această bară conține instrumentele utilizate frecvent, de exemplu, deschiderea și salvarea fișierelor (în cazul aplicației **Microsoft Excel**, deschiderea și salvarea agendelor de lucru), imprimarea foilor de calcul, copierea și inserarea datelor, prelucrarea datelor etc. Amintim că instrumentele standard simplifică accesul la comenzile utilizate frecvent din meniuri, iar destinația butoanelor respective poate fi concretizată fixând cursorul pe pictograma dorită.

4. Bara de instrumente de formatare. Această bară este similară celei din aplicația **Microsoft Word** și conține un set de instrumente care permit stabilirea fontului și dimensiunilor caracterelor, stilul de afișare, alinierea datelor în celulă etc.




5. Caseta pentru nume. În această casetă calculatorul afișează adresa (referința) celei selectate. De exemplu, în cazul foii de lucru din *figura 1.2*, această adresă este F4. Să reținem că în foaia de calcul celula selectată este evidențiată de calculator printr-un chenar îngroșat.

6. Caseta pentru formule. În această casetă calculatorul afișează conținutul celei selectate și oferă utilizatorului posibilitatea să efectueze modificările dorite. De exemplu, în cazul selectării celei F4 a foii de lucru din *figura 1.2*, în această casetă se afișează formula =D4*E4, care are semnificația “înmulțiți conținutul celei D4 la conținutul celei E4 și afișați valoarea obținută”. În mod similar, celula F10 conține formula =SUM(F4:F9), care are semnificația “adunați valorile din celulele F4, F5, F6, F7, F8 și F9 și afișați valoarea obținută”.

7. Fereastra de document, în care se afișează una dintre foile de calcul ale agendei de lucru. Întrucât foile de lucru sunt foarte mari, ferestrele de document conțin, în general, bare pentru defilarea conținutului atât pe orizontală, cât și pe verticală. Evident, pe ecran este afișată numai o mică parte a foii de lucru, iar selectarea zonei de afișare cade în sarcina utilizatorului.

De obicei, în cazul afișării pe ecran sau tipăririi la imprimantă, în foaia de calcul se afișează nu formulele, ci rezultatele evaluării lor. Astfel, în cazul celei F4 din *figura 1.2* pe ecran se afișează valoarea 25,00 sau, cu alte cuvinte, rezultatul evaluării formulei =D4*E4 din această celulă. Exact la fel, în cazul celei F10 pe ecran se afișează rezultatul evaluării formulei =SUM(F4:F9), deci valoarea 48,30.

8. Etichetele pentru selectarea foilor, care sunt plasate în partea de jos a ecranului. Pentru a afișa o foaie de calcul, se selectează eticheta respectivă. De exemplu, în cazul agendei de lucru din *figura 1.2* este selectată eticheta **Rechizite**, foaia respectivă fiind afișată pe ecran. Să reținem că agenda de lucru **Cumparaturi** din *figura 1.2* conține trei foi de calcul, și anume: **Rechizite**, **Alimente** și **Manuale**, comutarea acestora operându-se prin selectarea etichetei dorite.

Spre deosebire de editoarele de texte, în aplicațiile de calcul tabelar cursorul șoricelului în majoritatea cazurilor are forma unui semn masiv , care se poate schimba însă în funcție de instrumentul pe care îl folosim, de exemplu în  sau în .

Întrebări și exerciții

- 1 Numiți elementele unei foi de calcul.
- 2 Cum se identifică coloanele, rândurile și celulele unei foi de calcul?
- 3 Explicați termenul *referință/adresă de celulă*.
- 4 Indicați pe foaia de calcul din *figura 1.2* următoarele celule: B3, C5, C8, D3, F4, F10, G11 și A12.
- 5 Ce fel de date poate conține o celulă? Care este diferența dintre valori și formule?
- 6 Găsiți în foaia de calcul **Rechizite** din *figura 1.2* celulele care conțin valori și celulele care conțin formule. Se admite că agenda de lucru **Cumparaturi** este accesibilă pe calculatorul la care lucrați dvs.

- 7 Explicați modul de prelucrare automată a datelor din celulele unei foi de calcul.
- 8 Modificați numărul de creioane din celula E8 a foi de calcul **Rechizite** din *figura 1.2*. Observați cum se vor schimba imediat și valorile afișate în celulele F8 și F10. Cum se explică acest lucru?
- 9 Care este diferența dintre un tabel dintr-un document text și un tabel din cadrul unei foi de calcul?
- 10 Explicați termenul *calcul tabelar*.
- 11 Enumerați toate elementele ferestrei de aplicație din *figura 1.2*. Indicați destinația butoanelor ◀ și ▶ din colțul stânga-jos al acestei ferestre.
Care sunt asemănările și diferențele dintre ferestrele de aplicații **Microsoft Word** și **Microsoft Excel**?
- 12 Numiți câteva probleme care, în opinia dvs., ar putea fi soluționate cu ajutorul aplicațiilor de calcul tabelar.
- 13 Care este destinația comenzilor **New**, **Open**, **Close**, **Save**, **Save As** din meniul **File**?

1.2. TIPURI DE DATE

Termeni-cheie:

- tip de date
- tipul de date *număr*
- tipul de date *dată calendaristică*
- tipul de date *marcă de timp*
- tipul de date *text*
- tipul de date *logic*
- tipul de date *eroare*

Pentru prelucrarea corectă a datelor, aplicațiile de calcul tabelar trebuie să aibă informații despre natura lor. De exemplu, reprezintă oare datele supuse prelucrării texte sau numere, date calendaristice sau indicații ale ceasornicului etc. De asemenea, un program de calcul tabelar trebuie să cunoască modul în care aceste date vor fi reprezentate pe calculator, de exemplu, ca o succesiune de oceteți, ca numere întregi sau ca numere reale. Pentru a scuti utilizatorul de toate detaliile legate de reprezentarea internă a datelor și a asigura corectitudinea calculelor, în aplicațiile de calcul tabelar se utilizează noțiunea *tip de date*.

Numim tip de date o mulțime de valori și o mulțime de operații care pot fi efectuate cu acestea.

Aplicațiile de calcul tabelar utilizează mai multe tipuri de date, principalele dintre ele fiind următoarele:

1. Tipul de date *număr*. Mulțimea de valori a tipului de date *număr* include toate numerele reale care pot fi reprezentate pe calculator, iar mulțimea operațiilor ce pot fi efectuate cu ele include adunarea (notată prin semnul +), scăderea (-), înmulțirea (*) și împărțirea (/).

Valorilor de tip numeric li se poate atașa simbolul procentului sau al unităților monetare dolar, euro, leu ș.a.

Exemple:

7	215
-712	-3,14
234,3	428,56835
345,3289	89563,46935

2. Tipul de date *dată calendaristică*. În general, valorile acestui tip de date reprezintă tot numere, care sunt tratate însă ca date calendaristice. Aplicațiile de calcul tabelar reprezintă zilele, începând cu 1 ianuarie 1900 și terminând cu 31 decembrie 9999, prin numerele 1, 2, 3, 4 etc. În momentul afișării pe ecran, programul, folosind un calendar electronic, transformă numărul de ordine în data respectivă. Evident, numărul 1 va fi transformat în 1 ianuarie 1900, numărul 2 – în 2 ianuarie 1900, numărul 3 – în 3 ianuarie 1900 ș.a.m.d. Din contra, la introducerea datelor, calculatorul, folosind același calendar, calculează numărul de ordine al zilei indicate față de 1 ianuarie 1900. Pentru afișarea datelor calendaristice, există mai multe formate prestabilite, în țara noastră cele mai răspândite fiind ziua.luna.anul, ziua/luna/anul sau ziua-luna-anul.

Exemple:

01.01.04	16.02.13
01/01/04	16/02/13
01-ian-04	16-Feb-13

3. Tipul de date *marcă de timp*. Valorile acestui tip de date reprezintă numere fracționare de la 0 la 1, care sunt tratate de aplicație ca mărci de timp (indicații ale ceasornicului). De exemplu, numărului 0 îi corespunde ora 0.00, numărului 0,25 – ora 6.00, numărului 0,5 – ora 12.00, numărului 0,75 – ora 18.00 ș.a.m.d. Pentru afișarea mărcilor de timp există mai multe formate prestabilite, cele mai răspândite fiind ore:minute sau ore:minute:secunde.

Exemple:

06:00	21:04:39
6:30	09:04 PM
18:05	9:04 PM

Aplicațiile de calcul tabelar permit utilizarea concomitentă a datelor calendaristice și a mărcilor de timp, de exemplu, 01-ian-13 18:00. Valorile respective sunt reprezentate pe calculator cu un număr rațional, partea întreagă a căruia specifică data, iar cea fracționară – marca de timp. Accentuăm faptul că modul de scriere a datelor calendaristice și a mărcilor de timp diferă de la o țară la alta. Prin urmare, pentru fiecare calculator, formatul datelor respective va fi determinat de țara pentru care a fost configurată aplicația **Microsoft Excel**.

Întrucât tipurile de date *număr*, *dată calendaristică* și *marcă de timp* au pe calculator una și aceeași reprezentare internă, ele au și o denumire comună – **tipuri numerice**.

4. Tipul de date *text*. Mulțimea de valori a tipului de date *text* include șiruri formate din caractere imprimabile, iar una dintre operațiile admise este juxtapunerea (alipirea) lor.

Exemple:

Caiete
Radieră
Cheltuieli administrative
Costul manualelor

5. Tipul de date *logic*. Acest tip de date include valorile TRUE (adevărat) și FALSE (fals). Aceste valori pot fi obținute în urma evaluărilor anumitor formule din foaia de calcul.

6. Tipul de date *eroare*. Valorile acestui tip de date sunt generate de aplicația de calcul tabelar în cazurile când apar erori. Principalele valori de eroare sunt:

- #DIV/0! – împărțire cu zero;
- #N/A! – valoare lipsă;
- #NAME! – nume invalid;
- #NULL! – intersecție vidă între domenii;
- #NUM! – număr invalid;
- #REF! – referință invalidă;
- #VALUE! – valoare incorectă;
- ##### – indicator de afișare imposibilă.

Pentru exemplificare, în *figura 1.3* este reprezentată o foaie de calcul, valorile din celulele căreia aparțin următoarelor tipuri de date:

- a) tipul de date *text* – celulele B2, B4, B5, B6, B8, B9, ..., C4, C5 și C6;
- b) tipul de date *număr* – celulele C16, C17 și C18;
- c) tipul de date *dată calendaristică* – celulele C8 și C11;
- d) tipul de date *marcă de timp* – celulele C9 și C12.

	A	B	C	D
1				
2		Foaie de drum		
3				
4		Tipul unității de transport	Dacia-215D	
5		Numărul de înmatriculare	C IK 096	
6		Conducător auto	Rotaru Ion	
7				
8		Data plecării	12-Mai-03	
9		Ora plecării	07:00	
10				
11		Data sosirii	14-Mai-03	
12		Ora sosirii	19:00	
13				
14				
15		Indicațiile vitezometrului		
16		La plecare	345211	
17		La sosire	346189	
18		Distanța parcursă	978	
19				
20		Semnătura		
21				

Fig. 1.3. Tipuri de date într-o foaie de calcul

Tipul datelor din fiecare celulă a unei foi de calcul este stabilit de aplicația de calcul tabelar în momentul introducerii sau redactării valorilor respective și în procesul evaluării sau reevaluării formulilor. Alegerea tipurilor potrivite de date, care ar reflecta cât mai exact specificul problemelor ce vor fi soluționate cu ajutorul calculatorului, cade în sarcina utilizatorului.

Întrebări și exerciții

- ❶ Explicați termenul *tip de date*.
- ❷ Numiți mulțimea de valori ale următoarelor tipuri de date:
 - a) text;
 - b) număr;
 - c) dată calendaristică;
 - d) marcă de timp;
 - e) logic;
 - f) eroare.
- ❸ Determinați tipul de date pentru fiecare dintre valorile ce urmează:
 - a) Munteanu Vasile
 - b) 1998
 - c) Data 15.06.03
 - d) Ora exactă 18:00
 - e) 18:00
 - f) 21.03.03 07:00
 - g) 18:30:01
 - h) TRUE;
 - i) Total
 - j) #####
 - k) 3,14
 - l) 314
- ❹ Numiți tipul de date al valorilor din celulele foi de calcul **Rechizite** (fig. 1.2).
- ❺ Ce tipuri de date ar trebui utilizate pentru reprezentarea într-o foaie de calcul a următoarelor informații:
 - a) numele și prenumele elevului;
 - b) notele obținute de fiecare elev pe parcursul anului de învățământ;
 - c) înălțimea fiecărui elev;
 - d) greutatea fiecărui elev;
 - e) anul nașterii fiecărui elev;
 - f) clasa în care învață fiecare elev;
 - g) începutul și sfârșitul fiecărei lecții în orarul școlar;
 - h) data nașterii fiecărui elev?

1.3. INTRODUCEREA ȘI EDITAREA DATELOR

Termeni-cheie:

- introducerea valorilor
- introducerea formulilor
- selectarea obiectelor
- operații de editare

Aplicațiile de calcul tabelar oferă mai multe facilități de introducere și de editare a datelor. Pentru a introduce datele dorite într-o celulă, ea trebuie mai întâi selectată. Acest lucru se efectuează prin tehnicile standard utilizate în interfețele grafice:

- executarea unui clic-stânga pe celula dorită;
- deplasarea indicatorului celulei selectate (acest indicator reprezintă un chenar îngroșat) cu ajutorul tastelor ←, ↑, ↓, → sau al tastei **Tab**;
- indicarea adresei (referinței) celulei în fereastra de dialog afișată cu ajutorul comenzii **Edit, Go To** (Editare, Treci La).

Valorile pot fi introduse direct în celulă sau prin intermediul casetei de formule. Dacă se dorește utilizarea casetei de formule, mai întâi ea trebuie activată, executând în acest scop în casetă un clic-stânga. La **introducerea valorilor** vom respecta următoarele reguli:

- semnul plus (+) în fața oricărui număr va fi ignorat;
- numerele negative trebuie să fie precedate de semnul minus (-);
- lungimea textelor nu va depăși 255 de caractere;
- dacă dorim ca numărul introdus să fie interpretat ca text, plasăm un apostrof (‘) înaintea acestuia;
- în aceeași celulă pot fi introduse concomitent o dată și o marcă de timp, separate printr-un spațiu;
- introducerea unei valori se va încheia prin selectarea altei celule sau prin acționarea tastei **Enter**.

Formulele pot fi introduse, ca și valorile, fie direct în celule, fie prin intermediul casetei respective. În ambele cazuri, înaintea formulei, în mod obligatoriu, se introduce semnul egal (=).

Editarea datelor se efectuează cu ajutorul unor comenzi dedicate, majoritatea dintre ele fiind grupate în meniurile **Edit** (Editare) și **Insert** (Inserare). Comenzile utilizate frecvent pot fi lansate, de asemenea, cu ajutorul butoanelor respective din bara de instrumente standard. Destinația comenzilor de editare este prezentată în *tabelul 1.1*.

Tabelul 1.1

Comenzi pentru editarea datelor

Denumirea opțiunilor	Destinația
Meniul Edit (Editare)	
Undo (Anulează)	Anulează ultima comandă de editare executată până la activarea opțiunii în studiu.
Cut (Decupează)	Decupează obiectul selectat. Obiectul decupat sau conținutul acestuia este depus în memoria-tampon.
Copy (Copie)	Copie obiectul selectat sau conținutul acestuia în memoria-tampon.
Paste (Lipește)	Introduce obiectul din memoria-tampon sau conținutul acestuia în locul unde se află punctul de inserare.
Clear (Curăță)	Șterge conținutul sau atributele obiectului selectat.
Delete (Elimină)	Elimină din foaia de calcul obiectul selectat.
Delete Sheet (Elimină foaia)	Elimină foaia de calcul din agenda curentă.
Move or Copy Sheet (Mută sau copie foaia)	Mută sau copie foaia de calcul în cadrul agendei curente sau în cadrul altei agende de lucru indicate de utilizator.

Denumirea opțiunilor	Destinația
Meniul Insert (Inserare)	
Cells (Celule)	Inserează celule în vecinătatea obiectului selectat.
Rows (Rânduri)	Inserează rânduri în vecinătatea obiectului selectat.
Columns (Coloane)	Inserează coloane în vecinătatea obiectului selectat.
Worksheet (Foaie de calcul)	Inserează în agenda de lucru o foaie de calcul.

Tehnicile de editare a datelor sunt prezentate în *tabelul 1.2*. Să reținem că aplicarea tehnicilor din acest tabel presupune **selectarea obiectului** supus modificărilor. Un astfel de obiect poate fi conținutul unei celule sau celula propriu-zisă, o coloană sau un rând.

Tabelul 1.2

Tehnici de editare a datelor

Operația de editare	Consecutivitatea de acțiuni
Modificarea conținutului unei celule	<ul style="list-style-type: none"> – executăm dublu-clic pe celula dorită; – cursorul își schimbă forma în , simbolizând trecerea aplicației în regimul de redactare a conținutului din celulă; – modificăm conținutul celulei utilizând în acest scop facilitățile standard de editare a textelor; – după terminarea corectărilor, acționăm tasta Enter;
Copierea unei celule	<ul style="list-style-type: none"> – executăm clic pe celula sursă; – executăm comanda Copy; – executăm clic pe celula destinație; – executăm comanda Paste;
Mutarea conținutului unei celule	<ul style="list-style-type: none"> – executăm clic pe celula sursă; – executăm comanda Cut; – executăm clic pe celula destinație; – executăm comanda Paste;
Inserarea unei coloane (a unui rând)	<ul style="list-style-type: none"> – selectăm coloana (rândul) lângă care va fi inserată o coloană nouă (un rând nou), executând în acest scop un clic-stânga pe antetul respectiv; – executăm comanda Insert, Columns (Insert, Rows);
Eliminarea unei celule	<ul style="list-style-type: none"> – selectăm celula dorită; – executăm comanda Edit, Delete; – indicăm în caseta de dialog direcția în care va fi deplasat restul celulelor;
Eliminarea unei coloane (a unui rând)	<ul style="list-style-type: none"> – selectăm coloana (rândul) dorit; – executăm comanda Edit, Delete;
Ștergerea conținutului unei celule, a unei coloane sau a unui rând	<ul style="list-style-type: none"> – selectăm celula, coloana sau rândul dorit; – executăm comanda Edit, Clear; – selectăm opțiunea Contents;

Cu această ocazie amintim că **operațiile de editare** pot fi efectuate nu numai asupra unor obiecte separate (o coloană, un rând sau o celulă), dar și asupra unui grup de obiecte, de exemplu, un grup de coloane sau un grup de rânduri adiacente. Selectarea grupurilor respective se efectuează utilizând tehnicile de lucru cu șoricelul și tastele speciale ale tastaturii, de exemplu, glisând cursorul pe antetele coloanelor sau ale rândurilor dorite.

Întrebări și exerciții

- ❶ Cum poate fi selectată o celulă a foii de calcul? O coloană? Un rând?
- ❷ Explicați destinația comenzilor **Undo**, **Cut**, **Copy**, **Paste**, **Clear**, **Delete** ale meniului **Edit**.
- ❸ Care este diferența dintre comenzile **Cut** și **Copy**?
- ❹ Explicați destinația comenzilor **Cells**, **Rows**, **Columns** ale meniului **Insert**.
- ❺ Aflați cu ajutorul sistemului de asistență destinația și modul de utilizare a comenzii **Worksheet** din meniul **Insert**.
- ❻ Cum poate fi modificat conținutul unei celule?
- ❼ Numiți tipul de date al fiecărei valori din celulele foii de calcul **Salarii** (fig. 1.4).

	A	B	C	D	E	F	G
1							
2							
3							
4			Nume, prenume	Zile lucrate	Plata pe zi	Salariu	
5							
6							
7			Popescu Ion	20	225,8	4516	
8			Margine Vasile	15	132,15	1982,25	
9			Munteanu Cornel	19	148,3	2817,7	
10			Rotaru Irina	21	229,5	4819,5	
11							
12							
13			Total			14135,45	

Fig. 1.4. Foaia de calcul **Salarii**

- ❶ Introduceți în calculator foaia de calcul din *figura 1.4*. Accentuăm că în celulele F7, F8, F9 și F10 se introduc nu valorile indicate în desen, ci formulele de calcul, respectiv, =D7*E7, =D8*E8, =D9*E9 și =D10*E10. În celula F13 se introduce formula =SUM(E7:E10), destinată calculării sumei valorilor din celulele E7, E8, E9 și E10.
- ❷ Introduceți în foaia de calcul din *figura 1.4* următoarele modificări:
 - a) înlocuiți valoarea 225,8 din celula E7 cu valoarea 330,1;
 - b) înlocuiți prenumele Ion din celula C7 cu prenumele Sorin;
 - c) înlocuiți valoarea 19 din celula D9 cu valoarea 23;
 - d) eliminați coloana A;
 - e) eliminați rândurile 1 și 2.

Explicați schimbările care se produc în foaia de calcul, în special în rândul ce conține textul Total.
- ❸ Introduceți în calculator foaia de calcul din *figura 1.3*. Menționăm că în celula C18 se introduce nu valoarea 978, ci formula de calcul a distanței parcurse =C17-C16. Modificați foaia de calcul după cum urmează:

- a) inserați un rând deasupra denumirii Foaie de drum;
- b) inserați un rând între valorile Conducător auto și Data plecării;
- c) schimbați data plecării în 13-Mai-03;
- d) schimbați data sosirii în 15-Mai-03;
- e) înlocuiți consecutiv valoarea 346189 din celula C17 prin valorile 346203, 346715 și 346000.

Explicați schimbările care se produc în foaia de calcul.

1.4. FORMATAREA DATELOR

Termeni-cheie:

- obiect
- proprietăți ale obiectului
- format
- formatare
- format de coloană
- format de rând
- format de celulă

Foile de calcul conțin cele mai diverse **obiecte**: valori, celule, coloane, rânduri ș.a. Fiecare dintre aceste obiecte are anumite **proprietăți**. De exemplu, o celulă poate fi caracterizată prin dimensiuni, forma chenarului, culoarea fundalului pe care sunt afișate valorile respective. În mod similar, caracterele utilizate pentru reprezentarea conținutului unei celule se caracterizează prin fontul utilizat, de exemplu, *Times New Roman*, *Courier* sau *Arial*, dimensiuni și stilul de afișare *Regular*, *Bold* sau *Italic*. Ca și în cazul altor aplicații, proprietățile obiectelor utilizate în calculele tabelare se descriu cu ajutorul formatelor.

Formatul reprezintă un set de informații atașate fiecărui obiect, care descriu proprietățile lui. Procesul de stabilire și de modificare a acestor proprietăți se numește formatare.

Un **format de coloană (de rând)** include următoarele informații:

- lățimea coloanei (înălțimea rândului);
- regimul de afișare pe ecran.

Proprietățile coloanelor pot fi stabilite cu ajutorul comenzii **Column** (Coloană) din meniul **Format**. După lansarea comenzii **Format, Column**, aplicația de calcul tabelar afișează un submeniu care conține următoarele opțiuni:

Width (Lățimea) – stabilirea lățimii coloanelor. După activarea opțiunii **Width**, aplicația de calcul tabelar afișează pe ecran o casetă de dialog în care utilizatorul poate indica lățimea dorită. Această comandă acționează numai asupra coloanelor în care se află celulele selectate.



AutoFit Selection (Ajustare automată) – Lățimea coloanei va fi stabilită de aplicația de calcul tabelar în mod automat, astfel încât tot conținutul fiecărei celule să fie vizibil pe ecran.

Hide (Ascunde) – coloanele în care se află celulele selectate nu vor mai fi afișate pe ecran.

Unhide (Nu mai ascunde) – coloanele “ascunse” anterior cu ajutorul comenzii **Hide** vor fi din nou afișate pe ecran. Pentru a afișa coloanele dorite, înainte de lansarea comenzii **Unhide**, utilizatorul va selecta cel puțin câte o celulă din coloanele vecine cu cele ascunse.

Standard Width (Lățimea standard) – stabilește lățimea standard, care implicit va fi utilizată pentru toate coloanele din foaia de calcul. Să reținem că lățimea coloanelor se indică nu în milimetri sau centimetri, ci prin numărul maximal de cifre zecimale ce pot fi afișate în coloana respectivă.

Tot așa comanda **Format, Row** (Formatare, Rând) permite stabilirea înălțimii și a regimului de afișare a rândurilor unei foi de calcul. După lansarea acestei comenzi, aplicația de calcul tabelar afișează un submeniu care conține opțiunile **Height** (Înălțime), **AutoFit** (Ajustare automată), **Hide** (Ascunde) și **Unhide** (Nu mai ascunde).

Să reținem că utilizatorul poate schimba lățimea coloanelor și înălțimea rândurilor cu ajutorul șoricelului, plasând în acest scop cursorul în zona antetelor de coloană sau de rând. În momentul când cursorul este poziționat pe o linie ce desparte două coloane (două rânduri), el își schimbă forma din  în , indicând prin aceasta că lățimea coloanei (înălțimea rândului) poate fi modificată prin glisarea liniei respective.

Un **format de celulă** include următoarele informații:

- formatul numeric, care specifică modul în care va fi prelucrat conținutul celulei: ca text, număr, dată calendaristică sau marcă de timp;
- alinierea conținutului celulei pe verticală și pe orizontală;
- formatul caracterelor: fontul utilizat, stilul de afișare, dimensiunea caracterelor, culoarea de afișare, efectele speciale;
- aspectul chenarului;
- descrierea fundalului pe care va fi afișat conținutul celulei;
- regimul de protecție a fiecărei celule.

Formatarea celulelor se efectuează cu ajutorul comenzii **Format, Cells** (Formatare, Celule). După lansarea acestei comenzi, aplicația de calcul tabelar afișează fereastra de dialog cu același nume, care are mai multe pagini, fiecare dintre ele conținând elemente de control destinate formatării celulelor (*fig. 1.5*).

Formatarea caracterelor se efectuează cu ajutorul elementelor de control de pe pagina **Font** (*fig. 1.5*). Ca și în cazul unui editor de texte, utilizatorul poate selecta fontul dorit (Arial, Courier, Times New Roman ș.a.), stilul de afișare (Regular, *Italic*, **Bold**, **Bold Italic**), dimensiunile de caractere, culoarea caracterelor și efectele speciale. Accesul rapid la unele opțiuni din pagina **Font** se realizează prin intermediul barei de instrumente de formatare. Această bară conține listele derulante **Font**, **Font Size** (Dimensiunea fontului), **Font Color** (Culoarea fontului) și butoanele **Bold**, **Italic**, **Underline** (Subliniere). În același timp, elementele de control din bara de instrumente de formatare servesc și ca indicatoare de proprietăți. Ele își schimbă starea în funcție de proprietățile caracterelor și ale celulelor în care se află punctul de inserare.

Alinierea conținutului celulelor pe verticală și pe orizontală se efectuează cu ajutorul elementelor de control de pe pagina **Alignment** (Aliniere) a ferestrei de dialog **Format Cells** (*fig. 1.6*).

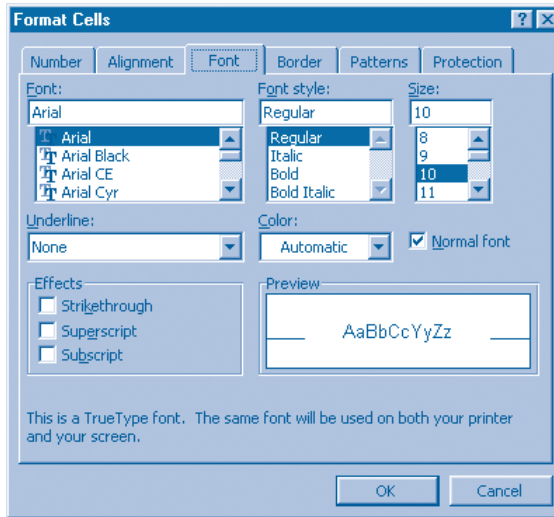


Fig. 1.5. Fereastra de dialog **Format Cells** (Formatarea celulelor), pagina **Font**

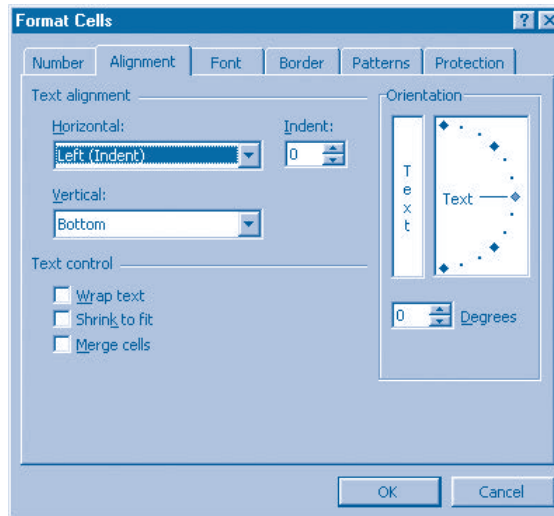


Fig. 1.6. Pagina **Alignment** (Aliniere)

Listele derulante de pe această pagină oferă utilizatorului diverse opțiuni de aliniere și de orientare a conținutului fiecărei celule, care, în general, sunt similare celor din aplicațiile destinate editării textelor. Casetele de marcare de pe această pagină permit aranjarea ordonată a textului (**Wrap text**), redimensionarea caracterelor conform dimensiunilor celulei (**Shrink to fit**) și comasarea celulelor vecine în scopul afișării complete a conținutului din celula curentă (**Merge cells**).

Pagina **Border** (Chenar) a ferestrei de dialog **Format Cells** (fig. 1.7) permite desena-rea chenarului celulelor selectate, oferind utilizatorului mai multe opțiuni: stilul liniei de chenar, culoarea chenarului, absența sau prezența unor laturi ale chenarului ș.a. Utilizarea rațională a acestor opțiuni permite punerea în evidență a celulelor ce conțin date impor- tante pentru utilizator.

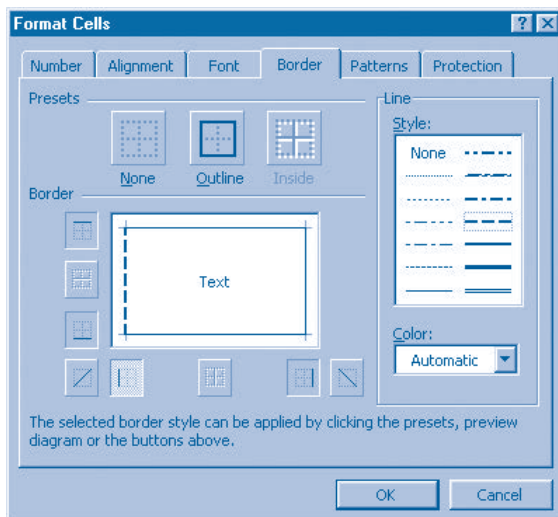


Fig. 1.7. Pagina **Border** (Chenar)

În mod similar, pagina **Patterns** (Modele) oferă utilizatorului posibilitatea să aleagă fundalul pe care va fi afișat conținutul celulelor supuse formătării (fig. 1.8).

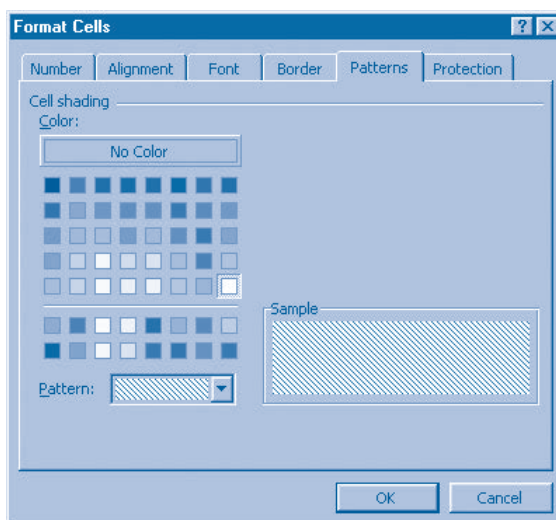


Fig. 1.8. Pagina **Patterns** (Modele)

Elementele de control ale paginii **Patterns** permit alegerea culorii fundalului (**Color**) și a modelului ce va fi desenat pe acest fundal (**Pattern**). Să reținem că alegerea nereușită a acestor formătări poate împiedica citirea corectă de către utilizator a datelor afișate pe ecran sau tipărite la imprimantă.

Întrebări și exerciții

- ❶ Explicați semnificația următorilor termeni: *obiect*, *proprietăți ale obiectului*, *format*, *formatare*.
- ❷ Numiți proprietățile următoarelor obiecte ale unei foi de calcul: rând, coloană, celulă, conținutul unei celule.
- ❸ Numiți proprietățile următoarelor obiecte ale unui document text: caracter, paragraf, pagină. Care dintre aceste proprietăți sunt comune și pentru obiectele din componența unei foi de calcul?
- ❹ Utilizând sistemul de asistență **Help**, aflați destinația elementelor de control de pe pagina **Protection** (Protecție) ale ferestrei de dialog **Format Cells**.
- ❺ Introduceți în calculator foaia de calcul din *figura 1.9*. Formatați rândurile, coloanele și celulele din această foaie exact la fel ca în *figura prezentată*.

	A	B	C	D	E	F	G	H	I
1									
2		Fonturi și stiluri		Alinierea textului		Chenare		Modele	
3		Arial		Stânga					
4		Courier		Centru					
5		Times New Roman		Dreapta					
6		Regular		Sus					
7		Bold		Centru					
8		<i>Italic</i>		Jos					
9		<i>Bold Italic</i>		Stânga-sus					
10		<u>Underline</u>		Dreapta-jos					
11									

Fig. 1.9. Formatarea celulelor

- ❻ Folosind ca model *figura 1.10*, formatați foaia de calcul **Rechizite**.

	A	B	C	D	E	F	G
1							
2							
3		Nr. crt.	Denumire	Preț unitar	Cantitate	Cost	
4		1	Caiet	2,50	10	25,00	
5		2	Mapă	5,25	2	10,50	
6		3	Riglă	1,60	1	1,60	
7		4	Pix	2,75	2	5,50	
8		5	Creion	1,30	4	5,20	
9		6	Radieră	0,25	2	0,50	
10			Total			48,30	
11							

Fig. 1.10. Foaia de calcul Rechizite

1.5. FORMATE NUMERICE

Termeni-cheie:

- format numeric
- categorie
- valoare memorată
- valoare afișată

E cunoscut faptul că fiecare celulă a unei foi de calcul poate conține o valoare sau o formulă. În procesul derulării aplicației de calcul tabelar, calculatorul evaluează toate formulele și afișează rezultatele obținute. Cu alte cuvinte, fiecare formulă produce o valoare care este afișată în celula respectivă și care poate fi folosită în alte formule. Pentru prelucrarea corectă a valorilor și a formulelor, aplicațiile de calcul tabelar tratează fiecare valoare ca aparținând unui anumit tip de date: *număr*, *text*, *dată calendaristică*, *marcă de timp*, *logic* sau *eroare*. Cunoșcând tipul de date din fiecare celulă, aplicațiile de calcul tabelar aleg în mod automat reprezentarea internă a datelor, care, fiind invizibilă pentru utilizator, asigură efectuarea corectă a operațiilor respective. De exemplu, calculatorul va aduna numere, însă nu va aduna texte, va înmulți, dar nu va împărți un număr la zero, nu va înmulți și nu va împărți texte etc.

Pe parcursul dezvoltării aplicațiilor de calcul tabelar s-a stabilit că numai cunoașterea tipului de dată al fiecărei valori este insuficientă pentru prelucrarea și afișarea corectă a informațiilor din foile de calcul. De exemplu, paginile dintr-un registru școlar sunt numerotate prin numerele naturale 1, 2, 3 etc., numărul de ordine al fiecărui elev înscris în registru de asemenea este redat prin numerele naturale 1, 2, 3 etc., însă notele medii ale fiecărui elev se exprimă prin numere reale pozitive din intervalul 1–10, având cel mult două cifre zecimale după virgulă. Evident, nu ne putem imagina un număr de pagină sau un număr de ordine egal cu 8,14, în timp ce utilizarea aceluiași număr pentru indicarea unei note medii este corectă. În toate cazurile din acest exemplu, valorile respective aparțin tipului de date *număr*, însă operațiile care pot fi efectuate asupra lor și modul în care ele vor fi afișate diferă.

Pentru a concretiza modul în care vor fi prelucrate și afișate datele din fiecare celulă, aplicațiile de calcul tabelar includ în formatul acestora o informație specială, denumită *format numeric*.

Formatul numeric este o parte componentă a formatului de celulă și reprezintă un set de informații care descriu modul în care vor fi prelucrate, afișate pe ecran și tipărite la imprimantă valorile respective.

Totodată, urmează să reținem că celelalte componente ale unui format de celulă – alinierea conținutului, formatul caracterelor, aspectul chenarului, descrierea fundalului și regimul de protecție – se referă numai la aspectul datelor într-o foaie de calcul, iar formatul numeric, împreună cu tipul de dată, definește modul în care va fi prelucrat conținutul fiecărei celule.

Pentru a atribui celulelor selectate formatul numeric dorit, se execută comanda **Format, Cells** și se alege pagina **Number** (Număr). După lansarea acestei comenzi,

aplicația de calcul tabelar afișează o fereastră de dialog cu același nume, din care utilizatorul selectează unul dintre formatele predefinite sau definește un format numeric propriu (fig. 1.11).

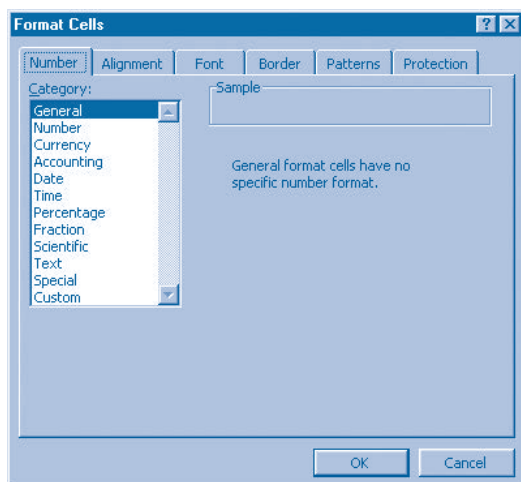


Fig. 1.11. Pagina **Number** a ferestrei **Format Cells**

Pentru a simplifica procesele de formatare, în aplicațiile de calcul tabelar formatele numerice sunt grupate pe categorii.

Categoria reprezintă un set de formate numerice caracteristice anumitor aplicații: numere întregi, numere reale, fracții ordinare, procente, sume de bani, date calendaristice, mărci de timp, numere de telefon etc.

Aplicația de calcul tabelar **Microsoft Excel** oferă utilizatorului următoarele categorii de formate numerice:

General (General) – această categorie nu stabilește celulei selectate un format numeric special, conținutul ei fiind afișat exact la fel cum a fost introdus de utilizator (fig. 1.11). De obicei, acest format este atribuit în mod automat tuturor celulelor în momentul creării unei noi foi de calcul.

Number (Număr) – această categorie permite selectarea unui format numeric pentru numerele reale, cu sau fără semn. Elementele de control de pe pagina respectivă permit stabilirea numărului de cifre zecimale după virgulă și modul de afișare a numerelor negative: cu semn sau în culoare roșie (fig. 1.12). Să amintim că numărul de cifre după virgulă se referă la modul de afișare, și nu la modul de păstrare a numărului în celula respectivă, aplicațiile de calcul tabelar memorând în celulă până la 15 cifre zecimale. De exemplu, în celula B2 este memorat numărul $\pi = 3,14159265358979$ (fig. 1.12, caseta pentru formule), care însă, având un format numeric cu două cifre zecimale după virgulă, afișează valoarea 3,14.

Currency (Valută) – pentru reprezentarea valorilor numerice cu semnificație de sume de bani. Elementele de control ale acestei categorii permit selectarea simbolului monetar (lei, euro, dolari SUA, lira sterlină, yenul japonez etc.), numărul de cifre zecimale după virgulă și modul de afișare a numerelor negative.

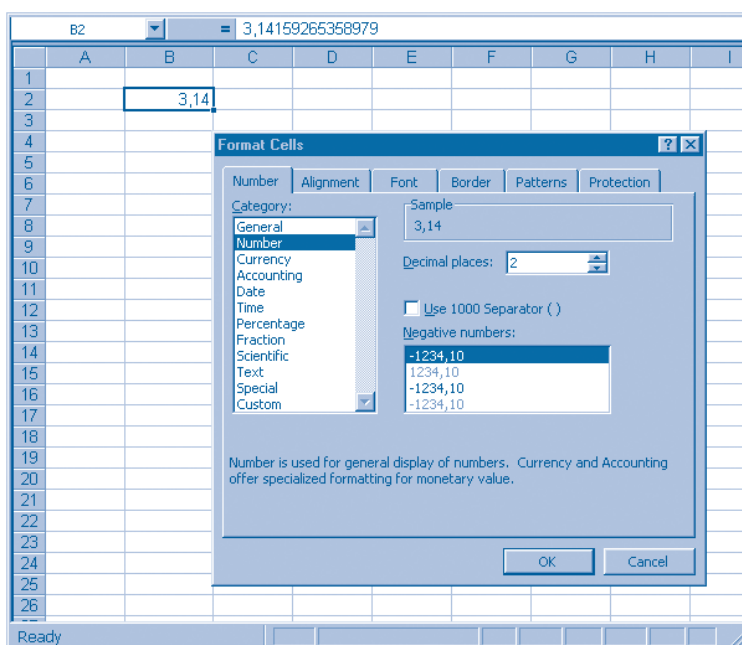


Fig. 1.12. Categoria **Number** (Număr)

Accounting (Contabilitate) – pentru reprezentarea valorilor numerice întâlnite frecvent în calculele de contabilitate. De obicei, numerele respective au semnificația de sume de bani, fiind aliniate conform virgulei care desparte partea întreagă de cea fracționară.

Date (Data) – această categorie permite selectarea unui format numeric pentru reprezentarea datelor calendaristice. Formatele prestabilite depind de țara pentru care a fost configurată aplicația de calcul tabelar, în Republica Moldova utilizându-se formatele ziua.luna.anul, ziua/luna/anul sau ziua-luna-anul.

Time (Timp) – pentru reprezentarea valorilor numerice cu semnificație de mărci de timp. Pentru afișarea mărcilor de timp, există mai multe formate prestabilite, cele mai răspândite fiind ore:minute sau ore:minute:secunde.

Percentage (Procentaj) – pentru reprezentarea valorilor numerice cu semnificație de procentaj. Aplicația de calcul tabelar va atașa numerelor respective simbolul %.

Fraction (Frație) – această categorie permite selectarea unui format numeric pentru reprezentarea fracțiilor ordinare.

Scientific (Științific) – pentru reprezentarea valorilor numerice într-o formă specială, utilizată în calculele științifice. Acest format se studiază în cursurile avansate de informatică.

Text – pentru reprezentarea valorilor numerice în formă de text. Acest lucru este util când se dorește prelucrarea separată a fiecărui caracter din scrierea unui număr.

Special – această categorie permite selectarea unui format numeric pentru reprezentarea unor numere speciale, de exemplu, a codurilor poștale, a numerelor de telefon, a codurilor de asigurare socială ș.a. Evident, formatele respective depind de țara pentru care a fost configurată aplicația de calcul tabelar.

Custom (Client) – pentru reprezentarea valorilor numerice în formate elaborate de utilizator. Modul de elaborare a formatelor numerice se studiază în cursurile avansate de informatică.

De obicei, imediat după crearea unei foi noi de calcul, tuturor celulelor li se atribuie în mod automat formatul numeric **General**. În procesul introducerii datelor, aplicația de calcul tabelar stabilește tipul lor, încercând să selecteze în mod automat și un format numeric potrivit. Uneori însă acest format nu corespunde intențiilor utilizatorului, el fiind obligat să indice explicit formatul numeric dorit. Acest lucru poate fi făcut până sau după introducerea datelor respective, lansând comanda **Format, Cells, Number**.

Să reținem că aplicațiile de calcul tabelar memorează și prelucrează valorile numerice cu o precizie de 15 cifre semnificative, iar formatele numerice intervin numai atunci când valorile respective sunt afișate pe ecran sau tipărite la imprimantă. Prin urmare, vom deosebi **valorile memorate**, adică valorile păstrate în celulele foi de calcul, și **valorile afișate**, adică valorile afișate pe ecran sau tipărite la imprimantă conform formatelor numerice atribuite celulelor respective.

Pentru exemplificare, în *figura 1.13* este prezentată o foaie de calcul care conține una și aceeași valoare $\pi = 3,14159265358979$, înscrisă în celule cu formate numerice diferite. Se observă că această valoare este afișată în celulele respective în funcție de formatul numeric atribuit fiecărei celule: ca număr întreg, ca număr real, ca o fracție ordinară, ca o valoare procentuală, ca text, ca dată calendaristică și ca marcă de timp. Amintim că în formă internă datele calendaristice sunt reprezentate prin partea întregă a unui număr real, numerotarea zilelor începând cu 1 ianuarie 1900, iar mărcile de timp (orele, minutele și secunde) se redau prin partea fracționară a numărului respectiv.

D11		= 3,14159265358979					
	A	B	C	D	E	F	G
1		Neformat		Formatat		Formatul numeric	
2							
3		3,14159265358979		3,14		Număr, 2 cifre după virgulă	
4		3,14159265358979		3,1416		Număr, 4 cifre după virgulă	
5		3,14159265358979		3		Număr, fără cifre după virgulă	
6		3,14159265358979		3,14159265358979		Text	
7		3,14159265358979		3,14 LEI		Valută, 2 cifre după virgulă	
8		3,14159265358979		03-Ian-00		Data calendaristică	
9		3,14159265358979		3:23:54		Marcă de timp	
10		3,14159265358979		314,16%		Procentaj, 2 cifre după virgulă	
11		3,14159265358979		3 1/8		Fracție, în optimi	
12							

Fig. 1.13. Formatarea numerelor

Întrebări și exerciții

- ❶ Explicați semnificația următorilor termeni: *format numeric*, *categorie*, *valoare memorată*, *valoare afișată*.
- ❷ Determinați când apare necesitatea atribuirii de formate numerice celulelor din componența unei foi de calcul.
- ❸ Numiți principalele categorii de formate numerice și explicați destinația lor.

- ④ Ce format numeric este atribuit în mod automat fiecărei celule din componența unei foi de calcul nou-create?
- ⑤ Explicați destinația fiecărei categorii de formate numerice prezentate în *figura 1.11*.
- ⑥ Indicați pe foaia de calcul din *figura 1.12* valoarea memorată și valoarea afișată.
- ⑦ Introduceți în calculator foaia de calcul din *figura 1.13*. Înscrieți în celulele din coloana D valoarea 3,14159265358979 și formatați celulele respective conform indicațiilor din coloana F. Explicați rezultatele afișate în celulele din coloana D. Indicați valorile memorate și valorile afișate.
- ⑧ Celulele D3, D4, ..., D11 ale foi de calcul din *figura 1.14* conțin una și aceeași formulă =B3. Această formulă copiază valoarea numerică din celula B3, denumită celulă de intrare, în celula în care ea se află. Pentru fiecare dintre celulele D3, D4, ..., D11 au fost stabilite formatele numerice indicate pe desen.

	A	B	C	D	E	F	G
1		Celula de intrare		Formatat		Formatul numeric	
2							
3		12345,1		12345,10		Număr, 2 cifre după virgulă	
4				12345,1000		Număr, 4 cifre după virgulă	
5				12345		Număr, fără cifre după virgulă	
6				12345,1		Text	
7				12 345,10 LEI		Valută, 2 cifre după virgulă	
8				18-Oct-33		Data calendaristică	
9				2:24:00		Marcă de timp	
10				1234510,00%		Procentaj, 2 cifre după virgulă	
11				12345 1/8		Fracție, în optimi	
12							

Fig. 1.14. Afișarea datelor în funcție de formatul numeric

Introduceți foia de calcul în calculator și determinați cum vor fi afișate valorile 1; 100; 1000; 1,879534; 32768,3146; 1504098987, înscrise consecutiv în celula de intrare B3.

- ⑨ Parcurgeți atent toate formatele numerice de pe pagina **Number** ale ferestrei de dialog **Format Cells** (*fig. 1.11*). Utilizând sistemul de asistență, aflați semnificația simbolurilor dd, mm, yy, # din componența formatelor numerice ce aparțin categoriei **Custom**.

2.1. OPERATORI ȘI OPERANZI

Termeni-cheie:

- formulă
- operand
- operator
- prioritățile operatorilor

Celulele unei foi de calcul pot conține valori sau formule. Valorile reprezintă datele supuse prelucrării, iar formulele indică operațiile care trebuie efectuate asupra valorilor din celule. Imediat după modificarea conținutului uneia dintre celule, aplicația de calcul tabelar parcurge toate celulele foi de calcul și evaluează formulele întâlnite, memorând în celulele respective valorile obținute.

Formulele reprezintă expresii matematice scrise într-o formă specială, care indică operațiile ce trebuie efectuate asupra valorilor din celule.

De obicei, aplicațiile de calcul tabelar afișează în celulele foi de calcul nu formulele propriu-zise, ci rezultatele evaluărilor respective. Întrucât conținutul celulei selectate este afișat în caseta pentru formule, utilizatorul poate vizualiza formulele respective, selectând consecutiv toate celulele foi de calcul supuse prelucrării. Dacă însă se dorește afișarea concomitentă a tuturor formulilor, se lansează comanda **Tools, Options, View** (Instrumente, Opțiuni, Aspect) și se marchează caseta **Formulas** (Formule). Pentru exemplificare, în figura 2.1. este prezentată foaia de calcul **Rechizite** până și după lansarea comenzii **Tools, Options, View, Formulas**.

În general, o formulă începe cu semnul “=” și este formată din operanzi, operatori și denumiri de funcții. Ordinea de efectuare a operațiilor poate fi concretizată cu ajutorul parantezelor rotunde (“(” și “)”).

Operanzii specifică valorile care vor participa în calcule. Ca operanzi, în aplicațiile de calcul tabelar, se utilizează constante, referințe (adrese de celule), denumiri de celule și funcții.

De exemplu, în formula

$$=A1+2$$

A1 este o referință, iar 2 este o constantă. În mod similar, în formula

a)

	A	B	C	D	E
1					
2	Datele supuse prelucrării				
3		1	2	Gimnaziu	
4		3	4	Liceu	
5					
6	Formule cu...				
7	operatori aritmetici	operatori relaționali	operatorul pentru texte		
8	=B3+C3	=B3<C3	=D3&D4		
9	=B3-C3	=B3>C3	=D4&D3		
10	=B4*C4	=B4>=C4	=D4&"1 Teoretic Călărași"		
11	=B4/C4	=B4<>C4	=D3&"1 nr. 2"		
12	=B4*C3	=C4>=B4	=D4&"1 "&"nr. 5"		
13	=(C4-B4)*2	=C4<=B4	=D4&D3&D4		
14					

b)

	A	B	C	D	E
1					
2	Datele supuse prelucrării				
3		1	2	Gimnaziu	
4		3	4	Liceu	
5					
6	Formule cu...				
7	operatori aritmetici	operatori relaționali	operatorul pentru texte		
8	3	TRUE	GimnaziuLiceu		
9	-1	FALSE	LiceuGimnaziu		
10	12	FALSE	Liceul Teoretic Călărași		
11	0,75	TRUE	Gimnaziul nr. 2		
12	9	TRUE	Liceul nr. 5		
13	1	FALSE	LiceuGimnaziuLiceu		
14					

Fig. 2.2. Operatori și operanzi într-o foaie de calcul:
a) afișarea formulelor; b) afișarea valorilor calculate

Atunci când într-o formulă se întâlnesc mai mulți operatori, ei vor fi evaluați într-o anumită ordine. Regulile de evaluare a unei formule sunt cele obișnuite în matematică:

- operațiile se efectuează conform priorității operatorilor;
- în cazul priorităților egale, operațiile se efectuează de la stânga la dreapta;
- mai întâi se calculează expresiile dintre paranteze.

Prioritățile operatorilor sunt indicate în *tabelul 2.1*.

Tabelul 2.1

Prioritățile operatorilor


Operator	Semnificație	Prioritate
^	ridicarea la putere	1 (maximă)
* și /	înmulțire și împărțire	2
+ și -	adunare și scădere	3
&	concatenare	4
=, <, >, <>, <=, >=	operatori relaționali	5 (minimă)

De exemplu, având două formule aparent identice, =2+3*4 și =(2+3)*4, rezultatele evaluării vor fi diferite. Astfel, formula =2+3*4 se evaluează în ordinea următoare:

$3 \times 4 = 12$, apoi $2 + 12 = 14$, iar rezultatul afișat în celulă va fi 14. Formula $=(2+3)*4$ va fi evaluată în ordinea indicată de paranteze: $2+3=5$, apoi $5 \times 4 = 20$, iar rezultatul afișat în celulă va fi 20.

Pentru a simplifica introducerea formulelor în celule, aplicațiile de calcul tabelar oferă tehnici speciale, cea mai simplă fiind specificarea referințelor prin selectarea celulelor respective. Pentru introducerea formulelor prin selectare:


– activăm caseta pentru formule;

– acționăm butonul  (**Edit Formula**);

– dacă dorim să introducem o referință, selectăm cu șoricelul celula dorită (celula selectată va fi înconjurată cu un chenar cu linie punctată, iar adresa ei va apărea în formulă);

– după introducerea unei referințe, continuăm formula prin selectarea de la tastatură a unui operator sau a unei paranteze;

– pentru terminarea formulei, acționăm butonul  sau tasta **Enter**;

– dacă renunțăm la introducerea formulei, acționăm butonul  (**Cancel Formula**).

Dacă în procesul introducerii sau evaluării unei formule apar erori, aplicația de calcul tabelar afișează un mesaj în care se specifică tipul erorii și, eventual, se propun anumite sugestii. Evident, utilizatorul ar trebui să evite acceptarea în mod mecanic a sugestiilor propuse de calculator, întrucât unele dintre ele ar putea să nu corespundă scopului urmărit.

Întrebări și exerciții

- 1 Care este destinația formulelor din componența unei foi de calcul?
- 2 Cum pot fi vizualizate formulele unei foi de calcul?
- 3 Explicați modul de introducere a formulelor prin selectare.
- 4 Numiți operatorii și operanzii din componența următoarelor formule:

a) $=A1+2+B4$	f) $=B2$
b) $=(A3+B5)/(A1-B3)+2$	g) $=B1<>B2$
c) $=B1\&"curent"$	h) $="luna"\&"mai"$
d) $=A1*(B1+B2)/2$	i) $=A1\&"numere"$
e) $=A1>B1$	j) $=A2=3$
- 5 Explicați semnificația următorilor termeni: *formulă*, *operator*, *operand*, *prioritățile operatorilor*.
- 6 Aranjați operatorii ce urmează în ordinea priorităților, începând cu prioritatea maximă:
 $<$, $+$, $*$, $\&$, $/$, $<>$, \wedge , $-$, $<=$.
- 7 Se admite că celula A1 conține valoarea 2, celula A2 – valoarea 3, celula B1 – valoarea Jos, iar celula B2 – valoarea Sus. Evaluați formulele ce urmează:

a) $=A1+A2$	f) $=A2$
b) $=A1+A2/A1-A2$	g) $=A1<>A2$
c) $=(A1+A2)/(A1-A2)$	h) $=B1\&B2$
d) $=A1*A1+A2/2$	i) $=B2\&B1$
e) $=A1*(A1+A2)/2$	j) $=B1\&"sau"\&B2$

- ⑧ Care este destinația formulelor din foaia de calcul din *figura 2.1*? Numiți operatorii și operanzii din componența acestor formule.
- ⑨ Introduceți în calculator foaia de calcul din *figura 2.2*. Înscrieți în celulele B3, C3 și D3 valorile, respectiv, 5, 6 și Școală. Observați cum se modifică valorile afișate în celulele care conțin formule. Explicați rezultatele afișate pe ecran.
- ⑩ Determinați tipurile de date ale valorilor din foaia de calcul din *figura 2.2*. Influențează oare tipul datelor rezultatele evaluării formulelor?
- ⑪ Înscrieți în celula B3 a foii de calcul din *figura 2.2* valoarea Sus. Aplicația de calcul tabelar va afișa în celulele B8 și B9 valoarea #VALUE!. Argumentați prin ce se explică acest lucru.
- ⑫ Introduceți în calculator foaia de calcul **Alimente** (*fig. 2.3*). Înscrieți în celulele F4, F5, F6, F7, F8 și F9 formulele necesare pentru a calcula costul fiecărui produs alimentar, iar în celula F10 – formula care calculează costul tuturor produselor procurate.

	A	B	C	D	E	F	G
1							
2							
3		Nr. crt.	Denumire	Preț unitar	Cantitate	Cost	
4		1	Lapte	3,20 LEI	3	9,60 LEI	
5		2	Pâine	2,60 LEI	2	5,20 LEI	
6		3	Crupe	5,80 LEI	4,5	26,10 LEI	
7		4	Sare	4,70 LEI	2	9,40 LEI	
8		5	Cașcaval	18,90 LEI	0,5	9,45 LEI	
9		6	Salam	42,30 LEI	0,3	12,69 LEI	
10			Total			72,44 LEI	
11							

Fig. 2.3. Foaia de calcul **Alimente**

2.2. ADRESAREA CELULELOR

Termeni-cheie:

- adresă relativă
- adresă absolută
- nume de celulă
- domeniu de celule

Din practică s-a stabilit că foile de calcul utilizate frecvent conțin foarte multe formule care diferă numai prin unele adrese de rând sau de coloană. De exemplu, în foaia de calcul **Rechizite** (*fig. 2.1*) formulele =D4*E4, =D5*E5, =D6*E6, ..., =D9*E9 din coloana F se deosebesc doar prin numărul de rând, care ia consecutiv valorile 4, 5, 6, ..., 9. Tot așa, formulele din coloana F a foii de calcul **Alimente** (*fig. 2.3*) se vor deosebi tot numai prin numerele de rânduri, formând aceeași secvență =D4*E4, =D5*E5, =D6*E6, ..., =D9*E9.

Pentru a simplifica introducerea formulelor, aplicațiile de calcul tabelar utilizează următoarele metode de referire (de adresare) a celulelor:

- adrese relative;
- adrese absolute;
- nume de celule;
- domenii de celule.

Adresele de celule formate prin juxtapunerea (alipirea) identificatorilor de rând și de coloană se numesc adrese relative.

Când o formulă este copiată, aplicația de calcul tabelar schimbă adresele relative din componența ei în așa mod, încât poziția relativă a operandilor față de celula care va conține formula copiată să rămână aceeași. De exemplu, presupunem că celula B1 conține formula =A1+2 (fig. 2.4). Se observă că poziția relativă a operandului A1 față de celula B1, care conține formula propriu-zisă, poate fi definită ca “celula care se află în același rând, dar în coloana vecină din stânga”. În procesul copierii formulei din celula B1, aplicația de calcul tabelar va modifica adresa relativă A1 în așa mod, încât poziția operandului față de celula în care este copiată formula să rămână fără schimbări. Prin urmare, în procesul copierii formulei =A1+2 din celula B1 în celulele B2, B3, ..., B10, adresa relativă A1 va fi schimbată în mod automat în A2, A3, ..., A10. În mod similar, în procesul copierii formulei =A1+2 din celula B1 în celulele C1, D1, E1, adresa relativă A1 va fi modificată în B1, C1, D1.

	A	B	C	D	E	F
1		=A1+2	=B1+2	=C1+2	=D1+2	
2		=A2+2				
3		=A3+2				
4		=A4+2				
5		=A5+2				
6		=A6+2				
7		=A7+2				
8		=A8+2				
9		=A9+2				
10		=A10+2				
11						

Fig. 2.4. Copierea formulelor ce conțin adrese relative

Întrucât în unele cazuri modificarea automată a adreselor nu este necesară, aplicațiile de calcul tabelar oferă utilizatorului posibilitatea să folosească adrese absolute.

Adresele de celule în care identificatorii de rând și de coloană sunt precedați de simbolul \$ se numesc adrese absolute.

De exemplu, adresele A1, B1, A2 sunt adrese relative, pe când adresele \$A\$1, \$B\$1, \$A\$2 sunt adrese absolute. Evident, când o formulă este copiată, adresele absolute nu vor fi modificate (fig. 2.5).

În general, copierea formulelor se efectuează prin utilizarea comenzilor **Edit**, **Copy** și **Edit**, **Paste** sau a butoanelor respective din bara de instrumente standard. Dacă însă se dorește copierea rapidă a unei formule în mai multe celule, putem utiliza tehnica “trage și lasă”, glisând pătrățelul ■ din colțul dreapta-jos al celulei ce conține formula de copiat peste celulele adiacente. Amintim că în momentul localizării “mănerului” ■ din colțul dreapta-jos, cursorul își schimbă forma din + în +, indicând prin aceasta trecerea aplicației în regimul de copiere.

	A	B	C	D	E	F
1		=\$A\$1+2	=\$A\$1+2	=\$A\$1+2	=\$A\$1+2	
2		=\$A\$1+2				
3		=\$A\$1+2				
4		=\$A\$1+2				
5		=\$A\$1+2				
6		=\$A\$1+2				
7		=\$A\$1+2				
8		=\$A\$1+2				
9		=\$A\$1+2				
10		=\$A\$1+2				
11						

Fig. 2.5. Copierea formulor ce conțin adrese absolute

Pentru a face formulele mai lizibile, aplicațiile de calcul tabelar permit a atribui celulelor un nume. Un **nume de celulă** trebuie să respecte următoarele reguli:

- să înceapă cu o literă sau cu caracterul “_” (liniuța de jos);
- să conțină numai litere, cifre, caracterul “.” (punct) sau “_”;
- să nu conțină spații;
- să nu fie asemănător unei referințe;
- să nu conțină mai mult de 255 de caractere.

Exemple de nume:

<i>corect</i>	<i>greșit</i>
Coeficient	lei/dolar
Cursul_valutar	tempe-ratura
temperatura	A2
Temperatura_Celsius	2S
Anul_2003	Anul 2003

Pentru a atribui celulei un nume, selectăm celula dorită, lansăm comanda **Insert, Name, Define** (Inserare, Nume, Definiere) și introducem în fereastra de dialog numele respectiv. De exemplu, în foaia de calcul **Alimente**, costul alimentelor procurate este calculat în lei, euro și dolari SUA. Recalcularea costurilor din lei în euro și dolari se efectuează utilizând coeficienții din celulele C13 și C14, denumite sugestiv Lei_Euro și Lei_Dolar (fig. 2.6, vezi p. 33).

Evident, fiecare nume de celulă reprezintă o adresă absolută, rămânând în procesul copierii fără schimbări.

Aplicațiile moderne de calcul tabelar permit referirea concomitentă a mai multor celule. De exemplu, formula =SUM(F4:F9) aplicată în baza foii de calcul **Alimente** calculează suma valorilor din celulele F4, F5, F6, F7, F8 și F9, celulele respective fiind indicate printr-o notație mai compactă F4:F9 (fig. 2.6, vezi p. 33).

Numim domeniu o submulțime de celule ale foii de calcul. Un domeniu poate fi adiacent (celulele sunt alăturate) sau neadiacent.

În cazul domeniilor adiacente, celulele respective pot fi adresate indicând adresele a oricare două celule care se află în colțurile opuse ale domeniului. Pentru separarea acestor adrese, se utilizează semnul “:” (două puncte). De exemplu, domeniul adiacent

a)

Lei_Euro		= 15,65							
	A	B	C	D	E	F	G	H	I
1									
2									
3		Nr. crt.	Denumire	Preț unitar, lei	Cantitate	Cost, lei	Cost, euro	Cost, dolari	
4		1	Lapte	3,20	3	9,60	0,61	0,66	
5		2	Pâine	2,60	2	5,20	0,33	0,36	
6		3	Crupe	5,80	4,5	26,10	1,67	1,80	
7		4	Sare	4,70	2	9,40	0,60	0,65	
8		5	Cașcaval	18,90	0,5	9,45	0,60	0,65	
9		6	Salam	42,30	0,3	12,69	0,81	0,87	
10		Total				72,44	4,63	4,99	
11									
12									
13		1 Euro =	15,65	lei					
14		1 Dolar =	14,52	lei					

b)

Lei_Euro		= 15,65							
	A	B	C	D	E	F	G	H	I
1									
2									
3		Nr. crt.	Denumire	Preț unitar, lei	Cantitate	Cost, lei	Cost, euro	Cost, dolari	
4		1	Lapte	3,2	3	=D4*E4	=F4/Lei_Euro	=F4/Lei_Dolar	
5		2	Pâine	2,6	2	=D5*E5	=F5/Lei_Euro	=F5/Lei_Dolar	
6		3	Crupe	5,8	4,5	=D6*E6	=F6/Lei_Euro	=F6/Lei_Dolar	
7		4	Sare	4,7	2	=D7*E7	=F7/Lei_Euro	=F7/Lei_Dolar	
8		5	Cașcaval	18,9	0,5	=D8*E8	=F8/Lei_Euro	=F8/Lei_Dolar	
9		6	Salam	42,3	0,3	=D9*E9	=F9/Lei_Euro	=F9/Lei_Dolar	
10		Total				=SUM(F4:F9)	=SUM(G4:G9)	=SUM(H4:H9)	
11									
12									
13		1 Euro =	15,65	lei					
14		1 Dolar =	14,52	lei					

Fig. 2.6. Utilizarea numelor de celule:
a) afișarea valorilor; b) afișarea formulelor

din figura 2.7a poate fi referit prin una dintre secvențele ce urmează: B2:E6, E6:B2, B6:E2 sau E2:B6. În cazul domeniilor neadiacente, se indică toate subdomeniile, adresele respective fiind separate prin semnul “;” (punct și virgulă). De exemplu, celulele domeniului neadiacent din figura 2.7b pot fi referite prin secvența B2:C7; E4:F5; E8:F8.

a)

	A	B	C	D	E	F	G
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							

b)

	A	B	C	D	E	F	G
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							

Fig. 2.7. Domenii de celule:
a) adiacente; b) neadiacente

Întrebări și exerciții

- ❶ Care este diferența dintre adresele relative și cele absolute?
- ❷ Determinați tipul adreselor ce urmează:
 - a) A1
 - b) B3
 - c) \$D\$8
 - d) C5
 - e) \$K\$12
 - f) K12
 - g) \$B\$21
 - h) D8
- ❸ Cum va arăta formula =A1+2 din celula B1 a foii de calcul din *figura 2.4*, dacă ea ar fi copiată în celulele C2, C3 și C4?
- ❹ Introduceți în calculator foaia de calcul din *figura 2.4*. Copiați formula din celula B1 în coloanele C, D, E și rândurile 2, 3 și 4 ale foii de calcul. Observați cum se modifică adresa relativă în procesul copierii acestei formule. Explicați rezultatele afișate pe ecran.
- ❺ Explicați în care cazuri se folosesc adresele relative și în care cazuri – cele absolute.
- ❻ Introduceți în calculator foaia de calcul din *figura 2.5*. Copiați formula din celula B1 în coloanele C, D, E și rândurile 2, 3 și 4 ale foii de calcul. Explicați rezultatele afișate pe ecran.
- ❼ Stabiliți în ce scop li se atribuie un nume unor celule din componența foilor de calcul.
- ❽ Care dintre numele de celule ce urmează sunt corecte:
 - a) Greutatea
 - b) Suma_catetelor
 - c) 31A
 - d) C5
 - e) \$K\$12
 - f) b^3
 - g) Perimetrul
 - h) AA3?
- ❾ Introduceți în calculator foaia de calcul din *figura 2.6*. Schimbați valorile din celulele Lei_Euro și Lei_Dolar în 15,95 și 14,78. Explicați rezultatele afișate pe ecran.
- ❿ Înlocuiți numele de celule Lei_Euro și Lei_Dolar din foaia de calcul din *figura 2.6* pin adresele celulelor respective. Comparați volumul de lucru în cazul utilizării numelor de celulă și a adreselor de celulă.
- ⓫ Indicați în *figura 2.7* următoarele domenii:
 - a) A1:E8
 - b) A1; B2; C3
 - c) A1:B3; D5:F6
 - d) B6:G10
 - e) G10:B6
 - f) B2; C2; D2; E2
 - g) B2; C3; D4; E5
 - h) F7:B2
- ⓬ Indicați pe foaia de calcul din *figura 2.6* domeniile referite în formulele din celulele F10, G10 și H10.

2.3. EVALUAREA FORMULELOR

Termeni-cheie:

- conversia tipurilor de date
- precedente de celulă
- dependente de celulă

Prin evaluarea unei formule se înțelege calculul valorii ei. Rezultatul furnizat de formula supusă evaluării depinde de valorile operanzilor și de operatorii care acționează asupra acestora. Atunci când aplicația de calcul tabelar evaluează o formulă, se așteaptă ca pentru fiecare operator să se întâlnească anumite tipuri de operanzi. De exemplu, în cazul operatorilor aritmetici \wedge , $+$, $-$, $*$, $/$ operanzii respectivi trebuie să fie de tip numeric, în cazul operatorului $\&$ ambii operanzi trebuie să fie de tip text, iar în cazul operatorilor relaționali $<$, $>$, $<>$, $<=$, $>=$ ambii operatori trebuie să aibă același tip, numeric, logic sau text. Tipul rezultatelor furnizate de operatori este indicat în tabelul 2.2.

Tabelul 2.2

Tipul rezultatelor furnizate de operatori

Operator	Tipul operanzilor	Tipul rezultatului
\wedge , $+$, $-$, $*$, $/$	numeric	numeric
$\&$	text	text
$=$, $<$, $>$, $<>$, $<=$, $>=$	tipuri identice	logic

Amintim că în reprezentarea internă, datele calendaristice și mărcile de timp sunt memorate ca numere, fapt ce permite prelucrarea lor cu ajutorul operatorilor aritmetici și al celor relaționali. De exemplu, diferența a două date calendaristice este egală cu numărul de zile ce le desparte, iar diferența a două mărci – cu intervalul respectiv de timp.

În cazul operatorilor relaționali, datele calendaristice și mărcile de timp sunt comparate ca numere, iar textele – ca șiruri de caractere ordonate lexicografic conform tabelului de cod ASCII.

Pentru exemplificare, în figura 2.8 sunt prezentate rezultatele evaluării unor formule ce conțin operatori aritmetici, operatori relaționali și operatorul pentru texte. Pentru a ilustra faptul că operatorii aritmetici acceptă date calendaristice, în formula $=D4-C4$ se scad, iar în formula $=C4>D4$ se compară două valori de acest tip. În mod similar, în formula $=D5-C5$ se scad, iar în formula $=C5+D5$ se adună două mărci de timp. În formula $=C6\&D6$ se concatenează, iar în formula $=C6>D6$ se compară două valori de tip text.

În cazurile în care tipurile de date necesare pentru efectuarea unei operații nu corespund celor așteptate (vezi tabelul 2.2), aplicația de calcul tabelar încearcă să convertească valorile respective, transformându-le în tipul dorit.

Prin conversia tipului de date al unei valori înțelegem transformarea ei în așa mod, încât ea să aparțină altui tip de date.

a)

	A	B	C	D	E	F	G
1							
2		Tipul operanzilor	Operanzi		Formule		
3		Număr	1,00	2,00	3,00	-1,00	
4		Data calendaristică	01.01.2003	10.01.2003	9	FALSE	
5		Marcă de timp	10:00	10:30	00:30	20:30	
6		Text	unu	doi	unudoi	TRUE	
7							
8							

b)

	A	B	C	D	E	F	G
1							
2		Tipul operanzilor	Operanzi		Formule		
3		Număr	1	2	=C3+D3	=C3-D3	
4		Data calendaristică	37622	37631	=D4-C4	=C4>D4	
5		Marcă de timp	0,416666666666667	0,4375	=D5-C5	=C5+D5	
6		Text	unu	doi	=C6&D6	=C6>D6	
7							
8							

Fig. 2.8. Operații cu numere, date calendaristice, texte și mărci de timp:
a) afișarea valorilor; b) afișarea formulelor

În procesul evaluării formulelor, aplicațiile de calcul tabelar efectuează, dacă-i necesar, următoarele conversiuni:

1. Conversiune în valori numerice. Textele, dacă ele pot fi interpretate ca numere, datele calendaristice și mărcile de timp sunt transformate în numere. Valorile logice FALSE și TRUE sunt transformate în numere, respectiv, 0 și 1. Dacă conversiunea nu este posibilă, aplicația de calcul tabelar generează valoarea de eroare #VALUE!.

2. Conversiunea în valori text. Atunci când așteaptă un text, aplicația de calcul tabelar transformă toate numerele și valorile logice în text.

3. Conversiunea în valori logice. Aplicația de calcul tabelar transformă numărul 0 în FALSE, iar toate celelalte numere în TRUE. În cazul în care se întâlnește un text ce diferă de "true" sau "false", se va genera valoarea de eroare #VALUE!.

Pentru exemplificare, în figura 2.9 sunt prezentate rezultatele evaluării unor formule, operanzii cărora, când acest lucru este posibil, sunt transformați în valori numerice.

Foaia de calcul din figura 2.9 conține un tabel în celulele căruia sunt înscrise formule ce efectuează adunarea operanzilor din antetele de rânduri și coloane. Antetele tabelului conțin câte un număr, o dată calendaristică, o marcă de timp, un text și o valoare logică. Aplicația de calcul tabelar afișează în fiecare celulă a tabelului rezultatul evaluării formulei respective. De exemplu, în cazul formulei =B3+C2 în celula C3 se afișează rezultatul adunării numerelor 1 și 2, iar în cazul formulei =B3+D2 în celula D3 se afișează rezultatul adunării numărului 1 cu data calendaristică 10.01.2003. În procesul evaluării, această dată este transformată în numărul 37631, apoi se calculează suma $1+37631=37632$, care ulterior este afișată în celula D3 ca data calendaristică 11.01.2003. În cazul evaluării formulei =B3+F2 din celula F3, aplicația de calcul tabelar încearcă să transforme textul "doi" într-un număr. Întrucât acest lucru este imposibil, în celula respectivă se afișează mesajul de eroare #VALUE!.

Pentru a facilita procesele de căutare și eliminare a erorilor, aplicațiile de calcul tabelar utilizează noțiunile *precedente* și *dependente* ale unei celule.

a)

	A	B	C	D	E	F	G	H
1								
2		Operatorul "+"	2,00	10.01.2003	10:30	doi	TRUE	
3		1,00	3,00	11.01.2003	10:30	#VALUE!	2	
4		01.01.2003	03.01.2003	12.01.2106	01.01.2003 10:30	#VALUE!	02.01.2003	
5		10:00	10:00	10.01.2003	20:30	#VALUE!	01.01.1900 10:00	
6		unu	#VALUE!	#VALUE!	#VALUE!	#VALUE!	#VALUE!	
7		FALSE	2	10.01.2003	10:30	#VALUE!	1	
8								

b)

	A	B	C	D	E	F	G	H
1								
2		Operatorul "+"	2	37631	0,4375	doi	TRUE	
3		1	=B3+C2	=B3+D2	=B3+E2	=B3+F2	=B3+G2	
4		37622	=B4+C2	=B4+D2	=B4+E2	=B4+F2	=B4+G2	
5		0,416666666666667	=B5+C2	=B5+D2	=B5+E2	=B5+F2	=B5+G2	
6		unu	=B6+C2	=B6+D2	=B6+E2	=B6+F2	=B6+G2	
7		FALSE	=B7+C2	=B7+D2	=B7+E2	=B7+F2	=B7+G2	
8								

Fig. 2.9. Conversia operanzilor în cazul operatorului "+":
a) afișarea valorilor; b) afișarea formulelor

Numim precedente (surse) ale unei celule toate celulele care sunt referite în formula pe care ea eventual o conține. Numim dependente (destinații) ale unei celule toate celulele în care ea este referită.

De exemplu, precedentele celulei B5 din foaia de calcul din figura 2.10 sunt celulele A2 și C2, iar precedentă celulei D8 este celula C5. Dependenta celulei A2 este celula B5, iar dependentele celulei C5 sunt celulele B8 și D8.

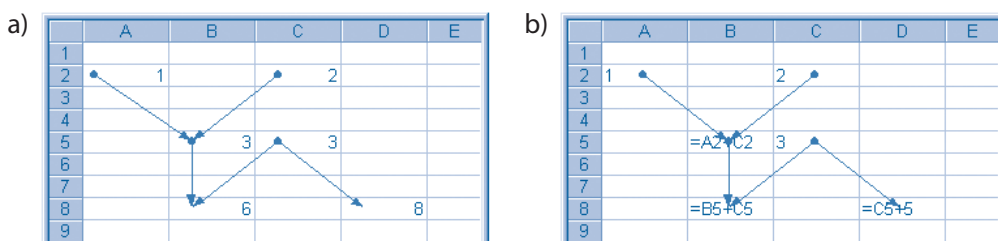


Fig. 2.10. Precedente și dependente de celule:
a) afișarea valorilor; b) afișarea formulelor

Vizualizarea precedentelor și dependentelor celulei selectate poate fi efectuată cu ajutorul comenzii **Tools, Auditing** (Instrumente, Revizie). Această comandă conține următoarele opțiuni:

Trace Precedents (Trasează precedentele) – afișează pe ecran una sau mai multe săgeți care simbolizează transferul de date din celulele-sursă către celula selectată.

Trace Dependents (Trasează dependentele) – afișează pe ecran una sau mai multe săgeți care simbolizează transferul de date din celula selectată către celulele-destinație.

Trace Error (Trasează eroarea) – afișează pe ecran una sau mai multe săgeți care simbolizează transferul de date din celulele-sursă către celula selectată ce conține valori de tipul *eroare* (#VALUE!, #DIV/0!, #NUM! ș.a.).

Remove All Arrows (Șterge toate săgețile) – șterge de pe ecran toate săgețile, afișate prin lansarea comenzilor precedente.

Pentru exemplificare, în *figura 2.10* este prezentată imaginea unei foi de calcul după lansarea comenzilor **Trace Precedents** și **Trace Dependents**.

Întrebări și exerciții

- ❶ Când demarează procesul de evaluare a formulelor?
- ❷ Numiți tipurile de date acceptate de operatorii aritmetici.
- ❸ Care sunt tipurile de date acceptate de operatorii relaționali?
- ❹ Pot fi oare comparate două valori de tip text?
- ❺ Introduceți în calculator foaia de calcul din *figura 2.8*. Înscrieți în celula D4 data calendaristică 12.09.51. Explicați rezultatele afișate pe ecran.
- ❻ Explicați termenul *conversiunea tipului de date*. Când apare necesitatea conversiunii unui tip de date?
- ❼ Cum se efectuează conversiunea datelor calendaristice și a mărcilor de timp în numere?
- ❽ Pot fi oare transformate în numere orice valori de tip text?
- ❾ Folosind ca model foaia de calcul din *figura 2.9*, elaborați tabele similare pentru operatorii aritmetici * și /, operatorii relaționali = și <>, operatorul pentru texte &. Introduceți tabelele respective în calculator și explicați rezultatele afișate pe ecran.
- ❿ În tabelul ce urmează sunt indicate tipurile de date ale operandilor și operatorii care acționează asupra acestora. Indicați tipurile de date ale rezultatelor furnizate de operatorii respectivi.

Tipul operandului 1	Operator	Tipul operandului 2	Tipul rezultatului
număr	+	text	
dată calendaristică	+	număr	
marcă de timp	*	număr	
text	&	logic	
text	>	număr	
număr	>=	text	
dată calendaristică	&	text	
dată calendaristică	&	marcă de timp	

- ⓫ Cum se definesc precedentele și dependentele unei celule?
- ⓬ Indicați în foaia de calcul din *figura 2.9* precedentele și dependentele fiecărei celule.
- ⓭ Introduceți în calculator foaia de calcul din *figura 2.10*. Utilizând comanda **Tools, Auditing** (Instrumente, Revizie), afișați pe ecran săgețile ce simbolizează transferul de date de la celulele sursă la celulele destinație.
- ⓮ Indicați în foaia de calcul **Alimente** (*fig. 2.6*) precedentele și dependentele fiecărei celule. Afișați pe ecran săgețile ce simbolizează transferul de date de la celulele sursă la celulele destinație.
- ⓯ Utilizând comanda **Trace Error** (Trasează eroarea), indicați în foaia de calcul din *figura 2.9* săgețile ce simbolizează transferul de date de la celulele sursă la una dintre celulele ce conține erori.

2.4. FUNCȚII

Termeni-cheie:

- funcție
- argument
- valoare returnată

Pentru a efectua calculele complexe într-un mod simplu și rapid, aplicațiile de calcul tabelar oferă utilizatorului posibilitatea să includă în formule **funcții** predefinite, destinația cărora este cunoscută în toate foile de calcul, și **funcții** definite de utilizator. Amintim că în matematică o funcție $f: A \rightarrow B$ se descrie prin trei elemente:

- 1) regula, procedul, legea de corespondență f prin intermediul căreia fiecărui element din A i se pune în corespondență un singur element din B ;
- 2) domeniul de definiție sau mulțimea valorilor variabilei independente x ;
- 3) subdomeniul sau domeniul valorilor (mulțimea în care funcția ia valori).

Prin notația matematică $y = f(x)$ se accentuează faptul că x este variabila independentă sau **argumentul funcției**, $f(x)$ este valoarea funcției în punctul x , iar y este variabila dependentă.

Funcțiile reprezintă formule predefinite, care primesc la intrare anumite valori, numite argumente, efectuează o prelucrare a acestora și returnează (întorc) valorile calculate.

De exemplu, în foaia de calcul **Alimente** (fig. 2.6b) se utilizează funcția SUM, care însumează valorile din celulele domeniului indicat între paranteze. Astfel, în formula =SUM(F4:F9) se adună valorile din celulele domeniului F4:F9, în formula =SUM(G4:G9) se adună valorile din celulele domeniului G4:G9, iar în formula =SUM(H4:H9) se adună valorile din celulele domeniului H4:H9. Evident, formula =SUM(F4:F9) este mult mai compactă decât formula =F4+F5+F6+F7+F8+F9, care descrie aceleași operații, însă fără utilizarea funcțiilor.

De obicei, orice funcție din aplicațiile de calcul tabelar este formată din două elemente:

- numele funcției, de exemplu SUM, care indică operațiile (procedeele) ce vor fi efectuate;
- unul sau mai multe argumente, separate prin punct și virgulă și incluse între paranteze rotunde (“(” și “)”).

De exemplu, în formula =SUM(F4:F9), notația SUM este numele funcției, iar notația F4:F9 reprezintă argumentul acesteia.

În calitate de argumente, funcțiile din aplicațiile de calcul tabelar acceptă valori (numere, date calendaristice, mărci de timp, texte, valori logice, valori de eroare), referințe de celule (adrese, domenii, nume de celule) sau alte funcții. Evident, dacă în scrierea unei formule funcția apare pe primul loc, ea va fi precedată de semnul =.

Funcțiile utilizate frecvent în aplicațiile de calcul tabelar sunt prezentate în *tabelul 2.3*.

Funcții utilizate frecvent în aplicațiile de calcul tabelar

Funcția	Exemple	Descriere
AVERAGE (Media)	AVERAGE(A1:A10)	Calculează media numerelor din celulele domeniului A1:A10.
	AVERAGE(1; 2; 3)	Calculează media numerelor 1; 2; 3.
COUNT (Numără)	COUNT(A1:A5; C1:C5)	Numără toate celulele din domeniul neadiacent A1:A5; C1:C5 care conțin valori numerice.
COUNTA (Numără A)	COUNTA(B3:D8)	Numără toate celulele din domeniul B3:D8 care nu sunt vide.
MAX (Maximum)	MAX(A1:B5)	Returnează valoarea maximă din celulele domeniului A1:B5.
	MAX(10; 20; A1:B5; D3:E8)	Returnează valoarea maximă din numerele 10; 20 și cele aflate în celulele domeniului neadiacent A1:B5; D3:E8.
MIN (Minimum)	MIN(C3:D8)	Returnează valoarea minimă din celulele domeniului C3:D8.
SUM (Suma)	SUM(A1:B10; C5:D10; F15)	Calculează suma numerelor din celulele domeniului neadiacent A1:B10; C5:D10; F15.
	SUM(10; 20,54; B5:D10)	Calculează suma numerelor 10; 20,54 și a celor aflate în celulele domeniului B5:D10.
PRODUCT (Produsul)	PRODUCT(B1:B5)	Calculează produsul numerelor aflate în celulele domeniului B1:B5.
POWER (Puterea)	POWER(A1; 2)	Calculează pătratul numărului din celula A1.
	POWER(A1; 3)	Calculează cubul numărului din celula A1.
IF (Dacă)	IF(A1=0; "zero"; "diferit de zero")	Dacă condiția A1=0 este adevărată (TRUE), returnează textul "zero". Dacă condiția A1=0 este falsă (FALSE), returnează textul "diferit de zero".
	IF(A1=0; SUM(B1:B5); B1*2)	Dacă condiția A1=0 este adevărată (TRUE), calculează SUM(B1:B5) și returnează suma respectivă. Dacă condiția A1=0 este falsă (FALSE), calculează B1*2 și returnează produsul respectiv.
DATE (Data)	DATE(2003, 9, 15)	Returnează numărul ce corespunde datei calendaristice 15.09.2003.
TIME (Timpul)	TIME(15; 20; 30)	Returnează numărul ce corespunde mărcii de timp 15:20:30.
CHAR (Caracterul)	CHAR(65)	Returnează caracterul codul căruia este 65.

Funcția	Exemple	Descriere
CODE (Codul)	CODE("A")	Returnează codul caracterului "A".
LEN (Lungimea)	LEN("Text")	Returnează numărul de caractere în cuvântul "Text".

O aplicație de calcul tabelar conține sute de funcții. Acestea sunt grupate pe categorii, de exemplu, funcții matematice, funcții pentru prelucrarea datelor calendaristice și a mărcilor de timp, funcții pentru prelucrarea textelor, funcții pentru calcule financiare ș.a. O listă completă și o descriere detaliată a fiecărei funcții poate fi afișată pe ecran sau tipărită la imprimantă cu ajutorul sistemului de asistență **Help**.

Pentru a simplifica introducerea funcțiilor, aplicațiile de calcul tabelar oferă utilizatorului facilitarea **Function Wizard** (Asistentul de funcții), care afișează informații detaliate despre funcțiile dorite și permite selectarea numelor de funcții și a argumentelor respective cu ajutorul șoricelului. Lansarea asistentului se efectuează prin acționarea butonului **=** (**Edit Formula**), amplasat lângă caseta pentru formule. În continuare, utilizatorul poate selecta numele funcției dintr-o listă derulantă, care este afișată în caseta pentru nume. După selectarea funcției dorite, aplicația de calcul tabelar afișează o fereastră de dialog în care utilizatorul poate introduce argumentele funcției (fig. 2.11).

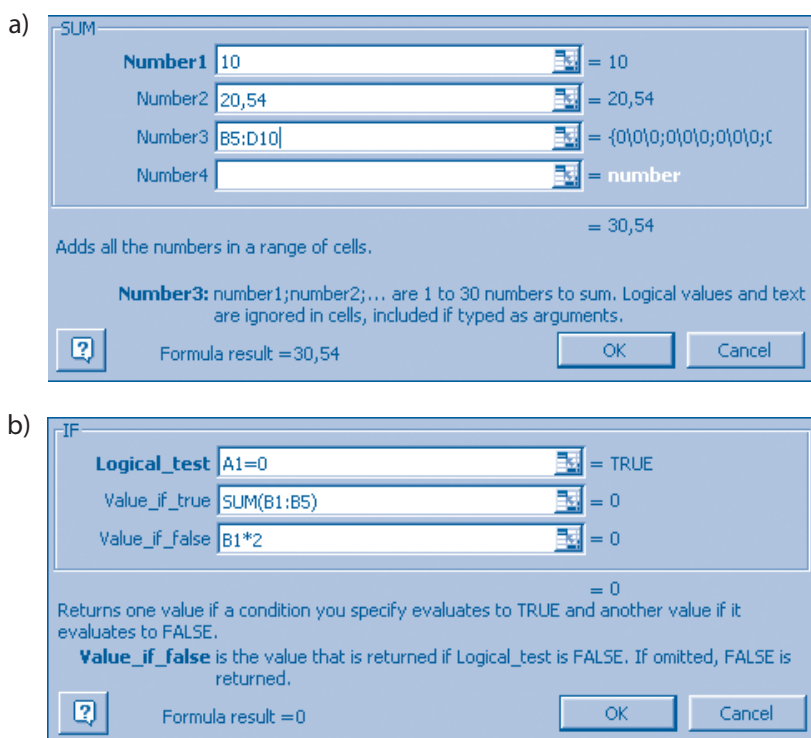


Fig. 2.11. Ferestre de dialog pentru introducerea funcțiilor:
a) funcția SUM; b) funcția IF

Dacă funcția dorită nu se regăsește în lista derulantă, se selectează opțiunea **More Functions** (Mai multe funcții), care afișează fereastra de dialog **Paste Function** (Încleie funcția). Această fereastră conține toate funcțiile aplicației de calcul tabelar, grupate pe categorii (fig. 2.12).

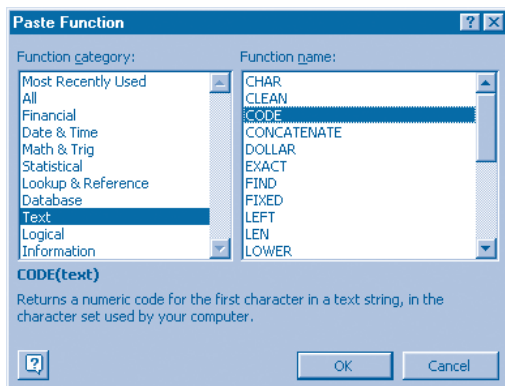



Fig. 2.12. Fereastra de dialog **Paste Function**

Același efect poate fi obținut lansând comanda **Insert, Function** (Inserare, Funcție) sau acționând butonul  (**Paste Function**).

Să reținem că adresele de formule care apar ca argumente de funcții pot fi relative sau absolute. Evident, în procesul copierii formulelor ce conțin funcții, adresele relative vor fi modificate în mod automat, fapt ce reduce substanțial volumul de lucru necesar pentru introducerea și editarea funcțiilor. Pentru exemplificare, în figura 2.13 este prezentată foaia de calcul **Note**, în care se calculează notele minime, notele medii și notele maxime pentru fiecare disciplină și pentru fiecare elev. Cu toate că această foaie

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1																
2		Nr. crt.	Nume, prenume	Limba română	Limba engleză	Matematica	Informatica	Fizica	Chimia	Biologia		Nota minimă	Nota medie	Nota maximă		
3	1	Bucur Elena	5	6	6	7	5	5	5	8		5	6,00	8		
4	2	Bunu Ion	7	7	5	5	5	6	9			5	6,29	9		
5	3	Ciobotaru Anton	8	8	9	10	10	6	6			6	8,14	10		
6	4	Ciobanu Cristina	9	9	6	7	5	5	7			5	6,86	9		
7	5	Dogaru Valentina	9	10	8	8	7	7	9			7	8,29	10		
8	6	Moraru Ionța	6	6	6	7	5	5	7			5	6,00	7		
9	7	Munteanu Alexandru	5	5	5	5	5	5	6			5	5,14	6		
10	8	Plugaru Raisa	6	5	5	5	5	7	7			5	5,71	7		
11	9	Rotaru Constantin	4	5	5	5	4	4	6			4	4,71	6		
12	10	Vulpe Victor	6	6	8	8	8	7	7			6	7,14	8		
13																
14			Nota minimă	4	5	5	5	4	4	6						
15			Nota medie	6,50	6,70	6,30	6,70	5,90	5,70	7,20						
16			Nota maximă	9	10	9	10	10	7	9						
17																
18																

Fig. 2.13. Foaia de calcul **Note**

de calcul conține 60 de formule, numai 6 dintre ele au fost introduse de la tastatură, restul fiind copiate. Astfel, formula =MIN(D3:J3) a fost introdusă în celula L3 și apoi copiată în toate celulele domeniului L4:L12. În mod similar, formula =AVERAGE(D3:J3) a fost introdusă în celula M3 și apoi copiată în toate celulele domeniului M4:M12, formula =MAX(D3:J3) a fost introdusă în celula N3 și apoi copiată în toate celulele domeniului N4:N12.

Pentru a înțelege mai bine modul de utilizare a funcției IF, vom reprezenta structura acesteia în forma

$$\text{IF}(\text{argument 1}; \text{argument 2}; \text{argument 3}).$$

În procesul evaluării, aplicația calculează argumentul 1, denumit *condiție*, și returnează valoarea unuia dintre celelalte două argumente. Dacă rezultatul evaluării condiției este TRUE, funcția IF returnează valoarea argumentului 2; dacă rezultatul evaluării condiției este FALSE, funcția IF returnează valoarea argumentului 3.

De exemplu, funcția IF(2=2; 3; 4) returnează valoarea 3, iar funcția IF(2<>2; 3; 4) returnează valoarea 4.

În foile de calcul, funcția IF se utilizează pentru selectarea datelor care satisfac anumite criterii. Pentru exemplificare, vom studia aplicarea acestei funcții la rezolvarea ecuațiilor de gradul I cu o singură necunoscută. Amintim că astfel de ecuații au forma

$$ax + b = 0.$$

Dacă a și b sunt numere reale și $a \neq 0$, atunci ecuația are o singură rădăcină $x = -\frac{b}{a}$; dacă $a = 0$ și $b \neq 0$, atunci ecuația nu are rădăcini; dacă $a = 0$ și $b = 0$, atunci ecuația are o mulțime infinită de rădăcini. Foia de calcul, destinată rezolvării ecuațiilor de gradul I, este prezentată în figura 2.14.

	A	B	C	D	E	F	G
1							
2	a	b		Ecuatia		Rădăcina	
3							
4	1	1		1x+1=0		-1	
5	2	3		2x+3=0		-1,5	
6	2	4		2x+4=0		-2	
7	0	1		0x+1=0		Nu are rădăcini	
8	0	0		0x+0=0		O mulțime infinită de rădăcini	
9	3	-6		3x-6=0		2	
10	-3	6		-3x+6=0		2	
11							
12							

Fig. 2.14. Foia de calcul **Ecuatii**

Rădăcinile ecuației de gradul I se calculează cu ajutorul formulei

$$=\text{IF}(A4=0; \text{IF}(B4=0; \text{"O mulțime infinită de rădăcini"}; \text{"Nu are rădăcini"}); -B4/A4),$$

înscrise în celula F4. Prima funcție IF analizează conținutul celulei A4, în care este memorat coeficientul a . Dacă în urma evaluării condiției $A4=0$ se obține valoarea logică TRUE, adică coeficientul $a = 0$, atunci se va trece la evaluarea funcției interioare IF; dacă însă $a \neq 0$, se returnează rădăcina ecuației $-\frac{b}{a}$. Funcția interioară IF analizează conținutul celulei B4, în care se păstrează coeficientul b . Dacă în urma evaluării condiției $B4=0$ se obține valoarea

logică TRUE, adică coeficientul $b = 0$, se returnează textul "O mulțime infinită de rădăcini", iar în caz contrar – textul "Nu are rădăcini".

Pentru a face foaia de calcul mai sugestivă, în celula D4 se afișează ecuația propriu-zisă, scrierea căreia este generată cu ajutorul formulei

$$=A4&"x"&IF(B4>=0;"+";"")&B4&"=0"$$

memorate în celula D4. Această formulă concatenează valoarea coeficientului a din celula A4, simbolul x , semnul $+$ în cazurile în care coeficientul b este pozitiv, valoarea coeficientului b din celula B4 și constanta $\text{text} = 0$. Funcția IF din această formulă returnează semnul $+$, dacă se respectă condiția $B4=0$ și un text vid în caz contrar.

Pentru a rezolva concomitent mai multe ecuații, formulele din celulele D4, F4 sunt copiate, respectiv, în celulele D5:D10 și F5:F10.

Întrebări și exerciții

- ❶ Explicați, cum se definesc funcțiile pe care le-ați studiat la orele de matematică?
 - ❷ Care este sensul termenului *funcție* în aplicațiile de calcul tabelar?
 - ❸ Care sunt părțile componente ale unei funcții de calcul tabelar?
 - ❹ Determinați numele și argumentele funcțiilor ce urmează:
 - a) MAX(1; 2; 3; A1)
 - b) SUM(6; B3; C8; D1:D10)
 - c) IF(B1=0; A1+5; A1-5)
 - d) MIN(A1:A10; C1:C10)
 - e) AVERAGE(B5:B15; C5:D25)
 - f) IF(A1=0; SUM(B1:G1); PRODUCT(B1:G1))
 - g) DATE(B5)
 - h) IF(C1<>0; LEN(A1); LEN(B1))
- Utilizând *tabelul 2.3*, explicați destinația acestor funcții.
- ❺ Aflați cu ajutorul sistemului de asistență **Help** destinația următoarelor funcții: CONCATENATE, LEFT, RIGHT, DAY, TODAY, YEAR, AND, OR, NOT, ABS, TYPE.
 - ❻ Care este destinația asistentului **Function Wizard**? Cum se introduc formulele în foaia de calcul cu ajutorul acestui asistent?
 - ❼ Introduceți în calculator foaia de calcul **Note** (*fig. 2.13*). Modificați această foaie conform datelor din registrul clasei în care învățați dvs. Explicați destinația fiecărei formule din foaia de calcul.
 - ❽ Afișați pe ecran precedentele și dependentele celulelor D14 și L3 ale foii de calcul **Note** (*fig. 2.13*). Explicați rezultatele afișate pe ecran.
 - ❾ Creați o foaie de calcul în care să fie afișate codurile literelor mari A, B, C, ..., Z ale alfabetului latin.
 - ❿ Creați o foaie de calcul în care să fie afișate caracterele ce corespund codurilor 97, 98, 99, ..., 122.
 - ⓫ În coloanele B și C ale foii de calcul **Piese** (*fig. 2.15*) se înscriu, respectiv, lungimile a și lățimile b a 20 de piese de formă dreptunghiulară. Introduceți în foaia de calcul **Piese** formulele necesare pentru calcularea perimetrului $2a + 2b$ și a ariei ab ale fiecărei piese.
 - ⓬ Completați foaia de calcul **Piese** (*fig. 2.15*) cu formulele necesare pentru determinarea piesei de arie maximă.
 - ⓭ Introduceți în calculator foaia de calcul **Ecuații** (*fig. 2.14*). Explicați destinația formulilor din celulele foii de calcul. Afișați pe ecran precedentele și dependentele formulilor din celulele D4 și F4.

	A	B	C	D	E	F
1						
2		Lungimea	Lățimea	Perimetrul	Aria	
3		a	b	$2a+2b$	ab	
4		2	3	10	6	
5		6	8	28	48	
6		
7						
8						
9						
10						
11						
12						
13						
14						

Fig. 2.15. Foaia de calcul **Piese**

- 14 Pentru a determina cât mai exact lungimea unei tije, s-au efectuat n măsurări repetate, $n \leq 20$. Rezultatele măsurărilor se notează prin $L_1, L_2, L_3, \dots, L_n$. Lungimea tije L se calculează ca media rezultatelor măsurărilor:

$$L = \frac{L_1 + L_2 + L_3 + \dots + L_n}{n}.$$

De asemenea, pentru fiecare măsurare se determină abaterea de la medie:

$$\Delta L_1 = L_1 - L; \Delta L_2 = L_2 - L; \Delta L_3 = L_3 - L; \dots; \Delta L_n = L_n - L.$$

Cunoașterea abaterilor de la medie este utilă în cazurile când se dorește eliminarea din calcul a rezultatelor unor măsurări care par anormale în raport cu celelalte.

Creați o foaie de calcul pentru a determina lungimea tije L , abaterea maximă ΔL_{\max} :

$$\Delta L_{\max} = \max(|\Delta L_1|, |\Delta L_2|, |\Delta L_3|, \dots, |\Delta L_n|)$$

și abaterea minimă ΔL_{\min} de la medie:

$$\Delta L_{\min} = \min(|\Delta L_1|, |\Delta L_2|, |\Delta L_3|, \dots, |\Delta L_n|).$$

- 15 Creați o foaie de calcul destinată rezolvării ecuațiilor de gradul II:

$$ax^2 + bx + c = 0, \quad a \neq 0.$$

Amintim că pentru a rezolva o astfel de ecuație, mai întâi se calculează discriminantul:

$$D = b^2 - 4ac.$$

În continuare, se analizează semnul discriminantului D . Dacă $D > 0$, atunci ecuația are două rădăcini diferite:

$$x_1 = \frac{-b + \sqrt{D}}{2a}; \quad x_2 = \frac{-b - \sqrt{D}}{2a}.$$

Dacă $D = 0$, atunci ecuația are două rădăcini egale:

$$x_1 = x_2 = \frac{-b}{2a}.$$

Dacă $D < 0$, atunci ecuația nu are rădăcini.

DIAGrame ȘI Obiecte Grafice

3.1. ELEMENTELE UNEI DIAGrame

Termeni-cheie:

- diagramă
- indicator de date
- serie de date
- axa categoriilor
- categorie de date
- axa valorilor

Datele numerice ale unei foi de calcul pot fi înțelese mai ușor dacă sunt reprezentate în formă grafică.

Diagrama reprezintă o imagine în care valorile datelor numerice sunt redată prin dimensiunile unor obiecte grafice.

Pentru exemplificare, în figura 3.1 este reprezentată o diagramă, construită în baza datelor din foaia de calcul **Note**.

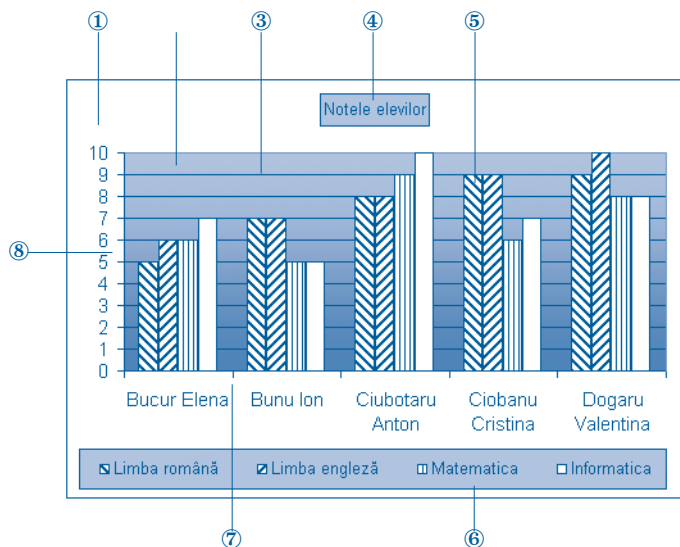


Fig. 3.1. Elementele unei diagrame

Ca și în cazul editoarelor de texte, în aplicațiile de calcul tabelar diagramele sunt tratate ca obiecte complexe care conțin următoarele elemente (fig. 3.1):

1. Zona pentru diagramă. În această zonă va fi desenată diagrama propriu-zisă. În cazul aplicațiilor de calcul tabelar, diagramele pot fi inserate direct în foaia de calcul în care se conțin datele respective sau în foi speciale, denumite *foi pentru diagrame*.

2. Zona de desenare. În această zonă sunt desenate obiectele grafice ce redau într-o formă sugestivă datele numerice din anumite celule ale foii de calcul.

3. Liniile de caroiaj. Aceste linii facilitează citirea și compararea valorilor numerice redade prin obiectele grafice din interiorul zonei de desenare.

4. Titlul diagramei. Titlul este un element opțional care redă într-o formă concisă conținutul diagramei.

5. Indicatori de date. Indicatorii redau prin dimensiunile lor valorile numerice din celulele foii de calcul și, în funcție de tipul diagramei, pot avea diverse forme: dreptunghiuri, linii, sectoare de cerc etc. Fiecare indicator de date reprezintă o valoare dintr-o anumită celulă a foii de calcul.

Pentru exemplificare, în figura 3.2 este prezentată foaia de calcul **Note** în care prin chenar îngroșat sunt indicate datele utilizate pentru crearea diagramei **Notele elevilor**.

L3		=MIN(D3:J3)														
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1																
2		Nr. crt.	Nume, prenume	Limba română	Limba engleză	Matematica	Informatica	Fizica	Chimia	Biologia		Nota minimă	Nota medie	Nota maximă		
3	1	Bucur Elena	5	6	6	7	5	5	8		5	6,00	8			
4	2	Bunu Ion	7	7	5	5	5	6	9		5	6,29	9			
5	3	Ciubotaru Anton	8	8	9	10	10	6	6		6	8,14	10			
6	4	Ciobanu Cristina	9	9	6	7	5	5	7		5	6,86	9			
7	5	Dogaru Valentina	9	10	8	8	7	7	9		7	8,29	10			
8	6	Moraru Ionița	6	6	6	7	5	5	7		5	6,00	7			
9	7	Munteanu Alexandru	5	5	5	5	5	5	6		5	5,14	6			
10	8	Plugaru Raisa	6	5	5	5	5	7	7		5	5,71	7			
11	9	Rotaru Constantin	4	5	5	5	4	4	6		4	4,71	6			
12	10	Vulpe Victor	6	6	8	8	8	7	7		6	7,14	8			
13																
14		Nota minimă	4	5	5	5	4	4	6							
15		Nota medie	6,50	6,70	6,30	6,70	5,90	5,70	7,20							
16		Nota maximă	9	10	9	10	10	7	9							
17																
18																

Fig. 3.2. Valorile numerice redade grafic în diagrama **Notele elevilor**

Confruntând diagrama din figura 3.1 și foaia de calcul din figura 3.2, observăm că indicatorii de date reprezintă notele la patru discipline școlare, indicatorii pentru fiecare disciplină fiind trasați în același mod.

Un grup de date aflate într-o coloană sau într-un rând ale foii de calcul este numit serie de date.

În figura 3.2 sunt evidențiate patru serii de date:

- notele școlare din domeniul D3:D7;
- notele școlare din domeniul E3:E7;

- notele școlare din domeniul F3:F7;
- notele școlare din domeniul G3:G7.

Aplicațiile de calcul tabelar atribuie fiecărei serii de date un nume, care este preluat din celula ce conține antetul valorilor respective. Astfel, seriei de date din coloana D i se atribuie denumirea Limba română, seriei de date din coloana E – denumirea Limba engleză, seriei de date din coloana F – denumirea Matematica, iar seriei de date din coloana G – denumirea Informatica.

Indicatorii care reprezintă valorile unei serii de date sunt trasați cu aceeași culoare, același model sau același simbol și formează o serie grafică de date.

De exemplu, în diagrama din *figura 3.1* apar patru modele de indicatori de date, care se disting prin faptul că sunt colorați în mod diferit: albastru pentru indicatorii seriei de date Limba română, roșu pentru indicatorii seriei de date Limba engleză, galben pentru indicatorii seriei de date Matematica și azuriu pentru indicatorii seriei de date Informatica. Să reținem că în cazul imprimării în alb-negru, culorile respective sunt substituite prin nuanțe de gri.

6. Legenda. Reprezintă o casetă care conține pentru fiecare serie de date câte o mostră de culoare, model sau simbol, însoțită de numele seriei. Vizualizând legenda, putem identifica mai ușor seriile de date.

7. Axa categoriilor. Pentru a construi o diagramă, aplicațiile de calcul tabelar grupează datele pe categorii.

Categoria reprezintă un grup de date corelate (legate) între ele. Fiecare categorie are un nume, care este preluat din antetul valorilor respective.

Diagrama din *figura 3.1* conține cinci **categorii de date**, denumirile cărora sunt preluate din foaia de calcul **Note**:

- Bucur Elena – domeniul D3:G3 (*fig. 3.2*);
- Bunu Ion – domeniul D4:G4;
- Ciubotaru Anton – domeniul D5:G5;
- Ciobanu Cristina – domeniul D6:G6;
- Dogaru Valentina – domeniul D7:G7.

De obicei, categoriile sunt trasate de-a lungul axei orizontale X , iar seriile – de-a lungul axei verticale Y , însă utilizatorul poate indica și o altă ordine.

8. Axa valorilor. De-a lungul acestei axe sunt trasate seriile de date.

Axele unei diagrame pot să conțină **reper** pentru valori, care sunt mici segmente ce intersectează axele și care marchează o categorie, scara sau seriile de date ale diagramei. Acestor repere li se pot atașa **etichete** care să le definească. De exemplu, axa valorilor din *figura 3.1* conține reperele marcate cu etichetele 0, 1, 2, ..., 10, iar axa categoriilor – reperele marcate cu denumirile de categorii Bucur Elena, Bunu Ion, ..., Dogaru Valentina. Pentru mai multă claritate, o diagramă poate conține și titluri de axe.

Pentru identificarea mai ușoară a elementelor unei diagrame, aplicațiile de calcul tabelar afișează, după selectarea elementului dorit, denumirea acestuia. În cazul indicatorilor de date, se afișează numele seriei, numele categoriei și valoarea corespunzătoare din foaia de calcul (*fig. 3.3*).

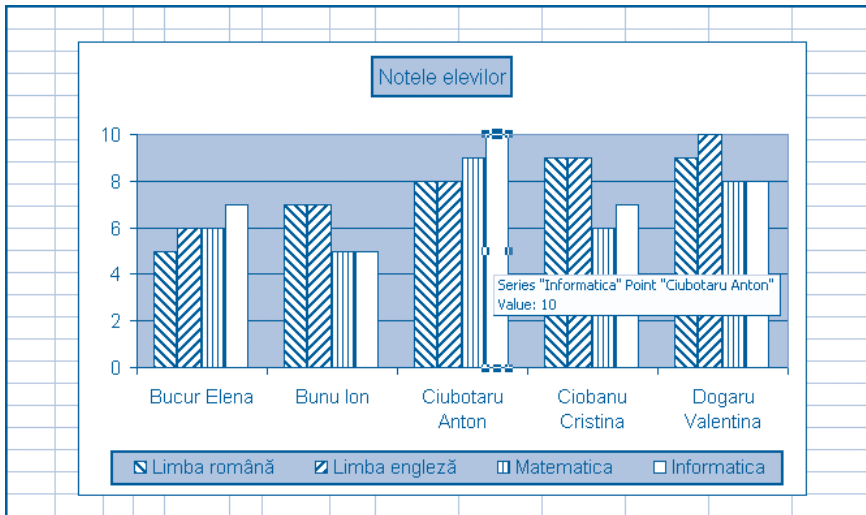


Fig. 3.3. Afișarea informațiilor despre indicatorii de date

Spre deosebire de procesoarele de texte, aplicațiile de calcul tabelar oferă utilizatorilor instrumente mult mai eficiente pentru crearea și editarea diagramelor. Amintim că în cazul editoarelor de texte, diagramele sunt create cu ajutorul unor aplicații speciale, care necesită introducerea datelor ce vor fi reprezentate în formă de diagramă în tabele speciale, denumite **foi de date**. De obicei, aceste foi nu sunt tratate ca părți componente ale textului, ele fiind invizibile pentru cititor. Dacă însă se dorește crearea unei diagrame în baza unui tabel deja inserat în text, informațiile respective trebuie copiate din text în foaia de date. În consecință, între tabelele unui text și diagramele respective nu există o legătură directă, iar modificarea tabelelor nu se răsfrânge asupra diagramelor.

În cazul aplicațiilor de calcul tabelar, orice modificare a datelor din foile de calcul implică actualizarea automată a diagramelor. De exemplu, dacă utilizatorul va schimba nota elevului Bunu Ion din 7 în 9, indicatorul respectiv de date își va schimba în mod automat lungimea, reflectând valorile actualizate ale datelor din foaia de calcul.

Întrebări și exerciții

- ❶ Indicați pe diagrama din figura 3.3. următoarele elemente:
 - a) zona pentru diagramă;
 - b) zona de desenare;
 - c) titlul diagramei;
 - d) legenda;
 - e) axa categoriilor;
 - f) reperele de pe axa categoriilor;
 - g) etichetele de reper de pe axa categoriilor;
 - h) axa valorilor;
 - i) reperele de pe axa valorilor;
 - j) etichetele de reper de pe axa valorilor;
 - k) liniile de caroiaj;

- l) seria grafică de date Limba română;
 m) seria grafică de date Informatica;
 n) categoria Ciubotaru Anton;
 o) indicatorul de date “seria Limba engleză, categoria Bunu Ion”;
 p) indicatorul de date “seria Informatica, categoria Dogaru Valentina”.
- ② Câți indicatori de date conține diagrama din *figura 3.1*? Câte categorii și câte serii de date sunt redată pe această diagramă?
- ③ Indicați pe foaia de calcul din *figura 3.2* valorile ce corespund următoarelor elemente ale diagramei **Notele elevilor** din *figura 3.1*:
- a) seria Matematica;
 b) categoria Ciubotaru Anton;
 c) seria Informatica;
 d) categoria Bunu Ion;
 e) indicatorul de date “seria Matematica, categoria Bunu Ion”;
 f) indicatorul de date “seria Informatica, categoria Bucur Elena”.
- ④ Explicați sensul termenilor *serie de date* și *categorie de date*.
- ⑤ Care este diferența dintre diagramele create cu ajutorul unui editor de texte și diagramele create cu ajutorul aplicațiilor de calcul tabelar?
- ⑥ Amintiți-vă cum se creează o diagramă cu ajutorul unui editor de texte și inserați într-un document text diagrama din *figura 3.1*. Măriți cu un punct toate notele din seria de date Matematica. Este redată oare această modificare în diagrama creată cu ajutorul editorului de texte?
- ⑦ Măriți cu un punct toate notele din seria de date Matematica (*fig. 3.2*). Observați cum se modifică indicatorii de date din diagrama **Notele elevilor** din *figura 3.1*. Prin ce se explică acest fapt?
- ⑧ Schimbați în foaia de calcul din *figura 3.2* numele Ciubotaru Anton prin Margine Vasile. În acest caz, se va modifica oare diagrama **Notele elevilor** din *figura 3.1*? Verificați experimental răspunsul dvs.
- ⑨ Selectați consecutiv toate elementele diagramei **Notele elevilor** din *figura 3.1*. Explicați mesajele afișate pe ecran.
- ⑩ Ca și în cazul editoarelor de texte, fiecare element al diagramelor create cu ajutorul aplicațiilor de calcul tabelar se caracterizează prin anumite proprietăți, de exemplu, dimensiuni, poziția în cadrul diagramei, fontul, stilul de afișare ș.a. Încercați să enumerați aceste proprietăți pentru fiecare dintre elementele diagramei din *figura 3.1*.

3.2. CREAREA DIAGRAMELOR

Termeni-cheie:

- tip de diagramă
- diagramă bidimensională
- diagramă tridimensională

În aplicația de calcul tabelar **Microsoft Excel** diagramele se creează lansând comanda **Insert, Chart** (Inserare, Diagramă) sau acționând butonul **Chart Wizard** (Asistentul pentru diagrame) din bara de instrumente standard.

În continuare, în scopuri didactice, vom folosi foaia de calcul **Note** din *figura 3.1* și vom construi diagrama **Notele elevilor**, aspectul final al căreia este prezentat în *figura 3.2*.

Imediat după lansare, asistentul pentru diagrame **Chart Wizard** afișează pe ecran prima pagină de dialog în care ne cere să alegem tipul diagramei pe care dorim să o construim (*fig. 3.4*).

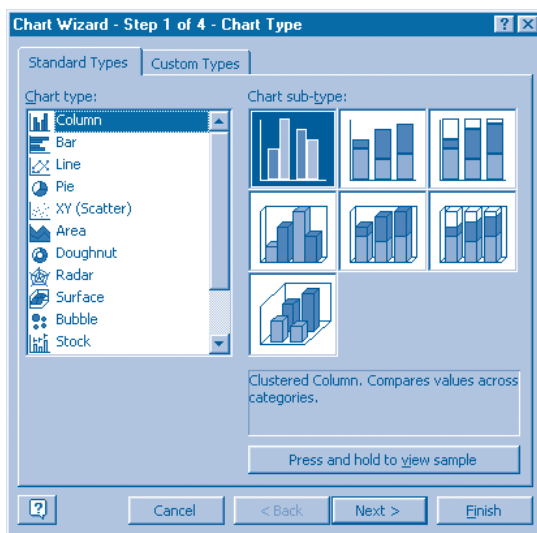


Fig. 3.4. Pagina de dialog **Chart Wizard – Step 1 of 4 – Chart Type**
(Pasul 1 din 4 – Tipul diagramei)

Caseta **Chart Type** a acestei ferestre oferă utilizatorului mai multe **tipuri de diagrame**. Se știe că tipul de diagramă se definește conform obiectului grafic folosit pentru reprezentarea valorilor numerice: coloane, bare, linii, sectoare de cerc etc. Cele mai utilizate tipuri de diagrame sunt:

Column – diagrame cu coloane;

Bar – diagrame cu bare;

Line – diagrame liniare;

Pie – diagrame circulare;

XY (Scatter) – diagrame XY.

Evident, fiecare tip de diagramă afișează datele în mod diferit. Aspectul grafic al fiecărui tip de diagramă poate fi previzualizat în partea dreaptă a ferestrei de dialog **Step 1 of 4**.

Alegerea tipului de diagramă se face conform recomandărilor ce urmează:

1. Diagramele cu coloane se utilizează pentru a reprezenta diferite serii de date care se schimbă în spațiu sau în timp. Pentru exemplificare amintim notele elevilor la anumite discipline școlare, numărul de absolvenți din fiecare promoție, veniturile lunare ale părinților, vânzările unei societăți comerciale etc.

2. Diagramele cu bare se aplică pentru a compara serii de date care nu se schimbă în timp. De exemplu, o astfel de diagramă poate fi folosită pentru a reprezenta capacitățile de prelucrare a diferitor calculatoare.

3. **Diagramele liniare** sunt cele mai potrivite pentru a reprezenta o tendință sau o relație dintre anumite valori pe o perioadă de timp. Drept exemplu amintim temperatura unui pacient din spital, cursul leului ș.a.

4. **Diagramele circulare** se folosesc pentru a evidenția raportul dintre părți și întreg. De exemplu, în cazul unei rețete culinare fiecare sector de cerc reprezintă cantitatea unui produs dintr-un anumit fel de mâncare.

5. **Diagramele XY** se folosesc pentru a reprezenta dependența dintre două serii de numere. Acest tip de diagrame este cel mai potrivit pentru a construi graficele funcțiilor studiate la orele de matematică, de exemplu, ale funcției liniare $y = ax + b$.

Caseta **Chart sub-type** (Subtipul diagramei) ne permite să alegem pentru anumite tipuri de diagrame reprezentări **bidimensionale** (în plan) sau **tridimensionale** (în spațiu). Evident, reprezentările tridimensionale au un impact vizual mai puternic.

După alegerea tipului și subtipului de diagramă dorit, se trece la pasul următor. După acționarea butonului **Next** (Fereastra următoare), pe ecran este afișată fereastra de dialog din *figura 3.5*.

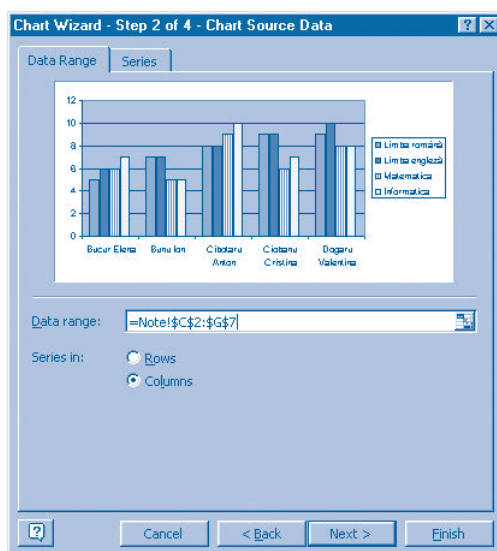


Fig. 3.5. Fereastra de dialog **Chart Wizard – Step 2 of 4 – Chart Source Data** (Pasul 2 din 4 – Sursa de date pentru diagramă)

În caseta de text **Data Range** (Domeniul de date) a acestei ferestre utilizatorul trebuie să introducă domeniul de celule al foii de calcul, care conține datele ce trebuie reprezentate pe diagramă. Domeniul poate fi definit prin scrierea în casetă a adreselor de celule sau direct prin glisarea cursorului peste celulele dorite. Evident, în cazul diagramei **Notele elevilor** din *figura 3.1*, vom glisa cursorul peste celulele C2:G7.

Atragem atenția că butoanele radio de pe pagina **Data Range** permit selectarea seriilor de date pe rânduri sau pe coloane, iar elementele de control de pe pagina **Series** permit editarea denumirilor de serii și modificarea domeniilor respective de valori.

După efectuarea alegerilor dorite și activarea butonului **Next**, pe ecran va fi afișată fereastra de dialog din *figura 3.6*.

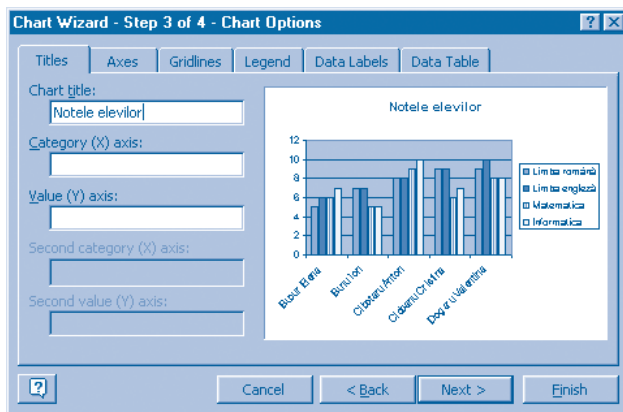


Fig. 3.6. Fereastra de dialog **Chart Wizard – Step 3 of 4 – Chart Options**
(Pasul 3 din 4 – Opțiuni pentru diagramă)

Această fereastră conține mai multe pagini care permit editarea elementelor diagramei în curs de construcție. Pentru început, vom indica pe pagina **Titles** (Titluri) denumirea diagramei, și anume: **Notele elevilor**. Menționăm că toate ferestrele de dialog ale asistentului pentru diagrame conțin butonul de comandă **Back** (Înapoi), la acționarea căruia se revine la fereastra precedentă, fereastră în care opțiunile alese anterior pot fi revăzute.

Ultima fereastră de dialog a asistentului pentru diagrame (fig. 3.7) ne permite să indicăm locul în care va fi inserată diagrama în curs de construcție – pe o foaie pentru diagrame (**As new sheet**) sau pe foia de calcul ce conține datele respective (**As object in**). În cazul foi de calcul **Notele elevilor**, inserăm diagrama în foaia de calcul **Note**.

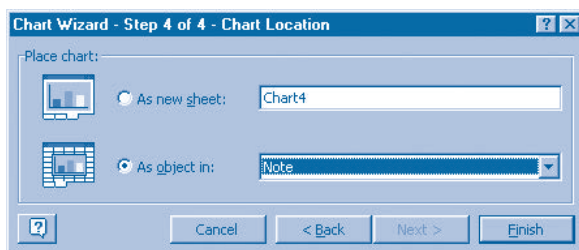


Fig. 3.7. Fereastra de dialog **Chart Wizard – Step 4 of 4 – Chart Location**
(Pasul 4 din 4 – Amplasarea diagramei)

Să reținem faptul că ferestrele de dialog ale asistentului pentru diagrame conțin mai multe pagini, care oferă utilizatorului o gamă extinsă de opțiuni referitoare la tipul și subtipul diagramei, selectarea domeniului de celule ce conțin valorile care vor fi redată în formă grafică, amplasarea elementelor diagramei, modul de trasare a fiecărui element grafic etc. Evident, memorizarea acestor opțiuni nu este necesară, întrucât informațiile respective se conțin în sistemul de asistență **Help** al aplicației de calcul tabelar. De la utilizator se cere doar cunoașterea etapelor principale ce trebuie parcurse pentru a crea o diagramă și utilizarea judicioasă a tuturor facilităților oferite de aplicațiile de calcul tabelar. Utilizând aceste facilități, utilizatorul trebuie să creeze diagrame simple, bine gândite, care ajută cititorul să înțeleagă semnificația datelor din foile de calcul. Se consideră că o diagramă

este bine proiectată numai atunci când cititorul nu trebuie să apeleze la valorile numerice din foaia de calcul pentru a descifra datele redată în formă grafică.

Întrebări și exerciții

- ❶ Numiți etapele principale ce trebuie parcurse pentru a crea o diagramă. Poate fi oare schimbată ordinea de parcurgere a acestor etape?
- ❷ Utilizând sistemul de asistență **Help**, determinați destinația fiecărui element de control de pe paginile ferestrelor de dialog ale asistentului pentru diagrame.
- ❸ Creați cu ajutorul asistentului pentru diagrame diagrama **Notele elevilor** (fig. 3.1). Explicați destinația fiecărei ferestre de dialog afișate pe ecran de această aplicație.
- ❹ În figura 3.8 este prezentată diagrama **Numărul absolvenților Gimnaziului "Constantin Stere"**. Ați construit această diagramă în procesul studierii editoarelor de texte. Creați acum diagrama în studiu cu ajutorul aplicației de calcul tabelar. Comparați facilitățile destinate creării diagramelor din cadrul editorului de texte și cele din cadrul aplicației de calcul tabelar.

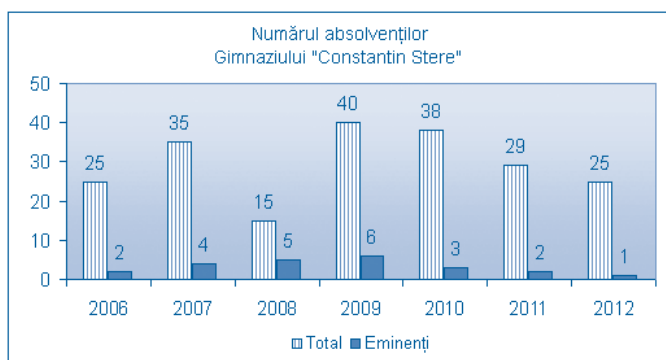


Fig. 3.8. Diagrama **Numărul absolvenților Gimnaziului "Constantin Stere"**

- ❺ Construiți diagrama liniară **Cursul dolarului american în raport cu leul moldovenesc** (fig. 3.9).

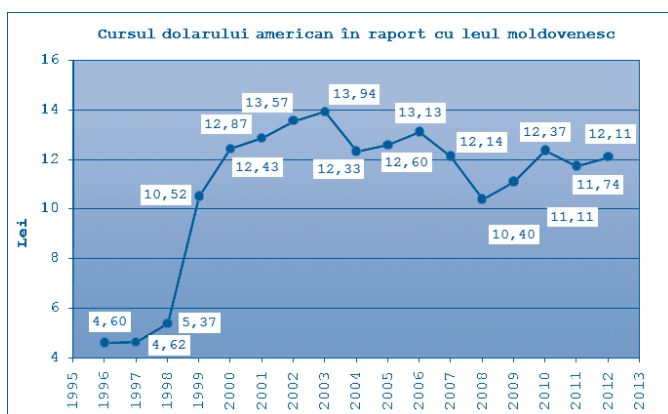


Fig. 3.9. Diagrama **Cursul dolarului american în raport cu leul moldovenesc**

- ⑥ În tabelul ce urmează sunt prezentate datele referitoare la venitul lunar al unei familii. Construieți în baza acestor date o diagramă cu coloane, o diagramă cu bare și o diagramă circulară. Care tip de diagramă este cel mai potrivit pentru redarea datelor referitoare la veniturile unei familii?

	Ianuarie	Februarie	Martie	Aprilie	Mai	Iunie
Mama	2 700	500	900	200	1 000	1 400
Tata	1 000	1 500	1 100	1 800	300	1 200

- ⑦ Care este deosebirea dintre diagramele bi- și tridimensionale? În ce situații este indicată folosirea diagramei tridimensionale?
- ⑧ Parcul auto al unei întreprinderi este compus din 60 de autoturisme, 20 de camioane și 15 autobuze. Construieți în baza acestor date două diagrame circulare – una bi-, iar alta tridimensională. Care subtip de diagramă este cel mai potrivit pentru redarea datelor referitoare la componența unui parc auto?
- ⑨ Construieți diagrame bi- și tridimensionale ce ar reprezenta grafic datele referitoare la venitul lunar al unei familii (vezi exercițiul 6). Selectați diagrama care, în opinia dvs., ar fi cea mai potrivită pentru reprezentarea acestor date. Argumentați alegerea dvs.
- ⑩ Dați o caracteristică succintă a fiecărui tip de diagramă. Fiecare caracteristică trebuie să includă:
- descrierea elementelor grafice folosite pentru reprezentarea valorilor numerice;
 - destinația și amplasarea axei valorilor și a axei categoriilor;
 - exemple de date pentru care aplicarea acestui tip de diagramă ar fi cea mai potrivită.

3.3. EDITAREA DIAGRAMELOR

Termeni-cheie:

- încapsularea obiectelor
- ierarhia obiectelor
- obiect al diagramei
- tehnici de editare

Aplicațiile de calcul tabelar tratează fiecare diagramă ca pe un obiect complex, care este format din alte obiecte, și anume: titlul diagramei, zona de desenare, seriile grafice de date, axa valorilor, axa categoriilor, legenda, liniile de caroiaj ș.a. Fiind părți componente ale diagramei, unele dintre aceste obiecte conțin, la rândul lor, alte obiecte, relativ mai simple. De exemplu, o serie grafică este formată din indicatori și etichete de date, iar axa valorilor, în afară de axa propriu-zisă, include titlul, reperele și etichetele care le definesc. Cu alte cuvinte, obiectele din componența unei diagrame pot fi incluse unul în altul.

Numim încapsulare metoda de creare a obiectelor complexe prin includerea în componența lor a altor obiecte, considerate mai simple. Obiectele incluse pot conține, la rândul lor, alte obiecte, considerate și mai simple.

Să reținem că **încapsularea obiectelor** este una dintre metodele de bază, frecvent utilizată în informatică. De exemplu, în aplicațiile de calcul tabelar agenda de lucru este un obiect complex care include obiecte mai simple, și anume, foile de calcul. La rândul său, orice foaie de calcul reprezintă un obiect compus, ce include alte obiecte, relativ mai simple: rânduri, coloane, celule, diagrame ș.a. În mod similar, orice document creat cu ajutorul editoarelor moderne de texte reprezintă, de asemenea, un obiect complex, format din obiecte mai simple: texte, tabele, diagrame, ecuații, imagini, secvențe sonore și secvențe video.

| **Numim ierarhie a obiectelor ordinea în care acestea sunt incluse unul în altul.**

Ierarhia obiectelor din componența diagramelor create cu ajutorul aplicațiilor de calcul tabelar este prezentată în *figura 3.10*.

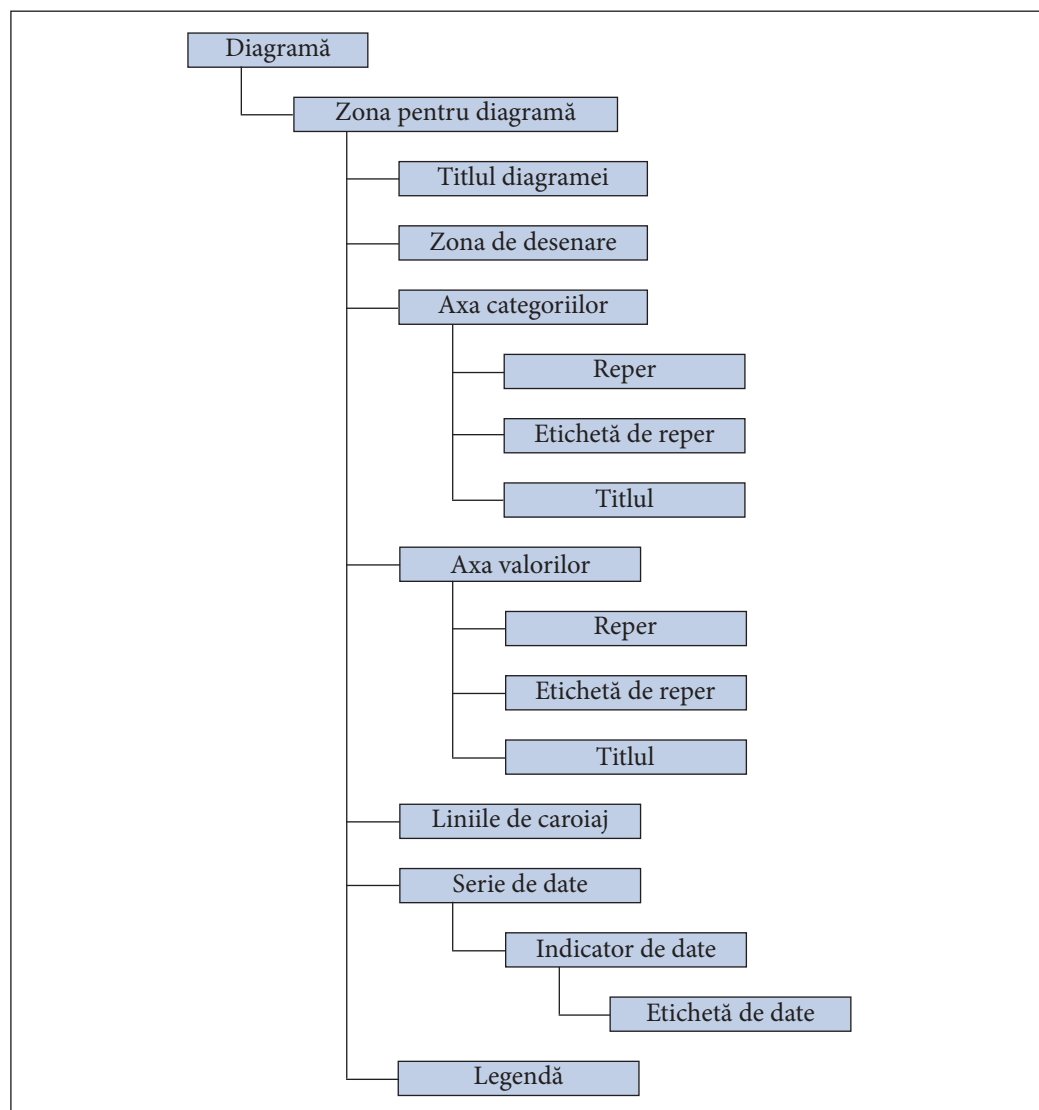


Fig. 3.10. Ierarhia obiectelor unei diagrame

Din *figura 3.10* se observă că pe nivelul unu al ierarhiei se află diagrama propriu-zisă, iar pe nivelul doi – zona pentru diagramă. Nivelul al treilea este format din obiectele incluse în zona pentru diagramă, și anume: titlul diagramei, zona de desenare, axa valorilor, axa categoriilor etc. Nivelul al patrulea este format de obiectele incluse în cele de nivelul trei: reperele, etichetele de reper, titlurile de axe, indicatorii de date. Ultimul nivel, al cincilea, conține etichetele de date, care sunt părți componente ale indicatorilor de date.

Cunoașterea ierarhiei obiectelor simplifică procesele de creare și de modificare a diagramei, permițând utilizatorului să identifice și să selecteze mai ușor obiectele supuse prelucrării.

Ca și în cazul altor programe, orice obiect, prelucrat cu ajutorul aplicațiilor de calcul tabelar, se caracterizează prin anumite proprietăți. Evident, nu sunt o excepție în acest sens nici **obiectele** din componența **diagramelor**, care se caracterizează prin dimensiuni, poziția ocupată în cadrul unei agende de lucru sau în cadrul unei foi de calcul, prezența sau absența chenarului, modul de desenare ș.a.m.d. De altfel, setul de informații care descriu proprietățile unui obiect se numește **format**, iar procesul de stabilire și de modificare a acestor proprietăți se numește **formatare**.

De obicei, după crearea unei diagrame utilizatorul dorește să insereze sau să elimine unele elemente ale acesteia și/sau să modifice proprietățile anumitor obiecte. De exemplu, titlul unei diagrame poate fi amplasat deasupra sau sub zona de desenare, iar textul respectiv poate fi scris cu ajutorul anumitor fonturi – **Arial**, **Times New Roman**, **Courier** ș.a. În mod similar, axa valorilor poate să aibă sau să nu aibă repere și etichete.

Pregătirea grafică a unei diagrame înainte de a începe operațiile de tipărire sau de inserare definitivă într-un document electronic se numește editare a diagramelor.

În procesul editării, asupra obiectelor din componența unei diagrame pot fi efectuate următoarele operații:

- inserare;
- redimensionare;
- mutare;
- eliminare;
- formatare.

Comenzile care asigură efectuarea acestor operații sunt grupate în meniurile **Edit** (Editare), **Format** (Formatare), **Chart** (Diagramă) și în meniurile contextuale. De exemplu, meniul **Edit** conține comanda **Clear** (Șterge), care, în funcție de obiectul selectat, oferă utilizatorului opțiunile **All** (Totul), **Series** (Seria) și **Formats** (Formatele). În mod similar, meniul **Format** conține comenzile **Selected Plot Area** (Zona selectată de desenare), **Selected Axis** (Axa selectată), **Selected Axis Title** (Titlul axei selectate), **Selected Data Series** (Seria selectată de date) ș.a.

Aplicațiile moderne de calcul tabelar oferă utilizatorului meniuri contextuale foarte detaliate, setul de opțiuni al cărora variază în funcție de obiectul aflat sub cursor. Amintim că lansarea comenzilor din astfel de meniuri se realizează prin efectuarea unui clic-dreapta pe obiectul supus editării și selectarea comenzii dorite.

În general, **tehnicele de editare** utilizate pentru prelucrarea diagramelor presupun parcurgerea următoarelor etape:

- selectarea obiectului supus editării;
- activarea comenzii dorite, utilizând în acest scop bara de meniuri, meniurile contextuale sau butoanele de pe bara de instrumente standard;
- introducerea, în caz de necesitate, a informațiilor suplimentare, cerute de aplicația de calcul tabelar.

Pentru exemplificare, în *figura 3.11* este prezentat procesul de formatare a zonei de desenare a diagramei **Notele elevilor**.

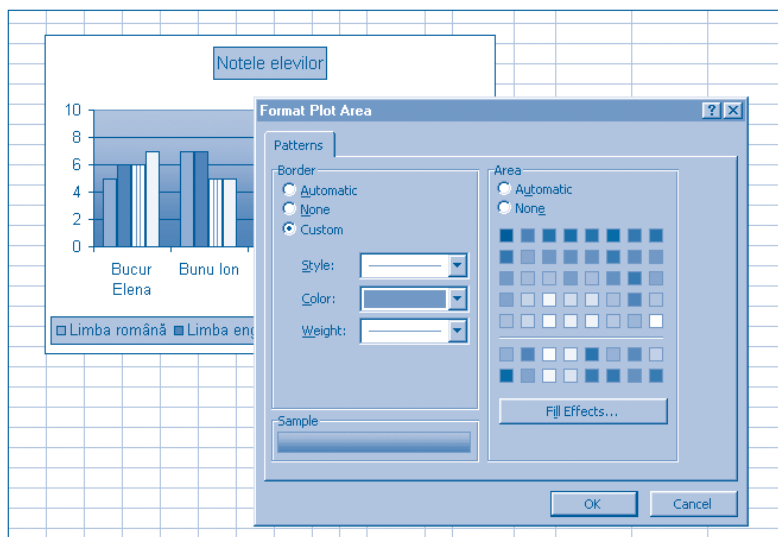


Fig. 3.11. Formatarea zonei de desenare

Pentru a reda zonei de desenare aspectul grafic dorit, utilizatorul va introduce în fereastra de dialog **Format Plot Area** (Formatarea zonei de desenare) informații referitoare la absența sau prezența chenarului, stilul liniei de chenar, prezența sau absența culorii de fundal, efectele de umplere ș.a.

Simplificarea proceselor de creare și de editare a diagramei se asigură prin respectarea următoarelor recomandări:

1. Înainte de a crea o diagramă, stabiliți domeniile de celule ale foii de calcul din care vor fi preluate datele respective.

2. Alegeți tipul de diagramă în funcție de informațiile pe care doriți să le puneți în evidență: cu coloane (variația datelor în spațiu sau în timp), cu bare (serii de date care nu se schimbă în timp), liniară (relațiile dintre părți și întreg), XY (dependența dintre două serii de numere).

3. Alegeți subtipul diagramei: bidimensională (în plan) sau tridimensională (în spațiu).

4. Includeți în diagramă numai acele elemente, care sunt absolut necesare pentru înțelegerea rapidă și corectă a datelor.

5. Exclueți din diagrame elementele care dublează informațiile deja existente sau care au numai funcții decorative.

6. Formatați fiecare obiect al diagramei, asortând chenarele, culorile de fundal, modelele de umplere, culorile de desenare, dimensiunile și fonturile utilizate.

Atragem atenția că folosirea mai multor culori este justificată numai în cazurile în care diagrama va fi tipărită la o imprimantă color sau inserată într-un document electronic. În cazul documentelor în alb-negru, se recomandă folosirea diferitor efecte de umplere a zonei de desenare și a indicatorilor de date, fapt ce simplifică discriminarea (deosebirea) lor.

Întrebări și exerciții

- ❶ Explicați termenii *încapsularea obiectelor* și *ierarhie a obiectelor*.
- ❷ Utilizând ca model *figura 3.10*, indicați pe desene separate ierarhia obiectelor diagramelor din *figurile 3.8* și *3.9*.
- ❸ Indicați pe un desen ierarhia obiectelor din componența documentelor create cu ajutorul editoarelor moderne de texte.
- ❹ Utilizând sistemul de asistență și meniurile contextuale, determinați proprietățile fiecărui obiect din componența diagramelor **Numărul absolvenților gimnaziului “Constantin Stere”** (*fig. 3.8*) și **Cursul dolarului american în raport cu leul moldovenesc** (*fig. 3.9*).

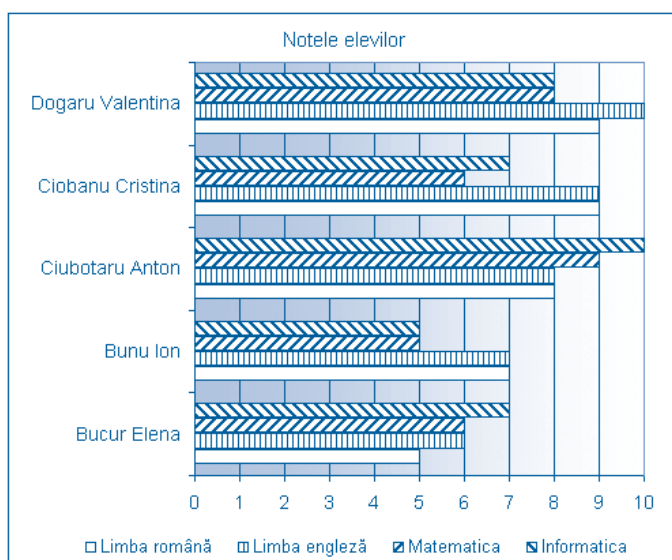


Fig. 3.12. Diagrama **Notele elevilor**

- ❺ Editați diagrama **Notele elevilor** conform modelului prezentat în *figura 3.12*.
- ❻ Activarea comenzii **Chart Option** a meniului **Chart** afișează pe ecran o fereastră de dialog care conține mai multe pagini. Utilizând sistemul de asistență, determinați destinația elementelor de control de pe fiecare pagină a acestei ferestre.
- ❼ Cum ar putea fi asortate culorile care sunt utilizate pentru redarea elementelor unei diagrame? Argumentați răspunsul dvs.
- ❽ În care cazuri nu este necesară includerea în diagramă a următoarelor elemente:
 - titlul diagramei;
 - titlurile de axe;

- liniile de caroiaj;
- legenda;
- reperele;
- etichetele de reper.

Argumentați răspunsul dvs.

- 9 Transformați diagrama **Cursul dolarului american în raport cu leul moldovenesc** (fig. 3.9) în una tridimensională. În opinia dvs., care diagramă este mai sugestivă: cea bidimensională sau cea tridimensională?
- 10 Analizați atent diagramele incluse în manualele de geografie și de istorie. Determinați tipul fiecărei diagrame și numiți elementele din componența lor. Determinați formatarea fiecărui obiect din componența acestor diagrame. Folosind o aplicație de calcul tabelar, încercați să creați și să formatați exact la fel una din astfel de diagrame.

3.4. HĂRȚI ȘI OBIECTE GRAFICE

Termeni-cheie:

- hartă
- obiecte grafice
- instrumente de desenare

Pentru o reprezentare mai sugestivă a datelor numerice ce caracterizează anumite regiuni geografice, aplicațiile de calcul tabelar oferă utilizatorului posibilitatea să insereze în foile de calcul diagrame speciale, denumite hărți. De obicei, pe o astfel de diagramă valorile numerice sunt redade prin atribuirea fiecărei zone geografice a unei culori distincte sau prin plasarea în zonele respective a unor indicatori de date.

Harta reprezintă o diagramă specială în care indicatorii de date sunt asociați anumitor zone geografice: continente, țări, regiuni, județe, raioane etc.

Pentru exemplificare, în *figura 3.13* este prezentată o hartă ce conține informații despre populația Republicii Moldova. Culorile fiecărei zone geografice sunt selectate de aplicația de calcul tabelar în funcție de numărul de locuitori, oferindu-i astfel cititorului posibilitatea să înțeleagă mai ușor specificul fiecărei unități administrativ-teritoriale.

Impactul vizual al hărților este mult mai puternic decât cel al diagramelor obișnuite, întrucât cititorul nu mai este nevoit să stabilească o legătură între zonele geografice și datele numerice ce le corespund.

Ca și în cazul diagramelor obișnuite, aplicațiile de calcul tabelar desenează hărțile în baza informațiilor din celulele foilor de calcul. Aceste informații trebuie să fie organizate în forma unui tabel cu cel puțin două coloane. Prima coloană a tabelului va conține în mod obligatoriu denumirile zonelor geografice ce trebuie să apară pe hartă. Pentru exemplificare, în *figura 3.14* este prezentată foaia de calcul ce conține datele în baza cărora a fost creată harta **Populația Republicii Moldova**.

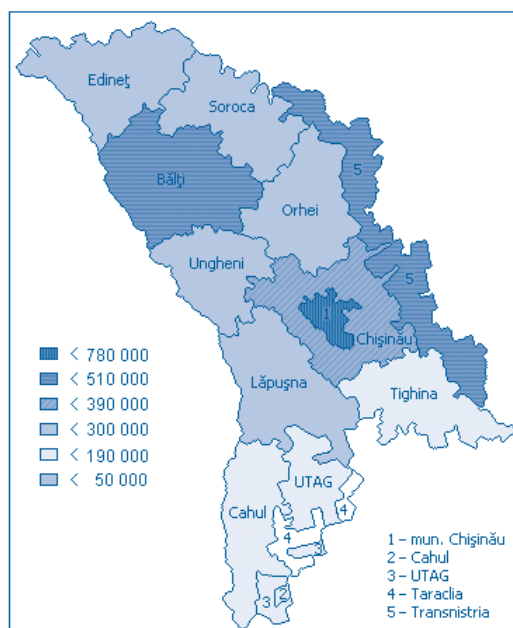


Fig. 3.13. Harta **Populația Republicii Moldova**

D9		= 299,5			
	A	B	C	D	E
1					
		Nr. crt.	Unități administra-tiv-teritoriale	Populație (mii persoane)	
2					
3		1	Mun. Chișinău	779,0	
4		2	Bălți	502,7	
5		3	Cahul	190,9	
6		4	Chișinău	382,6	
7		5	Edineț	281,7	
8		6	Lăpușna	277,1	
9		7	Orhei	299,5	
10		8	Soroca	276,2	
11		9	Taraclia	45,9	
12		10	Tighina	169,6	
13		11	Ungheni	260,8	
14		12	UTA Găgăuzia	159,2	
15					

Fig. 3.14. Foaia de calcul **Populația Republicii Moldova**

Evident, orice schimbare a datelor din foaia de calcul implică actualizarea automată a hărților create în baza acestora. De exemplu, dacă utilizatorul va schimba valoarea 299,5 din celula D9 cu valoarea 301,5, culoarea zonei geografice Orhei (fig. 3.13) va fi schimbată în mod automat.

În cazul aplicației de calcul tabelar **Microsoft Excel**, inserarea hărților se efectuează cu ajutorul unui program special, denumit **Microsoft Map** (Harta Microsoft). Meniurile acestui program conțin comenzi destinate selectării indicatorilor de date, formatării elementelor din componența hărților, inserarea și eliminarea etichetelor și a textelor

explicative, selectarea culorilor ș.a. Studiarea detaliată a fiecărei comenzi poate fi făcută cu ajutorul sistemului de asistență **Help**, care conține un capitol special, dedicat creării și editării hărților.

Să reținem că programele destinate creării și editării hărților necesită calculatoare puternice, acestea fiind performante și foarte scumpe. În consecință, aceste programe pot fi instalate numai pe unele calculatoare din cabinetele școlare de informatică.

Ca și în cazul editoarelor de texte, în foile de calcul pot fi inserate diverse obiecte: imagini, secvențe sonore, secvențe video. În funcție de modul de creare a obiectelor care pot fi incluse într-o foaie de calcul, deosebim:

- 1) obiecte create în cadrul aplicației **Microsoft Excel**;
- 2) obiecte create cu ajutorul altor aplicații.

De exemplu, din prima grupă fac parte diagramele și hărțile, iar din grupa a doua – desenele create cu ajutorul aplicației **Paint**, secvențele sonore înregistrate cu ajutorul aplicației **Sound Recorder** ș.a. Obiectele respective se inserează prin una dintre metodele ce urmează:

- a) prin intermediul memoriei-tampon;
- b) prin apelul aplicației în care se va crea obiectul dorit;
- c) prin citirea obiectului dorit din fișierul indicat de utilizator.

Modul de inserare a obiectelor este similar celui din editoarele de texte: se activează meniul **Insert** (Inserare) și se selectează comanda **Object** (Obiect). În continuare, utilizatorul poate indica fișierul din care va fi preluat obiectul dorit sau lansa aplicația în care el va fi creat.

Din practică s-a stabilit că, în afară de diagrame, în foile de calcul se utilizează și alte reprezentări grafice, în special desenele și imaginile.
















Numim obiecte grafice desenele și imaginile create în cadrul aplicațiilor de calcul tabelar sau preluate din alte aplicații.




Amintim că obiectele grafice create și prelucrate cu ajutorul calculatoarelor pot fi formate din microzone (grafica orientată pe puncte) sau din obiecte mai simple (grafica orientată pe obiecte). De obicei, imaginile create în grafica orientată pe puncte sunt preluate din alte aplicații, de exemplu, din aplicația **Paint**. În cazul graficii orientate pe obiecte, aplicațiile de calcul tabelar conțin programe speciale care permit inserarea următoarelor obiecte:

- linii;
- pătrate și dreptunghiuri;
- cercuri și elipse;
- arce de cerc sau de elipse;
- forme neregulate;
- casete explicative;
- figuri predesenate etc.

În aplicația de calcul tabelar **Microsoft Excel**, inserarea acestor obiecte se efectuează cu ajutorul elementelor de control ale barei de instrumente de desenare **Drawing** (Desenare), care este identică celei a editorului de texte **Microsoft Word**. Destinația butoanelor barei de instrumente de desenare este prezentată în *tabelul 3.1*.

Instrumentele de desenare

Butonul	Denumirea	Destinația
	Draw (Desenează)	Meniul conține un șir de comenzi ce asigură gruparea obiectelor, modificarea ordinii de suprapunere, selectarea formelor predesenate, alinierea obiectelor etc.
	Select Objects (Selectare obiecte)	Cursor pentru selectarea obiectelor grafice. La acționarea acestui buton, cursorul aplicației de calcul tabelar se transformă într-o săgeată dublă, invitând utilizatorul să selecteze unul sau mai multe obiecte grafice.
	Free Rotate (Rotire liberă)	Rotirea obiectului selectat. După acționarea acestui buton, marcajul obiectului selectat se transformă în cerceulețe mici care pot fi glisate la unghiul dorit.
	AutoShapes (Forme predesenate)	Desenarea formelor predefinite: linii, săgeți, conectori, figuri geometrice, scheme logice etc.
	Line (Linie)	Desenarea liniilor drepte.
	Arrow (Săgeată)	Desenarea săgeților.
	Rectangle (Dreptunghi)	Desenarea dreptunghiurilor.
	Oval (Elipsă)	Desenarea elipselor.
	Text Box (Casetă de text)	Inserarea casetelor de text. Textul inserat poate fi formatat independent de celelalte obiecte ale foii de calcul.
	Insert WordArt (Inserare text artistic)	Lansează aplicația WordArt , care permite inserarea unor fragmente de text, prezentate într-o formă artistică.
	Fill Color (Umple cu culoare)	Umplerea unei zone închise cu culoarea selectată. Suplimentar, utilizatorul poate selecta unul dintre modelele de umplere propuse.
	Line Color (Culoarea liniei)	Stabilește culoarea liniilor de contur ale obiectului selectat.
	Font Color (Culoarea fontului)	Stabilește culoarea caracterelor din textul selectat.
	Line Style (Stilul de linie)	Stabilește grosimea și forma liniilor de contur ale obiectului selectat.
	Dash Style (Stilul de întrerupere)	Stabilește modul de trasare a liniilor de contur ale obiectelor selectate: continue, întrerupte, punctate etc.

Butonul	Denumirea	Destinația
	Arrow Style (Stilul de săgeată)	Stabilește forma extremităților segmentelor de dreaptă selectate.
	Shadow (Umbrire)	Adaugă umbre obiectelor selectate.
	3-D (Imagini tridimensionale)	Transformă imaginile bidimensionale ale obiectelor selectate în imagini tridimensionale.

După inserarea obiectelor grafice dorite, utilizatorul poate schimba proprietățile lor. Aceste proprietăți sunt individuale pentru fiecare obiect grafic și sunt afișate pe ecran cu ajutorul meniurilor contextuale. Pentru aceasta, cursorul se poziționează pe obiectul supus formatării și se execută un clic-dreapta. În funcție de obiectul selectat, aplicația de calcul tabelar afișează un meniu contextual, care conține opțiunea **Format xxxxxx ...**, unde **xxxxxx** reprezintă denumirea obiectului – formă predesenată, text artistic, casetă de text etc. Evident, acest meniu conține și toate comenzile necesare pentru copierea, decuparea sau gruparea obiectelor.

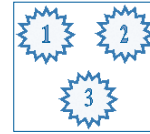
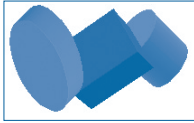
Întrebări și exerciții

- ❶ Prin ce se deosebește harta creată cu ajutorul aplicațiilor de calcul tabelar de o hartă obișnuită?
- ❷ Numiți indicatorii de date ce pot fi folosiți în hărțile create cu ajutorul aplicațiilor de calcul tabelar.
- ❸ Găsiți în manualele de istorie și de geografie hărți care ar putea fi create cu ajutorul aplicațiilor de calcul tabelar. Ce fel de date numerice sunt reprezentate pe astfel de hărți? Numiți indicatorii de date utilizați pe hărțile găsite de dvs.
- ❹ Care este deosebirea dintre o diagramă și o hartă, creată cu ajutorul aplicațiilor de calcul tabelar?
- ❺ Creați în baza datelor din foaia de calcul **Populația Republicii Moldova** (fig. 3.14) o diagramă cu bare sau cu coloane. Confrunțați diagrama creată cu harta din figura 3.13. Cum credeți, care dintre aceste imagini este mai sugestivă? Argumentați răspunsul dvs.
- ❻ Determinați dacă pe calculatorul cu care lucrați dvs. este instalat un program destinat creării și editării hărților.
- ❼ Amintiți-vă metodele de codificare a imaginilor în grafica orientată pe puncte și în grafica orientată pe obiecte. Care sunt avantajele și dezavantajele fiecăreia dintre aceste metode?
- ❽ Numiți cel puțin două surse din care ar putea fi preluate imagini, realizate în grafica orientată pe puncte.
- ❾ Afișați pe ecran bara de instrumente de desenare și explicați destinația fiecărui element de control.
- ❿ Numiți operațiile care pot fi efectuate asupra obiectelor grafice.
- ⓫ Explicați cum pot fi formate obiectele grafice din componența unei foi de calcul.
- ⓬ Inserați în foaia de calcul **Notele elevilor** (fig. 3.2) obiectele grafice



în așa mod, încât ele să indice clasamentul elevilor în funcție de reușita lor.

16 Inserați într-o foaie de calcul obiectele grafice ce urmează:



Formați aceste obiecte conform modelului prezentat mai sus.

3.5. CONSTRUIREA GRAFICELOR

Termeni-cheie:

- grafic
- diagramă XY

Una dintre problemele frecvent întâlnite în matematică, fizică, chimie, tehnică, medicină, economie etc. constă în construirea graficelor.

Graficul reprezintă un desen pe care este redată variația unei mărimi sau raportul dintre două mărimi variabile.

Pentru exemplificare, amintim graficele funcțiilor matematice $f: R \rightarrow R$, graficul mișcării mecanice $d = vt$, graficul fierberii apei, variația temperaturii aerului pe parcursul unei zile sau al unei luni ș.a.

În aplicațiile de calcul tabelar, graficele pot fi construite în baza valorilor din celulele foii de calcul ca **diagrame de tipul XY**. Valorile respective pot fi introduse în foaia de calcul de către utilizator, de exemplu, ca rezultate ale unor măsurări (temperatura, viteza, distanța, masa etc.), sau calculate în mod automat cu ajutorul aplicației de calcul tabelar. Pentru exemplificare, în *figura 3.15* este prezentată foaia de calcul **Temperatura**, care conține datele referitoare la variația temperaturii pe parcursul primelor zece ore ale uneia dintre zilele lunii martie.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1														
2		t, ore	0	1	2	3	4	5	6	7	8	9	10	
3		T, grade	2,5	-0,3	-2,8	-3,4	-2,8	0,0	2,9	5,5	7,5	8,8	9,5	
4														
5														

Fig. 3.15. Foaia de calcul **Temperatura**

Graficul, construit în baza datelor din foaia de calcul **Temperatura**, este prezentat în *figura 3.16*. Accentuăm că posibilitățile asistentului pentru diagrame nu permit reprezentarea pe grafice a săgeților ce simbolizează orientarea axelor de coordonate. În consecință, elementele respective au fost inserate ca obiecte grafice distincte.

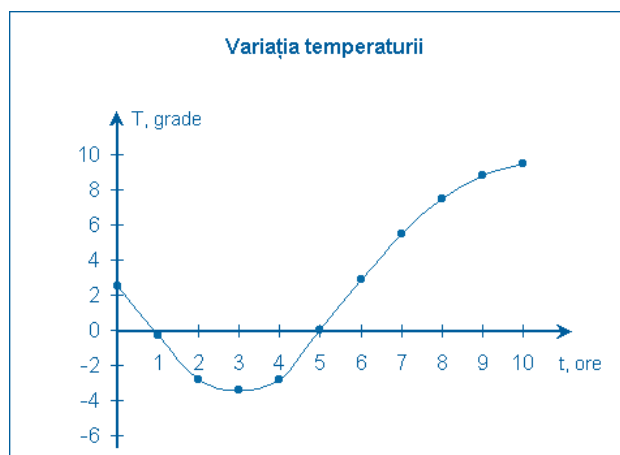


Fig. 3.16. Graficul **Variația temperaturii**

Graficele funcțiilor matematice pot fi construite în mod similar, însă valorile, necesare pentru trasarea graficului, pot fi calculate în mod automat cu ajutorul aplicației de calcul tabelar. De exemplu, presupunem că se dorește construirea graficului funcției

$$f(x) = x^2 - 5x + 3.$$

Evident, pentru a construi graficul respectiv, aplicația de calcul tabelar are nevoie de valorile argumentului x și de valorile funcției $y = x^2 - 5x + 3$, organizate în formă de tabel. În principiu, valorile respective pot fi calculate “manual” de către utilizator și introduse în celulele foii de calcul de la tastatură, însă o astfel de metodă este ineficientă. Tabelul respectiv poate fi completat în mod automat de însăși aplicația de calcul tabelar, înscriind în celulele foii de calcul formule de tipul $=x*x-5*x+3$.

În *figura 3.17* este prezentată foaia de calcul **Funcția de gradul doi**, care conține un tabel format din două rânduri. Primul rând al acestui tabel conține valorile argumentului x , de la 0 până la 5 cu pasul 0,5. Rândul al doilea conține valorile y , calculate conform formulei prezentate mai sus.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1														
2		x	0	0,5	1	1,5	2	2,5	3	3,5	4	4,5	5	
3		y	3	0,75	-1	-2,25	-3	-3,25	-3	-2,25	-1	0,75	3	
4														
5														

Fig. 3.17. Foaia de calcul **Funcția de gradul doi**

Pentru a înscrie în celulele tabelului formulele de calcul, inițial introducem în celula C3 formula $=C2*C2-5*C2+3$, pe care, în continuare, o copiem în celulele D3:M3 ale foii de calcul.

Graficul funcției $y = x^2 - 5x + 3$, creat în baza datelor din foaia de calcul **Funcția de gradul doi**, este prezentat în *figura 3.18*.

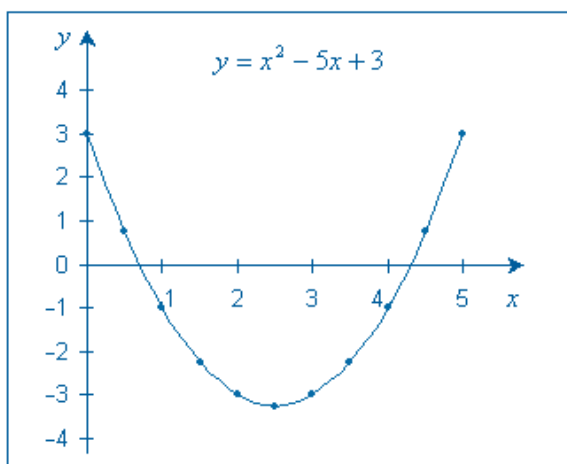


Fig. 3.18. Graficul unei funcții de gradul doi

Ca și în cazul graficului precedent, săgețile care indică direcția axelor de coordonate au fost inserate în graficul funcției de gradul doi ca obiecte grafice distincte.

De regulă, pentru a construi un grafic, trebuie să parcurgem următoarele etape:

- 1) alcătuim tabelul ce conține valorile funcției;
- 2) construim în baza tabelului o diagramă de tipul **XY**;
- 3) inserăm în diagramă obiectele grafice care indică direcțiile axelor de coordonate;
- 4) inserăm, în caz de necesitate, elemente suplimentare, care simplifică citirea graficului (liniile de caroaiaj, denumirile de axe, titlul graficului, unitățile de măsură etc.).

Crearea tabelului ce conține valorile funcției necesită o atenție sporită din partea utilizatorului, întrucât alegerea nereușită a domeniului de valori ale argumentului poate perturba funcționarea corectă a asistentului pentru diagrame.

Întrebări și exerciții

- ❶ Găsiți în manualele de fizică câteva grafice. Numiți obiectele din componența fiecărui grafic.
- ❷ În care cazuri este justificată reprezentarea datelor în formă de grafic?
- ❸ În tabelul ce urmează este prezentată variația temperaturii unui kilogram de apă, supus încălzirii. Construiți în baza acestor date graficul fierberii apei.

<i>t</i>, min.	0	1	2	3	4	5	6	7	8
<i>T</i>, °C	60	65	72	80	88	95	100	100	100

- ❹ În tabelul ce urmează este prezentat drumul *d* parcurs de un corp peste anumite intervale de timp *t*. Reprezentați datele din tabel în formă de grafic.

<i>t</i>, min.	0	1	2	3	4	5
<i>d</i>, m	0	48	96	144	192	240

- ❺ Construiți în baza datelor ce urmează graficul topirii și graficul solidificării naftalinei.

4.1. ELEMENTELE UNEI BAZE DE DATE

Termeni-cheie:

- date elementare
- date compuse
- bază de date
- listă de date
- înregistrare
- câmp

Informația supusă prelucrării este reprezentată în calculator în formă de date. Deosebim două categorii de date: date elementare și date compuse.

Datele elementare sunt indivizibile în raport cu prelucrările la care urmează a fi supuse. Exemple de astfel de date: numerele întregi, numerele reale, cuvintele unui text, obiectele grafice predesenate etc.

Datele compuse sunt alcătuite din mai multe date elementare și pot fi divizate în date mai simple. Exemple: informația din registrul clasei, care conține numele și prenumele elevului, anul nașterii, domiciliul etc., informația dintr-o notă de plată, în care se indică denumirea mărfii procurate, cantitatea, costul, taxele ș.a.m.d.

Evident, o dată poate fi considerată elementară dintr-un punct de vedere și compusă din altul, în raport cu prelucrările la care ea va fi supusă. De exemplu, în cazul calculelor aritmetice, orice număr întreg poate fi considerat ca fiind o dată elementară, însă în cazul reprezentării acestui număr pe calculator, el poate fi tratat ca fiind o dată compusă, formată din date mai simple – cifre binare sau cifre zecimale.

În prezent, volumul datelor prelucrate cu ajutorul calculatoarelor moderne este foarte mare, fapt ce complică stocarea, extragerea, prelucrarea și utilizarea informațiilor necesare. Pentru a simplifica aceste procese, au fost elaborate instrumente speciale, denumite baze de date.

Baza de date este un ansamblu de informații, organizate într-un mod special, fapt ce facilitează stocarea și extragerea lor.

De exemplu, dacă dorim să păstrăm pe calculator informațiile referitoare la elevii din clasa în care învățăm, putem crea o bază de date, ce va conține pentru fiecare persoană câte o dată complexă, formată din mai multe date elementare – numele, prenumele, anul nașterii,

domiciliul etc. În mod similar putem crea o bază de date ce va conține informații despre toți elevii din școala în care învățăm sau despre toți elevii din orașul în care locuim. În continuare, am putea extrage din această bază de date informațiile dorite, indicând numai numele sau anul de naștere al elevului, adresa domiciliului sau clasa în care învață.

Bazele de date sunt utilizate în cele mai diverse domenii: gestionarea întreprinderilor, evidența populației, controlul vamal, evidența mijloacelor de transport, activitățile bancare etc. De obicei, crearea și utilizarea unor baze mari de date se efectuează cu ajutorul programelor dedicate, care se studiază în cursurile avansate de informatică.

Aplicațiile de calcul tabelar conțin mijloace simple și eficiente pentru crearea unor baze de date relativ mici, suficiente însă pentru a satisface necesitățile utilizatorilor individuali și ale instituțiilor nu prea mari, așa cum sunt, de exemplu, primăriile, școlile, bibliotecile, magazinele etc. În cadrul aplicațiilor de calcul tabelar, bazele de date se realizează în formă de liste.

Lista de date reprezintă o structură tabelară cu rânduri și coloane. Fiecare rând al listei este numit înregistrare, iar fiecare coloană – câmp al bazei de date. Primul rând al tabelului conține denumirile de câmpuri.

Pentru exemplificare, în figura 4.1 este prezentată o foaie de calcul, care conține datele referitoare la participanții unui concurs de informatică, organizate în formă de listă de date.

	A	B	C	D	E	F	G	H	I
1									
2		Nr. crt.	Nume	Prenume	Anul nașterii	Clasa	Localitatea	Punctaj	
3		1	Munteanu	Ion	1995	12	Chișinău	263	
4		2	Vulpe	Petru	1995	12	Orhei	290	
5		3	Plugaru	Raisa	1996	11	Ungheni	531	
6		4	Rotaru	Constantin	2000	7	Ocnita	208	
7		5	Moraru	Ionița	1997	10	Chișinău	412	
8		6	Cibotaru	Mihai	1997	10	Cahul	585	
9		7	Munteanu	Constantin	2000	7	Călărași	301	
10		8	Margine	Vasile	1996	11	Orhei	531	
11		9	Bucur	Elena	1999	8	Tiraspol	115	
12		10	Olaru	Ana	1995	12	Ungheni	382	
13		11	Munteanu	Ana	1999	8	Călărași	531	
14		12	Cobzaru	Dorin	2000	7	Chișinău	585	
15		13	Cibotaru	Anton	1998	9	Orhei	314	
16		14	Melinte	Victor	1998	9	Ocnita	600	
17		15	Munteanu	Alexandru	1996	11	Chișinău	189	
18									

Fig. 4.1. Foaia de calcul **Lista participanților**

Această listă conține cincisprezece înregistrări, câte una pentru fiecare participant, și șapte câmpuri cu denumirile Nr. crt., Nume, Prenume, Anul nașterii, Clasa, Localitatea, Punctaj. Evident, fiecare înregistrare poate fi tratată ca o dată complexă, formată din datele elementare, înscrise în câmpurile ce-i aparțin. De exemplu, înregistrarea din rândul 5 conține datele elementare 3, Plugaru, Raisa, 1996, 11, Ungheni, 531, înscrise în celulele B5:H5 ale foii de calcul.

Pentru a crea o listă de date, mai întâi introducem, într-un rând, denumirile de câmpuri și, în continuare, înregistrările propriu-zise. Fiecare rând al domeniului ocupat de listă va conține o înregistrare, iar fiecare coloană va conține un câmp. În procesul creării listelor se vor respecta următoarele recomandări:

- fiecare listă de date va fi creată într-o foaie separată de calcul;
- între denumirile de câmpuri și înregistrări nu trebuie să existe rânduri goale;
- celulele dintr-o anumită coloană trebuie să conțină date de același tip;
- pentru a evita confuziile, denumirile de câmpuri vor fi formate diferit de datele ce urmează;
- datele din fiecare celulă a listei nu vor fi precedate de spații;
- celulele adiacente listei vor rămâne goale;
- fiecare înregistrare va fi numerotată, în acest scop utilizându-se prima sau ultima coloană a listei de date.

Se consideră că fiecare listă reprezintă o bază de date distinctă, care va fi prelucrată independent de listele aflate în alte foi de calcul ale aceleiași agende de lucru.

Întrebări și exerciții

- ❶ Explicați termenii *date elementare* și *date compuse*. Dați exemple de astfel de date.
- ❷ Ce reprezintă o bază de date? Care este destinația bazelor de date?
- ❸ Numiți câteva domenii în care se aplică bazele de date.
- ❹ Indicați pe *figura 4.1* următoarele elemente:
 - denumirile de câmpuri;
 - înregistrările care conțin numele Munteanu;
 - înregistrările care conțin prenumele Ana;
 - înregistrările care conțin punctajul 531;
 - înregistrările care conțin localitatea Chișinău.
- ❺ Introduceți în calculator foaia de calcul **Lista participanților** (*fig. 4.1*). Verificați dacă sunt respectate toate recomandările formulate în acest paragraf.
- ❻ Creați după modelul ce urmează foaia de calcul **Lista cărților**. Introduceți în foaia de calcul datele despre toate cărțile de care vă folosiți dvs. (câte o înregistrare pentru fiecare carte).

Nr. crt.	Autor	Titlu	Editura	Anul	Pagini
1.	Gremalschi Anatol	Informatica	Știința	2013	170
2.

- ❼ Creați după modelul ce urmează foaia de calcul **Prietenii**.

Nr. crt.	Nume	Prenume	Data nașterii	Telefon	Adresa
1.	Andronic	Vitalie	12-Sept-2000	324567	Decebal, 13, ap. 4
2.

Introduceți în foaia de calcul datele despre prietenii dvs. (câte o înregistrare pentru fiecare persoană).

- ❽ Creați foaia de calcul **Lista telefoanelor**, care trebuie să conțină informațiile dintr-o agendă de telefoane (câte o înregistrare pentru fiecare număr de telefon frecvent utilizat).

4.2. GESTIONAREA LISTELOR

Termeni-cheie:

- formular de date
- vizualizarea înregistrărilor
- căutarea înregistrărilor
- editarea înregistrărilor
- inserarea înregistrărilor
- eliminarea înregistrărilor

După creare, bazele de date servesc ca “depozite”, din care putem extrage informațiile necesare sau în care putem stoca informații noi. Evident, odată cu trecerea timpului, bazele de date trebuie actualizate, eliminând înregistrările devenite inutile, modificând celulele anumitor înregistrări sau adăugând înregistrări noi. Prin urmare, gestionarea unei baze de date presupune efectuarea următoarelor operații:

- căutarea anumitor înregistrări;
- inserarea înregistrărilor;
- eliminarea înregistrărilor;
- modificarea câmpurilor din componența anumitor înregistrări.

Întrucât în cadrul foilor de calcul listele de date nu au un statut special, actualizarea lor poate fi făcută utilizând tehnicile standard, aplicabile pentru oricare alt gen de informații:

- selectăm celula dorită;
- edităm conținutul celulei selectate.

Natural, în cazul unor liste ce conțin sute sau mii de înregistrări, căutarea manuală a celulelor ce vor fi supuse modificărilor necesită un volum mare de muncă și devine o operație oboșitoare.

Pentru a simplifica procesele de gestionare a listelor de date, aplicațiile de calcul tabelar conțin comenzi speciale, destinate prelucrării fiecărei înregistrări ca a unui ansamblu unitar de date. Majoritatea acestor comenzi sunt grupate în fereastra de dialog **Form** (Formular), care poate fi afișată pe ecran cu ajutorul comenzii cu același nume din meniul **Data** (Date). După selectarea unei celule arbitrare din cadrul listei de date și lansarea comenzii **Data**, **Form**, aplicația de calcul tabelar parcurge antetele de coloană și generează un formular ce conține casete de text marcate cu denumirile câmpurilor respective. De exemplu, în cazul foii de calcul **Lista participanților** (fig. 4.1), **formularul de date** va avea aspectul prezentat în figura 4.2.





Casetele de text din componența formularelor de date au o destinație dublă. În primul rând, ele servesc pentru afișarea informațiilor din câmpurile respective ale înregistrării curente. În rândul al doilea, aceste casete permit editarea informațiilor afișate, scutind astfel utilizatorul de parcurgerea oboșitoare a tuturor celulelor din componența listelor de date. Amintim că pentru parcurgerea rapidă a înregistrărilor putem folosi bara de defilare, afișată în centrul formularelor de date. Numărul înregistrării curente va fi afișat în colțul dreapta-sus al formularului, simplificându-se astfel selectarea înregistrărilor dorite.






Fig. 4.2. Formularul de date **Lista participanților**

Destinația elementelor de control ale formularelor de date este prezentată în tabelul 4.1.

Tabelul 4.1

Destinația elementelor de control ale formularelor de date

Butonul	Denumirea	Destinația
	New (Nou)	Șterge toate informațiile din casetele de text și pregătește baza de date pentru introducerea unei înregistrări noi. În colțul dreapta-sus al formularului apare mesajul New Record (Înregistrare nouă), care invită utilizatorul să introducă în casetele de text datele respective. Trecerea dintr-o casetă în alta se efectuează prin acționarea tastei Tab sau prin clicuri-stângă. Înregistrarea nou-formată este introdusă în baza de date numai după acționarea tastei Enter .
	Delete (Șterge)	Șterge înregistrarea curentă din baza de date. Înainte de a efectua operația de ștergere, aplicația de calcul tabelar afișează un mesaj prin care preîntâmpină utilizatorul că înregistrarea curentă va fi pierdută irevocabil.
	Restore (Restabilește)	Restabilește informația inițială din casetele de text ale formularului de date. Restabilirea informației este necesară în cazul introducerii în înregistrarea curentă a unor modificări nedorite și poate fi făcută numai până la acționarea tastei Enter .
	Find Previous (Caută precedenta)	Încarcă în formularul de date înregistrarea precedentă, care satisface, în cazul când au fost indicate, criteriile de căutare.

Butonul	Denumirea	Destinația
	Find Next (Caută următoarea)	Încarcă în formularul de date înregistrarea următoare, care satisface, în cazul când au fost indicate, criteriile de căutare.
	Criteria (Criterii)	Șterge toate informațiile din casetele de text și pregătește baza de date pentru căutarea anumitor înregistrări. În colțul dreapta-sus al formularului de date apare mesajul Criteria , care invită utilizatorul să introducă în casetele de text criteriile de căutare. Acestea pot fi anumite valori, de exemplu, Munteanu, sau condiții de genul =2000. După acționarea unuia dintre butoanele Find Prev sau Find Next , în formularul de date va fi încărcată înregistrarea care satisface criteriile de căutare.
	Clear (Curăță)	Șterge criteriile de căutare din casetele de text ale formularului de date.
	Form (Formular)	Comutează formularul de date din regimul de căutare în regimul de afișare a înregistrărilor.
	Close (Închide)	Închide formularul de date.

Pentru a facilita procesele de actualizare a bazelor de date, recomandăm cititorului să utilizeze următoarele operații-tip, destinate gestionării listelor din cadrul foilor de calcul.

Vizualizarea înregistrărilor:

- 1) selectăm o celulă arbitrară din componența listei;
- 2) lansăm comanda **Data, Form**, care va afișa pe ecran formularul de date cu prima înregistrare din listă;
- 3) pentru a trece la înregistrarea următoare, acționăm butonul **Find Next**;
- 4) pentru a trece la înregistrarea precedentă, acționăm butonul **Find Prev**.

Căutarea unei înregistrări:

- 1) selectăm o celulă arbitrară din componența listei;
- 2) lansăm comanda **Data, Form**, care va afișa pe ecran formularul de date cu prima înregistrare din listă;
- 3) acționăm butonul **Criteria**;
- 4) introducem în una sau în mai multe casete de text criteriile de căutare;
- 5) pentru a trece la înregistrarea următoare, care satisface criteriile de căutare, acționăm butonul **Find Next**;
- 6) pentru a trece la înregistrarea precedentă, care satisface criteriile de căutare, acționăm butonul **Find Prev**;
- 7) după găsirea înregistrării dorite, acționăm butonul **Form**.

Editarea unei înregistrări:

- 1) încărcăm în formularul de date înregistrarea dorită;
- 2) introducem în casetele de text modificările necesare;

- 3) acționăm tasta **Enter**;
- 4) trecem la editarea altei înregistrări sau acționăm butonul **Close**.

Inserarea înregistrărilor noi:

- 1) selectăm o celulă arbitrară din componența listei;
- 2) lansăm comanda **Data, Form**, care va afișa pe ecran formularul de date cu prima înregistrare din listă;
- 3) acționăm butonul **New**;
- 4) introducem în casetele de text datele înregistrării noi;
- 5) pentru a insera înregistrarea în listă, acționăm tasta **Enter**;
- 6) introducem în casetele de text datele unei alte înregistrări noi etc.;
- 7) pentru a termina procesul de inserare, acționăm butonul **Close**.

Eliminarea unor înregistrări:

- 1) selectăm o celulă arbitrară din componența listei;
- 2) lansăm comanda **Data, Form**, care va afișa pe ecran formularul de date cu prima înregistrare din listă;
- 3) încercăm în formular înregistrarea dorită;
- 4) acționăm butonul **Delete**.

Utilizarea eficientă a comenzilor din cadrul formularelor de date necesită anumite abilități, care se formează soluționând cât mai multe probleme practice, legate de gestionarea listelor de date.

Întrebări și exerciții

- ❶ Numiți operațiile utilizate frecvent în gestionarea bazelor de date.
- ❷ Ce acțiuni declanșează comanda **Data, Form**? De unde sunt preluate denumirile casetelor de text din cadrul unui formular de date?
- ❸ Explicați destinația elementelor de control ale unui formular de date.
- ❹ Deschideți foaia de calcul **Lista participanților** și afișați pe ecran formularul de date din *figura 4.2*. Ce informații conțin casetele de text imediat după afișarea formularului?
- ❺ Utilizând bara de defilare a formularului de date din *figura 4.2*, vizualizați toate înregistrările din foaia de calcul **Lista participanților**. Observați cum se modifică informațiile afișate în casetele de text. Indicați în foaia de calcul celulele, informația cărora este afișată în aceste casete.
- ❻ Afișați pe ecran formularul de date din *figura 4.2*. Comutați formularul de date în regimul de căutare (butonul **Criteria**) și introduceți în caseta de text **Nume** cuvântul Munteanu. Observați cum se modifică conținutul tuturor casetelor de text după acționarea butoanelor **Find Prev** și **Find Next**. Indicați în foaia de calcul înregistrările, informația cărora este afișată în aceste casete.
- ❼ Vizualizați, fără a parcurge toată lista de date din *figura 4.1*, înregistrările care se referă la:
 - a) participanții cu numele Munteanu;
 - b) participanții care învață în clasa a 7-a;
 - c) participanții născuți în anul 2000;
 - d) participantele cu prenumele Ana;
 - e) participanții care învață în clasa a 10-a;
 - f) participanții născuți în anul 1996;

- g) participanții care au acumulat mai mult de 200 de puncte;
 h) participanții care au acumulat mai mult de 500 de puncte.
- 8 Modificați punctajul participantului Cobzaru Dorin (fig. 4.1) din 585 în 588, iar cel al lui Melinte Victor – din 600 în 598.
- 9 Eliminați din foaia de calcul **Lista participanților** (fig. 4.1) înregistrările nr. 3, 7 și 10. Renumerotați înregistrările rămase în listă.
- 10 Inserați în foaia de calcul **Lista participanților** (fig. 4.1) înregistrările din tabelul ce urmează:

Nr. crt.	Nume	Prenume	Anul nașterii	Clasa	Localitatea	Punctaj
16.	Perju	Ion	1995	12	Bălți	149
17.	Mardare	Lucia	1999	8	Soroca	523
18.	Botnaru	Elena	1997	10	Bălți	487

- 11 Explicați, cum pot fi efectuate următoarele operații:
- vizualizarea tuturor înregistrărilor dintr-o listă de date;
 - căutarea anumitor înregistrări;
 - inserarea unor înregistrări noi;
 - eliminarea anumitor înregistrări;
 - modificarea unor informații din componența înregistrărilor existente?
- 12 Care metodă de gestionare a listelor este mai simplă: cea bazată pe selectarea fiecărei celule sau cea bazată pe utilizarea formularelor de date? Argumentați răspunsul dvs.

4.3. SORTAREA ÎNREGISTRĂRILOR

Termeni-cheie:

- sortare
- cheie de sortare
- ordine de sortare
- ordine ascendentă
- ordine descendentă

Informațiile dintr-o bază de date pot fi prelucrate mai ușor atunci când utilizatorul dispune de posibilitatea de a aranja înregistrările într-o anumită ordine. De exemplu, înregistrările din baza de date **Lista participanților** (fig. 4.1) pot fi aranjate în ordine alfabetică după valorile din câmpurile Nume, Prenume sau în ordine numerică după valorile din câmpul Punctaj. Aranjarea în ordine alfabetică este utilă în procesul înregistrării participanților, iar aranjarea în ordine numerică – pentru a determina învingătorii concursului.

Operația de rearanjare a înregistrărilor dintr-o listă după valorile unuia sau a mai multor câmpuri se numește sortare.

Aplicațiile de calcul tabelar dispun de instrumente speciale, destinate sortării înregistrărilor. În cazul aplicației **Microsoft Excel**, sortarea înregistrărilor se efectuează cu ajutorul comenzii **Data, Sort** (Date, Sortare). După lansarea acestei comenzi, pe ecran este afișată fereastra de dialog **Sort** (fig. 4.3), în care utilizatorul indică opțiunile dorite de sortare.



Fig. 4.3. Fereastra de dialog **Sort** (Sortare)

În fereastra de dialog **Sort** pot fi indicate până la trei **chei de sortare**, fiecare cheie reprezentând o denumire de câmp din antetul listei. **Ordinea de sortare** – după valori crescătoare sau după valori descrescătoare – se precizează pentru fiecare cheie cu ajutorul butoanelor radio **Ascending** și **Descending**.

Amintim că în cazul valorilor de tip text, înregistrările vor fi aranjate în ordine lexicografică, conform tabelului de codificare instalat pe calculatorul respectiv. Evident, în cazul textelor ce nu conțin simboluri matematice, valorile respective vor fi aranjate în ordine alfabetică.

Pentru exemplificare, în figura 4.4 este prezentată foaia de calcul **Lista participanților**, după sortarea bazei de date conform cheii **Nume**, în ordine crescătoare.

	A	B	C	D	E	F	G	H
1								
2		Nr. crt.	Nume	Prenume	Anul nașterii	Clasa	Localitatea	Punctaj
3		9	Bucur	Elena	1999	8	Tiraspol	115
4		6	Ciubotaru	Mihai	1997	10	Cahul	585
5		13	Ciubotaru	Anton	1998	9	Orhei	314
6		12	Cobzaru	Dorin	2000	7	Chișinău	585
7		8	Margine	Vasile	1996	11	Orhei	531
8		14	Melinte	Victor	1998	9	Ocnița	600
9		5	Moraru	Ionița	1997	10	Chișinău	412
10		1	Munteanu	Ion	1995	12	Chișinău	263
11		7	Munteanu	Constantin	2000	7	Călărași	301
12		11	Munteanu	Ana	1999	8	Călărași	531
13		15	Munteanu	Alexandru	1996	11	Chișinău	189
14		10	Olaru	Ana	1995	12	Ungheni	382
15		3	Plugaru	Raisa	1996	11	Ungheni	531
16		4	Rotaru	Constantin	2000	7	Ocnița	208
17		2	Vulpe	Petru	1995	12	Orhei	290
18								

Fig. 4.4. Sortarea listei după cheia **Nume**

Confruntând *figura 4.1*, în care este prezentată lista până la sortare, și *figura 4.4*, în care este prezentată aceeași listă după sortare, se observă că înregistrările listei au fost rearanjate în ordine alfabetică numai după valorile text din câmpul Nume. Întrucât câmpul Prenume nu a fost indicat în calitate de a doua cheie de sortare (*fig. 4.3*), valorile respective nu au fost luate în considerare. De exemplu, în listă apar două înregistrări cu numele de familie Ciubotaru și patru înregistrări cu numele de familie Munteanu, care nu sunt aranjate în ordine alfabetică după valorile din câmpul Prenume. Pentru a sorta înregistrările după ambele chei – Nume și Prenume –, în fereastra de dialog **Sort** se indică ambele câmpuri și ordinea dorită de sortare (*fig. 4.5*).

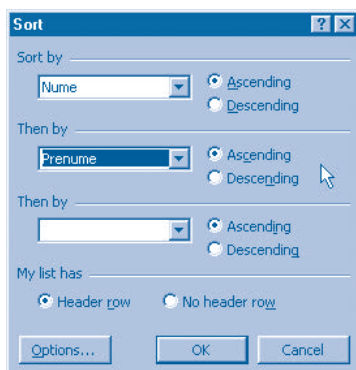


Fig. 4.5. Indicarea a două chei de sortare

Baza de date, sortată după cheile Nume și Prenume, este prezentată în *figura 4.6*.

	A	B	C	D	E	F	G	H
1								
2		Nr. crt.	Nume	Prenume	Anul nașterii	Clasa	Localitatea	Punctaj
3		9	Bucur	Elena	1999	8	Tiraspol	115
4		13	Ciubotaru	Anton	1998	9	Orhei	314
5		6	Ciubotaru	Mihai	1997	10	Cahul	585
6		12	Cobzaru	Dorin	2000	7	Chișinău	585
7		8	Margine	Vasile	1996	11	Orhei	531
8		14	Melinte	Victor	1998	9	Ocnița	600
9		5	Moraru	Ionita	1997	10	Chișinău	412
10		15	Munteanu	Alexandru	1996	11	Chișinău	189
11		11	Munteanu	Ana	1999	8	Călărași	531
12		7	Munteanu	Constantin	2000	7	Călărași	301
13		1	Munteanu	Ion	1995	12	Chișinău	263
14		10	Olaru	Ana	1995	12	Ungheni	382
15		3	Plugaru	Raisa	1996	11	Ungheni	531
16		4	Rotaru	Constantin	2000	7	Ocnița	208
17		2	Vulpe	Petru	1995	12	Orhei	290
18								

Fig. 4.6. Sortarea listei după cheile **Nume** și **Prenume**

Confruntând lista sortată după cheia Nume (*fig. 4.4*) și lista sortată după cheile Nume și Prenume (*fig. 4.6*), se observă că în cazurile în care numele de familie coincid, înregistrările au fost rearanjate în ordine alfabetică, ținându-se cont și de prenumele participanților la concurs. Astfel, în cazul numelui de familie Ciubotaru, primul apare în listă Ciubotaru Anton,

urmat de Ciubotaru Mihai. În mod similar, participanții cu numele de familie Munteanu sunt aranjați în ordinea Alexandru, Ana, Constantin și Ion.

Prin urmare, aranjarea înregistrărilor se efectuează în ordinea în care au fost indicate cheile de sortare din fereastra de dialog **Sort** (fig. 4.5):

- 1) mai întâi înregistrările sunt sortate după prima cheie;
- 2) în continuare, înregistrările sunt sortate, dacă a fost indicată, după cheia a doua;
- 3) în sfârșit, înregistrările sunt sortate, dacă a fost indicată, după cheia a treia.

În caz de necesitate, utilizatorul poate indica opțiuni suplimentare de sortare. Astfel, după acționarea butonului **Options** (Opțiuni), pe ecran va fi afișată fereastra de dialog **Sort Options** (Opțiuni de sortare), în care poate fi precizată ordinea de sortare: după lunile anului sau după zilele săptămânii, pe rânduri sau pe coloane ș.a.

În general, **ordinea de sortare ascendentă** aranjează înregistrările în felul următor:

- numerele de la cel mai mic negativ la cel mai mare pozitiv;
- datele calendaristice și mărcile de timp în ordine cronologică;
- textele în ordine alfabetică;
- valorile logice cu FALSE întâi și apoi cu TRUE;
- valorile de eroare în ordinea în care sunt întâlnite;
- spațiile.

Evident, **ordinea de sortare descendentă** aranjează înregistrările exact în ordinea inversă celei ascendente, cu excepția faptului că spațiile sunt plasate tot pe ultima poziție.

Să reținem că aplicațiile de calcul tabelar oferă utilizatorului posibilitatea să sorteze întreaga listă sau numai o anumită porțiune a ei. Pentru sortarea unei părți a listei, înainte de lansarea comenzii **Data, Sort**, se selectează numai înregistrările dorite.

Întrebări și exerciții

- 1 Explicați termenul *sortarea înregistrărilor*. Când apare necesitatea de a sorta înregistrările unei baze de date?
- 2 Care este destinația cheilor de sortare? De unde sunt preluate denumirile eventualelor chei de sortare?
- 3 Cum se indică ordinea de sortare a înregistrărilor unei baze de date? Poate fi oare această ordine modificată?
- 4 Explicați cum vor fi aranjate în ordine ascendentă următoarele informații:
 - numerele;
 - datele calendaristice;
 - mărcile de timp;
 - textele;
 - valorile logice;
 - erorile.Cum vor fi aranjate aceste informații în ordine descendentă?
- 5 Contează oare ordinea în care sunt indicate cheile de sortare din fereastra de dialog **Sort** (fig. 4.5)? Argumentați răspunsul dvs.
- 6 Deschideți foaia de calcul **Lista participanților** (fig. 4.1). Sortați înregistrările bazei de date din această foaie de calcul după cum urmează:

- a) după cheia Nume, în ordine ascendentă;
- b) după cheia Nume, în ordine descendentă;
- c) după cheile Nume și Prenume, în ordine ascendentă;
- d) după cheile Nume și Prenume, în ordine descendentă;
- e) după cheia Punctaj, în ordine ascendentă;
- f) după cheia Punctaj, în ordine descendentă;
- g) după cheia Anul nașterii, în ordine ascendentă;
- h) după cheia Anul nașterii, în ordine descendentă;
- i) după cheia Clasa, în ordine ascendentă;
- j) după cheia Clasa, în ordine descendentă;
- k) după cheile Anul nașterii și Clasa, în ordine ascendentă;
- l) după cheile Anul nașterii și Clasa, în ordine descendentă;
- m) după cheia Clasa, în ordine ascendentă, și cheia Punctaj, în ordine descendentă.

Explicați rezultatele afișate pe ecran.

- 7 Aranjați înregistrările din foaia de calcul **Lista cărților** (vezi exercițiul 6 din paragraful 4.1) după cum urmează:
 - a) în ordinea alfabetică a autorilor;
 - b) în ordinea alfabetică a titlurilor;
 - c) în ordinea alfabetică a editurilor;
 - d) în ordinea cronologică a anilor în care au fost editate;
 - e) în ordinea crescătoare a numărului de pagini.
- 8 Dispuneți înregistrările din foaia de calcul **Prieteni** (vezi exercițiul 7 din paragraful 4.1) după cum urmează:
 - a) în ordine alfabetică, după nume și prenume;
 - b) în ordine cronologică, conform datelor de naștere;
 - c) în ordine alfabetică, după denumirile de străzi.

- 9 Deschideți foaia de calcul **Note** (fig. 2.13). Această foaie conține trei domenii distincte de celule:

B2:J12 – lista elevilor și notele la unele discipline școlare;

L2:N12 – notele minimă, medie și maximă pentru fiecare elev;

C14:J16 – notele minimă, medie și maximă pentru fiecare disciplină școlară.

Întrucât fiecare dintre aceste domenii este înconjurat de celule vide, aplicația **Microsoft Excel** le tratează ca pe liste distincte de date. Sortați lista elevilor după cum urmează:

- a) în ordine alfabetică, după nume și prenume;
- b) în ordine ascendentă, conform notelor la una dintre disciplinele școlare, de exemplu, la informatică;
- c) în ordine ascendentă, conform notelor minime din coloana L;
- d) în ordine descendentă, conform notelor medii din coloana M;
- e) în ordine ascendentă, conform notelor maxime din coloana N.

Încercați să copiați în altă foaie de calcul datele care se referă la notele minimă, medie și maximă pentru fiecare disciplină școlară. Sortați aceste date în ordine ascendentă conform notelor medii.

4.4. SELECTAREA ÎNREGISTRĂRILOR

Termeni-cheie:

- selectarea înregistrărilor
- filtrarea înregistrărilor
- condiții de selectare
- copierea înregistrărilor selectate

Bazele de date sunt utilizate pentru păstrarea unor cantități mari de informații. În procesul prelucrării acestor informații apare necesitatea selectării înregistrărilor care satisfac anumite condiții. De exemplu, în cazul foii de calcul **Lista participanților** (fig. 4.1), prezintă interes extragerea înregistrărilor care se referă la participanții ce au obținut mai mult de 500 de puncte, la competitorii care învață în clasa a 8-a, la competitorii dintr-o anumită localitate ș.a.m.d.

Operația de alegere dintr-o bază de date a unor înregistrări care satisfac anumite condiții se numește selectarea sau filtrarea înregistrărilor.

În cazul aplicației **Microsoft Excel**, selectarea înregistrărilor se efectuează cu ajutorul comenzii **Filter** (Filtru) din meniul **Data**. Această comandă are trei subcomenzi: **AutoFilter** (Filtru automat), **Show All** (Afișează tot) și **Advanced Filter** (Filtru avansat). După selectarea unei celule arbitrare din interiorul listei de date și activizarea comenzii **Data, Filter, AutoFilter**, aplicația de calcul tabelar atașează fiecărei denumiri de câmp câte o listă derulantă ce conține toate valorile care apar în coloana respectivă (fig. 4.7).

	A	B	C	D	E	F	G	H
1								
2		Nr. crt	Nume	Prenume	Anul nașterii	Clasa	Localitatea	Punctaj
3		1	Munteanu	Ion	(All)	12	Chișinău	263
4		2	Vulpe	Petru	(Top 10...)	12	Orhei	290
5		3	Plugaru	Raisa	(Custom...)	11	Ungheni	531
6		4	Rotaru	Constantin	1995	7	Ocnîța	208
7		5	Moraru	Ionița	1996	10	Chișinău	412
8		6	Ciubotaru	Mihai	1997	10	Cahul	585
9		7	Munteanu	Constantin	1998	7	Călărași	301
10		8	Margine	Vasile	1999	11	Orhei	531
11		9	Bucur	Elena	1999	8	Tiraspol	115
12		10	Olaru	Ana	1995	12	Ungheni	382
13		11	Munteanu	Ana	1999	8	Călărași	531
14		12	Cobzaru	Dorin	2000	7	Chișinău	585
15		13	Ciubotaru	Anton	1998	9	Orhei	314
16		14	Melinte	Victor	1998	9	Ocnîța	600
17		15	Munteanu	Alexandru	1996	11	Chișinău	189
18								

Fig. 4.7. Filtrarea automată a înregistrărilor

Pentru a selecta înregistrările dorite, utilizatorul va indica pentru unul sau mai multe câmpuri ale listei de date **condițiile de selectare**. Acestea pot fi:

- All** – toate valorile care apar în celulele de sub denumirile de câmp;
- Top 10** – primele zece din toate valorile care apar în celulele de sub denumirile de câmp, aranjate în ordine crescătoare;
- Custom** – condiții de selectare definite de utilizator;
- o singură valoare din cele care apar în celulele de sub denumirea de câmp.

Imediat după introducerea condițiilor de selectare, aplicația de calcul tabelar parcurge lista de date, verifică condițiile de selectare pentru fiecare câmp și “ascunde” înregistrările care nu satisfac condițiile indicate. Prin urmare, în lista de date vor fi afișate numai înregistrările care satisfac condițiile de selectare impuse fiecărui câmp. Pentru a se reveni la afișarea tuturor înregistrărilor și eliminarea condițiilor de selectare, se activează comanda **Data, Filter, Show All** (Date, Filtru, Afișează tot).

De exemplu, presupunem că se dorește selectarea tuturor înregistrărilor care se referă la participanții la concursul de informatică, născuți în anul 2000. Pentru a selecta aceste înregistrări, indicăm în lista derulantă a câmpului Anul nașterii (fig. 4.7) valoarea 2000. Imediat după alegerea acestei opțiuni, pe ecran va fi afișată lista din figura 4.8.

	A	B	C	D	E	F	G	H	I
1									
2		Nr. crt	Nume	Prenume	Anul nașter	Clasa	Localitatea	Punctaj	
6		4	Rotaru	Constantin	2000	7	Ocnia	208	
9		7	Munteanu	Constantin	2000	7	Călărași	301	
14		12	Cobzaru	Dorin	2000	7	Chișinău	585	
18									
19									

Fig. 4.8. Selectarea înregistrărilor conform condiției “Anul nașterii = 2000”

În mod similar, dacă se dorește selectarea tuturor înregistrărilor care se referă la participanții din Chișinău, indicăm în lista derulantă a câmpului Localitatea valoarea respectivă. Înregistrările selectate de aplicația de calcul tabelar sunt prezentate în figura 4.9.

	A	B	C	D	E	F	G	H	I
1									
2		Nr. crt	Nume	Prenume	Anul nașter	Clasa	Localitatea	Punctaj	
3		1	Munteanu	Ion	1995	12	Chișinău	263	
7		5	Moraru	Ionita	1997	10	Chișinău	412	
14		12	Cobzaru	Dorin	2000	7	Chișinău	585	
17		15	Munteanu	Alexandru	1996	11	Chișinău	189	
18									
19									

Fig. 4.9. Selectarea înregistrărilor conform condiției “Localitatea = Chișinău”

Pentru a selecta înregistrările care se referă la participanții din Chișinău, născuți în anul 2000, vom indica valorile respective în două liste derulante: valoarea 2000 în lista Anul nașterii și valoarea Chișinău în lista Localitatea. Evident, aplicația de calcul tabelar va afișa pe ecran o singură înregistrare (fig. 4.10).

	A	B	C	D	E	F	G	H	I
1									
2		Nr. crt	Nume	Prenume	Anul nașter	Clasa	Localitatea	Punctaj	
14		12	Cobzaru	Dorin	2000	7	Chișinău	585	
18									
19									

Fig. 4.10. Selectarea înregistrărilor conform condițiilor “Anul nașterii = 2000” și “Localitatea = Chișinău”

Dacă utilizatorul alege din lista derulantă atașată fiecărui câmp opțiunea **Custom** (La comandă), aplicația de calcul tabelar afișează pe ecran fereastra de dialog **Custom AutoFilter** (Filtru automat la comandă), în care pot fi indicate condiții de selectare definite de utilizator. De exemplu, dacă se dorește selectarea înregistrărilor care se referă la participanții ce au obținut 500 și mai multe puncte, în fereastra de dialog se va introduce condiția "Punctaj is greater than or equal to 500" (fig. 4.11). Înregistrările selectate de aplicația de calcul tabelar sunt prezentate în figura 4.12.

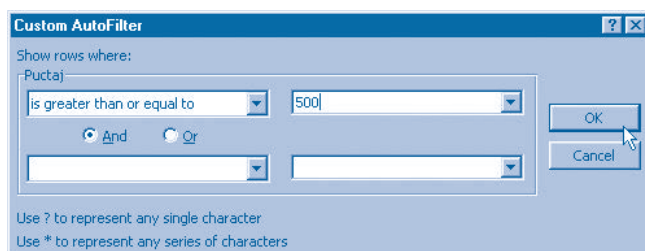


Fig. 4.11. Condiții de selectare definite de utilizator

	A	B	C	D	E	F	G	H	I
1									
2		Nr. crt	Nume	Prenume	Anul nașter	Clasa	Localitatea	Punctaj	
5		3	Plugaru	Raisa	1996	11	Ungheni	531	
8		6	Ciubotaru	Mihai	1997	10	Cahul	585	
10		8	Margine	Vasile	1996	11	Orhei	531	
13		11	Munteanu	Ana	1999	8	Călărași	531	
14		12	Cobzaru	Dorin	2000	7	Chișinău	585	
16		14	Melinte	Victor	1998	9	Ocnița	600	
18									
19									

Fig. 4.12. Selectarea înregistrărilor conform condiției "Punctaj \geq 500"

Criteriile de selectare definite de utilizator pot include una sau două condiții reunite prin operația logică ȘI (**And**) / SAU (**Or**). De obicei, astfel de condiții se utilizează atunci când dorim să afișăm înregistrările ce conțin valori dintr-un anumit domeniu.

În general, înregistrările selectate pot fi supuse unor prelucrări ulterioare, de exemplu, sortate după valorile anumitor câmpuri, inserate într-o altă listă, tipărite la imprimantă etc. În astfel de cazuri se recomandă copierea înregistrărilor selectate într-o altă zonă a foi de lucru sau chiar într-o altă foaie.

Pentru a **copia înregistrările selectate**, se efectuează următoarele operații:

- 1) selectăm înregistrările dorite folosind comanda **Data, AutoFilter**;
- 2) selectăm domeniul de celule ce conține lista cu înregistrările filtrate;
- 3) executăm comanda **Copy** din meniul **Edit** (când există un filtru activ, această comandă copiază numai înregistrările selectate);
- 4) selectăm celula stânga-sus a domeniului destinație;
- 5) executăm comanda **Paste** din meniul **Edit**.

Aplicația de calcul tabelar va copia înregistrările selectate la noua locație fără ca lista inițială să sufere schimbări. Amintim că pentru a se reveni la afișarea tuturor înregistrărilor din lista inițială și eliminarea tuturor condițiilor de selectare, se execută comanda **Data, Filter, Show All**.

Întrebări și exerciții

- ❶ Care este destinația listelor derulante, atașate de aplicația de calcul tabelar fiecărei denumiri de câmp a listei de date?
- ❷ Explicați termenul *selectarea (filtrarea)* înregistrărilor. În ce loc al foii de calcul sunt memorate înregistrările selectate?
- ❸ Care este destinația opțiunilor **All**, **Top 10** și **Custom** din componența listelor derulante atașate de aplicația de calcul tabelar denumirilor de câmpuri ale listelor de date?
- ❹ Explicați destinația elementelor de control ale ferestrei de dialog **Custom AutoFilter** (fig. 4.11).
- ❺ Cum pot fi copiate înregistrările selectate într-un alt loc al foii de calcul?
- ❻ Explicați destinația opțiunilor **AutoFilter** și **Show All** ale comenzii **Filter** din componența meniului **Data**.
- ❼ Utilizând sistemul de asistență, încercați să determinați destinația opțiunii **Advanced Filter** din componența comenzii **Data, Filter**.
- ❽ Contează oare ordinea în care sunt indicate condițiile de selectare din listele derulante atașate fiecărei denumiri de câmp? Argumentați răspunsul dvs.
- ❾ Deschideți foaia de calcul **Lista participanților** (fig. 4.1). Selectați din lista de date înregistrările care sunt conforme următoarelor condiții:
 - a) participanții selectați locuiesc în Orhei;
 - b) participanții selectați sunt născuți în anul 1995;
 - c) participanții selectați sunt născuți în anul 1995 și locuiesc în Orhei;
 - d) participanții selectați locuiesc în Orhei sau în Chișinău;
 - e) participanții selectați au numele de familie Munteanu sau Ciubotaru;
 - f) participanții selectați învață în clasa a 7-a și locuiesc în Ocnița;
 - g) participanții selectați au acumulat 550 și mai multe puncte;
 - h) numele de familie al participanților selectați începe cu litera M.Explicați rezultatele afișate pe ecran.
- ❿ Deschideți foaia de calcul **Lista cărților** (vezi exercițiul 6 din paragraful 4.1). Selectați din lista de date înregistrările care sunt conforme următoarelor condiții:
 - a) cărțile au fost editate în anii 1995–2003;
 - b) cărțile au fost publicate de Editura ȘTIINȚA;
 - c) fiecare carte conține cel mult 200 de pagini.
- ⓫ Selectați din lista de date din componența foii de calcul **Prietenii** (vezi exercițiul 7 din paragraful 4.1) înregistrările care satisfac următoarele condiții:
 - a) persoanele s-au născut în anul indicat de utilizator;
 - b) persoanele au prenumele indicat de utilizator;
 - c) persoana are telefonul indicat de utilizator.
- ⓬ Selectați din foaia de calcul **Lista participanților** (fig. 4.1) înregistrările care se referă la participanții ce au acumulat 500 și mai multe puncte. Copiați înregistrările selectate în altă foaie de calcul. Ordonăți lista astfel obținută după cheia **Punctaj**, în ordine descendentă.
- ⓭ Creați în baza datelor din foaia de calcul **Lista participanților** (fig. 4.1) baze separate de date ce vor conține:
 - a) participanții care învață în clasa a 7-a;
 - b) participanții care locuiesc în Chișinău;
 - c) participanții născuți în anul 1995;
 - d) participanții din clasa a 11-a care au acumulat 500 și mai multe puncte.Ordonăți aceste liste după cheile **Nume** și **Prenume**, în ordine ascendentă.

5.1. ALGORITMI ȘI EXECUTANȚI

Termeni-cheie:

- algoritm
- executant
- comandă manuală
- comandă prin program
- program
- limbaj de programare

Prelucrarea informației cu ajutorul calculatoarelor presupune existența unor programe care se încarcă în memoria internă a acestora. Cunoaștem deja că orice program de calculator reprezintă o succesiune de cuvinte binare care indică ordinea (secvența) calculelor. Programele pentru calculatoare se elaborează în baza algoritmilor.

În mod intuitiv, algoritmul reprezintă o mulțime finită de instrucțiuni care, fiind executate într-o ordine bine stabilită, produc în timp finit un rezultat. Procesul de elaborare a algoritmilor se numește algoritmizare.

Amintim că în viața cotidiană instrucțiunile reprezintă anumite indicații date cuiva în vederea îndeplinirii unui lucru. Exemple de instrucțiuni:

- 1) înainte de a porni calculatorul, verificați integritatea cablului de rețea;
- 2) dacă $a \neq 0$, calculați $x = b/a$;
- 3) pentru a afla rădăcinile ecuației de gradul doi, calculați mai întâi discriminantul $D = b^2 - 4ac$.

Cuvântul **algoritm** provine de la numele marelui matematician al Evului Mediu *Al-Khwarizmi Muhammed ibn Musa* (cca 780–850), care a folosit pentru prima oară reguli precise și clare pentru efectuarea operațiilor aritmetice de bază, studiate astăzi în clasele primare: adunarea, scăderea, înmulțirea și împărțirea. Mai târziu, aceste reguli apar sub denumirea de algoritm în cartea “Elementele lui Euclid”. Se consideră că algoritmul lui Euclid pentru calculul celui mai mare divizor comun a două numere naturale este unul dintre primii algoritmi cunoscuți în matematică.

Exemplu de algoritm:

Date inițiale: lungimile a, b ale catetelor triunghiului dreptunghic.

1. Calculați $x = a^2$.

2. Calculați $y=b^2$.

3. Calculați $z=x + y$.

4. Calculați $c = \sqrt{z}$.

Rezultat: lungimea ipotenuzei c .

Evident, acest algoritm poate fi executat de orice persoană care cunoaște operațiile aritmetice de bază, operațiile cu puteri și operațiile cu radicali.

În prezent, sensul cuvântului *algoritm* s-a extins, acest termen fiind utilizat în mai multe domenii ale științei și tehnicii pentru a descrie o succesiune de instrucțiuni destinate soluționării anumitor probleme. În informatică noțiunea de algoritm se asociază în mod obligatoriu cu noțiunea de executant.

Executantul reprezintă un obiect care poate îndeplini anumite comenzi. Mulțimea acestor comenzi formează repertoriul executantului.

De exemplu, televizorul poate fi tratat ca un obiect care îndeplinește următoarele comenzi: conectarea la rețea, deconectarea de la rețea, mărirea volumului, micșorarea volumului, recepționarea canalului nr. 1, recepționarea canalului nr. 2 ș.a.m.d. Comenzile respective sunt comunicate televizorului prin intermediul butoanelor de pe pupitrul de comandă. În mod similar, calculatorul reprezintă un obiect care îndeplinește următoarele comenzi: adunarea, scăderea, înmulțirea, împărțirea și compararea numerelor, citirea datelor de la tastatură, afișarea datelor pe ecran, scrierea datelor pe un disc magnetic, citirea datelor de pe un disc optic etc.

Definirea exactă a unui executant include:

- 1) descrierea setului (repertoriului) de comenzi pe care executantul le poate îndeplini;
- 2) descrierea mediului în care lucrează executantul.

În continuare vom studia executanții **Cangurul** și **Furnica**, elaborați în scopuri didactice pentru școlile din țara noastră.

Executantul Cangurul

Acest executant reprezintă un program de calculator, care derulează sub sistemul de operare **Windows**. Fereastra aplicației **Cangurul** este prezentată în *figura 5.1*.

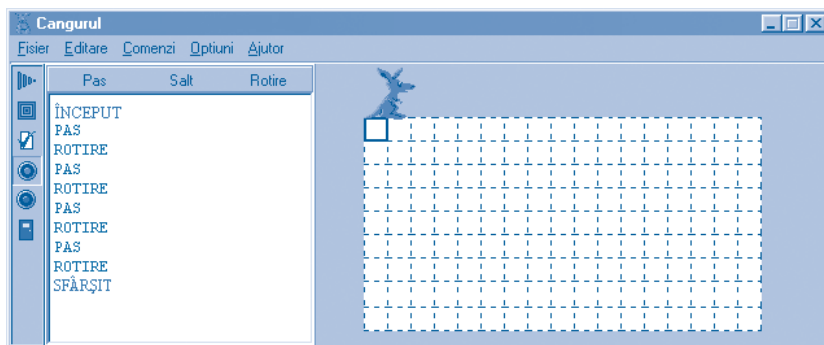


Fig. 5.1. Fereastra aplicației **Cangurul**

Executantul propriu-zis este simbolizat prin pictograma unui cangur, care poate îndeplini următoarele comenzi:

PAS – Cangurul se deplasează cu un pătrățel, trasând segmentul respectiv de dreaptă;
SALT – Cangurul se deplasează cu un pătrățel, însă nu desenează nimic;
ROTIRE – Cangurul se rotește cu 90° după acele de ceasornic.

Fereastra aplicației **Cangurul** conține următoarele elemente:

1) **Bara de meniuri**, care include meniurile standard Fișier, Editare, Comenzi, Opțiuni, Ajutor;

2) **Centrul de comandă**, care include butoanele Pas, Salt, Rotire, Execută, Stop, Control, Executare manuală, Executare automată, Ieșire;

3) **Zona de editare** a programelor;

4) **Mediul de lucru** al Cangurului, care reprezintă un câmp dreptunghiular, liniat în pătrățele.

Deosebim două moduri de comandă a executanților: comandă manuală și comandă prin program.

Modul de comandă manuală presupune introducerea separată a fiecărei comenzi și îndeplinirea ei de către executant.

De exemplu, presupunem că se dorește desenarea unui pătrat. În cazul comenzii manuale, utilizatorul va acționa butoanele Pas, Rotire și Salt din componența Centrului de comandă în așa mod, încât Cangurul să deseneze pătratul respectiv.

Modul de comandă prin program presupune memorarea în prealabil a unei secvențe de comenzi și executarea lor în regim automat, fără intervenția utilizatorului.

În cazul modului de comandă prin program, utilizatorul va memora în zona de editare a programelor secvența de instrucțiuni (comenzi) care asigură desenarea pătratului (fig. 5.1). Să reținem că începutul și sfârșitul fiecărui program se indică cu ajutorul cuvintelor auxiliare ÎNCEPUT și SFÂRȘIT.

Programul reprezintă un algoritm scris în limbajul executantului. Procesul de elaborare a programelor se numește programare.

Evident, programele pot fi salvate pe disc și utilizate de mai multe ori. În cazul aplicației **Cangurul**, salvarea și deschiderea programelor se efectuează cu ajutorul comenzilor grupate în meniul Fișier.

În general, programele pentru executanți se scriu cu ajutorul unor limbaje speciale, denumite **limbaje de programare**. Aceste limbaje conțin instrucțiuni și cuvinte auxiliare. De obicei, comenzile executantului sunt în același timp și instrucțiuni ale limbajului de programare. De exemplu, limbajul de programare al executantului **Cangurul** conține instrucțiunile PAS, SALT, ROTIRE și cuvintele auxiliare ÎNCEPUT și SFÂRȘIT.

Executantul Furnica

În această aplicație executantul este simbolizat prin imaginea unei furnici, care se poate deplasa pe un câmp divizat în pătrățele (fig. 5.2). În unele pătrățele ale câmpului pot fi inserate caractere imprimabile. De obicei, se cere elaborarea unui program, care aranjează aceste caractere într-o anumită ordine.

Executantul poate îndeplini comenzile SUS, JOS, DREAPTA, STÂNGA, care deplasează Furnica din pătrățelul curent în unul dintre pătrățelele vecine. Dacă în pătrățelul vecin se

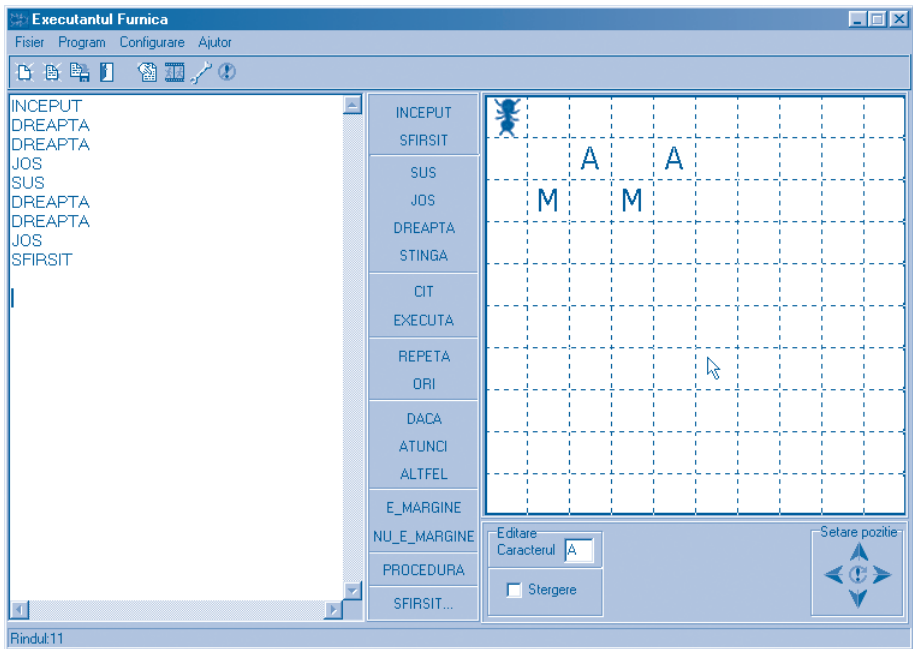


Fig. 5.2. Fereastra aplicației **Furnica**

afă un caracter, el va fi împins, când acest lucru este posibil, în direcția mișcării. Începutul și sfârșitul programelor se indică cu ajutorul cuvintelor auxiliare ÎNCEPUT și SFÂRȘIT.

Fereastra aplicației **Furnica** (fig. 5.2) conține următoarele elemente:

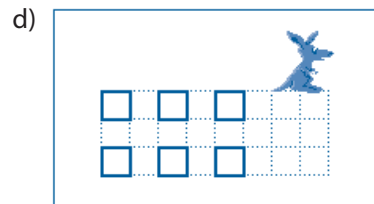
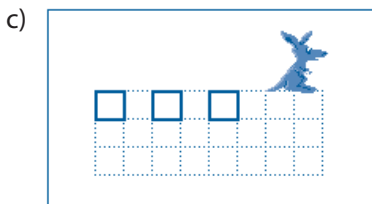
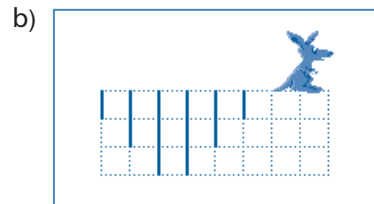
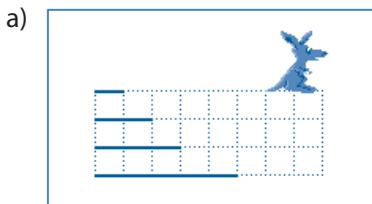
- 1) **Bara de meniuri**, care include meniurile standard Fișier, Program, Configurare, Ajutor;
- 2) **Butoanele** Nou, Deschide, Salvare, Ieșire, Analiza sintactică, Execută, Configurare, Stop;
- 3) **Zona de editare** a programelor;
- 4) **Mediul de lucru** al Furnicii, care reprezintă un câmp dreptunghiular, liniat în pătrățele;
- 5) **Centrul de comandă**, care include butoanele SUS, JOS, DREAPTA, STÂNGA, simbolizate prin săgeți;
- 6) **Caseta de inserare** a caracterelor imprimabile;
- 7) **Butoanele** ÎNCEPUT, SFÂRȘIT, SUS, JOS, ..., PROCEDURA, destinate simplificării proceselor de editare a programelor.

În modul de comandă manuală, utilizatorul acționează săgețile care simbolizează comenzile SUS, JOS, DREAPTA, STÂNGA. În modul de comandă prin program, utilizatorul introduce mai întâi instrucțiunile respective în zona de editare. După definitivare, programul poate fi lansat în execuție și/sau salvat pe disc. Pentru exemplificare, în figura 5.2 este prezentat programul care aranjează literele din celulele mediului de lucru al Furnicii în așa mod, încât ele să formeze cuvântul MAMA.

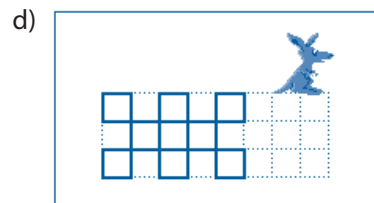
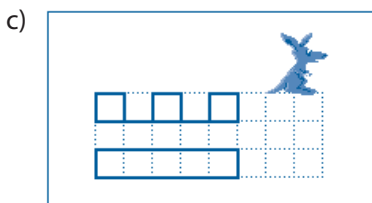
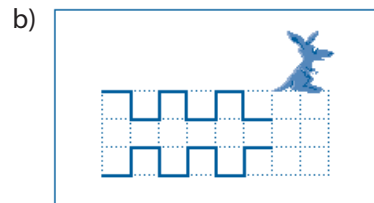
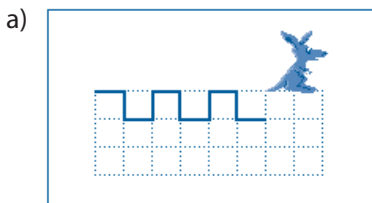
Accentuăm faptul că, indiferent de tipul executantului, programele servesc ca “date inițiale” pentru Centrul de comandă, care parcurge instrucțiunile programului în curs de derulare și generează în baza lor comenzi pentru executant. În paragrafele ce urmează vom vedea că limbajele de programare conțin instrucțiuni speciale, care pot genera sute și mii de comenzi, descriind astfel într-o formă compactă prelucrări foarte complexe.

Întrebări și exerciții

- ❶ Amintiți-vă cel puțin trei algoritmi pe care i-ați studiat la lecțiile de matematică.
- ❷ Pot fi oare rețetele culinare tratate ca algoritmi? Argumentați răspunsul dvs.
- ❸ Încercați să formulați algoritmi destinați adunării, scăderii, înmulțirii și împărțirii numerelor zecimale. Formulați astfel de algoritmi pentru adunarea și scăderea numerelor binare.
- ❹ Încercați să alcătuiți o listă completă a comenzilor pe care le poate executa televizorul dvs.
- ❺ Alcătuiți o listă completă a comenzilor pe care le poate executa un *player* (aparat pentru redarea înregistrărilor).
- ❻ Ce fel de informații trebuie să conțină o descriere completă a unui executant?
- ❼ Descrieți repertoriul de comenzi și mediul de lucru al executantului **Cangurul**.
- ❽ Descrieți repertoriul de comenzi și mediul de lucru al executantului **Furnica**.
- ❾ Care este diferența dintre algoritmi și programe? Argumentați răspunsul dvs.
- ❿ Prin ce se deosebește modul de comandă prin program de modul de comandă manuală?
- ⓫ Lansați în execuție aplicația **Cangurul**. Aflați cu ajutorul sistemului de asistență destinația tuturor elementelor de control din fereastra acestei aplicații.
- ⓬ Lansând comenzile PAS, SALT, ROTIRE, desenați figurile ce urmează:

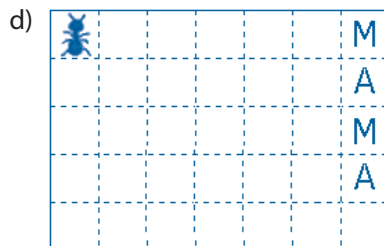
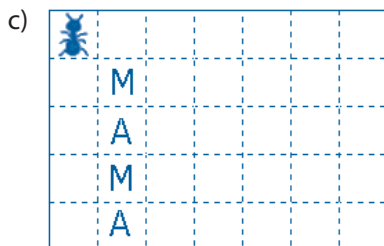
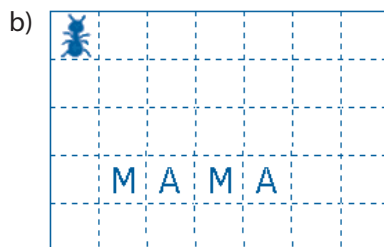
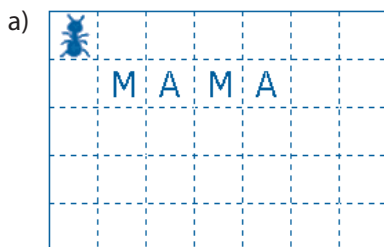


- ⓬ Scrieți programele pentru desenarea figurilor ce urmează:



Salvați aceste programe pe discul magnetic.

- 14) Lansați în execuție aplicația **Furnica**. Aflați cu ajutorul sistemului de asistență destinația tuturor elementelor de control din fereastra acestei aplicații.
- 15) Lansând comenzile SUS, JOS, DREAPTA, STÂNGA, aranjați caracterele din *figura 5.2* conform modelelor ce urmează:



- 16) Scrieți programele destinate aranjării caracterelor din *figura 5.2* conform modelelor prezentate în exercițiul 15. Salvați aceste programe pe discul magnetic.
- 17) Reprezentați pe un desen figura trasată de executantul **Cangur** după execuția programului ce urmează:

1	2	3
ÎNCEPUT PAS PAS ROTIRE PAS	PAS ROTIRE PAS PAS ROTIRE	PAS PAS ROTIRE SFÂRȘIT

- 18) Reprezentați pe un desen amplasarea literelor din *figura 5.2* după execuția programului ce urmează:

1	2	3
ÎNCEPUT JOS DREAPTA DREAPTA STÂNGA STÂNGA JOS	JOS DREAPTA DREAPTA SUS DREAPTA SUS JOS	STÂNGA STÂNGA STÂNGA SUS DREAPTA DREAPTA SFÂRȘIT

- 19) Care este rolul Centrului de comandă în procesul execuției unui program?
- 20) Cine dă comenzi executantului în modul de comandă manuală? Dar în modul de comandă prin program?

5.2. SUBALGORITMI

Termeni-cheie:

- subprogram
- programul principal
- procedură
- apel de procedură
- rafinare succesivă

Ca și în cazul limbajelor folosite de oameni, un limbaj de programare se definește prin sintaxa și semantica lui. E cunoscut faptul că sintaxa este un set de reguli care guvernează alcătuirea propozițiilor, iar semantica este un set de reguli care determină înțelesul, semnificația propozițiilor respective. Evident, în cazul limbajelor de programare echivalentul propoziției este programul.

Sintaxa programelor poate fi descrisă cu ajutorul **formatelor**, care reprezintă modele de utilizare a instrucțiunilor, a simbolurilor și a cuvintelor auxiliare. De exemplu, în cazul executanților **Cangurul** și **Furnica**, programele scrise până în prezent aveau formatul ce urmează:

```
ÎNCEPUT  
Instrucțiunea_1  
Instrucțiunea_2  
...  
Instrucțiunea_k  
SFÂRȘIT
```

unde *Instrucțiunea_k*, $k = 1, 2, 3$ ș.a.m.d. poate fi oricare din comenzile PAS, SALT, ROTIRE în cazul executantului **Cangurul** și SUS, JOS, DREAPTA, STÂNGA în cazul executantului **Furnica**.

Analiza programelor de o reală importanță practică a scos în evidență faptul că foarte multe dintre ele conțin secvențe de instrucțiuni care execută aceleași operații. De exemplu, presupunem că se dorește desenarea a opt pătrate, aranjate conform *figurii 5.3*.

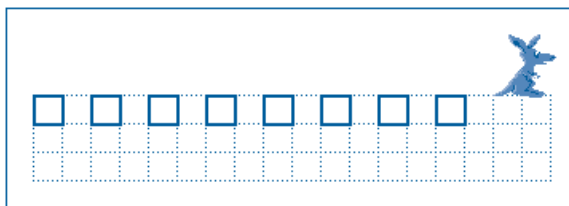


Fig. 5.3. Desenarea pătratelor

Soluționând această problemă prin metoda forței brute, am putea scrie un program în care fiecare pătrat va fi desenat trasându-i-se laturile respective cu ajutorul instrucțiunilor PAS și ROTIRE. Evident, un astfel de program va fi foarte lung, iar scrierea lui va reprezenta o muncă obositoare. O soluție mai elegantă constă în elaborarea unui program

auxiliar pentru desenarea unui pătrat de dimensiunile dorite și folosirea ulterioară a acestui program pentru desenarea unui număr arbitrar de pătrate. În astfel de cazuri, programul auxiliar se numește **subprogram**, iar programul care îl apelează – **program principal** sau, pur și simplu, **program**.

În cazul executanților **Cangurul** și **Furnica**, subprogramele se numesc **proceduri** și au următorul format:

```
PROCEDURA Nume
Instrucțiunea_1
Instrucțiunea_2
...
Instrucțiunea_k
SFÂRȘITUL PROCEDURII
```

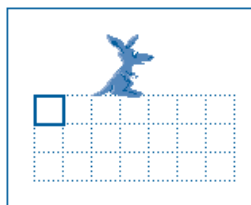
Numele procedurii reprezintă o secvență formată din litere și cifre, care începe, în mod obligatoriu, cu o literă. Cuvântul **PROCEDURA** nu este o instrucțiune, ci un cuvânt auxiliar, care comunică Centrului de comandă numele procedurii și marchează locul de unde începe descrierea ei. Sfârșitul acestei descrieri este indicat de cuvintele auxiliare **SFÂRȘITUL PROCEDURII**. Instrucțiunile încadrate între cuvintele auxiliare **PROCEDURA** și **SFÂRȘITUL PROCEDURII** formează **corpul procedurii**.

Exemplu:

Procedura Pătrat

```
PROCEDURA Pătrat
PAS
ROTIRE
PAS
ROTIRE
PAS
ROTIRE
PAS
ROTIRE
SALT
SALT
SFÂRȘITUL PROCEDURII
```

Figura desenată



Procedura **Pătrat** desenează un pătrat, poziția inițială a Cangurului fiind în colțul stânga-sus. Poziția finală a Cangurului a fost aleasă în așa mod, încât ea coincide cu poziția inițială a pătratului următor (*fig. 5.3*).

Instrucțiunile din componența procedurilor vor fi executate numai atunci când în procesul derulării programului principal se va întâlni instrucțiunea **apel de procedură**. Această instrucțiune are următorul format:

```
EXECUTĂ Nume
```

unde cuvântul *Nume* reprezintă numele procedurii apelate.

Când Centrul de comandă întâlnește un apel de procedură, execuția programului principal se suspendează și începe îndeplinirea instrucțiunilor din componența procedurii apelate. După terminarea acestor instrucțiuni, se reia executarea programului principal. Prin urmare, programul `Opt_pătrate` pentru desenarea pătratelor din *figura 5.3* va include instrucțiunile ce urmează:

```
ÎNCEPUT
EXECUTĂ Pătrat
EXECUTĂ Pătrat
EXECUTĂ Pătrat
EXECUTĂ Pătrat
EXECUTĂ Pătrat
EXECUTĂ Pătrat
EXECUTĂ Pătrat
EXECUTĂ Pătrat
SFÂRȘIT
```

Evident, programul principal trebuie să cunoască descrierea tuturor procedurilor care vor fi apelate. În funcție de tipul executantului, procedurile respective pot fi incluse în programul principal sau înscrise în fișiere separate pe discurile magnetice.

În cazul executanților **Cangur** și **Furnica**, procedurile se includ la începutul programului principal. Pentru acești executanți, formatul general al unui program este:

```
PROCEDURA Nume_1
...
SFÂRȘITUL PROCEDURII
PROCEDURA Nume_2
...
SFÂRȘITUL PROCEDURII
...
ÎNCEPUT
Instrucțiunea_1
Instrucțiunea_2
...
Instrucțiunea_k
SFÂRȘIT
```

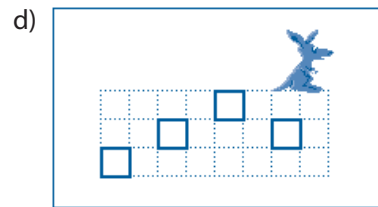
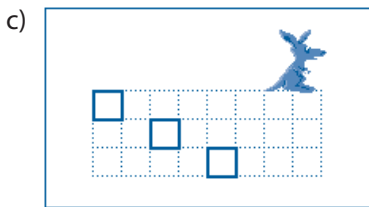
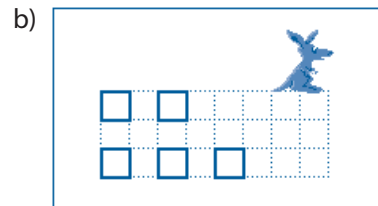
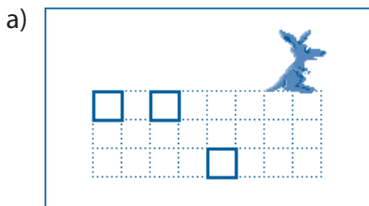
Din formatul prezentat se observă că un program este format din descrierea subprogramelelor *Nume_1*, *Nume_2* ș.a.m.d., după care urmează instrucțiunile programului principal, încadrate între cuvintele auxiliare ÎNCEPUT și SFÂRȘIT. Instrucțiunile respective și cuvintele care le încadrează formează corpul programului. Prin urmare, un program este format din **descrierile de subprograme** și **corpul programului** propriu-zis.

Amintim faptul că programul reprezintă un algoritm scris în limbajul executantului. Evident, clasificarea programelor în subprograme și programe principale se aplică și în cazul algoritmilor. În procesul soluționării unei probleme, utilizatorul elaborează algoritmi și subalgoritmi necesari, scriind programele și subprogramele respective pentru fiecare tip concret de executant.

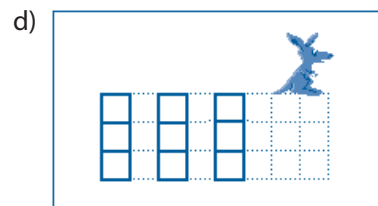
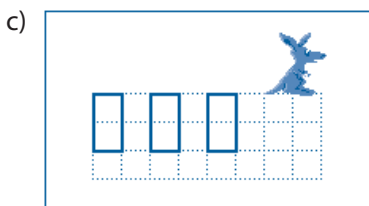
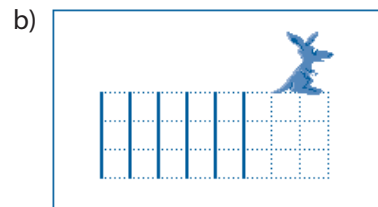
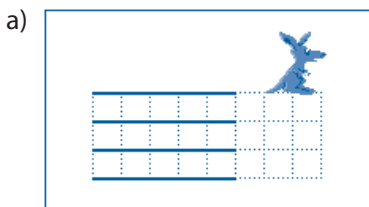
În general, subprogramele reprezintă subalgoritmi, destinați soluționării anumitor probleme întâlnite frecvent în procesul elaborării algoritmului principal. Metoda de soluționare a problemelor complexe prin divizarea lor în probleme mai simple se numește **metoda rafinării succesive**. De exemplu, în cazul *figurii 5.3*, problema inițială – desenarea celor opt pătrate – a fost divizată în opt subprobleme identice – desenarea unui singur pătrat.

Întrebări și exerciții

- ❶ Când apare necesitatea utilizării subalgoritmilor? Dați exemple.
- ❷ Care este diferența dintre un algoritm și un subalgoritm?
- ❸ Care este formatul procedurilor scrise în limbajul executanților **Cangurul** și **Furnica**? Cum se apelează aceste proceduri?
- ❹ Utilizând procedura **Pătrat**, elaborați programele necesare pentru desenarea figurilor ce urmează:



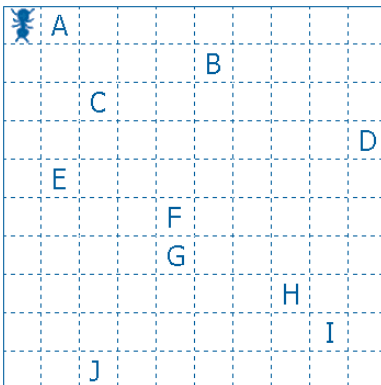
- ❺ Identificați în figurile ce urmează fragmentele care se repetă. Scrieți procedurile necesare pentru desenarea acestor fragmente.



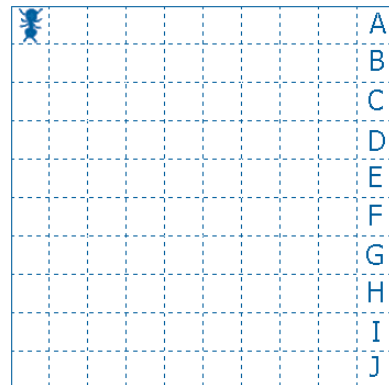
Utilizând procedurile elaborate, scrieți programele necesare pentru desenarea figurilor prezentate mai sus.

- ⑥ Aplicând metoda rafinării succesive, scrieți programele pentru desenarea figurilor c și d din exercițiul 12 al paragrafului 5.1.
- ⑦ Scrieți în limbajul executantului **Furnica** următoarele proceduri:
 Stânga_5 – deplasarea Furnicii în stânga cu cinci poziții;
 Dreapta_5 – deplasarea Furnicii în dreapta cu cinci poziții;
 Sus_5 – deplasarea Furnicii în sus cu cinci poziții;
 Jos_5 – deplasarea Furnicii în jos cu cinci poziții.
 Utilizând aceste proceduri, elaborați un program care deplasează Furnica în așa mod, încât traiectoria mișcării să reprezinte un pătrat cu laturile: a) 5×5 ; b) 6×6 ; c) 8×8 .
- ⑧ În figurile ce urmează este prezentat aspectul inițial și aspectul final al câmpului de lucru al executantului **Furnica**.

Aspectul inițial



Aspectul final



Scrieți o procedură care mută caracterul imprimabil dintr-un anumit rând în ultima celulă a rândului. Scrieți un program care mută toate caracterele imprimabile în ultima coloană a câmpului.

5.3. ALGORITMI REPETITIVI. CICLU CU CONTOR

Termeni-cheie:

- schemă logică
- algoritm liniar
- instrucțiune compusă
- instrucțiune simplă
- ciclu cu contor
- algoritm repetitiv

Pentru o reprezentare mai sugestivă, algoritmi pot fi descriși cu ajutorul schemelor logice. **Schema logică** reprezintă un desen, care conține următoarele simboluri grafice:

- START – punctul de pornire a procesului de execuție a algoritmului;
- STOP – punctul de oprire a procesului de execuție a algoritmului;
- Instrucțiune* – execuția unei instrucțiuni;
- Nume* – apelul unui subalgoritm;

→ – linia orientată, care indică ordinea în care trebuie executate instrucțiunile algoritmului.

Pentru exemplificare, în *figura 5.4* este prezentată schema logică a procedurii Pătrat și schema logică a programului Opt_pătrate.

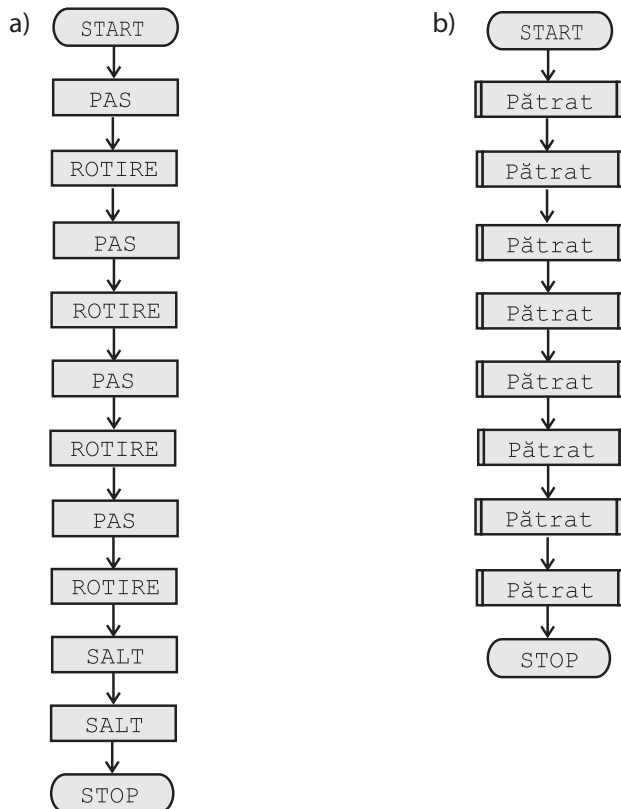


Fig. 5.4. Scheme logice:
 a – procedura Pătrat; b – programul Opt_pătrate

Din analiza schemelor logice din *figura 5.4* se observă că procesul de execuție a unui algoritm poate fi simbolizat printr-o deplasare imaginară dintr-un simbol grafic în altul în direcția indicată de liniile respective.

Algoritmii instrucțiunile cărora sunt executate în ordinea apariției lor în text se numesc algoritmi liniari.

Evident, în cazul algoritmilor liniari drumul imaginar parcurs de la simbolul grafic START până la simbolul grafic STOP reprezintă o linie ce nu se autointersectează (fig. 5.4).

În procesul elaborării algoritmilor s-a observat că unele secvențe de instrucțiuni deseori trebuie executate de mai multe ori. De exemplu, în cazul procedurii Pătrat (fig. 5.4a), secvența de instrucțiuni PAS, ROTIRE se execută de patru ori, iar instrucțiunea apel de procedură din programul Opt_pătrate – de opt ori. Pentru a simplifica procesele de elaborare a algoritmilor, în astfel de cazuri se poate utiliza instrucțiunea REPETĂ. Formatul și schema logică a acestei instrucțiuni sunt date în figura 5.5.

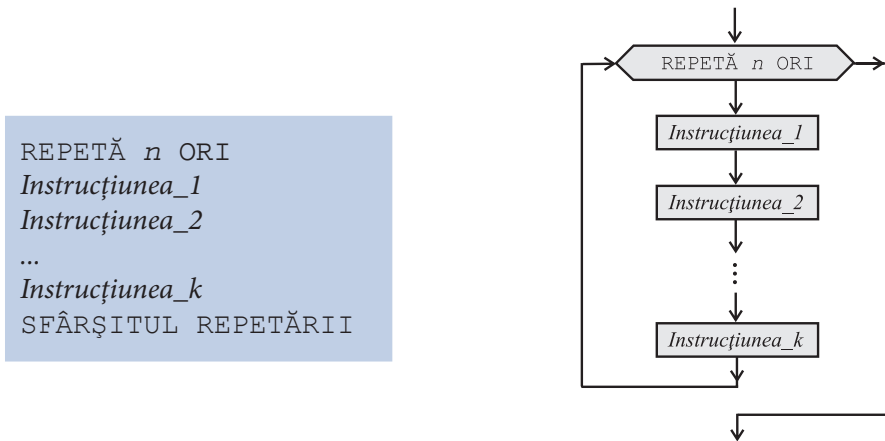


Fig. 5.5. Formatul și schema logică a instrucțiunii REPETĂ

În descrierea instrucțiunii REPETĂ n reprezintă numărul dorit de repetări, iar cuvintele REPETĂ, ORI, SFÂRȘITUL REPETĂRII sunt cuvinte auxiliare.

Instrucțiunea REPETĂ se notează pe câteva linii și include în componența sa alte instrucțiuni. Instrucțiunile de felul acesta se numesc **instrucțiuni compuse**, spre deosebire de **instrucțiunile simple** PAS, SALT, ROTIRE, SUS, JOS, DREAPTA, STÂNGA, apel de procedură, studiate în paragrafele precedente.

În procesul execuției instrucțiunii REPETĂ, Centrul de comandă va îndeplini de n ori secvența de instrucțiuni încadrată între cuvintele auxiliare. Utilizând această instrucțiune, putem transcrie programele ce conțin secvențe de instrucțiuni, care trebuie executate de mai multe ori, într-o formă mai compactă. De exemplu, programul Opt_pătrate poate fi transcris în forma:

1

```
PROCEDURA Pătrat
REPETĂ 4 ORI
PAS
ROTIRE
SFÂRȘITUL REPETĂRII
SALT
SALT
SFÂRȘITUL PROCEDURII
```

2

```
ÎNCEPUT
REPETĂ 8 ORI
EXECUTĂ Pătrat
SFÂRȘITUL REPETĂRII
SFÂRȘIT
```

Instrucțiunea REPETĂ n ORI se numește **ciclu cu contor**, deoarece la execuția ei se repetă ciclic aceeași secvență de instrucțiuni, iar numărul de repetări n este cunoscut în momentul scrierii programului. Secvența de instrucțiuni, încadrată între liniile ce conțin cuvintele auxiliare REPETĂ și SFÂRȘITUL REPETĂRII, se numește **corpul ciclului**.

Algoritmii ce conțin secvențe de instrucțiuni care în procesul execuției se îndeplinesc de mai multe ori se numesc algoritmi repetitivi.

Schemele logice, care reprezintă în mod grafic procesele de execuție a procedurii Pătrat și a programului Opt_pătrate, sunt prezentate în figura 5.6. În aceste scheme se utilizează simbolul grafic REPETĂ, din care, spre deosebire de simbolurile grafice studiate anterior, pleacă două linii orientate: prima spre instrucțiunile din corpul ciclului, iar a doua – spre instrucțiunea care va fi executată imediat după terminarea ciclului.

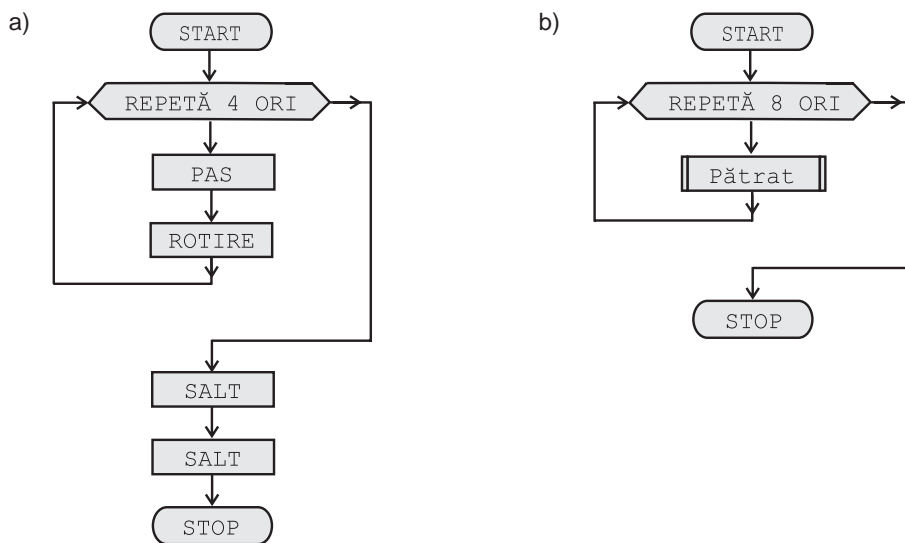


Fig. 5.6. Schemele logice ale algoritmilor repetitivi:
a – procedura Pătrat; b – programul Opt_pătrate

Din analiza schemelor logice ale algoritmilor repetitivi se observă că drumul imaginat parcurs de la simbolul grafic START până la simbolul grafic STOP reprezintă o linie ce conține cel puțin o buclă. Această buclă include simbolul grafic REPETĂ și toate simbolurile grafice ce corespund instrucțiunilor din corpul ciclului (fig. 5.6).

Fiind o instrucțiune compusă, instrucțiunea REPETĂ poate include în corpul său alte instrucțiuni de acest tip, formându-se astfel o structură imbricată. În consecință, programele ce conțin un număr relativ mic de instrucțiuni pot descrie succesiuni foarte lungi de acțiuni. Pentru exemplificare, prezentăm în continuare un program care impune Cangurul să se deplaseze de 100 de ori de-a lungul marginii de sus a zonei de desenare:

```

ÎNCEPUT
REPETĂ 100 ORI
REPETĂ 15 ORI
SALT
  
```

SFÂRȘITUL REPETĂRII
 ROTIRE
 ROTIRE
 SFÂRȘITUL REPETĂRII
 SFÂRȘIT

Întrebări și exerciții

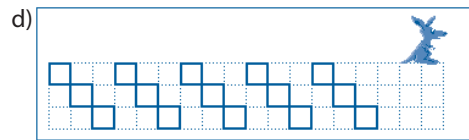
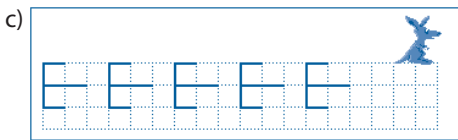
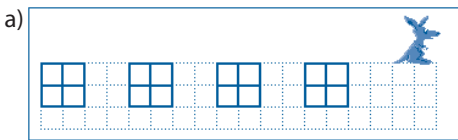
1 Explicați semnificația următoarelor simboluri grafice:



f) →

Care este semnificația liniilor orientate din componența schemelor logice?

- 2 Desenați schemele logice ale algoritmilor elaborați pentru desenarea figurilor din exercițiile 4, 5 și 8 din paragraful 5.2.
- 3 Indicați pe schemele logice din figura 5.4 drumul imaginar ce simbolizează procesul de execuție a algoritmilor respectivi.
- 4 Explicați termenul *algoritm liniar*. Dați exemple.
- 5 Care este formatul instrucțiunii REPETĂ. Dați câteva exemple de scriere a acestei instrucțiuni.
- 6 Când este rezonabilă folosirea instrucțiunii REPETĂ? Dați exemple.
- 7 Explicați termenul *algoritm repetitiv*. Dați exemple.
- 8 Indicați pe schemele logice din figura 5.6 drumul imaginar ce simbolizează procesul de execuție a algoritmilor respectivi.
- 9 Indicați pe schemele logice din figura 5.6 buclele care simbolizează ciclurile cu contor.
- 10 Enumerați instrucțiunile simple pe care le cunoașteți. Prin ce se deosebește o instrucțiune simplă de o instrucțiune compusă?
- 11 Elaborați algoritmi repetitivi pentru desenarea figurilor ce urmează:



Calculați câte comenzi va executa Cangurul în procesul desenării acestor figuri.

12 Câte comenzi va executa Cangurul în procesul derulării următoarelor programe:

a) ÎNCEPUT
 REPETĂ 10 ORI
 SALT
 ROTIRE

b) ÎNCEPUT
 REPETĂ 10 ORI
 REPETĂ 20 ORI
 SALT

```

ROTIRE
SALT
SFÂRȘITUL REPETĂRII
SFÂRȘIT

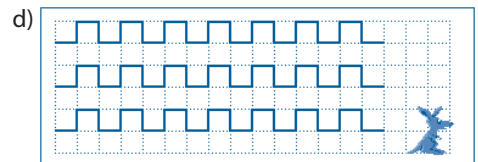
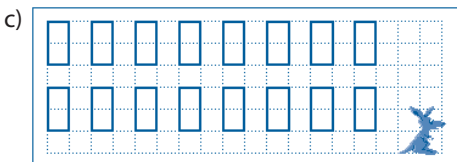
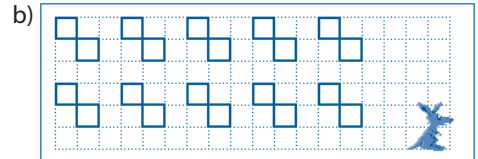
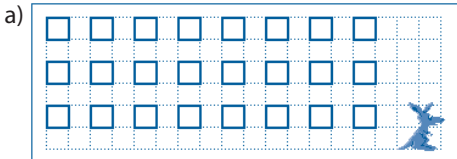
```

```

ROTIRE
ROTIRE
SALT
SFÂRȘITUL REPETĂRII
SFÂRȘITUL REPETĂRII
SFÂRȘIT

```

- 13 Utilizând ciclurile imbricate, elaborați algoritmi repetitivi pentru desenarea figurilor ce urmează:



Calculați câte comenzi va executa Cangurul în procesul desenării acestor figuri.

- 14 Scrieți în limbajul executantului **Furnica** următorii subalgoritmi repetitivi:

Stânga_8 – deplasarea Furnicii în stânga cu opt poziții;

Dreapta_8 – deplasarea Furnicii în dreapta cu opt poziții;

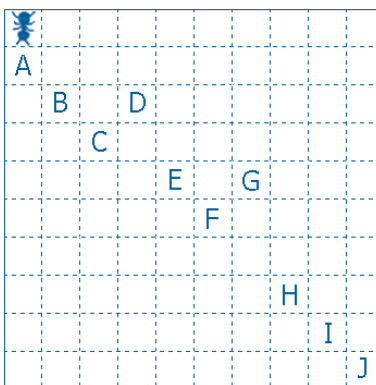
Sus_8 – deplasarea Furnicii în sus cu opt poziții;

Jos_8 – deplasarea Furnicii în jos cu opt poziții.

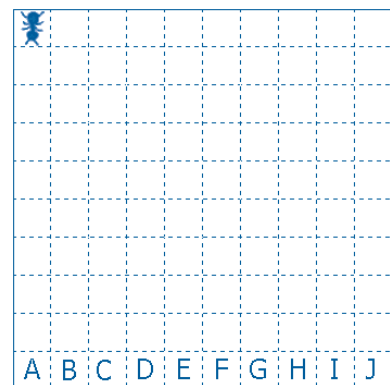
Utilizând acești subalgoritmi, elaborați un program care deplasează Furnica în așa mod, încât traiectoria mișcării să reprezinte un pătrat cu laturile: a) 8×8 ; b) 9×9 ; c) 10×10 .

- 15 În figurile ce urmează este prezentat aspectul inițial și aspectul final al câmpului de lucru al executantului **Furnica**. Utilizând ciclurile cu contor, scrieți o procedură care mută caracterul imprimabil dintr-o anumită coloană în ultima celulă a coloanei.

Aspectul inițial



Aspectul final



Utilizând ciclurile cu contor, scrieți un program care mută toate caracterele imprimabile în ultimul rând al câmpului.

5.4. ALGORITMI REPETITIVI. CICLU CU CONDIȚIE

Termeni-cheie:

- condiție
- ciclul cu condiție
- algoritm cu conexiune inversă
- eroare de execuție

Până în prezent, toți algoritmi elaborați de noi pentru comanda executanților **Cangurul** și **Furnica** nu analizau situația din mediul de lucru al executanților. Cu alte cuvinte, Centrul de comandă dirija acțiunile executanților fără să verifice dacă comenzile respective garantează realizarea scopului propus sau dacă executanții sunt în stare să îndeplinească comenzile primite. Acest mod de elaborare a algoritmilor complică esențial soluționarea multor probleme întâlnite frecvent în practica cotidiană.

De exemplu, presupunem că poziția inițială a Furnicii este necunoscută. Se cere să elaborăm un program care deplasează Furnica în colțul stânga-sus al mediului de lucru. Evident, instrucțiunile studiate până acum nu sunt suficiente pentru a soluționa această problemă, întrucât nu cunoaștem câte comenzi de tipul SUS, STÂNGA trebuie să fie date executantului.

Pentru a evita astfel de situații, executanții sunt dotați cu **senzori***, care comunică Centrului de comandă anumite informații din mediul de lucru. De exemplu, executanții **Cangurul** și **Furnica** sunt dotați cu senzori, care descoperă prezența obstacolelor în direcția unor eventuale deplasări. Informațiile respective sunt transmise Centrului de comandă prin intermediul variabilelor logice E_LINIE și E_MARGINE, care pot lua valorile FALS sau ADEVĂRAT.

În cazul algoritmilor repetitivi, analiza anumitor situații din mediul de lucru se realizează cu ajutorul instrucțiunii compuse CÂT. Această instrucțiune asigură execuția ciclică a unei secvențe de instrucțiuni atâta timp cât în mediul de lucru se respectă anumite condiții. Formatul și schema logică a instrucțiunii CÂT sunt date în figura 5.7.

```
CÂT Condiție  
Instrucțiunea_1  
Instrucțiunea_2  
...  
Instrucțiunea_k  
SFÂRȘITUL CICLULUI
```

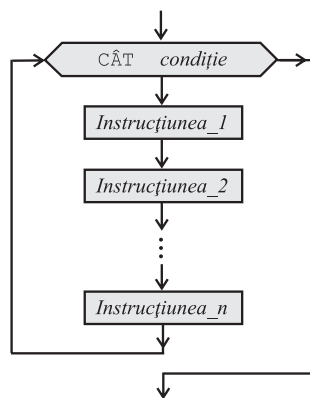


Fig. 5.7. Formatul și schema logică a instrucțiunii CÂT

* Sensor – dispozitiv care sesizează (observă, descoperă) un anumit fenomen.

În descrierea instrucțiunii CÂT *Condiție* este o expresie logică, iar cuvintele CÂT și SFÂRȘITUL CICLULUI sunt cuvinte auxiliare.

Condițiile reprezintă expresii logice care indică anumite situații din mediul de lucru al executantului.

În limbajele de programare a executanților **Cangurul** și **Furnica**, expresiile logice utilizate pentru a indica anumite situații din mediul de lucru sunt foarte simple:

Cangurul

```
E_LINIE
E_MARGINE
NU E_LINIE
NU E_MARGINE
```

Furnica

```
E_MARGINE
NU E_MARGINE
```

În cazul unor executanți mai sofisticăți, în componența condițiilor pot intra variabile care comunică Centrului de comandă informații despre direcția și viteza mișcării altor obiecte, prezența obstacolelor, temperaturii, umidității, nivelului de radiație etc., iar expresiile logice pot fi formate utilizând operațiile relaționale =, ≠, >, ≥, <, ≤ și operațiile logice NU, ȘI, SAU.

Pentru exemplificare, vom examina următoarea problemă. Presupunem că se dorește deplasarea Cangurului spre una dintre marginile câmpului de lucru, însă poziția inițială a acestuia nu este cunoscută. Evident, în astfel de cazuri utilizatorul nu poate folosi instrucțiunea REPETĂ, întrucât nu se cunoaște numărul de pași pe care trebuie să-i facă Cangurul. Cu ajutorul instrucțiunii CÂT, astfel de probleme se soluționează foarte ușor:

```
CÂT NU E_MARGINE
SALT
SFÂRȘITUL CICLULUI
```

În procesul execuției instrucțiunii CÂT, Centrul de comandă va analiza mai întâi condiția NU E_MARGINE. Dacă această condiție are valoarea ADEVĂRAT, se execută instrucțiunile din corpul ciclului, în cazul exemplului de mai sus – instrucțiunea SALT. După îndeplinirea instrucțiunii SALT, senzorul din dotarea Cangurului explorează pătrățelul vecin și actualizează valoarea variabilei E_MARGINE. Dacă condiția NU E_MARGINE ia valoarea FALS, execuția ciclului se termină și se trece la instrucțiunea ce urmează imediat după cuvintele auxiliare SFÂRȘITUL CICLULUI.

Instrucțiunea CÂT se numește **ciclu cu condiție**, deoarece la execuția ei se repetă ciclic aceeași secvență de instrucțiuni, iar numărul de repetări se stabilește în procesul derulării programului în funcție de valorile curente ale condiției respective.

Implementarea instrucțiunii CÂT presupune existența dintre Centrul de comandă și executant a două canale de transmitere a informației:

- 1) canalul direct, destinat transmiterii comenzilor de la Centrul de comandă la executant;
- 2) canalul invers, destinat transmiterii informațiilor colectate cu ajutorul senzorilor de la executant la Centrul de comandă.

Algoritmii ce conțin secvențe de instrucțiuni, execuția cărora depinde de informațiile colectate în mediul de lucru al executanților, se numesc algoritmi cu conexiune inversă.

Majoritatea algoritmilor utilizați în informatica modernă sunt algoritmi cu conexiune inversă, fapt ce permite executanților să se adapteze la situațiile concrete din mediul de lucru. Pentru exemplificare, prezentăm în continuare un program foarte mic, care desenează o figură destul de complicată – o spirală (fig. 5.8):

```
ÎNCEPUT  
CÂT NU E_LINIE  
CÂT NU E_LINIE  
PAS  
SFÂRȘITUL CICLULUI  
ROTIRE  
SFÂRȘITUL CICLULUI  
SFÂRȘIT
```

Ciclul interior, corpul căruia conține instrucțiunea PAS, asigură desenarea unui segment de dreaptă, lungimea căruia depinde de configurația liniilor trasate anterior. Imediat cum sensorul Cangurului depistează în pătrățelul vecin o linie, variabila NU_E_LINIE obține valoarea FALS și execuția acestui ciclu se termină. Instrucțiunea imediat următoare rotește Cangurul cu 90°, pregătindu-l astfel pentru trasarea unei linii noi. Ciclul exterior asigură trasarea liniilor ce formează spirala până când se ajunge la situația în care toate pătrățelele vecine sunt deja ocupate.

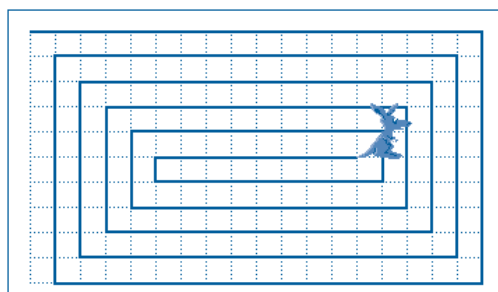


Fig. 5.8. Spirala desenată de Cangur

Teoretic, o spirală ar putea fi trasată și fără utilizarea instrucțiunii CÂT, folosind pentru organizarea ciclurilor instrucțiunea REPETĂ. Însă în acest caz, utilizatorul va fi nevoit să calculeze în mod manual lungimea fiecărei linii din componența spiralei și să scrie câte patru cicluri cu contor pentru fiecare spirală. Evident, în cazul unui număr mare de spire, acest lucru, practic, este imposibil.

Din practică s-a stabilit că în procesul elaborării și depanării programelor apar situații când executantul primește comenzi pe care nu le poate îndeplini. De exemplu, programul

```
ÎNCEPUT  
REPETĂ 1000 ORI  
PAS  
SFÂRȘITUL REPETĂRII  
SFÂRȘIT
```

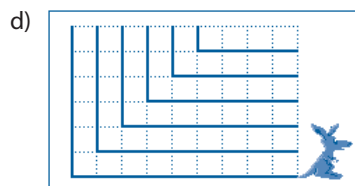
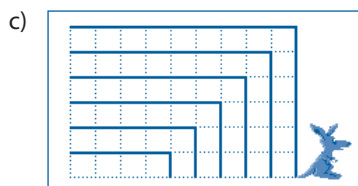
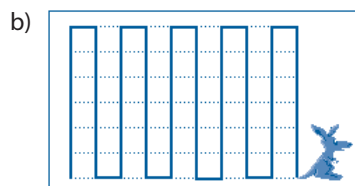
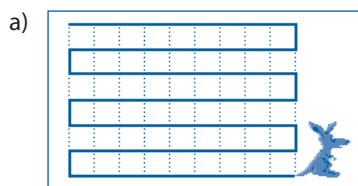
este corect din punct de vedere sintactic, însă derularea lui completă este imposibilă, întrucât dimensiunile câmpului de lucru sunt mult mai mici de 1000×1000 . După lansarea în execuție a acestui program, indiferent de poziția inițială, se va ajunge în situația în care comanda PAS nu mai poate fi îndeplinită, întrucât Cangurul se află deja la marginea câmpului. Prin urmare, în procesul derulării programului de mai sus apare o eroare de execuție sau un refuz.

Eroarea de execuție (refuzul) apare atunci când în procesul derulării unui program executantul nu poate îndeplini comanda primită.

Reacția Centrului de comandă la o eroare de execuție depinde de tipul executantului și posibilitățile programului respectiv. De exemplu, executanții **Cangurul** și **Furnica** afișează un mesaj de eroare și opresc derularea programului. În cazul unor executanți mai sofisticăți, refuzul este tratat ca o condiție specială, care declanșează execuția unor subalgoritmi de corectare a erorilor apărute. Pentru exemplificare, amintim sistemele de asistență a editoarelor de texte și a aplicațiilor de calcul tabelar, care, în cazul unei erori, afișează pe ecran sugestii ce îl ajută pe utilizator să lanseze comenzile respective corecte.

Întrebări și exerciții

- ❶ Care este destinația senzorilor cu care sunt dotați executanții?
- ❷ Pentru ce se utilizează condițiile? Cine atribuie valori acestor variabile?
- ❸ Reprezentați pe un desen fluxul de informații dintre Centrul de comandă și executant. Care este destinația canalului direct și a canalului invers?
- ❹ Prin ce se deosebesc algoritmi cu conexiune inversă de algoritmi fără astfel de conexiuni?
- ❺ Ar putea oare calculatorul să conducă un automobil fără ca să utilizeze algoritmi cu conexiune inversă? Argumentați răspunsul dvs.
- ❻ Desenați schema logică a programului pentru desenarea unei spirale. Indicați pe această schemă drumul imaginar ce simbolizează procesul de execuție a programului și buclele care corespund ciclurilor cu condiție.
- ❼ Încercați să elaborați un program care ar desena o spirală fără utilizarea ciclului cu condiție. Estimați numărul de instrucțiuni ale unui astfel de program.
- ❽ Utilizând ciclul cu condiție, elaborați programe pentru desenarea figurilor ce urmează:



Desenați schemele logice ale programelor elaborate. Indicați pe fiecare schemă drumul imaginar ce simbolizează procesul de execuție a programului și buclele care corespund ciclurilor cu condiție.

- 9 Elaborati un program care, fără să cunoască poziția inițială, deplasează Furnica în una dintre următoarele poziții:
- a) colțul stânga-sus;
 - b) colțul dreapta-sus;
 - c) colțul stânga-jos;
 - d) colțul dreapta-jos.
- Se consideră că toate pătrățelele mediului de lucru sunt libere.
- 10 Elaborati un program care deplasează neconținut Furnica în așa mod, încât traiectoria mișcării ei să reprezinte un pătrat, laturile căruia coincid cu marginile mediului de lucru. Inițial Furnica se află în colțul stânga-sus al mediului de lucru.

5.5. ALGORITMI CU RAMIFICĂRI

Termeni-cheie:

- ramificator
- algoritm cu ramificări

Foarte des, în procesul execuției unui program, apare necesitatea de a da anumite comenzi executantului în funcție de situația din mediul de lucru. Cunoaștem deja că informațiile referitoare la situația din mediul de lucru sunt colectate cu ajutorul senzorilor, datele respective fiind reprezentate în program prin variabile de genul `E_LINIE`, `E_MARGINE` etc. În paragraful precedent am folosit astfel de variabile pentru a scrie condițiile instrucțiunilor compuse CÂT.

În general, condițiile care descriu situația din mediul de lucru pot fi analizate și cu ajutorul altei instrucțiuni compuse – instrucțiunea DACĂ. Formatul și schema logică a instrucțiunii DACĂ sunt date în *figura 5.9*.

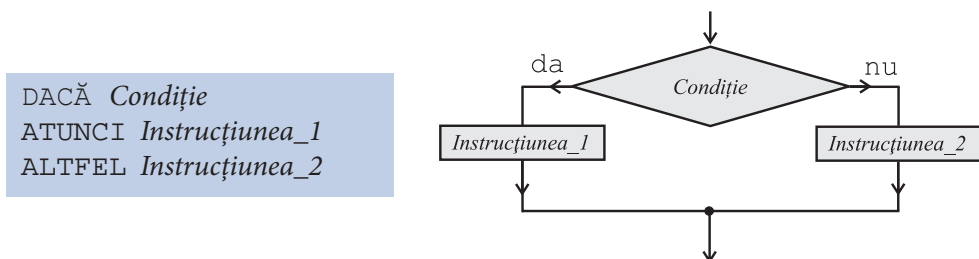


Fig. 5.9. Formatul și schema logică a instrucțiunii DACĂ

În descrierea instrucțiunii DACĂ *Condiție* este o expresie logică, iar cuvintele DACĂ, ATUNCI și ALTFEL sunt cuvinte auxiliare.

Ca și în cazul instrucțiunii CĂT, în limbajele de programare a executanților **Cangurul** și **Furnica** putem utiliza următoarele expresii logice:

```
E_LINIE
E_MARGINE
NU E_LINIE
NU E_MARGINE
```

Evident, în cazul unor executanți mai sofisticati, în componența condițiilor pot intra și alte variabile, iar expresiile logice pot fi formate utilizând operațiile relaționale =, ≠, >, ≥, <, ≤ și operațiile logice NU, ȘI, SAU.

În procesul execuției instrucțiunii DACĂ, Centrul de comandă va analiza mai întâi condiția respectivă. Dacă această condiție are valoarea ADEVĂRAT, se execută *Instrucțiunea_1*, iar în caz contrar (condiția are valoarea FALS), se execută *Instrucțiunea_2*.

Instrucțiunea compusă DACĂ se numește **ramificator**, deoarece drumul imaginar, care simbolizează procesul de execuție, va trece, în funcție de valorile curente ale condiției analizate, prin simbolul grafic *Instrucțiunea_1* sau prin simbolul grafic *Instrucțiunea_2*, rombul reprezentând punctul de ramificare.

Algoritmii ce conțin secvențe de instrucțiuni care vor fi executate numai pentru anumite valori ale condițiilor indicate se numesc algoritmi cu ramificări.

Pentru exemplificare, prezentăm un program care desenează un pătrat, laturile căruia coincid cu marginile câmpului de lucru:

```
ÎNCEPUT
REPETĂ 100 ORI
DACĂ E_MARGINE
ATUNCI ROTIRE
ALTFEL PAS
SFÂRȘITUL REPETĂRII
SFÂRȘIT
```

Corpul ciclului REPETĂ al acestui program conține instrucțiunea compusă DACĂ, care va fi executată de 100 de ori. La fiecare execuție se verifică condiția E_MARGINE și, în funcție de valoarea curentă, se execută instrucțiunea ROTIRE sau instrucțiunea PAS. Evident, instrucțiunea ROTIRE se va executa numai atunci când valoarea condiției va fi ADEVĂRAT sau, cu alte cuvinte, când Cangurul va ajunge la marginea câmpului de lucru.

Schema logică a programului dat este prezentată în *figura 5.10*. Din această schemă se observă că la fiecare execuție repetată a instrucțiunii DACĂ, alegerea uneia dintre instrucțiunile ROTIRE sau PAS se va efectua în funcție de poziția curentă a Cangurului față de marginea câmpului de lucru.

Întrebări și exerciții

- ❶ Care este destinația instrucțiunii DACĂ? Scrieți formatul și desenați schema logică a acestei instrucțiuni.

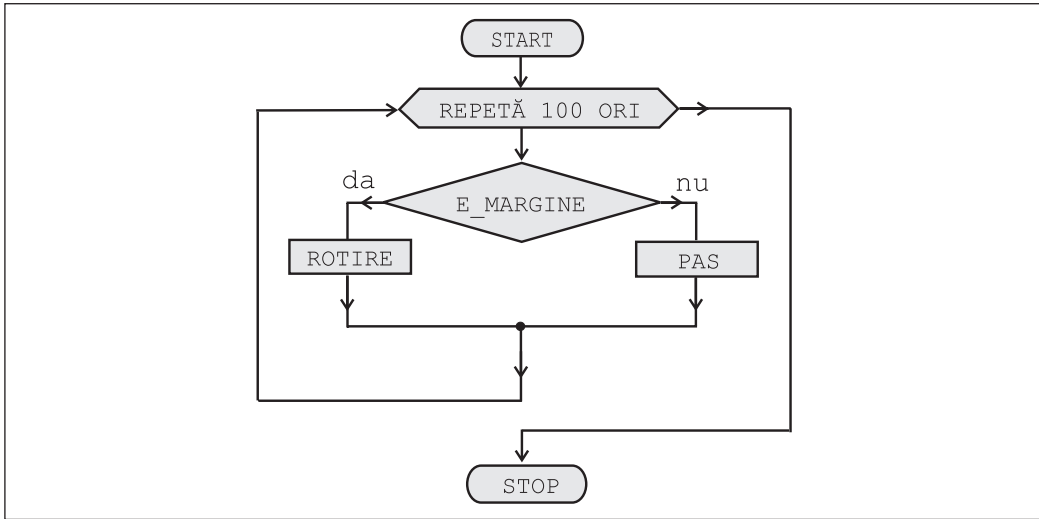


Fig. 5.10. Schema logică a algoritmului cu ramificații

- 2 Cum se execută instrucțiunea DACĂ? Poate fi oare înlocuită instrucțiunea DACĂ cu instrucțiunea CĂT?
- 3 Este oare, în opinia dvs., algoritmul cu ramificații un algoritm cu conexiune inversă? Argumentați răspunsul.
- 4 Numiți cel puțin trei probleme din matematică, algoritmi de soluționare a cărora utilizează ramificatori.
- 5 Care este diferența dintre algoritmi liniari și algoritmi cu ramificații?
- 6 Care este diferența dintre algoritmi repetitivi și algoritmi cu ramificații?
- 7 Elaborați două programe care desenează o spirală în interiorul unui dreptunghi, dimensiunile cărui nu sunt cunoscute (fig. 5.11). Pentru a analiza situația de pe câmpul de lucru, în primul program se va utiliza instrucțiunea DACĂ, iar în al doilea – instrucțiunea CĂT. Care program este mai eficient, cel ce utilizează instrucțiunea DACĂ sau cel ce utilizează instrucțiunea CĂT? Argumentați răspunsul dvs.

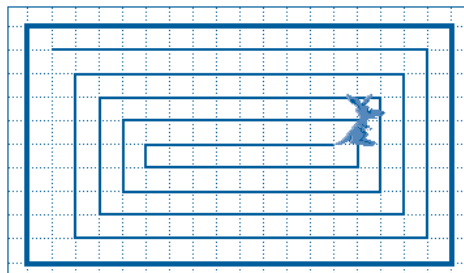


Fig. 5.11. Spirala desenată în interiorul unui dreptunghi

- 8 Desenați schemele logice ale programelor destinate trasării spiralei în interiorul unui dreptunghi (fig. 5.11). Indicați pe aceste scheme drumurile imaginare ce simbolizează procesele de execuție și punctul de ramificare.
- 9 Cangurul se află la începutul unui coridor, lățimea cărui este de două pătrățele (fig. 5.12). Elaborați un program care desenează o linie ce va trece exact prin mijlocul coridorului.

Se admite că la momentul scrierii programului, dimensiunile coridorului, cu excepția lățimii, nu sunt cunoscute.

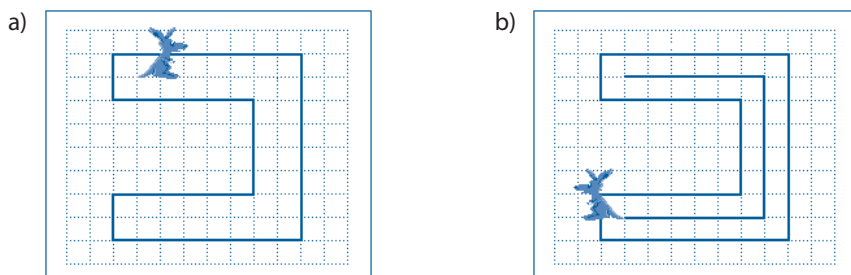


Fig. 5.12. Trasarea liniilor în interiorul unui coridor:
a – poziția inițială; b – poziția finală

5.6. ALGORITMUL DE FUNCȚIONARE A CALCULATORULUI

Termeni-cheie:

- unități funcționale
- dispozitivul central de comandă
- dispozitivul aritmetic și logic
- comenzi utilizate frecvent
- algoritmul de funcționare

Un calculator numeric conține următoarele **unități funcționale** (fig. 5.13): procesorul, memoria internă, memoria externă, dispozitivul de intrare și dispozitivul de ieșire. La rândul său, procesorul este format din **dispozitivul central de comandă** și **dispozitivul aritmetic și logic**.

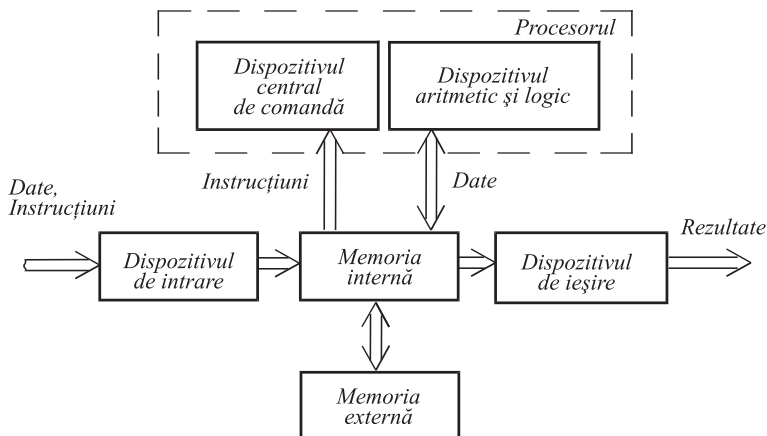


Fig. 5.13. Schema funcțională a calculatorului

Cu excepția dispozitivului central de comandă, toate unitățile funcționale ale calculatorului pot fi tratate ca executanți ce îndeplinesc anumite comenzi. Pentru exemplificare, în *tabelul 5.1* sunt prezentate **comenzile frecvent utilizate** ale executanților din componența unui calculator personal.

Tabelul 5.1

**Comenzile frecvent utilizate ale executanților
din componența unui calculator personal**

Nr. crt.	Executantul	Comenzile executantului
1.	Tastatura	Așteaptă acționarea unei taste Citește codul tastei apăsată Blochează tastatura
2.	Monitorul	Șterge ecranul Afișează caracterul indicat Afișează elementul grafic indicat Stabilește culoarea de fundal
3.	Unitatea de disc magnetic	Înscrie datele pe disc Citește datele de pe disc
4.	Imprimanta	Tipărește caracterul indicat Avans de linie Avans de pagină
5.	Unitatea de disc optic	Citește datele de pe disc Extrage discul
6.	Memoria internă	Citește datele din locația indicată Înscrie datele în locația indicată
7.	Dispozitivul aritmetic și logic	Adună Scade Înmulțește Împarte Compară ȘI SAU NU

Dispozitivul central de comandă asigură dirijarea executanților conform programului înscris în memoria internă a calculatorului. Programul constă dintr-un set de instrucțiuni, codificate în formă de cuvinte binare, în care se indică operația ce trebuie executată și amplasamentul (locul) operanzilor. De exemplu, într-o instrucțiune aritmetică se indică operația ce trebuie efectuată (adunarea, scăderea, înmulțirea sau împărțirea) și amplasamentul operanzilor în memoria internă. Într-o instrucțiune de introducere a informației se indică dispozitivul de intrare (tastatura, cititorul de documente) și locul din memoria internă unde va fi stocată informația introdusă. Într-o instrucțiune de extragere a informației se indică locul din memoria internă ce conține informația respectivă și dispozitivul de ieșire (monitorul sau imprimanta). În mod similar, în cazul citirii sau scrierii informației

pe un disc magnetic, în instrucțiunea respectivă se indică amplasamentul informației în memoria internă și unitatea de disc.

Algoritmul de funcționare a dispozitivului central de comandă și, implicit, a calculatorului în ansamblu, poate fi descris în felul următor:

```
CÂT INSTRUCȚIUNEA EXTRASĂ DIN MEMORIA INTERNĂ ≠ STOP  
EXTRAGE O INSTRUCȚIUNE DIN MEMORIA INTERNĂ  
DECODIFICĂ INSTRUCȚIUNEA EXTRASĂ  
TRANSMITE COMENZI EXECUTANȚILOR  
SFÂRȘITUL CICLULUI
```

Din algoritmul prezentat observăm că dispozitivul central de comandă realizează principiul de comandă prin program, transmițând executanților din componența calculatorului comenzi, generate în baza instrucțiunilor, extrase din memoria internă. Accentuăm faptul că repertoriul de instrucțiuni al unui calculator modern include atât instrucțiuni pentru prelucrarea informației (adunarea, scăderea, înmulțirea, împărțirea etc.), cât și instrucțiuni pentru apelurile de subprograme, realizarea algoritmilor repetitivi și a algoritmilor cu ramificații. Un astfel de repertoriu permite descrierea compactă a unor prelucrări foarte complexe, care, împreună cu vitezele de operare foarte mari (milioane de instrucțiuni pe secundă), asigură aplicarea eficientă a calculatoarelor în toate domeniile științei și tehnicii moderne.

Întrebări și exerciții

- 1 Determinați repertoriul de comenzi al fiecărei unități periferice (tastatura, imprimanta, monitorul, cititorul de documente etc.) din laboratorul de informatică.
- 2 Utilizând descrierile tehnice ale calculatoarelor din laboratorul de informatică, determinați repertoriul de instrucțiuni al calculatorului la care lucrați dvs. Indicați instrucțiunile de prelucrare a informației, instrucțiunile de intrare-ieșire și instrucțiunile de control.
- 3 Explicați destinația fiecărei unități funcționale din componența unui calculator (fig. 5.13).
- 4 Care este rolul dispozitivului central de comandă din componența procesorului?
- 5 Formulați algoritmul de funcționare a unui calculator numeric. Enumerați comenzile pe care le transmite dispozitivul central de comandă executanților din componența unui calculator.
- 6 De ce depinde capacitatea de prelucrare a unui calculator? Argumentați răspunsul dvs.
- 7 Ce operații efectuează dispozitivul central de comandă în procesul executării unui program? Cum poate fi întreruptă derularea unui program?
- 8 Cum se realizează principiul de comandă prin program în cazul unui calculator numeric? Care este destinația calculatoarelor instalate pe automobilele moderne? Ce fel de unități periferice are un astfel de calculator? Ce fel de programe derulează pe aceste calculatoare?

5.7. GENERALITĂȚI DESPRE ALGORITMI

Termeni-cheie:

- reprezentarea algoritmilor
- clasificarea algoritmilor
- proprietățile algoritmilor
- gândirea algoritmică
- cultura informațională

Pentru a simplifica procesele de elaborare a algoritmilor, s-a convenit ca aceștia să fie reprezentați (descriși, notați) cu ajutorul unor mijloace standard, cele mai răspândite fiind:

1) limbajul de comunicare între oameni, cu utilizarea, în caz de necesitate, a formulelor matematice;

2) schemele logice;

3) limbajele algoritmice (limbajele de programare).

De exemplu, algoritmul de soluționare a ecuațiilor de gradul I $ax + b = 0$ poate fi descris în limba română în felul următor:

1. Citește de la tastatură coeficienții a și b .
2. Dacă $a \neq 0$, atunci calculează rădăcina $x = -b / a$, afișează pe ecran mesajul "Ecuația are o singură rădăcină" și valoarea x . Stop.
3. Dacă $a = 0$ și $b = 0$, afișează pe ecran mesajul "Ecuația are o mulțime infinită de rădăcini". Stop.
4. Dacă $a = 0$ și $b \neq 0$, atunci afișează pe ecran mesajul "Ecuația nu are sens". Stop.

Avantajul principal al modului de descriere a algoritmilor cu ajutorul unui limbaj de comunicare între oameni constă în faptul că descrierile respective sunt înțelese de orice persoană care cunoaște instrucțiunile și operațiile ce apar în textul algoritmului. Cu regret, calculatoarele moderne nu pot interpreta univoc limbajele vorbite de oameni, iar algoritmi descriși în acest limbaj nu pot fi utilizați ca programe de calculator.

În cazul schemelor logice, algoritmi sunt descriși cu ajutorul simbolurilor grafice *start*, *stop*, *instrucțiune*, *apel de subalgoritm*, *repetă*, *cât*, *dacă* ș.a. Pentru exemplificare, în figura 5.14 este prezentată schema logică a algoritmului pentru soluționarea ecuațiilor de gradul I.

Analiza schemelor logice ne permite să efectuăm mai ușor **clasificarea algoritmilor** în algoritmi liniari, algoritmi repetitivi și algoritmi cu ramificații. Schemele logice sunt foarte sugestive, însă, ca și în cazul limbajelor de comunicare între oameni, descrierile respective nu pot fi interpretate univoc de calculatoarele moderne. Prin urmare, schemele logice nu pot fi utilizate ca programe de calculator.

Limbajele algoritmice reprezintă limbaje artificiale, create în scopul descrierii exacte a algoritmilor prin utilizarea unui vocabular, a unei sintaxe și a unei semantici bine definite. Pentru exemplificare, prezentăm în continuare algoritmul de soluționare a ecuațiilor de gradul I, scris în limbajul algoritmic PASCAL:

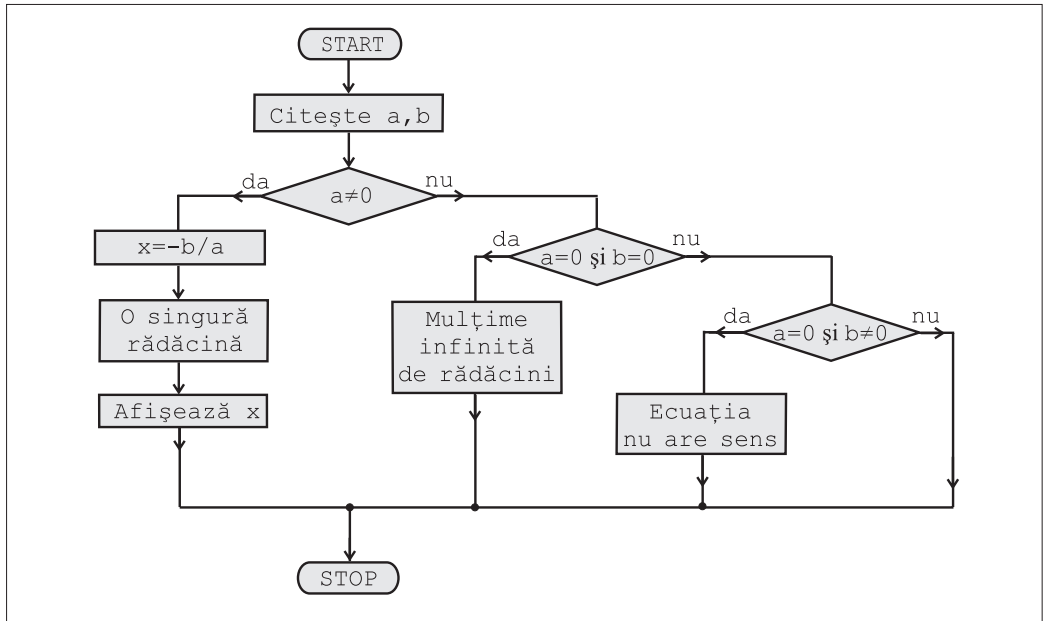


Fig. 5.14. Schema logică a algoritmului pentru soluționarea ecuațiilor de gradul I

Program Exemplu;

var a, b, x : real;

begin

 readln(a, b);

if a <> 0 **then**

begin

 x := -b/a;

 writeln('Ecuția are o singură rădăcină');

 writeln(x);

end;

if (a=0) **and** (b=0) **then**

 writeln('Ecuția are o mulțime infinită de rădăcini');

if (a=0) **and** (b <> 0) **then**

 writeln('Ecuția nu are sens');

end.

Cuvintele engleze din această descriere au următoarea semnificație:

program – program;

var – variabilă;

begin – început;

end – sfârșit;

if – dacă;

then – atunci;

readln – citește;

writeln – scrie;

and – și.

Constatăm că limbajul algoritmic PASCAL conține instrucțiuni și cuvinte auxiliare, pe care le-am studiat în paragrafele precedente în procesul elaborării programelor pentru comanda executanților **Cangurul** și **Furnica**.

Avantajul principal al limbajelor algoritmice constă în faptul că ele asigură o descriere univocă a algoritmilor, sunt înțelese de oameni și pot fi traduse relativ ușor în limbajul executanților. Evident, studierea unui limbaj algoritmic, ca și studierea unei limbi noi, necesită un anumit efort din partea utilizatorului. În școlile din țara noastră, limbajul algoritmic PASCAL se studiază începând cu clasa a 9-a.

Limbajele de programare reprezintă limbaje algoritmice, adaptate pentru descrierea algoritmilor într-o formă înțeleasă de executanți. De exemplu, programele elaborate în paragrafele precedente au fost scrise în limbajul executanților **Cangurul** și **Furnica**. Acest limbaj de programare conține, în afară de instrucțiunile REPETĂ, CÂT, DACĂ, apel de procedură ș.a., întâlniri în toate limbajele algoritmice, și instrucțiuni specifice anume acestor executanți: PAS, SALT, ROTIRE, SUS, JOS etc.

Indiferent de forma în care sunt reprezentați algoritmi, ei dispun de un șir de proprietăți comune, care le deosebesc de instrucțiunile, recomandările sau planurile elaborate pentru soluționarea unor probleme concrete. S-a stabilit că orice algoritm are trei proprietăți distincte:

1) **determinismul** – cunoașterea cu exactitate în fiecare moment al execuției algoritmului a următoarei operații de executat, precum și modul de execuție a fiecărei operații;

2) **universalitatea** – algoritmul este aplicabil pentru soluționarea tuturor problemelor pentru care a fost elaborat;

3) **finitudinea** – algoritmul este finit în spațiu (ca descriere) și timp (ca execuție).

De obicei, în procesul elaborării, utilizatorul schițează algoritmul într-un limbaj de comunicare între oameni, de exemplu, în limba română. Pe parcurs, pentru o descriere sugestivă a prelucrărilor preconizate, pot fi utilizate și schemele logice. Evident, algoritmul în versiunea finală va fi scris într-un limbaj algoritmic sau, în cazul unui executant concret, direct în limbajul respectiv de programare.

În general, elaborarea algoritmilor reprezintă un proces de creație care presupune că utilizatorul posedă așa-numita gândire algoritmică.

Prin gândire algoritmică înțelegem capacitatea persoanei de a elabora algoritmi pentru soluționarea problemelor pe care ea le întâlnește în viața cotidiană.

Evident, gândirea algoritmică se formează și se dezvoltă în procesul studierii mai multor discipline școlare, rolul central revenindu-i informaticii. Să reținem că dacă un secol în urmă sarcina principală a instituțiilor de învățământ consta în cultivarea științei de carte, astăzi această sarcină s-a extins, incluzând formarea **culturii informaționale** și dezvoltarea gândirii algoritmice.

Întrebări și exerciții

- ❶ Care sunt mijloacele principale de reprezentare a algoritmilor? Care sunt avantajele și dezavantajele fiecărui mod de notare a algoritmilor?
- ❷ Încercați să descrieți procesul de elaborare a unui algoritm. Ce fel de mijloace se folosesc pentru a descrie un algoritm în curs de elaborare?
- ❸ Există oare vreo diferență între un limbaj algoritmic și un limbaj de programare? Argumentați răspunsul dvs.
- ❹ Explicați termenii *gândire algoritmică* și *cultură informațională*. Utilizând un server de căutare, găsiți în Internet informații referitoare la acești termeni.
- ❺ Enumerați proprietățile de bază ale algoritmilor. Verificați dacă programele elaborate pentru comanda executanților **Cangurul** și **Furnica** posedă aceste proprietăți. Elaborați un algoritm pentru soluționarea ecuațiilor de gradul II. Reprezentați acest algoritm în formă de schemă logică.

BIBLIOGRAFIE

1. Cerchez, Emanuela, Șerban, Marinela, *Informatica pentru gimnaziu*, Iași, Editura POLIROM, 2005, 232 p.
2. Cerchez, Emanuela, Șerban, Marinela, *PC pas cu pas*, Iași, Editura POLIROM, 2008, 296 p.
3. Curteanu, Silvia, *EXCEL prin exemple*, Iași, Editura POLIROM, 2004, 352 p.
4. Eder, Bernhard, Kodym, Willibald, Lechner, Franz, *Excel: Calcul tabelar*, București, Editura ALL EDUCATIONAL, 2002, 112 p.
5. Garabet M.E., Voicu A.E., Huțanu E., *Fizică, biologie, chimie pentru gimnaziu utilizând Microsoft Office*, București, Editura ALL EDUCATIONAL, 2001, 224 p.
6. Harvey, Greg, *Excel pentru toți*, București, Editura Teora, 1996, 416 p.
7. Heathcote P.M., *Excel 2000... Pentru copii*, București, Editura ALL EDUCATIONAL, 2003, 64 p.
8. Ionescu, Bogdan, Ionescu, Iuliana, Oancea, Mirela, *Inițiere în Microsoft Office*, București, Editura InfoMega, 2006, 475 p.
9. Ionescu, Bogdan, Ionescu, Iuliana, *Tehnologia Aplicațiilor Office – Excel 2007*, București, Editura InfoMega, 2011, 502 p.
10. Ivanov, Lilia, Gremalschi, Anatol, Căpățână, Gheorghe ș.a., *Informatică. Curriculum pentru învățământul gimnazial (clasele VII–IX)*, Ministerul Educației al Republicii Moldova, 2010.
11. Mateiaș, Dorina, Matei, Rodica, *Tainele informaticii*. Manual de informatică pentru clasele V–VIII, Pitești, Editura Paralela 45, 2008, 280 p.
12. Mârșanu R., Voicu A.E., *Tehnologia informației. Informatică – Tehnologii asistate de calculator*, București, Editura ALL EDUCATIONAL, 2004, 112 p.
13. Reisner, Trudi, *Excel sub Windows 95 pentru începători*, București, Editura Teora, 1996, 112 p.
14. Босова Л.Л., *Информатика: Учебник для 7 класса*, М.: БИНОМ, Лаборатория знаний, 2012, 237 с.
15. Босова Л.Л., Босова А.Ю., *Информатика и ИКТ. Учебник для 9 класса*. В 2 ч., М.: БИНОМ, Лаборатория знаний, 2012, ч. 1 – 244 с., ч. 2 – 79 с.
16. Горячев А.В., Макарина Л.А. и др., *Информатика. 7 класс*. В 2 кн., М.: 2012, кн. 1 – 256 с., кн. 2 – 144 с.
17. Макарова Н.В., *Информатика и ИКТ. Практикум. 8–9 класс*, СПб.: Издательство «Питер», 2010, 384 с.
18. Первин Ю.А., *Информатика дома и в школе*. Книга для ученика, Серия «Основы информатики», СПб.: БХВ-Петербург, 2003, 352 с.
19. Первин Ю.А., *Информатика дома и в школе*. Книга для учителя, Серия «Основы информатики», СПб.: БХВ-Петербург, 2003, 144 с.
20. Семакин И.Г., Залогова Л.А., Русаков С.В., Шестакова Л.В., *Информатика. Базовый курс*. Учебник для 9 класса, М.: БИНОМ, Лаборатория знаний, 2012, 341 с.
21. Угринович Н.Д., *Информатика и ИКТ. Учебник для 9 класса*, М.: БИНОМ, Лаборатория знаний, 2009, 295 с.

Acest manual este proprietatea Ministerului Educației, Culturii și Cercetării
al Republicii Moldova

Gimnaziul/Liceul _____				
Manualul nr. _____				
Anul de folosire	Numele de familie și prenumele elevului	Anul școlar	Aspectul manualului	
			la primire	la restituire
1				
2				
3				
4				
5				

- Dirigintele verifică dacă numele elevului este scris corect.
- Elevul nu trebuie să facă niciun fel de însemnări în manual.
- Aspectul manualului (la primire și la restituire) se va aprecia folosind termenii: *nou, bun, satisfăcător, nesatisfăcător*.

Imprimare la Tipografia „BALACRON” SRL,
str. Calea Ieșilor, 10; MD-2069
Chișinău, Republica Moldova
Comanda nr.