

Ministerul Educației al Republicii Moldova

Anatol Gremalschi
Sergiu Corlat
Andrei Braicov

INFORMATICĂ

XII

Manual pentru clasa a XII-a

Știința
2015

Elaborat în baza curriculumului disciplinar în vigoare și aprobat prin Ordinul ministrului educației (nr. 399 din 25 mai 2015). Editat din mijloacele financiare ale *Fondului special pentru manuale*.

Comisia de evaluare: *Gheorghe Chistrugă*, prof. școlar, gr. did. superior, Liceul Teoretic „Mihai Eminescu”, Drochia; *Elena Doronina*, prof. școlar, gr. did. superior, Liceul Teoretic „Natalia Gheorghiu”, Chișinău; *Svetlana Rudoî*, prof. școlar, gr. did. superior, Liceul Teoretic „Mihai Eminescu”, Chișinău

Recenzenți: *Tatiana Baci*, dr. în psihologie, Universitatea Pedagogică de Stat „Ion Creangă”; *Alexei Colibneac*, prof. universitar, șef catedră „Grafică”, Academia de Muzică, Teatru și Arte Plastice, maestru în arte; *Tatiana Cartaleanu*, dr. în filologie, Universitatea Pedagogică de Stat „Ion Creangă”

Acest manual este proprietatea Ministerului Educației al Republicii Moldova

Liceul/Gimnaziul _____				
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 controlează dacă numele elevului este scris corect.

Elevul nu trebuie să facă niciun fel de însemnări în manual.

Aspectul manualului (la primire și restituire) se va aprecia cu calificativele: *nou, bun, satisfăcător, nesatisfăcător*.

Redactor: *Mariana Belenciuc*

Corector: *Maria Cornesco*

Redactor tehnic: *Nina Duduciuc*

Machetare computerizată și copertă: *Olga Ciuntu*

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.asm.md; prini_stiinta@yahoo.com;

www.stiinta.asm.md

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

Editura se obligă să achite deținătorilor de copyright, care încă nu au fost contactați, costurile de reproducere a imaginilor folosite în prezenta ediție.

Descrierea CIP a Camerei Naționale a Cărții

Gremalschi, Anatol

Informatică: Manual pentru clasa a 12-a/Anatol Gremalschi, Sergiu Corlat, Andrei Braicov; Min. Educației al Rep. Moldova. – Ch.: Î.E.P. Știința, 2015 (Tipografia „BALACRON” SRL) – 144 p.

ISBN 978-9975-67-984-8

004 (075.3)

Imprimare la Tipografia „BALACRON” SRL, str. Calea Ieșilor, 10; MD-2069, Chișinău, Republica Moldova
Comanda nr. 643

CUPRINS

Partea I. CALCUL NUMERIC

Capitolul 1. Elemente de modelare

- 1.1. Noțiune de model. Clasificarea modelelor 5
- 1.2. Modelul matematic și modelarea matematică 7
- 1.3. Soluții analitice și soluții de simulare 9
- 1.4. Etapele rezolvării problemei la calculator 12

Capitolul 2. Erori în calculul numeric

- 2.1. Numere aproximative. Eroarea absolută și relativă 16
- 2.2. Sursele erorilor de calcul 17

Capitolul 3. Metode numerice de rezolvare a ecuațiilor algebrice și transcendente

- 3.1. Separarea soluțiilor ecuațiilor algebrice și transcendente 22
- 3.2. Metoda bisecției 25
- 3.3. Metoda coardelor 27
- 3.4. Metoda Newton 31

Capitolul 4. Metode numerice de calcul al determinanților și rezolvarea sistemelor de ecuații liniare

- 4.1. Determinanți numerici 36
- 4.2. Metoda Cramer de rezolvare a sistemelor de ecuații liniare 41
- 4.3. Metoda Gauss de rezolvare a sistemelor de ecuații liniare 45

Capitolul 5. Integrarea numerică

- 5.1. Metoda dreptunghiurilor pentru calculul aproximativ al integralei definite 51
- 5.2. Variații ale metodei dreptunghiurilor 55
- 5.3. Formula trapezelor 58

Partea a II-a. BAZE DE DATE

Capitolul 6. Concepte generale despre baze de date

- 6.1. Noțiuni și concepte despre date și despre baze de date 63
 - 6.1.1. Date elementare și structuri de date 63
 - 6.1.2. Baze de date 64
- 6.2. Tipuri de baze de date 65

Capitolul 7. Elaborarea și gestiunea bazelor de date

- 7.1. Elaborarea unei baze de date 68
 - 7.1.1. Aspecte generale 68
 - 7.1.2. Proiectarea entităților unei baze de date relaționale 69
 - 7.1.3. Principii de proiectare 71
- 7.2. Sisteme de gestiune a bazelor de date (SGBD) 72
 - 7.2.1. Concepte generale despre sisteme de gestiune a bazelor de date 72
 - 7.2.2. Sistemul de gestiune a bazelor de date Microsoft Office Access 73
 - 7.2.3. Structura bazei de date *Liceu* 73

Capitolul 8. Tabelele – principalele obiecte ale unei baze de date

- 8.1. Crearea tabelor 76
 - 8.1.1. Crearea structurii unui tabel 76
 - 8.1.2. Proprietățile câmpurilor tabelului 78
- 8.2. Stabilirea relațiilor dintre tabele 81
- 8.3. Modificarea tabelor 82
 - 8.3.1. Introducerea și editarea datelor 82

8.3.2. Modificarea aspectului tabelului	84
8.3.3. Modificarea structurii tabelului	85
8.3.4. Caracteristica <i>Lookup</i> a câmpurilor	85
8.4. Crearea expresiilor Access	86
8.4.1. Operatori Access	86
8.4.2. Funcții Access	87
Capitolul 9. Interogări	
9.1. Noțiuni generale despre interogări	91
9.2. Interogări de selecție	94
9.2.1. Criterii de selecție	95
9.2.2. Interogări cu parametri	96
9.3. Interogări de acțiune	97
9.3.1. Interogări care generează tabele	97
9.3.2. Interogări de excludere a unor înregistrări	98
9.3.3. Interogări de modificare a unor înregistrări	98
9.3.4. Interogări care adaugă înregistrări în tabele existente	99
9.4. Interogări de totalizare	100
9.4.1. Interogări de creare a câmpurilor rezultante (calculate)	100
9.4.2. Interogări de grupare și totalizare	100
9.4.3. Interogări încrucișate	101
Capitolul 10. Formulare și rapoarte	
10.1. Formulare	103
10.1.1. Crearea unui formular cu ajutorul programului de asistență	103
10.1.2. Crearea sau modificarea formularelor în regimul <i>Design View</i> (regim de proiectare)	105
10.1.3. Subformulare	108
10.2. Rapoarte	109
10.2.1. Crearea unui raport cu ajutorul programului de asistență	109
10.2.2. Modificarea rapoartelor în regimul <i>Design View</i>	111
10.2.3. Crearea diagramelor în cadrul rapoartelor	111
10.3. Întreținerea bazelor de date (opțional)	113
10.3.1. Compactarea și repararea unei baze de date	113
10.3.2. Crearea copiilor de rezervă	113
10.3.3. Asigurarea securității datelor	113
Partea a III-a. WEB DESIGN	
Capitolul 11. Documente WEB	
11.1. Noțiuni și concepte	116
11.2. Tipurile documentelor Web	116
11.3. Proiectarea și realizarea unui document Web	118
11.4. Crearea documentelor Web cu ajutorul aplicațiilor de oficiu	120
Capitolul 12. Limbajul HTML	
12.1. Structura generală a unui document HTML	122
12.1.1. Despre documente HTML	122
12.1.2. Structura generală a unui document HTML	123
12.2. Formatarea textului	124
12.3. Liste	129
12.4. Referințe	132
12.5. Imagini	136
12.6. Tabele	139

CAPITOLUL 1

ELEMENTE DE MODELARE

După studierea acestui capitol, veți fi capabili să:

- descifrați sensul termenilor *model*, *modelare*;
- clasificați modelele obiectelor, proceselor, fenomenelor;
- utilizați și să elaborați modele matematice;
- distingeți soluțiile analitice și cele de simulare;
- explicați metodele de obținere a soluțiilor analitice și a celor de simulare;
- explicați interacțiunea dintre modelul matematic, algoritm și program;
- planificați procesul de rezolvare a problemei la calculator.

1.1. Noțiune de model. Clasificarea modelelor

Imitarea unor procese, obiecte sau fenomene este caracteristică societății umane pe tot parcursul istoriei sale. Primele desene, realizate de oamenii epocii de piatră pe pereții peșterilor, erau în același timp și primele încercări de a reproduce obiectele și fenomenele reale prin imagini (*fig. 1.1*).

Globul-machetă al planetei noastre este și el o imitație a unui corp real. El ne aduce la cunoștință date despre forma și mișcarea Terrei, amplasarea continentelor și oceanelor, a țărilor și orașelor (*fig. 1.2*). Dar elementele machetei nu constituie în întregime obiectul inițial. Astfel, în cazul globului-machetă avem de-a face doar cu un corp sferic, prin centrul căruia trece o axă, care permite rotirea, iar pe suprafață avînd imprimate diverse informații despre planeta Pămînt. Globul-machetă redă anumite trăsături ale corpului

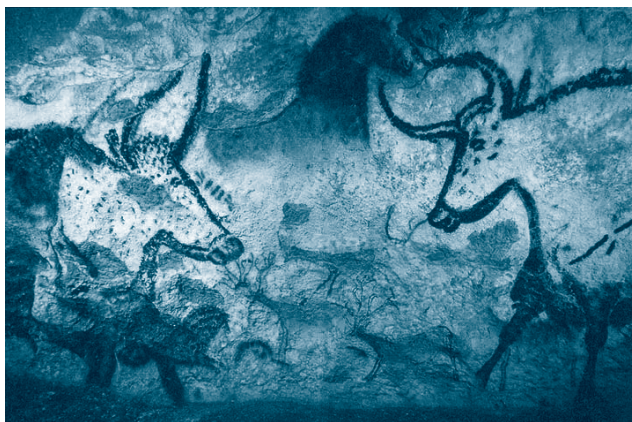


Fig. 1.1. Imagine zoomorfă din paleolitic



Fig. 1.2. Globul-machetă al Pămîntului

cosmic real, dar se deosebește de el: diferă dimensiunile, proprietățile fizice, structura etc. Globul-machetă este o reprezentare simplificată a Terrei, care permite studierea doar a anumitor particularități ale ei – este doar un *model*.

Modelul este un sistem material sau ideal, logico-matematic cu ajutorul căruia pot fi studiate, prin analogie, proprietățile și operațiile efectuate asupra sistemului inițial, care, în general, este mai complex.

Modelele sînt utilizate din cele mai vechi timpuri pentru examinarea fenomenelor și proceselor complexe, de exemplu, moleculele, atomii, sistemul solar, universul în ansamblu, un reactor atomic, un zgîrie-nor etc. Un model reușit este mai comod pentru cercetări decît obiectul real. Mai mult chiar, anumite obiecte și fenomene nici nu pot fi studiate în original. De exemplu, sînt greu de efectuat experimente ce țin de economia unei țări, sînt imposibile experiențele cu planetele sistemului solar, cele care presupun revenirea în timp etc. Un alt aspect important al modelării îl constituie posibilitatea de a pune în evidență doar acei factori, acele proprietăți ale obiectului real, care sînt esențiale pentru obiectul studiat.

De asemenea, modelul permite instruirea în vederea utilizării corecte a obiectului real, verificînd diferite moduri de a reacționa pe modelul acestui obiect. Experiențele cu obiectul real pot fi imposibile sau foarte periculoase (durata mare a procesului în timp, riscul de a deteriora obiectul). În cazurile cercetării obiectelor dinamice, caracteristicile cărora depind de timp, o importanță primordială capătă problema prognozării stării obiectului sub acțiunea unor anumiți factori.

În general, un model bine construit permite obținerea unor cunoștințe noi despre obiectul original supus cercetării.

Procesul de construire a modelului se numește modelare.

Există cîteva tipuri de modelare, ce pot fi unite în două grupe mari: *modelarea materială* și *modelarea ideală*.

În cazul *modelării materiale*, cercetarea originalului se efectuează prin redarea cu ajutorul unui alt obiect material, mai simplu decît cel real, denumit model, a caracteristicilor geometrice, fizice, dinamice, funcționale de bază ale originalului. Exemple: machetele clădirilor, avioanelor, automobilelor și ale vehiculelor militare etc.

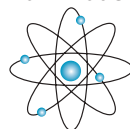
În cazul *modelării ideale*, cercetarea originalului se efectuează prin reprezentarea proprietăților lui cu ajutorul anumitor concepte, scheme, planuri, structuri, care există doar în imaginația omului.

În general, *modelarea ideală* se bazează pe o concepție intuitivă despre obiectul cercetărilor. De exemplu, experiența de viață a fiecărui om poate fi considerată drept un model personificat al lumii înconjurătoare. Atunci cînd modelarea nu este intuitivă și se folosesc anumite simboluri, cum ar fi diverse scheme, grafice, formule, ea este numită *simbolică*. În categoria modelelor simbolice un loc aparte îl ocupă *modelarea matematică*, în care examinarea obiectului se realizează prin intermediul unui model formulat în termeni și noțiuni matematice, cu folosirea anumitor metode matematice. Un exemplu clasic al modelării matematice îl constituie descrierea și cercetarea legilor de bază ale mecanicii lui Newton cu ajutorul instrumentelor matematice.

Pentru studierea unui proces sau fenomen nu este suficient să fie construit modelul respectiv. În general, modelul descrie anumite legături, relații, caracteristici ale originalului, însă scopul modelării constă în obținerea, în baza informațiilor furnizate de model, a unor date noi despre original.

Întrebări și exerciții

- 1 Explicați noțiunea de model. Dați exemple de obiecte ale lumii înconjurătoare și modele ale acestora. Pentru fiecare exemplu, indicați caracteristicile obiectului original, ce sînt redată de model.
- 2 Formulați definiția noțiunii de model material. Dați exemple de modele materiale. Argumentați necesitatea utilizării modelării materiale în diferite domenii ale științei și tehnicii.
- 3 Formulați definiția noțiunii de model ideal. Exemplificați. Motivați necesitatea utilizării modelelor ideale.
- 4 În anul 1911, cercetătorul E. Rutherford a propus „modelul planetar (nuclear) al atomului”. Explicați sensul cuvîntului „model” în contextul propunerii lui Rutherford, precum și motivul pentru care el este numit planetar.
- 5 Explicați sensul următoarei afirmații: „Pictogramele interfeței grafice a utilizatorului reprezintă modele ale obiectelor prelucrate de produsele software”.



1.2. Modelul matematic și modelarea matematică

Unul dintre scopurile de bază ale informaticii, ca știință interdisciplinară, constă în elaborarea metodelor de rezolvare a problemelor complicate de cercetare și de calcul cu ajutorul tehnicii computaționale. Inițial, informatica se dezvoltă ca o ramură a matematicii aplicate. Primele probleme, abordate în cadrul informaticii, erau, de asemenea, pur matematice, iar soluționarea lor se reducea la efectuarea unui volum mare de calcule complexe. În prezent însă, informatica a devenit o știință independentă, cu propriile metode și obiecte de cercetare, care se bazează pe legăturile matematice. Informatica studiază și soluționează probleme complexe din domeniul matematicii, fizicii, chimiei, biologiei, economiei, ecologiei, filologiei și al sociologiei. Dar, indiferent de domeniul din care provine problema, în procesul de soluționare a ei, informatica se bazează pe matematică. În consecință, pentru a dezlega o anumită problemă, înainte de a utiliza calculatorul propriu-zis, este necesară descrierea fenomenelor și proceselor care apar în problema supusă rezolvării cu ajutorul noțiunilor matematice. În particular, acestea pot fi funcții, ecuații, inecuații, sisteme de ecuații etc.

Modelul matematic reprezintă descrierea unui proces sau a unui fenomen cu ajutorul noțiunilor matematice.

Exemplul 1: Se consideră două automobile. Unul dintre ele se mișcă rectiliniu, traiectoria respectivă fiind definită prin ecuația $x = 1/2$. Al doilea automobil se deplasează pe o traiectorie circulară, descrisă de o circumferință unitară cu centrul în originea sistemului de coordonate. Se cere determinarea coordonatelor punctelor posibile de impact ale automobilelor.

Abstractizînd problema, obținem următoarea formulare: un punct se mișcă pe o traiectorie dată de ecuația $x = 1/2$. Al doilea punct se mișcă pe o traiectorie dată de ecuația $x^2 + y^2 = 1$. Se cere să se găsească soluțiile sistemului de ecuații:

$$\begin{cases} x = \frac{1}{2} \\ x^2 + y^2 = 1. \end{cases}$$

Grafic problema este reprezentată de schema alăturată.

Algoritmul de rezolvare e următorul:

1. Se consideră $x = 1/2$.

2. Din ecuația a doua a sistemului se calculează

$$y_1 = \sqrt{1 - \frac{1}{4}}; \quad y_2 = -\sqrt{1 - \frac{1}{4}}.$$

3. Soluția problemei este $\left(\frac{1}{2}, \frac{\sqrt{3}}{2}\right); \left(\frac{1}{2}, -\frac{\sqrt{3}}{2}\right)$.

Exemplul 2: Pe o masă netedă se află o bilă metalică, fixată de un arc (fig. 1.3). Arcul se comprimă, fără a-i deteriora elasticitatea, apoi se eliberează. Se cere determinarea coordonatei bilei peste t secunde.

Dacă k – coeficientul de elasticitate a arcului, m – masa bilei, x – mărimea deformației arcului, atunci, în baza legii lui Hooke și a legii 2 a lui Newton, modelul matematic al sistemului bilă-arc va avea forma:

$$ma = -kx, \text{ unde } a - \text{acelerația.}$$

Poziția bilei după comprimarea arcului (deformația inițială) se notează prin x_0 (fig. 1.4). Se cere determinarea poziției bilei (mărimea deformației) peste t secunde. Pentru aceasta modelul precedent se va transforma astfel încît să fie prezentată dependența valorii x (a deformației) de timp. Pentru deformări mici și lipsa forței de frecare se va folosi legea deplasărilor armonice

$$x(t) = x_0 \cos \sqrt{\frac{k}{m}} t.$$

Formula dată permite determinarea poziției bilei x (deformația arcului) în orice moment de timp t , în situația în care sînt cunoscute valorile k , m și x_0 .

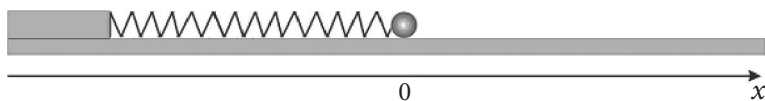
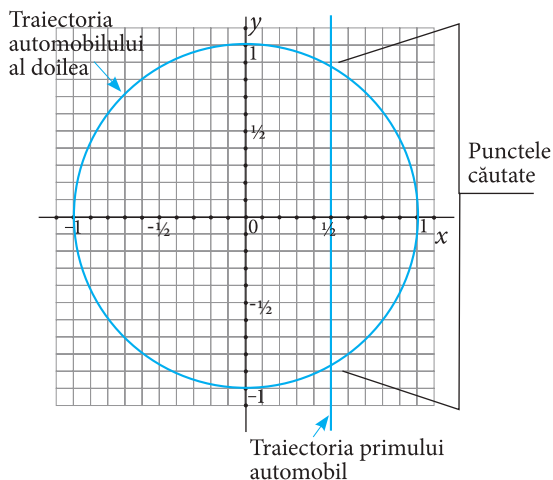


Fig. 1.3. Starea inițială a sistemului



Fig. 1.4. Starea sistemului după comprimarea arcului

Folosind formula dedusă mai sus, se poate rezolva problema generală de determinare a deformației în orice moment de timp cu ajutorul următorului program:

```

program cn01;
var
  t, x0,k,m,x: real;
begin
  readln(x0,k,m,t);
  x:=x0*cos(sqrt(k/m*t));
  writeln('x=', x:0:6);
end.

```

Întrebări și exerciții

- ❶ Definiți noțiunea de model matematic. Este el oare un model material?
- ❷ Indicați câteva domenii de utilizare a modelelor matematice. Argumentați necesitatea întrebuirii lor.
- ❸ Motivați necesitatea transpunerii modelelor matematice în programe de calculator și a folosirii acestora din urmă.
- ❹ Scrieți un program care, pentru problema din *exemplul 2*, va calcula deformația arcului x cu intervale de o secundă, din momentul inițial pînă în momentul t (t are o valoare întreagă). Valorile x_0, k, m, t se vor introduce de la tastatură.
- ❺ Izotopul radioactiv plutoniu-235 are perioada de înjumătățire de 26 de minute. În această perioadă jumătate din cantitatea inițială a izotopului dispare prin descompunere în alte elemente chimice. Descrieți modelul matematic care permite calculul numărului ciclurilor de înjumătățire necesare pentru dispariția a k procente din cantitatea inițială a izotopului.
- ❻ Efectul unui medicament se calculează conform formulei $r_k = \alpha r_{k-1} + 0,4^k$, unde r_k este concentrația substanțelor active peste k ore după administrarea lui. Inițial $r_0 = 1$, iar $0 < \alpha < 1$. Din formulă rezultă că r_k atinge valoarea maximă r_m după ce au trecut m ore, ulterior începînd să scadă. Scrieți un program ce va determina peste cîte ore efectul medicamentului în studiu va atinge valoarea maximă. Numărul real α se citește de la tastatură.

1.3. Soluții analitice și soluții de simulare

Mai întii vom examina următoarea problemă:

Exemplul 1: Un bazin cu volumul 100 m^3 , care inițial conține 20 m^3 de apă, se umple cu același lichid folosind pompa ce are capacitatea de pompare de $15 \text{ m}^3/\text{oră}$. Totodată, din bazin în dispozitivul de filtrare se scurg $5 \text{ m}^3/\text{oră}$. Se cere să se determine peste cîte ore va fi umplut bazinul.

Desigur, există mai multe metode de rezolvare a acestei probleme. Una dintre cele mai simple este modelarea procesului de acumulare a apei în bazin peste fiecare oră, cu ajutorul unui tabel.



Timp (ore)	Cantitatea de apă ce a fost pompată	Cantitatea de apă ce s-a scurs	Cantitatea de apă din bazin
0	0	0	20
1	15	5	30
2	30	10	40
3	45	15	50
4	60	20	60
5	75	25	70
6	90	30	80
7	105	35	90
8	120	40	100

Soluția problemei a fost obținută printr-un număr relativ mare de calcule consecutive, care au reconstruit dinamic volumul de apă în bazin peste fiecare oră. Sigur, nu este cea mai eficientă metodă: se utilizează un număr considerabil de rezultate intermediare, numărul de operații realizate este la fel nemotivat de mare.

O altă soluție se bazează pe o formulă care permite calculul direct al rezultatului final. Timpul necesar pentru umplerea bazinului este determinat de diferența dintre volumul total al bazinului și cantitatea de apă care se conținea inițial în el raportată la debitul apei timp de o oră:

$$t = \frac{100 - 20}{15 - 5} = 8.$$

Prima dintre metodele efectuate utiliza un proces iterativ, care determina volumul de apă după fiecare interval elementar de timp (1 oră) și calcula rezultatele noi, folosind datele obținute la etapa precedentă. Cu alte cuvinte, a fost realizat procesul de modelare a etapelor de umplere a bazinului. Acest proces se numește *simulare*.

Simularea este o tehnică de rezolvare a problemelor, bazată pe utilizarea unor modele matematice și logice ce descriu comportarea unui sistem real în spațiu și/sau în timp.

Model de simulare este modelul de rezolvare a problemei în baza tehnicii de simulare. Soluțiile obținute prin procesul de simulare se numesc soluții de simulare.

De obicei, modelele de simulare se folosesc atunci când este necesară determinarea stării sistemului cercetat atât în momentul când se obține soluția finală, cât și în momentele intermediare de timp.

Cea de a doua metodă din exemplul de mai sus a realizat calculul direct al soluției prin utilizarea unei formule:

$$t = \frac{V_{\text{bazin}} - V_{\text{inițial}}}{C_{\text{pompa}} - C_{\text{scurgere}}}, \quad \text{unde}$$

- V_{bazin} – volumul total al bazinului,
- $V_{\text{inițial}}$ – cantitatea inițială de apă în bazin,
- C_{pompa} – capacitatea de pompare a pompei,
- C_{scurgere} – capacitatea găurii de scurgere.

Formula permite calculul timpului necesar pentru umplerea oricărui bazin, cu orice cantitate inițială de apă și cu orice capacitate de pompare a pompei, depășind capacitatea de scurgere, care de asemenea poate varia.

Metoda de rezolvare a problemelor bazată pe utilizarea unor formule ce permit calculul direct al rezultatului final, fără a cerceta stări și rezultate intermediare, se numește metodă analitică. Soluțiile obținute cu ajutorul acestei metode sînt numite soluții analitice.

Exemplul 2: Să se scrie un program care calculează suma primilor n termeni ai progresiei geometrice, avînd primul termen cu valoarea a_1 , ($a_1 > 0$) și rația q , ($q < 1$).

Din matematică este cunoscută formula $S_n = \frac{a_1(q^n - 1)}{q - 1}$, care permite determinarea directă a sumei, fără a calcula valoarea fiecărui termen al progresiei. Totodată se știe că termenul cu indicele n , ($n \geq 2$) poate fi calculat folosind formula recurentă $a_n = q \times a_{n-1}$. Atunci rezultatul poate fi calculat iterativ, prin calculul recurent al termenilor progresiei și adunarea lor consecutivă $S_n = a_1 + a_2 + \dots + a_{n-1} + a_n$.

Calcul direct al sumei	Calcul iterativ al sumei
<pre> program cn02; var S,a,q : real; n : integer; begin readln(a, q, n); S:=a*(exp(n*ln(q))-1)/(q-1); writeln('S=', S:0:6); end. </pre>	<pre> program cn03; var S,a,q : real; i, n : integer; begin readln(a, q, n); S:=0; for i:=1 to n do begin S:=S+a; a:=a*q; end; writeln('S=', S:0:6); end. </pre>

Fiecare dintre metodele prezentate mai sus are avantajele și neajunsurile sale. Pentru unele probleme este foarte complicat sau practic imposibil de determinat formula analitică (de exemplu, coordonatele unei comete sau ale unui asteroid în funcție de timp), pentru altele este destul de dificil de simulat un model adecvat, chiar și folosind un număr foarte mare de calcule intermediare.

Soluția analitică permite calculul imediat al rezultatului final, dar nu permite cercetarea dinamicii construirii acestuia. Utilizarea unui proces iterativ (a soluției de simulare) permite construirea dinamică a rezultatului în funcție de datele folosite în problemă și controlul soluției la fiecare iterație realizată. În același timp, pentru obținerea soluțiilor de simulare este necesar un număr mai mare de operații decît în cazul soluțiilor analitice.

Alegerea modului de determinare a soluției este influențată de mai mulți factori, principalii dintre ei fiind:

- posibilitatea de determinare a soluției analitice;
- necesitatea cercetării soluțiilor (stărilor) intermediare;
- timpul necesar pentru realizarea calculelor (în cazul soluțiilor de simulare);
- eroarea soluției de simulare (diferența dintre mărimea calculată și cea exactă).

Întrebări și exerciții

- 1 Definiți noțiunea de simulare. Ce este o soluție de simulare?
- 2 Ce înțelegeți prin metodă analitică de rezolvare a unei probleme? Ce reprezintă o soluție analitică? Care sînt proprietățile soluțiilor analitice?
- 3 Enumerați proprietățile soluțiilor de simulare. Care dintre aceste proprietăți implică utilizarea calculatorului pentru a găsi astfel de soluții?
- 4 Determinați o metodă de simulare pentru calcularea elementului cu numărul n din șirul de numere 1, 2, 3, 5, 8, 13, 21, Există oare o metodă analitică pentru determinarea elementului cu numărul n ?
- 5 Rezolvați problemele ce urmează prin metoda simulării:
 - a) În timpul zilei o buburuză urcă pe un stîlp 5 m, iar în timpul nopții coboară 3 m. Ascensiunea începe dimineața. Înălțimea stîlpului este de 15 m. Peste cîte zile va ajunge buburuza în vîrfurile stîlpului?
 - b) În condițiile punctului precedent, ascensiunea începe dimineața de la înălțimea de 6 m.
 - c) În condițiile punctului precedent, ascensiunea începe odată cu căderea nopții.
- 6 Elaborați un program care calculează suma primilor n termeni ai progresiei aritmetice, avînd primul termen cu valoarea a_1 , ($a_1 > 0$) și rația r , ($r > 0$).

1.4. Etapele rezolvării problemei la calculator

Instrumentele informatice permit rezolvarea problemelor atît prin metode analitice, cît și prin metode de simulare. Indiferent de metoda aplicată, rezolvarea oricărei probleme include mai multe etape, fiecare dintre ele avînd același grad de importanță.

Analiza problemei. Este etapa de studiu al conținutului problemei. Se stabilește setul de date inițiale, se determină care este rezultatul ce urmează să fie obținut, care sînt relațiile dintre datele inițiale și rezultat. Tot la această etapă sînt stabilite restricțiile suplimentare asupra datelor inițiale și a rezultatului.

Elaborarea modelului matematic al problemei. La această etapă datele inițiale sînt descrise prin structuri matematice. Folosind limbajul matematic, se descriu relațiile care permit obținerea rezultatului din datele inițiale. În funcție de problemă, aceste relații pot fi recurente (este creat un model de simulare) sau să permită calculul direct al rezultatului (model analitic). Tot aici are loc (dacă este necesar) divizarea problemei în *subprobleme* și elaborarea separată a modelelor matematice pentru fiecare din ele.

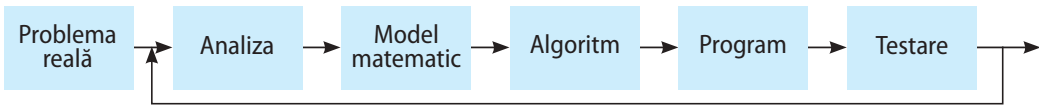
Elaborarea algoritmului. În cazul rezolvării informatice a unei probleme, algoritmul conține setul de instrucțiuni necesare pentru soluționarea problemei, descrise într-o formă prestabilită (pseudocod, schemă logică etc.), precum și ordinea executării acestora (pașii algoritmului). Dacă problema a fost divizată în subprobleme, algoritmul, suplimentar la descrierea subalgoritmilor, stabilește modul și condițiile de apel al acestora.

Scrierea programului. Pentru rezolvarea automatizată a problemei, cu ajutorul calculatorului algoritmul trebuie transpus într-o formă înțeleasă de calculator – **program**, folosind un **limbaj de programare**. Pașii algoritmului sînt prezentați cu ajutorul instrucțiunilor limbajului de programare, iar ordinea executării lor – de consecutivitate și structura instrucțiunilor limbajului. Datele inițiale și intermediare sînt descrise folosind structurile de date, acceptate de limbajul de programare. În procesul de scriere a programului pot să apară erori sintactice și/sau semantice. Procesul de corectare a lor este de

asemenea o parte a etapei de scriere a programului. Etapa se consideră încheiată atunci când compilarea sau interpretarea programului finalizează fără erori.

Testarea programului. O compilare reușită nu garantează rezolvarea corectă a problemei. Pentru verificarea corectitudinii programului se execută o serie de teste care stabilesc corectitudinea rezultatelor generate de program în funcție de seturi de date inițiale simple, medii și extreme. Dacă pentru toate testele efectuate programul prezintă rezultate corecte, se poate presupune că problema a fost rezolvată corect. Dacă în procesul de testare se obțin rezultate care diferă de cele corecte, urmează ca rezolvarea problemei să fie reluată, începând cu etapa de *analiză a problemei*.

Procesul de rezolvare a unei probleme la calculator poate fi ilustrat cu ajutorul următoarei scheme:



Exemplu:

Problemă:

În condiții de laborator, o populație de viruși, formată inițial din N unități și plasată într-un mediu steril, se micșorează în fiecare oră cu 50 de procente, dacă numărul virușilor la începutul orei este par, sau crește cu o unitate, dacă numărul virușilor la începutul orei este impar. În momentul când numărul virușilor devine mai mic decât cantitatea critică de supraviețuire C , populația dispăre integral.

Cerință: Scrieți un program care va stabili timpul necesar, în ore, pentru distrugerea în laborator a unei populații din N ($N < 32\,000$) viruși, având cantitatea critică de supraviețuire C ($1 < C < N$).

Analiza problemei

Populația formată dintr-un număr par de viruși în fiecare oră se micșorează de două ori. Populația formată dintr-un număr impar crește cu o unitate și se transformă într-o populație pară. Creșterea repetată este imposibilă, înjumătățirea se repetă însă cel puțin o dată la două ore. Prin urmare, valoarea C ($C > 1$) va fi atinsă într-un număr finit de ore.

Modelul matematic

Numărul populației în momentul de timp $t = 0$ este dat de numărul $N_0 = N$.

Numărul populației după t ore de sterilizare ($t > 0$) este dat de formula recurentă:

$$N_t = \begin{cases} \frac{N_{t-1}}{2}, & \text{dacă } N_{t-1} \text{ este par} \\ N_{t-1} + 1, & \text{dacă } N_{t-1} \text{ este impar} \end{cases}$$

Algoritm

Pasul 0. Inițializare: Se introduc valorile N , C . $t \leftarrow 0$.*

* Aici și în continuare expresia $a \leftarrow b$ semnifică: a primește valoarea lui b .

Pasul 1.

a) Trecerea unei ore: $t \leftarrow t+1$.

b) Remodelarea populației: dacă $N \bmod 2 = 0$, $N \leftarrow N \div 2$, altfel $N \leftarrow N+1$.

Pasul 2. Verificarea condiției de supraviețuire: dacă $N < C$, se afișează valoarea t . SFÎRȘIT. În caz contrar, se revine la *pasul 1*.

Program

```
program cn04;
var N,C,t: integer;
begin
  readln(N, C); t:=0;
  while (N>=C) do begin
    t:=t+1;
    if N mod 2 = 1 then N:=N+1 else N:= N div 2;
  end;
  writeln(t);
end.
```

Testare

Pentru verificarea corectitudinii rezolvării problemei au fost folosite următoarele seturi de date:

Date inițiale	Rezultat	Date inițiale	Rezultat	Date inițiale	Rezultat	Date inițiale	Rezultat
512 2	9	31999 16	15	331 330	2	332 330	1
25768 235	10	31999 2	18	31997 2	19	3 2	3

Întrebări și exerciții

- 1 Enumerați etapele rezolvării unei probleme la calculator. Explicați necesitatea fiecărei etape.
- 2 Care sînt metodele de descriere a algoritmului unei probleme? Exemplificați.
- 3 Care este impactul divizării unei probleme în subprobleme elementare? Dați exemple de probleme ce pot fi divizate în subprobleme. Indicați două sau mai multe probleme ce conțin subprobleme identice.
- 4 Pentru următoarele probleme realizați modelul matematic:
 - a) Sînt cunoscute coordonatele a trei vîrfuri ale unui dreptunghi cu laturile paralele axelor de coordonate. Se cere determinarea coordonatelor celui de-al patrulea vîrf.
 - b) În condițiile punctului precedent, laturile dreptunghiului pot fi poziționate arbitrar față de axele de coordonate.
- 5 Pentru următoarele probleme descrieți un algoritm de rezolvare, folosind metodele de descriere cunoscute:
 - a) Este dat un șir din cel mult 100 de numere întregi. Se cere să se aranjeze elementele șirului în ordine crescătoare.
 - b) Este dat un șir din cel mult 100 de numere întregi. Se cere determinarea elementului cu valoare maximă din șir și a numărului de repetări ale lui printr-o singură parcurgere a șirului.
- 6 Elaborați programe pentru rezolvarea problemelor din exercițiile 4 și 5.
- 7 Alcătuiți seturi de teste pentru programele realizate în exercițiul 6.

Test de evaluare

I Stabiliți valoarea de adevăr a următoarelor afirmații:

1. Modelul este o copie fidelă a originalului, care păstrează toate proprietățile acestuia:
a) adevărat; b) fals.
2. Modelul matematic este un model ideal:
a) adevărat; b) fals.
3. Modelul matematic poate fi utilizat în exclusivitate pentru rezolvarea problemelor matematice:
a) adevărat; b) fals.
4. Soluția de simulare a unei probleme permite calculul direct al rezultatului din datele inițiale ale problemei:
a) adevărat; b) fals.
5. În procesul de rezolvare a unei probleme la calculator, elaborarea modelului matematic al problemei precede elaborarea algoritmului:
a) adevărat; b) fals.
6. Rezultatul corect, obținut pentru un set de date de intrare, garantează corectitudinea rezultatelor furnizate de program pentru orice alt set de date:
a) adevărat; b) fals.

II Selectați varianta corectă a definiției:

1. Modelarea este procesul de:
a) utilizare a modelului;
b) construcție a modelului;
c) prezentare a modelului;
d) descompunere a originalului în componente elementare.
2. Modelul matematic este:
a) descrierea unei noțiuni matematice prin intermediul limbajului uman;
b) totalitatea de caracteristici geometrice ale unui model material;
c) descrierea unui proces sau fenomen prin intermediul noțiunilor matematice;
d) modelul material al unui corp sau figuri geometrice.
3. Metoda analitică de rezolvare a problemelor este metoda de rezolvare:
a) care permite calculul rezultatului final, prin cercetarea stărilor și rezultatelor intermediare;
b) prin identificarea rezultatului corect dintr-o listă finită de soluții posibile;
c) care permite calculul direct al rezultatului final, fără a cerceta stări și rezultate intermediare;
d) prin selectarea aleatorie a unui rezultat posibil.

III Descrieți pe etape procesul de rezolvare la calculator a problemei de determinare a valorii a^b pentru $0 < a, b < 10$, întregi.

CAPITOLUL 2

ERORI ÎN CALCULUL NUMERIC

După studierea acestui capitol, veți fi capabili să:

- identificați mărimile exacte și cele aproximative;
- calculați erorile absolute și relative;
- determinați sursele de erori în problemele de modelare;
- identificați tipurile de erori posibile în funcție de natura problemei rezolvate.

2.1. Numere aproximative. Eroarea absolută și relativă

Numărul a se numește aproximare a numărului A dacă valorile lor se deosebesc neînsemnat și a poate înlocui A în calcule numerice. Dacă este adevărată relația $a < A$, a este numit aproximare prin lipsă, dacă $a > A$ – aproximare prin adaos. De exemplu, pentru π numărul 3,14 va fi aproximare prin lipsă, iar 3,142 – aproximare prin adaos. Într-adevăr $3,14 < \pi < 3,142$. Valorile aproximative apar în procesul măsurărilor realizate, iar diferența valorii aproximative de cea exactă se poate datora unei varietăți de factori: condițiilor de temperatură, presiunii, umidității, calității instrumentelor de măsurare, calificării persoanei care execută măsurarea etc.

Info+

Reprezentarea numerelor în calculator implică utilizarea numerelor aproximative cu un număr finit de cifre. Cifrele reținute se numesc *cifre semnificative*, prima fiind obligatoriu diferită de 0.

Exemplu: 0,04708 are patru cifre semnificative. Primele două zerouri doar fixează poziția virgulei zecimale.

Aproximarea $a (a_n a_{n-1} a_{n-2} \dots a_0)$ a numărului exact A are k cifre semnificative exacte $a_i a_{i-1} \dots a_{i-k+1}$ dacă:

$$|A - a| \leq \frac{1}{2} 10^{i-k+1}.$$

Prin eroare Δ_a se înțelege diferența $A - a$ (uneori și $a - A$). În funcție de valorile a și A , Δ_a poate fi negativă sau pozitivă. Pentru a obține numărul exact A , se adaugă la a valoarea erorii Δ_a :

$$A = a + \Delta_a.$$

De multe ori este cunoscută valoarea aproximativă a fără a se cunoaște semnul erorii. În aceste cazuri se utilizează *eroarea absolută*, care se definește în felul următor:

Eroare absolută Δ a valorii aproximative a se consideră modulul diferenței dintre valoarea exactă A și valoarea aproximativă a :

$$\Delta = |A - a|.$$

Eroarea absolută nu este un indice suficient pentru estimarea exactității calculului sau a măsu-

rărilor. Fie că în urma măsurării lungimii a două bare, cu lungimi de 20 m și 6 m, respectiv, au fost obținute rezultatele de măsurare 20,5 m și 6,2 m.

Cu toate că valoarea absolută a erorii în primul caz este mai mare, este evident că prima măsurare a fost realizată mult mai exact decât a doua. Pentru a determina calitatea măsurării (a calculului), se raportează mărimea erorii absolute la o unitate de lungime (sau, în caz general, la o unitate de măsură respectivă).

Eroare relativă δ a valorii aproximative se consideră raportul dintre eroarea absolută Δ și modulul numărului exact A ($A \neq 0$):

$$\delta = \frac{\Delta}{|A|}.$$

Exemplu: O bară are lungimea de 100 cm. În urma unei măsurări a fost stabilită o valoare a lungimii egală cu 101 cm. Distanța dintre punctele A și B este de 3 000 m. Ca rezultat al măsurării s-a obținut distanța de 2 997 m. Se cere determinarea erorii relative a fiecărei măsurări și măsurarea mai exactă.

Rezolvare:

pentru prima măsurare $\Delta = |100 - 101| = 1$ cm, $\delta = \frac{1}{100} = 0,01$;

pentru măsurarea a doua $\Delta = |3\,000 - 2\,997| = 3$ m, $\delta = \frac{3}{3\,000} = 0,001$.

Deoarece eroarea relativă a celei de-a doua măsurări este mai mică, această măsurare este mai exactă.

Întrebări și exerciții

- 1 Explicați noțiunea de eroare. Dați exemple de apariție a erorilor în situații reale.
- 2 Definiți noțiunea de eroare absolută. Exemplificați.
- 3 Definiți noțiunea de eroare relativă. Exemplificați. Cum se va modifica formula de calcul a erorii relative, dacă eroarea trebuie indicată în % de la valoarea exactă a mărimii cercetate?
- 4 Lungimea traseului dintre două localități, afișată pe indicatorul de drum, este de 230 km. Parcurgând acest traseu cu autovehiculul, ați fixat variația indicațiilor dispozitivului de măsurare a distanței – 230,7 km. Considerând datele indicatorului ca fiind exacte, determinați eroarea absolută și eroarea relativă a măsurării.
- 5 O bară cu lungimea exactă de 100 cm a fost măsurată cu o eroare absolută de 2 cm. Care sînt valorile posibile, obținute în procesul de măsurare?
- 6 Volumul exact al unui vas este de 20 l. Măsurările de volum au fost efectuate cu o eroare relativă de 0,001. Care sînt valorile posibile ale volumului măsurat?

2.2. Sursele erorilor de calcul

Erorile care apar în timpul rezolvării problemelor pot proveni din diferite surse. Cunoașterea surselor de apariție a erorilor permite ocolirea lor și minimizarea efectului cumulativ al erorilor. Cele mai des întîlnite tipuri de erori sînt:

a) *erori de problemă*;

- b) erori de metodă;
- c) erori ale datelor de intrare;
- d) erori de aproximare;
- e) erori de rotunjire.

Erori de problemă. Această categorie de erori apare în situațiile când modelul matematic ales pentru rezolvarea problemei nu descrie complet procesul real cercetat. Astfel, în *exemplul 2* (1.2. Modelul matematic și modelarea matematică), pentru a construi modelul matematic al sistemului *bilă-arc*, s-a utilizat ecuația oscilațiilor armonice pentru *deformații mici și în lipsa forței de frecare*. Prin urmare, rezultatul obținut prin utilizarea formulei va fi diferit de cel exact, diferența fiind cu atât mai semnificativă, cu cât e mai mare forța de frecare și deformarea arcului în sistemul real.

Erori de metodă. Este o categorie de erori generată de imposibilitatea determinării unei metode exacte de rezolvare a problemei sau de restricțiile care impun utilizarea unei metode mai puțin exacte. În acest caz problema inițială este rezolvată printr-o metodă euristică, care poate genera diferențe esențiale între rezultatul calculat și cel exact.

Un exemplu elocvent este utilizarea metodei Greedy pentru rezolvarea *problemei rucsacului*.

Erori ale datelor de intrare. Deseori procesul de modelare matematică se bazează pe rezultatele unor experiențe, adică pe niște seturi de mărimi numerice, obținute în urma măsurărilor. Aceste mărimi nu sînt exacte (ex.: distanța, masa, viteza).

Fie că un corp se mișcă pe o traiectorie descrisă pe segmentul $[0,1]$ de funcția $f(x) = x^2 + x + 1$. Se știe că valoarea argumentului x se calculează cu o eroare absolută care nu depășește $0,01$. Prin urmare, dacă z este valoarea exactă a argumentului, atunci $|z - x| < 0,01$.

Ne amintim!

Problema rucsacului:

Fie un rucsac de volum x și n obiecte de volume $v_i, i = 1 \dots n$ și costuri $c_i, i = 1 \dots n$. Se cere să se pună în rucsac obiecte din setul propus astfel, încît costul total al obiectelor puse în rucsac să fie maxim posibil.

Dacă pentru rezolvare este folosită metoda Greedy, în majoritatea cazurilor soluția obținută va fi diferită de cea optimă. Astfel, pentru un set din cinci obiecte cu volumele 5, 7, 13, 20, 10, costurile respective de 4, 8, 15, 23, 5 și un rucsac cu volumul de 30 de unități va genera soluția cu valoarea 27 (obiectele 3, 2, 1). (Sortarea obiectelor este realizată în descrescerea raportului cost/volum.) În realitate soluția optimă are valoarea 31 (obiectele 2 și 4).

În condițiile date se poate stabili în ce măsură eroarea la măsurarea valorilor lui x influențează rezultatul calculului:

$$\begin{aligned} |f(x) - f(z)| &= |x^2 + x + 1 - z^2 - z - 1| = \\ &= |x^2 + x - z^2 - z| = |(x - z)(x + z + 1)|. \end{aligned}$$

Deoarece $x + z + 1 \leq 3$ și $|z - x| < 0,01$, rezultă $|f(x) - f(z)| \leq 0,03$.

Diferența dintre valoarea funcției de argument exact (z) și valoarea funcției de argument măsurat (x) este o mărime constantă. De aici reiese că erorile de calcul nu depind de x , ci numai de exactitatea cu care acesta este măsurat, iar funcția este stabilă la erori.

Erori de aproximare. Este o categorie de erori generate de anumite definiții și noțiuni matematice. Prezența lor este acceptată în special în problemele care folosesc noțiunea de limită, convergență etc. Apariția acestui tip de erori este motivată de însăși structura definiției care conține elemente de aproximare.

Exemplu: Șirul $\{x_n\}$ este convergent și are limita x dacă pentru orice $\varepsilon > 0$, $\varepsilon \rightarrow 0$, există un rang $n(n_\varepsilon)$, astfel încît $\forall n > n_\varepsilon, |x - x_n| < \varepsilon$.

Din punctul de vedere al analizei numerice, ε indică precizia cu care x_n aproximează x .

În procesul de calcul al limitei se utilizează o aproximarare consecutivă, care se apropie tot mai mult de limita exactă, fără a o atinge. Stoparea procesului de calcul are loc atunci cînd deviația (eroarea) devine mai mică decît eroarea maximă admisibilă (ε).

De reținut că rezultatele obținute prin calcul numeric sînt acceptate doar ca rezultate cantitative și nu pot servi drept demonstrație pentru anumite afirmații matematice.

Erori de rotunjire. Este un tip aparte de erori, generate de faptul că în procesul prelucrării mărimilor numerice în calculator ele pot fi păstrate doar cu un anumit număr de semne zecimale după virgulă. Drept exemplu poate servi constanta π , valorile funcțiilor trigonometrice etc.

Fie $A = \{a_1, a_2, a_3, \dots, a_n\}$ mulțimea tuturor numerelor care pot fi reprezentate în calculator. (Se consideră că $a_1 < a_2 < a_3 < \dots < a_n$.)

Oricare din numerele $\frac{a_i + a_{i+1}}{2}$ lipsește în mulțimea inițială A . În cazul apariției unei asemenea situații de calcul, apare eroarea legată de înlocuirea rezultatului real prin numărul cel mai apropiat de el din mulțimea A . Această procedură este numită **rotunjire**.

Funcția de rotunjire aplicată la calculatoare se definește în felul următor:

$$\forall x \in \mathbb{R}, a_i \in A, a_{i+1} \in A \quad x \in (a_i, a_{i+1})$$

$$rd(x) = a_i \text{ dacă } x \in \left(a_i, \frac{a_i + a_{i+1}}{2} \right),$$

$$rd(x) = a_{i+1} \text{ dacă } x \in \left[\frac{a_i + a_{i+1}}{2}, a_{i+1} \right).$$

Utilizarea funcției de rotunjire generează abateri de la legile de bază ale operațiilor aritmetice, care nu mai sînt asociative, distributive.

Ne amintim!

Se consideră că funcția $f(x): I \rightarrow \mathbb{R}$ posedă proprietatea Lipschits dacă există o constantă $m > 0$, astfel încît: $|f(x) - f(z)| \leq m^* |z - x|$, $\forall x, z \in I$.

Se consideră că determinarea valorilor funcției $f(x)$ este stabilă la erori, dacă $f(x)$ posedă proprietatea Lipschits.

În alți termeni, stabilitatea funcției la erori presupune variații mici ale valorilor funcției la variații mici ale argumentului.

Info+

Apropierea valorilor termenilor șirului $1/n$ de limita șirului - 0.

$n = 1$	$1/n = 1$
$n = 2$	$1/n = 0,5$
...	
$n = 501$	$1/n = 0,001996$
$n = 502$	$1/n = 0,001992$
$n = 503$	$1/n = 0,001988$
...	
$n = 999$	$1/n = 0,001001$
$n = 1000$	$1/n = 0,001000$
$n = 1001$	$1/n = 0,000999$
...	

Exemplu: Fie că în procesul de calcul se poate opera cu cel mult 3 cifre după virgulă. În acest caz pentru numerele

$$a = 0,2334, b = 0,2331, c = 0,233$$

$$\begin{aligned} \text{se obține } rd(rd(a+b)+c) &= \\ &= rd(rd(0,4665)+0,233) = \\ &= rd(0,467+0,233) = \mathbf{0,7}; \\ rd(a+rd(b+c)) &= \\ &= rd(0,2334+rd(0,4661)) = \\ &= rd(0,2334+0,466) = \\ &= rd(0,6994) = \mathbf{0,699}. \end{aligned}$$

În cazul rotunjirii rezultatelor de calcul, erorile nu depășesc după modul valoarea de $0,5 \times 10^{-n}$ (aceste erori se numesc *erori de rotunjire absolute*), unde n este numărul de semne semnificative (care pot fi percepute) în procedura de calcul. Erorile de rotunjire pot fi atât pozitive, cât și negative. În cazul alternării lor, are loc procesul de compensare, ca rezultat eroarea finală nu crește odată cu numărul de calcule.

Întrebări și exerciții

- 1 Identificați principalele surse de erori. Exemplificați.
- 2 Dați exemple de erori generate de imposibilitatea formulării exacte a problemei. Motivați imposibilitatea formulării exacte.
- 3 Care funcții posedă proprietatea de stabilitate a calculului valorilor față de erori?
- 4 Este oare stabil față de erori calculul valorilor funcției liniare? Dar al funcției $y = \sqrt{x}$, $x \in [1,2]$?
- 5 Explicați esența erorilor de aproximare.
- 6 Soluția unei probleme se calculează iterativ după formula: $x_0 = 0$, $x_{i+1} = \sqrt{2 + x_i}$.
Va atinge oare șirul soluțiilor calculate valoarea soluției exacte? Stabiliți experimental sau analitic soluția exactă. După câte iterații eroarea absolută a soluției calculate va deveni mai mică de 0,0001?
- 7 Din care motiv rezultatele obținute prin metode numerice nu pot servi drept demonstrații ale afirmațiilor matematice?
- 8 Care este cauza apariției erorilor de rotunjire? Exemplificați.
- 9 Scrieți un program care determină produsul și cîtul numerelor 1,00000001 și 0,999999999. Valorile vor fi stocate în variabile, avînd tipul **real**. Analizați rezultatele obținute. Explicați cauzele apariției erorii.

Test de evaluare

1. Eroarea absolută Δ a mărimii aproximative a pentru valoarea exactă A este dată de formula:
 - a) $\Delta = |A - a|$;
 - b) $\Delta = |A| - |a|$;
 - c) $\Delta = |a| - |A|$;
 - d) $\Delta = a - A$.
2. Frecvența pe care emite un post de radio este de 105,2 MHz. În procesul de scanare a frecvențelor, stația de radio a stabilit pentru postul dat frecvența de 105,25 MHz. Determinați eroarea absolută și eroarea relativă cu care a fost stabilită frecvența de stația de radio. Care este efectul real al erorii, observat în procesul de lucru al stației?
3. Determinați câte cifre semnificative are fiecare din următoarele numere:
 - a) 0,375;
 - b) 0,000672;
 - c) -0,1233;
 - d) -0,00222222;
 - e) 0,010101.
4. Erorile de aproximare sînt erorile care apar din cauza:
 - a) modelului matematic incomplet;
 - b) insuficienței datelor de intrare;
 - c) specificului reprezentării numerelor în calculator;
 - d) metodei aproximative de rezolvare;
 - e) definițiilor și noțiunilor matematice care conțin elemente de aproximare.
5. În rezultatele calculelor realizate pentru procesarea tranzacțiilor financiare se poate opera cu cel mult 2 cifre după virgulă. Pentru numerele $a = 0,113$, $b = 0,162$, $c = 0,21$ calculați:
 - a) $rd(rd(a+b)+c)$;
 - b) $rd(a+rd(b+c))$.Care tip de erori este cauza diferențelor rezultatelor obținute?

Studiu de caz

Fie un container cu volum de 250 de unități și 5 obiecte avînd volumele de 120, 40, 40, 100, 150 de unități și costurile respective de 150, 60, 80, 120, 180. Se cere să se pună în container obiecte din setul propus astfel, încît costul total al obiectelor puse în container să fie maxim posibil.

- a) Determinați soluția problemei, utilizînd metoda Greedy. (Ordonarea obiectelor va fi realizată după descreșterea raportului cost/volum.)
- b) Determinați soluția exactă a problemei, folosind metoda trierii.
- c) Determinați eroarea absolută și eroarea relativă ale soluției particulare, obținute prin metoda Greedy.
- d) Explicați cauzele apariției erorilor.

CAPITOLUL 3

METODE NUMERICE DE REZOLVARE A ECUAȚIILOR ALGEBRICE ȘI TRANSCENDENTE

După studierea acestui capitol, veți fi capabili să:

- recunoașteți prezența soluțiilor unei ecuații algebrice sau transcendente pe un interval dat;
- separați intervalele domeniului de definiție a unei funcții $f(x)$, care vor conține exact o soluție a ecuației $f(x) = 0$;
- utilizați algoritmi de rezolvare a ecuațiilor algebrice și transcendente prin metoda bisecției, metoda coardelor și metoda tangențelor;
- elaborați programe de rezolvare a ecuațiilor algebrice și transcendente prin metoda bisecției, metoda coardelor și metoda tangențelor;
- combinați metodele studiate pentru elaborarea algoritmilor eficienți de rezolvare a ecuațiilor algebrice și transcendente și a programelor care realizează acești algoritmi.

3.1. Separarea soluțiilor ecuațiilor algebrice și transcendente

A rezolva ecuația algebrică sau transcendentă (în continuare *ecuația*) $f(x) = 0$ înseamnă a determina acele valori ale variabilei x pentru care egalitatea $f(x) = 0$ este una adevărată. În cazul când ecuația are o structură simplă, soluțiile ei pot fi determinate exact și relativ ușor prin metodele analitice, care se studiază în cadrul cursului liceal de matematică. Dacă însă structura ecuației este complicată, procedura de determinare a soluțiilor devine destul de anevoioasă. Mai mult decât atât, atunci când ecuația modelează anumite situații, fenomene care depind de mai mulți parametri, iar valoarea acestora este cunoscută doar aproximativ, noțiunea de soluție exactă își pierde în general sensul. Din acest motiv, este

util de a cunoaște și metodele de calcul aproximativ al soluțiilor ecuațiilor și algoritmi care realizează aceste metode.

Fie dată ecuația

$$f(x) = 0, \quad (1)$$

$f(x)$ fiind definită și continuă pe un oarecare interval $a \leq x \leq b$.

Ne amintim!

Dacă funcția $f(x)$ are forma unui polinom sau poate fi adusă la această formă, ecuația $f(x) = 0$ se numește **algebrică**.

Exemplu: $3x - 7 = 0$.

În caz contrar – când $f(x)$ nu este un polinomială –, ecuația se numește **transcendentă**.

Exemplu: $\sin(2x) + \sqrt{\cos^2 x + e^x} = 0$.

Orice valoare ξ , pentru care expresia $f(\xi) = 0$ este adevărată, se numește zerou al funcției $f(x)$ sau soluție a ecuației $f(x) = 0$.

În cele ce urmează se va presupune că ecuația (1) are soluții distincte (izolate), adică pentru fiecare soluție a ecuației există o vecinătate a sa, care nu conține alte soluții.

Astfel, rezolvarea prin metode numerice a unei ecuații se divide în două etape:

1. Separarea intervalelor pe care ecuația are o singură soluție.

2. Micșorarea pe cât mai mult posibil a fiecărui din aceste intervale (dacă se pune problema determinării tuturor soluțiilor) sau a unuia din ele (dacă trebuie de determinat doar una din soluții).

Metoda analitică. Pentru separarea analitică a soluțiilor vor fi folosite proprietățile derivatei.

Dacă soluțiile ecuației $f'(x)=0$ pot fi ușor calculate, atunci, pentru a separa soluțiile $f(x)=0$, este necesar:

1. să se determine soluțiile distincte $a \leq x_1 \leq x_2 \leq \dots, \leq x_n \leq b$ ale ecuației $f'(x)=0$;

2. considerînd $a = x_0$ și $b = x_{n+1}$, să se calculeze valorile $f(x_0), f(x_1), \dots, f(x_{n+1})$. Segmentele $[x_i, x_{i+1}]$, $i = 0, \dots, n$, pentru care $f(x_i) \times f(x_{i+1}) < 0$ vor conține câte cel puțin o soluție a ecuației $f(x)=0$.

Exemplul 1: Să se determine numărul de soluții ale ecuației $e^x + x = 0$

$$f'(x) = e^x + 1; \quad f'(x) > 0 \quad \forall x \in \mathbb{R}.$$

Întrucît $\lim_{x \rightarrow -\infty} f(x) = -\infty$, $\lim_{x \rightarrow \infty} f(x) = \infty$, ecuația inițială are o singură soluție.

Exemplul 2: Să se separe rădăcinile ecuației $x^3 - 9x^2 + 24x - 19 = 0$ pe segmentul $[0, 8]$.

Rezolvare:

$$f(x) = x^3 - 9x^2 + 24x - 19;$$

$$f'(x) = 3x^2 - 18x + 24.$$

Rezolvînd ecuația $f'(x) = 0$, se obțin soluțiile $x_1 = 2$, $x_2 = 4$.

x	$f(x)$	Semn $f(x)$
0	-19	-
2	1	+
4	-3	-
8	109	+

Deci ecuația va avea trei soluții, câte una pe fiecare din segmentele $[0, 2]$, $[2, 4]$, $[4, 8]$.

Metoda grafică. O altă posibilitate de separare a rădăcinilor ecuației $f(x) = 0$ este cercetarea directă a graficului funcției $f(x)$. Pentru construcția acestuia pot fi folosite atît aplicații software specializate¹, cît și programe simple, elaborate cu ajutorul instrumentelor unui limbaj de programare.

Ne amintim!

Teoremă. Dacă funcția $f(x)$, continuă pe segmentul $[a, b]$, primește la extremitățile lui valori de semn diferit ($f(a) \times f(b) < 0$), atunci pe acest segment există cel puțin un punct ξ , astfel încît $f(\xi) = 0$. Dacă pe $[a, b]$ există derivata $f'(x)$, continuă, care are un semn constant, atunci ξ este soluție unică a ecuației $f(x) = 0$ pe acest segment.

¹ MatCAD, Matematica, Derive etc.

Separarea grafică a soluțiilor unei ecuații pe un segment dat poate fi realizată și local, cu ajutorul unei aplicații de calcul tabelar. Este suficient să se construiască un tabel cu două coloane. Prima coloană va reprezenta o divizare a segmentului în segmente elementare de lungimi egale. Cea de-a doua coloană va conține o formulă care calculează valoarea funcției $f(x)$ pentru valorile respective din prima coloană. În baza datelor din coloana cu valorile $f(x)$ se construiește o diagramă liniară, care reprezintă graficul funcției analizate.

Exemplu: $f(x) = x^{\cos(2x)} + 3\sin(x)$. Soluțiile se caută pe segmentul $[0,2, 10]$ (fig. 3.1 a, 3.1 b).

	A	B	C	D	E	F
1	x	y				
2	0,2	0,823102167				
3	0,4	1,696399241				
4	0,6	2,524907747				
46	9,2	5,503155521				
47	9,2	8,048154414				
48	9,4	9,448504359				
49	9,6	7,844007308				
50	9,8	4,209027748				
51	10	0,927006056				

Fig. 3.1a. Funcția $f(x)$ reprezentată tabelar în foaia de calcul. Colana A conține valorile x de la 0,2 la 10 cu pasul 0,2. Coloana B conține valorile $f(x)$

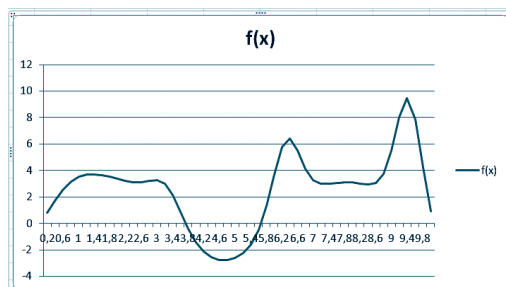


Fig. 3.1b. Funcția $f(x)$ reprezentată grafic în baza datelor din tabel. Pot fi ușor identificate două segmente oarecare, ce conțin exact câte o soluție a ecuației $f(x)=0$, de exemplu: $[3, 5]$ și $[5, 6]$

Întrebări și exerciții

- 1 Ce numim soluție a unei ecuații?
- 2 Ce condiții trebuie să satisfacă funcția $f(x)$, pentru ca pe un segment dat să fie cel puțin o rădăcină a ecuației $f(x)=0$? Dar pentru existența exact a unei soluții?
- 3 Determinați analitic numărul de soluții reale ale ecuației:
 - a) $x^5 - 5x + 7 = 0$;
 - b) $x^3 - 9x^2 + 24x - 13 = 0$.
- 4 Elaborați un program, care pentru funcția $f(x)$, continuă pe $[a, b]$, realizează o divizare a segmentului în n segmente de lungime egală și afișează toate segmentele la extremitățile cărora funcția are valori de semn opus:
 - a) $f(x) = x^3 - 7x^2 + 12x - 37$ pe $[-10, 10]$, $n = 100$;
 - b) $f(x) = \sin(3x) + 3\cos(x) - 1$ pe $[-2, 2]$, $n = 100$.
- 5 Separați, folosind cea mai potrivită metodă, soluțiile ecuațiilor:
 - a) $\frac{x^4}{4} + x^3 - \frac{x^2}{2} - 3x - 8 = 0$;
 - b) $2x^3 - 6x^2 - 48x + 17 = 0$;
 - c) $x^4 - 14x^2 - 24x - 4 = 0$;
 - d) $\frac{1}{2} - e^{-x^2} = 0$;
 - e) $e^x(x^2 - 2x + 2) = 0$;
 - f) $x[\sin(\ln(x)) - \cos(\ln(x))] = 0$;
 - g) $\ln(x) + \sqrt{\sin(3x) + 7} - x^2 = 0$.

3.2. Metoda biseecției

Fie dată funcția $f(x)$, continuă pe segmentul $[a, b]$, și $f(a) \times f(b) < 0$.
Se cere să se determine pe segmentul $[a, b]$ o soluție a ecuației

$$f(x) = 0. \tag{1}$$

Proprietățile funcției asigură existența a cel puțin unei soluții pe acest segment.

Una dintre cele mai simple metode de determinare a unei soluții a ecuației $f(x) = 0$ este *metoda biseecției*. Metoda presupune determinarea punctului de mijloc c al segmentului $[a, b]$, apoi calculul valorii $f(c)$. Dacă $f(c) = 0$, atunci c este soluția exactă a ecuației. În caz contrar, soluția este căutată pe unul dintre segmentele $[a, c]$ și $[c, b]$. Ea va aparține segmentului pentru care semnul funcției în extremități este diferit (fig. 3.2).

Dacă $f(a) \times f(c) > 0$, atunci soluția e căutată în continuare pe segmentul $[a_1, b_1]$, unde a_1 primește valoarea c , iar b_1 – valoarea b . În caz contrar, a_1 primește valoarea a , iar b_1 – valoarea c . Procesul de divizare se reia pe segmentul $[a_1, b_1]$, repetindu-se pînă cînd nu se obține soluția exactă sau (în majoritatea absolută a cazurilor!) devierea soluției calculate c_i de la cea exactă nu devine suficient de mică.

În urma divizărilor succesive se obține consecutivitatea segmentelor

$$[a_0, b_0], [a_1, b_1], [a_2, b_2], \dots, [a_p, b_p], \dots$$

Pentru fiecare dintre ele are loc relația $f(a_i) \times f(b_i) < 0, i = 0, 1, 2, \dots$ (2)

Estimarea erorii. Deoarece soluția exactă ξ a ecuației este un punct al segmentului $[a_p, b_p]$, rezultă că diferența dintre soluția exactă și cea calculată nu depășește lungimea acestui segment. Prin urmare, localizarea soluției pe un segment cu lungimea ε asigură o eroare de calcul a soluției ce nu depășește valoarea ε :

$$|\xi - c_i| < \varepsilon = |b_i - a_i|.$$

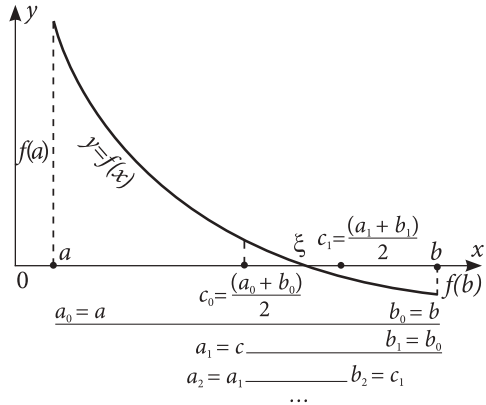


Fig. 3.2. Calculul consecutiv al segmentelor, care conțin soluția ecuației $f(x)=0$

ALGORITMIZAREA METODEI

Pornind de la descrierea matematică a metodei, putem separa două cazuri distincte de oprire a procesului de calcul al soluției ecuației $f(x) = 0$ pentru metoda biseecției:

A1. Algoritmul de calcul pentru un număr prestabilit n de divizări consecutive:

Pasul 0. Inițializare: $i \leftarrow 0$.

Pasul 1. Determinarea mijlocului segmentului $c \leftarrow \frac{a+b}{2}$.

Pasul 2. Reducerea segmentului ce conține soluția: dacă $f(c) = 0$, atunci soluția calculată este $x = c$. SFÎRȘIT.

În caz contrar, dacă $f(a) \times f(c) > 0$, atunci $a \leftarrow c; b \leftarrow b$, altfel $a \leftarrow a; b \leftarrow c$.

Pasul 3. $i \leftarrow i + 1$. Dacă $i = n$, atunci soluția calculată este $x = \frac{a+b}{2}$. SFÎRȘIT.

În caz contrar, se revine la *pasul 1*.

A2. Algoritmul de calcul pentru o precizie² ε dată:

Pasul 1. Determinarea mijlocului segmentului $c \leftarrow \frac{a+b}{2}$.

Pasul 2. Dacă $f(c) = 0$, atunci soluția calculată este $x = c$. SFÎRȘIT.

În caz contrar, dacă $f(a) \times f(c) > 0$, atunci $a \leftarrow c$; $b \leftarrow b$, altfel $a \leftarrow a$; $b \leftarrow c$.

Pasul 3. Dacă $|b - a| < \varepsilon$, atunci soluția calculată este $x = \frac{a+b}{2}$. SFÎRȘIT.

În caz contrar, se revine la *pasul 1*.

Exemplul 1: Să se determine o rădăcină a ecuației $x^4 + 2x^3 - x - 1 = 0$ pe segmentul $[0, 1]$ pentru 16 divizări consecutive.

Deoarece numărul de aproximări succesive este fixat, iar extremitățile segmentului cunoscute, atribuirile se realizează nemijlocit în program.

```
program cn05;
var a,b,c: real;
    i,n: integer;

function f(x: real): real;
begin f:=sqr(sqr(x))+2*x*sqr(x)-x-1;end;
begin a:=0; b:=1; n:=16;
      for i:=1 to n do
        begin c:=(b+a)/2;
              writeln('i=',i:3,' x=',c:10:8,' f(x)=',f(c):12:8);
              if f(c)=0 then break3
              else if f(c)*f(a)>0 then a:=c else b:=c;
        end;
      end.
```

Rezultate:

```
i= 1 x=0.50000000 f(x)= -1.18750000
i= 2 x=0.75000000 f(x)= -0.58984375
...
i= 15 x=0.86679077 f(x)= 0.00018565
i= 16 x=0.86677551 f(x)= 0.00009238
```

Exemplul 2: Să se determine o rădăcină a ecuației $6\cos(x) + 8\sin(x) = 0$ pe segmentul $[2, 4]$ cu precizia $\varepsilon=0,00017$.

```
program cn06;
var a,b,c,eps: real;

function f(x: real): real;
begin f:=6*cos(x)+8*sin(x);end;
begin a:=2; b:=4; eps:=0.00017;
      repeat
        c:=(b+a)/2;
        writeln('x=',c:10:8,' f(x)=',f(c):12:8);
        if f(c)=0 then break
        else if f(c)*f(a)>0 then a:=c else b:=c;
      until abs(b-a)<eps;
      end.
```

² În contextul dat precizia ε semnifică o eroare de calcul, care nu depășește valoarea ε .

³ Instrucțiune de salt necondiționat care întrerupe execuția instrucțiunii ciclice în care se conține.

Rezultate:

```
x=3.00000000 f(x) = -4.81099492
x=2.50000000 f(x) = -0.01908454
...
x=2.49829102 f(x) = -0.00199471
x=2.49816895 f(x) = -0.00077401
```

Întrebări și exerciții

- 1 În ce cazuri se folosesc metode aproximative de determinare a soluțiilor ecuațiilor algebrice? Descrieți metoda biseecției. Care sînt prioritățile ei? Dar neajunsurile? Formula pentru estimarea erorii, dedusă în paragraful curent, este $|\xi - c_i| < \varepsilon = |b_i - a_i|$. Exprimați diferența dintre valoarea exactă și cea calculată printr-o formulă care depinde doar de extremitățile segmentului inițial și numărul de divizări realizate.
- 2 Descrieți algoritmul metodei biseecției.
- 3 Determinați prin metoda biseecției soluțiile ecuațiilor:
 1. $e^x - x^2 = 0$ pe $[-1, -0,5]$;
 2. $x^3 - x - 1 = 0$ pe $[1, 2]$;
 3. $x^3 + 3x^2 - 3 = 0$ pe $[-3, -2]$;
 4. $x^5 - x - 2 = 0$ pe $[1, 2]$.a) pentru 10, 20, 40 de divizări ale segmentului inițial;
b) cu precizia $\varepsilon = 0,001; 0,0001; 0,00001$;
c) în condițiile punctului precedent determinați numărul de divizări necesare pentru a obține precizia cerută.
- 4 Determinați prin metoda biseecției soluțiile ecuațiilor de pe intervalele separate în exercițiile 3, 4, 5, p. 24, pentru 10, 20, 30 de divizări consecutive.

3.3. Metoda coardelor

Metoda biseecției, cu toată simplitatea ei, nu este eficientă în cazurile cînd rezultatul trebuie obținut printr-un număr redus de iterații, cu o exactitate înaltă. Astfel stînd lucrurile, este mai potrivită *metoda coardelor*, care constă în divizarea segmentului în părți proporționale, proporția fiind dată de punctul de intersecție al coardei care unește extremitățile segmentului cu axa $0x$.

Fie dată funcția $f(x)$, care posedă următoarele proprietăți:

1. $f(x)$ continuă pe segmentul $[a, b]$ și $f(a) \times f(b) < 0$.
2. Pe segmentul $[a, b]$ există $f'(x) \neq 0$; $f''(x) \neq 0$, continui, iar semnul lor pe $[a, b]$ este constant.

Proprietățile enumerate garantează existența soluției unice a ecuației $f(x) = 0$ pe $[a, b]$.

Metoda coardelor presupune alegerea în calitate de aproximare a soluției punctul determinat de intersecția dreptei ce trece prin punctele $(a, f(a))$ și $(b, f(b))$ cu axa $0x$.

Pentru realizarea metodei se stabilește extremitatea e a segmentului $[a, b]$ prin care se va duce o serie de coarde (fig. 3.3). Această extremitate este determinată de condiția:

$$f(e) \times f''(e) > 0.$$

Cealaltă extremitate a segmentului $[a, b]$ se consideră aproximare inițială a soluției: x_0 . Prin punctele $(e, f(e))$ și $(x_0, f(x_0))$ se construiește o coardă. Se determină punctul x_1

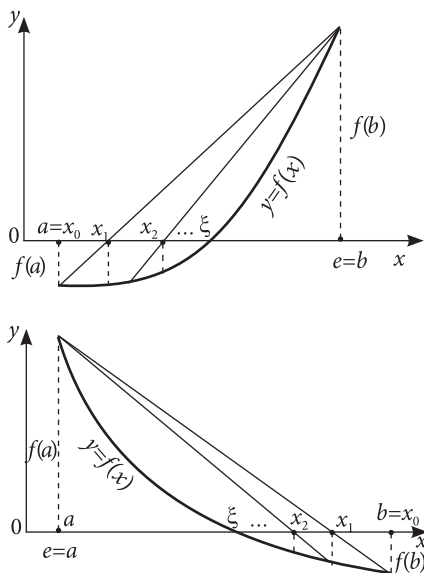


Fig. 3.3. Aproximarea succesivă de soluția ecuației prin metoda coardelor

Ne amintim!

Teorema Lagrange

Fie $f: [a, b] \rightarrow \mathbb{R}$, continuă și derivabilă, pe $[a, b]$. Atunci există $c \in (a, b)$, astfel încât:

$$f(b) - f(a) = (b-a)f'(c).$$

în care coarda intersectează axa Ox . Punctul x_1 este considerat următoarea aproximare a soluției.

Procesul se repetă, coarda următoare fiind dusă prin punctele $(e, f(e))$ și $(x_1, f(x_1))$. Astfel se obține șirul de aproximări $x_0, x_1, x_2, \dots, x_i, x_{i+1}, \dots, x_n, \dots$, limita căruia este soluția exactă a ecuației $f(x) = 0$.

Punctele e și x_0 sînt cunoscute. Folosind ecuația dreptei ce trece prin două puncte, putem determina aproximarea x_1 ($f(x_1) = 0$):

$$\frac{x - x_0}{e - x_0} = \frac{y - f(x_0)}{f(e) - f(x_0)},$$

de unde

$$x_1 = x_0 - \frac{f(x_0)}{f(e) - f(x_0)}(e - x_0).$$

În general, avînd calculată aproximarea x_{i-1} , putem determina următoarea aproximare x_i prin formula recurentă:

$$x_i = x_{i-1} - \frac{f(x_{i-1})}{f(e) - f(x_{i-1})}(e - x_{i-1}), \quad (3)$$

$$i = 1, 2, \dots$$

Se demonstrează că șirul de valori $x_1, x_2, \dots, x_p, x_{i+1}, \dots, x_n, \dots$ calculate după formula (3) converge către soluția ξ a ecuației $f(x) = 0$.

Eroarea metodei

Faptul că șirul aproximărilor succesive prin metoda coardelor converge către soluția exactă implică următoarea concluzie: eroarea soluției calculate va fi invers proporțională cu numărul de iterații efectuate. Totuși se poate determina și formula care permite estimarea erorii de calcul.

Fie $f(x)$ satisface condițiile (1), (2). Dacă ξ - soluția exactă a ecuației $f(x) = 0$ pe segmentul $[a, b]$, iar M_1 și m_1 - marginea superioară și inferioară a $f'(x)$ pe același segment, din teorema Lagrange și formula recurentă pentru calculul aproximărilor succesive rezultă:

$$|\xi - x_i| \leq \left| \frac{M_1 - m_1}{m_1} \right| \times |x_i - x_{i-1}|, \text{ sau } \left| \frac{M_1 - m_1}{m_1} \right| \times |x_i - x_{i-1}| \leq \varepsilon. \quad (4)$$

Prin urmare, dacă se cere calculul soluției cu o exactitate dată ε , calculele se vor repeta conform formulei (3) pînă cînd inegalitatea (4) nu va deveni una adevărată.

ALGORITMIZAREA METODEI

Aplicarea metodei coardelor necesită o cercetare prealabilă a funcției $f(x)$, pentru stabilirea extremității fixe, din care vor fi trasate coardele. Numărul n de aproximări succesive ale soluției poate fi indicat în enunțul problemei sau determinat de o condiție.

În ambele cazuri calculul se realizează conform formulei (3). Condiția de oprire în primul caz va fi aplicarea repetată de n ori a formulei (3); în cel de al doilea – îndeplinirea condiției (4).

Determinarea extremității fixe. Pentru a evita calculul $f''(x)$, se va folosi următorul procedeu: se determină semnul $f(x)$ în punctul c de intersecție cu axa Ox al dreptei care trece prin punctele $(a, f(a))$ și $(b, f(b))$. **Fixă va fi acea extremitate e a segmentului $[a, b]$, pentru care se îndeplinește condiția: $f(e) \times f(c) < 0$.**

A1. Algoritm de calcul pentru un număr prestabilit n de aproximări succesive:

Pasul 1. Determinarea extremității fixe e și a aproximării x_0 :

$$c \leftarrow a - \frac{f(a)}{f(b) - f(a)}(b - a);$$

dacă $f(c) \times f(a) < 0$, atunci $e \leftarrow a$, $x_0 \leftarrow b$, altfel $e \leftarrow b$, $x_0 \leftarrow a$; $i \leftarrow 0$.

Pasul 2. Calculul x_{i+1} conform formulei $x_{i+1} = x_i - \frac{f(x_i)}{f(e) - f(x_i)}(e - x_i)$.

Pasul 3. Dacă $i + 1 = n$, atunci soluția calculată $x \leftarrow x_i$. SFÎRȘIT.

În caz contrar, $i \leftarrow i + 1$ și se revine la *pasul 2*.

A2. Algoritm de calcul pentru o exactitate ε dată:

Deoarece în formula de estimare a erorii figurează mărimile M_1 și m_1 , atunci când valorile lor nu sînt indicate în enunțul problemei, este necesară descrierea analitică a $f'(x)$ și calcularea M_1 și m_1 .

Pasul 1. Determinarea extremității fixe e și a aproximării x_0 :

$$c \leftarrow a - \frac{f(a)}{f(b) - f(a)}(b - a);$$

dacă $f(c) \times f(a) < 0$, atunci $e \leftarrow a$, $x_0 \leftarrow b$, altfel $e \leftarrow b$, $x_0 \leftarrow a$; $i \leftarrow 0$.

Pasul 2. Calculul x_{i+1} conform formulei $x_{i+1} = x_i - \frac{f(x_i)}{f(e) - f(x_i)}(e - x_i)$.

Pasul 3. Dacă $\left| \frac{M_1 - m_1}{m_1} \right| \times |x_{i+1} - x_i| \leq \varepsilon$, atunci soluția calculată $x \leftarrow x_i$. SFÎRȘIT.

În caz contrar, $i \leftarrow i + 1$ și se revine la *pasul 2*.

Exemplul 1: Fie dată funcția $f(x) = \ln(x \sin x)$. Să se calculeze soluția aproximativă a ecuației $f(x) = 0$ pe segmentul $[0,5; 1,5]$ pentru 10 aproximări succesive, utilizînd metoda coardelor.

Pentru acest exemplu, preprocesarea matematică nu este necesară.

Deoarece numărul de aproximări succesive este fixat, iar extremitățile segmentului cunoscute, atribuirile respective vor fi realizate direct în corpul programului.

```

program cn07;
var  a,b,e,c,x: real;
     n,i: integer;

function f(x:real):real;
begin f:=ln(x*sin(x));end;

```

```

begin a:=0.5; b:=1.5; n:=10;
      {determinarea extremitatii fixe e si a aproximarii initiale x0}
      c:=(f(a))/(f(b)-f(a))*(b-a);
      if f(c)*f(a)>0 then begin e:=b; x:=a; end
      else begin e:=a; x:=b; end;
      {calculul iterativ al solutiei}
      for i:=1 to n do
        begin x:= x-(f(x))/(f(e)-f(x))*(e-x);
              writeln(x:10:8, ' ', f(x):12:8);
        end;
end.

```

Rezultate:

```

i= 1 x=1.27995775 f(x)= 0.20392348
i= 2 x=1.18251377 f(x)= 0.09028687
...
i= 9 x=1.11427651 f(x)= 0.00016577
i=10 x=1.11420523 f(x)= 0.00006678

```

Exemplu 2: Fie dată funcția $f(x) = x^4 - 3x^2 + 7,5x - 1$. Să se calculeze soluția aproximativă a ecuației $f(x) = 0$ pe segmentul $[-0,5; 0,5]$ cu exactitatea $\varepsilon = 0,0001$, utilizând metoda cordelor. Pentru funcția dată pe $[-0,5; 0,5]$ M_1 și m_1 sînt, respectiv, egale cu 10 și 5. Pentru simplitate, atribuirile necesare vor fi realizate direct în corpul programului.

```

program cn08;
var
  Msup,minf,a,b,e,x,xnou,xvechi,eps: real;
function f(x:real):real;
begin
  f:=sqr(sqr(x))-3*sqr(x)+7.5*x-1;
end;
begin
  a:=-0.5; b:=0.5; eps:=0.0001;
  Msup:=10; minf:=5;
  {determinarea extremitatii fixe si a aproximarii initiale}
  x:=(f(a))/(f(b)-f(a))*(b-a);
  if f(x)*f(a)>0 then begin e:=b; xnou:=a; end
  else begin e:=a; xnou:=b; end;
  {calculul iterativ al solutiei}
  repeat
    xvechi:=xnou;
    xnou:= xvechi-(f(xvechi))/(f(e)-f(xvechi))*(e-xvechi);
    writeln(' x=',xnou:10:8, ' f(x)=',f(xnou):12:8);
  until abs((Msup-minf)/minf*(xnou-xvechi))<eps;
end.

```

Rezultate:

```

x=0.22500000 f(x)= 0.53818789
x=0.15970438 f(x)= 0.12191694
...
x=0.14130134 f(x)= 0.00026052
x=0.14127062 f(x)= 0.00005579

```

Întrebări și exerciții

- ❶ Explicați esența metodei coardelor. Descrieți metoda grafic.
- ❷ Cum depinde extremitatea fixă de semnul $f''(x)$?
- ❸ Descrieți procesul de determinare a extremității fixe. Cum poate fi omis calculul $f''(x)$?
- ❹ Descrieți pe pași algoritmul metodei coardelor pentru un număr fix de iterații.
- ❺ Determinați soluțiile ecuațiilor, utilizând metoda coardelor, pentru 10, 20, 30 de iterații:
 - a) $x^3 - 0,2x^2 + 0,2x - 2,1 = 0$ pe $[1, 2]$;
 - b) $5x^3 - 20x + 3 = 0$ pe $[0, 1]$;
 - c) $e^x - x^2 = 0$ pe $[-1, 0]$.
- ❻ Separați soluțiile ecuațiilor care urmează. Rezolvați ecuațiile folosind metoda coardelor:
 - a) $\text{tg}(0,55x + 0,1) - x^2 = 0$ pentru 5, 25 de iterații;
 - b) $x^3 - 0,2x^2 + 0,5x + 1,5 = 0$ pentru 3, 9, 27 de iterații.
- ❼ Determinați prin metoda coardelor soluțiile ecuațiilor de pe intervalele separate în exercițiile 3, 4, 5, p. 24, pentru 10, 20, 30 de iterații. Comparați rezultatele cu cele obținute în exercițiul 4, p. 27. Explicați diferențele.
- ❽ Fie dată funcția $f(x) = x[\sin(\ln(x)) - \cos(\ln(x))]$. Determinați soluția ecuației $f(x) = 0$ pe segmentul $[2, 3]$ cu exactitatea $\varepsilon = 0,0001$, utilizând metoda coardelor.

3.4. Metoda Newton

Fie dată funcția $f(x)$, care posedă următoarele proprietăți:

1. $f(x)$, continuă, pe segmentul $[a, b]$ și $f(a)f(b) < 0$.
2. Pe segmentul $[a, b]$ există $f'(x) \neq 0, f''(x) \neq 0$, continui, și semnul lor pe $[a, b]$ este constant.

Urmează să se rezolve ecuația $f(x) = 0$ pentru $x \in [a, b]$. Se va încerca rezolvarea problemei prin trasarea consecutivă a unor tangente la graficul funcției, prima dintre ele fiind construită prin extremitatea $E_0(x_0, y_0)$ a segmentului $[a, b]$, extremitate pentru care se respectă condiția: $f(x_0) \times f''(x_0) > 0$.

Fie că tangenta cu numărul i intersectează axa Ox în punctul x_i (fig. 3.4). Următoarea tangentă ($i+1$) va fi trasată prin punctul E_{i+1} cu coordonatele $(x_i, f(x_i))$ și va intersecta axa absciselor în punctul x_{i+1} . Șirul de valori $x_0, x_1, x_2, \dots, x_i, x_{i+1}, \dots$ va converge către soluția ecuației $f(x) = 0$. Această metodă de calcul al soluției ecuației $f(x) = 0$ este numită *metoda tangentelor* sau *Newton*, după numele matematicianului care a introdus-o.

Pentru a calcula valorile $x_1, x_2, \dots, x_i, \dots$, se va folosi ecuația tangentei la funcția ce trece printr-un punct dat:

$$y - f(x_i) = f'(x_i)(x - x_i). \quad (5)$$

În caz general ecuația (5) reprezintă tangenta la funcția $f(x)$, care trece prin punctul $(x_i, f(x_i))$. Ea va intersecta axa absciselor în punctul cu coordonatele $(x_{i+1}, 0)$. În consecință se obține:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}. \quad (6)$$

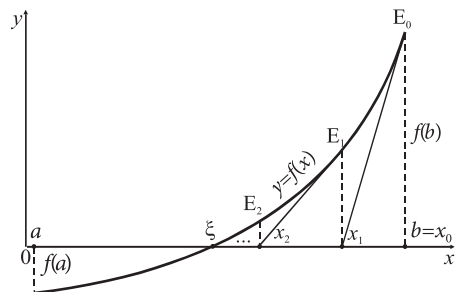


Fig. 3.4. Convergența șirului de valori $x_0, x_1, x_2, \dots, x_i, x_{i+1}, \dots$ către soluția exactă ξ

Eroarea metodei

Procesul iterativ de calcul poate fi oprit fie după repetarea unui număr prestabilit de ori, fie după atingerea unei exactități cerute.

Eroarea se va estima conform formulei:

$$\varepsilon = |\xi - x_{i+1}| \leq \frac{M_2}{2m_1} (x_{i+1} - x_i)^2, \quad (7)$$

unde

x_i, x_{i+1} – două aproximări succesive ale soluției calculate,

M_2 – supremul $f''(x)$ pe $[a, b]$,

m_1 – infimumul $f'(x)$ pe $[a, b]$.

ALGORITMIZAREA METODEI

Numărul de aproximări succesive în procesul de calcul poate fi stabilit apriori sau determinat de o condiție. Mai întâi se stabilește extremitatea segmentului care va servi drept aproximare inițială. Calculul aproximării următoare se realizează în ambele cazuri conform formulei (6). Condiția de oprire a calculelor va fi în primul caz generarea aproximării cu indicele cerut; în cel de al doilea – îndeplinirea condiției (7).

A1. Algoritmul de calcul pentru un număr dat de aproximări succesive:

Pentru a realiza acest algoritm, este suficient să fie cunoscute descrierile analitice pentru $f(x)$ și $f'(x)$. Dacă descrierea $f'(x)$ nu este indicată în enunț, urmează să fie calculată. Aproximarea inițială se deduce utilizând procedeul similar determinării extremității fixe pentru metoda coardelor.

Pasul 1. Determinarea aproximării inițiale x_0 : $c \Leftarrow a - \frac{f(a)}{f(b) - f(a)}(b - a)$;

dacă $f(c) \times f(a) < 0$, atunci $x_0 \Leftarrow a$, altfel $x_0 \Leftarrow b$; $i \Leftarrow 0$.

Pasul 2. Se calculează x_{i+1} conform formulei $x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$.

Pasul 3. Dacă $i+1 = n$, atunci soluția calculată $x \Leftarrow x_{i+1}$. SFÎRȘIT.

În caz contrar, $i \Leftarrow i+1$, apoi se revine la *pasul 2*.

A2. Algoritmul de calcul pentru o exactitate ε dată:

În formula de estimare a erorii figurează mărimile M_2 și m_1 . Atunci când valorile lor nu sînt indicate în enunțul problemei, este necesară o preprocesare matematică pentru stabilirea M_2 și m_1 . Suplimentar sînt necesare descrierile analitice pentru $f(x)$ și $f'(x)$.

Pasul 1. Determinarea aproximării inițiale x_0 : $c \Leftarrow a - \frac{f(a)}{f(b) - f(a)}(b - a)$;

dacă $f(c) \times f(a) < 0$, atunci $x_0 \Leftarrow a$, altfel $x_0 \Leftarrow b$; $i \Leftarrow 0$.

Pasul 2. Se calculează x_{i+1} conform formulei $x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$.

Pasul 3. Dacă $\frac{M_2}{2m_1} (x_{i+1} - x_i)^2 \leq \varepsilon$, atunci soluția calculată $x \Leftarrow x_{i+1}$. SFÎRȘIT.

În caz contrar, $i \Leftarrow i+1$ și se revine la *pasul 2*.

Exemplul 1³: Fie dată funcția $f(x) = x^3 - 2x^2 + x - 3$. Să se scrie un program care va calcula soluția ecuației $f(x) = 0$ pe segmentul [2; 15] pentru 10 aproximări succesive, utilizând metoda Newton.

Preprocesarea matematică: Se determină $f'(x)$:

$$f(x) = x^3 - 2x^2 + x - 3; \quad f'(x) = 3x^2 - 4x + 1.$$

Deoarece numărul de aproximări succesive este fixat, iar extremitățile segmentului cunoscute, atribuirile necesare se vor realiza direct în corpul programului.

```

program cn09;
var a, b, x, c : real;
    i, n: integer;
function f(z:real):real;
begin f:=z*z*z-2*z*z+z-3; end;
function fd1(z:real):real;
begin fd1:=3*z*z-4*z+1; end;
begin a:=2.1; b:=15; n:=10; i:=0;
      c:=a-(f(a))/(f(b)-f(a))*(b-a);
      if f(c)*f(a)<0 then x:=a else x:=b;
      while i<n do
begin i:=i+1;
      x:=x-f(x)/fd1(x);
      writeln('i=',i:2,' x=',x:15:12, ' f=',f(x):15:12);
end;
end.

```

Rezultate:

i= 1	x= 10.23214285700	f=869.11072454000
i= 2	x= 7.06207637180	f=256.52261987000
...		
i= 9	x= 2.17455942470	f= 0.00000009329
i=10	x= 2.17455941030	f= 0.00000000001

Exemplul 2: Fie dată funcția $f(x) = \cos^2(x) - \frac{x}{4}$. Să se scrie un program care va calcula soluția aproximativă a ecuației $f(x) = 0$ pe segmentul [2,4; 3] cu exactitatea $\varepsilon = 0,0001$, utilizând metoda Newton. Pentru funcția dată pe segmentul [2,4; 3] M_2 și m_1 sînt, respectiv, egale cu 2 și 0,03.

Preprocesarea matematică: $f(x) = \cos^2(x) - \frac{x}{4}$; $f'(x) = -\sin(2x) - \frac{1}{4}$.

Deoarece ε este dat, extremitățile segmentului și valorile M_2 , m_1 - cunoscute, atribuirile vor fi realizate direct în program.

```

program cn10;
var a, b, xn, xv, M2, m1, e, c : real;
function f(z:real):real;
begin f:=cos(z)*cos(z)-z/4; end;

```

³ Pentru toate exemplele și exercițiile propuse se presupune îndeplinirea condițiilor 1. și 2. (p. 31) de $f(x)$.

```

function fdl(z:real):real;
  begin fdl:=-sin(2*z)-1/4; end;
begin  a:=2.4; b:=3; M2:=2; m1:=0.03; e:=0.0001;
      c:=a-(f(a))/(f(b)-f(a))*(b-a);
      if f(c)*f(a)<0 then begin
        xn:=a; xv:=b;
        end
      else begin xn:=b; xv:=a; end;
  while M2*sqr(xn-xv)/(2*m1)>e do
  begin  xv:=xn;
        xn:=xv-f(xv)/fdl(xv);
        writeln(' x=',xn:15:12, ' f=',f(xn):15:12);
  end;
end.

```

Rezultate:

x=	2.47538619170	f=	-0.00078052066
x=	2.47646766320	f=	-0.00000027700
x=	2.47646804730	f=	0.00000000000

Întrebări și exerciții

- 1 Descrieți sensul geometric al metodei Newton.
- 2 Cum poate fi stabilită aproximarea inițială a metodei?
- 3 Explicați de ce, la alegerea corectă a punctului inițial, șirul aproximărilor obținute prin metoda Newton converge către soluția exactă a ecuației.
- 4 Scrieți un program care va determina pe fiecare din intervalele propuse soluțiile ecuațiilor care urmează, pentru 2, 4, 6 iterații, utilizând metoda Newton:
 - a) $4x^4 + 8x^3 - 3x^2 - 7x + 3 = 0$ pe $[-1,7; -1,58]$, $[-1,53; -1,4]$, $[0,4; 0,52]$, $[0,58; 0,8]$;
 - b) $\sqrt{x + \sqrt{x + \sqrt{x}}} - 12 = 0$ pe $[100; 150]$;
 - c) $(2 - x^2) \cos(x) + 2x \sin(x) = 0$ pe $[-4,5; -4]$ $[4; 4,5]$;
 - d) $\ln\left(\frac{1}{x} + \ln\left(\frac{1}{x} + \ln\left(\frac{1}{x}\right)\right)\right) - 2 = 0$ pe $[0,1; 0,5]$.
- 5 Modificați programul elaborat în exercițiul 4, pentru a calcula soluțiile ecuațiilor, utilizând metoda Newton, pentru $\varepsilon = 0,00001$:
 - a) $2\cos^2(x) - e^{x/2} = 0$ pe $[0,1; 0,74]$, $M_2=4$, $m_1=0,5$;
 - b) $x^5 - 4x + 9 = 0$ pe $[-2; -1]$, $M_2=150$, $m_1=1$.
- 6 Fie dată funcția $f(x) = \sin^2(x) - \frac{x}{2}$. Calculați soluția aproximativă a ecuației $f(x) = 0$ pe segmentul $[0,5; 0,7]$ cu exactitatea $\varepsilon = 0,00001$, utilizând metoda Newton.
- 7 Separați soluțiile, apoi calculați soluțiile ecuațiilor, folosind metoda Newton, pentru $\varepsilon = 0,00001$:
 - a) $x^5 - 80x^2 + 89 = 0$;
 - b) $e^x - x^2 = 0$.

Test de evaluare

I Selectați răspunsul corect:

1. A rezolva ecuația $f(x) = 0$ înseamnă a determina punctele:
 - a) de intersecție $f(x)$ cu axa Oy ;
 - b) de intersecție $f(x)$ cu axa Ox ;
 - c) în care $f(x)$ nu este definită.
2. A separa soluțiile ecuației $f(x) = 0$ înseamnă a determina:
 - a) un interval pe axa Ox , care va conține toate soluțiile ecuației $f(x) = 0$;
 - b) toate intervalele pe axa Ox , care au proprietatea de a conține exact câte o soluție a ecuației $f(x) = 0$;
 - c) toate intervalele pe axa Ox , care au proprietatea de a nu conține nicio soluție a ecuației $f(x) = 0$;
 - d) domeniul de valori al funcției $f(x)$.
3. Metoda biseției de rezolvare a ecuațiilor algebrice și transcendente se bazează pe divizarea intervalului pe care este căutată soluția ecuației în părți:
 - a) egale;
 - b) proporționale, determinate de coarde, trasate consecutiv;
 - c) proporționale, determinate de tangente, trasate consecutiv;
 - d) proporționale, determinate de coarde și tangente, trasate consecutiv.
4. Formula recurentă de calcul a aproximării x_i a soluției ecuației $f(x) = 0$ prin metoda coardelor este:
 - a) $x_{i+1} = x_i + \frac{f(x_i)}{f(e) - f(x_i)} (e - x_i)$;
 - b) $x_{i+1} = x_i - \frac{f(x_i)}{f(e) - f(x_i)} (e - x_i)$;
 - c) $x_{i+1} = x_i - \frac{f(e) - f(x_i)}{f(x_i)} (e - x_i)$;
 - d) $x_{i+1} = x_i - \frac{f(x_i)}{f(e) + f(x_i)} (e - x_i)$.
5. Aproximarea inițială în cazul aplicării metodei tangentelor este cea extremitate e a intervalului $[a, b]$ pe care este căutată soluția, pentru care este adevărată relația:
 - a) $f(e) \times f''(e) < 0$;
 - b) $f(e) \times f''(e) > 0$;
 - c) $f(e) \times f''(e) = 0$;
 - d) $f(e) \times f''(e) \neq 0$.

II Calculați folosind programele elaborate anterior:

1. Soluția ecuației $\frac{-6x^2 + 3^{x - \ln(3x^2 - 2x + 7)}}{4x^2 - x + 12} = 0$ pe $[10; 15]$ prin metoda biseției, pentru 5, 10, 20 de divizări ale segmentului inițial și prin metoda coardelor, pentru 5, 10, 20 de iterații. Explicați diferența dintre rezultatele obținute.
2. Soluția ecuației $x^2 - \sin 5x = 0$ pe $[0,5; 0,6]$ prin metoda tangentelor, pentru 2, 4, 6 divizări ale segmentului inițial și cu exactitatea $\varepsilon: 0,001; 0,00001; 0,0000001$.
Considerați: $m_1 = 1,2; M_2 = 6,2$.

METODE NUMERICE DE CALCUL AL DETERMINANȚILOR ȘI REZOLVAREA SISTEMELOR DE ECUAȚII LINIARE

După studierea acestui capitol, veți fi capabili să:

- descrieți metodele numerice de calcul al determinanților și algoritmi pentru implementarea acestor metode;
- descrieți metodele numerice de rezolvare a sistemelor de ecuații liniare;
- descrieți algoritmi de rezolvare a sistemelor de ecuații liniare prin metoda Cramer și metoda Gauss;
- elaborați programe pentru calculul determinanților de ordinul n ;
- elaborați programe pentru rezolvarea sistemelor de ecuații liniare de ordinul n ;
- combinați metodele studiate pentru elaborarea algoritmilor eficienți de rezolvare a sistemelor de ecuații liniare și a programelor care realizează acești algoritmi.

4.1. Determinanți numerici

Fie dată o matrice pătratică arbitrară de ordinul n :

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ & & \dots & \\ a_{n,1} & a_{n,2} & \dots & a_{n,n} \end{pmatrix}. \quad (1)$$

Fiecărei din matricele de acest tip îi este asociată o valoare numerică numită *determinant*. Determinantul poate fi definit în mod inductiv. Notăția folosită pentru determinantul matricei A este $\det(A)$.

Pentru a defini determinantul unei matrice de ordinul n , se va folosi noțiunea de *minor*.

Se numește minor de ordinul $n-1$ al elementului a_{ij} al matricei A de rang $n(n>1)$ determinantul matricei de rang $n-1$, obținută din matricea A prin excluderea rîndului i și a coloanei j . Vom nota minorul elementului a_{ij} prin A_{ij} , unde i indică rîndul, iar j – coloana la intersecția cărora se află elementul a_{ij} .

Astfel, pentru a calcula în matricea din exemplu minorul $A_{1,2}$ al elementului $a_{1,2}$, se exclude din matrice linia 1 și coloana 2:

$$A = \begin{pmatrix} 1 & 3 & 7 \\ 0 & 0 & 2 \\ -1 & 2 & 1 \end{pmatrix}; \quad A_{1,2} = \det \begin{pmatrix} 0 & 2 \\ -1 & 1 \end{pmatrix} = 2.$$

Se numește determinant al matricei A de rang n valoarea expresiei $\sum_{j=1}^n (-1)^{1+j} a_{1,j} A_{1,j}$.

Conform definiției

$$\Delta = \det(A) = \begin{vmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \dots & \dots & \dots & \dots \\ a_{n,1} & a_{n,2} & \dots & a_{n,n} \end{vmatrix} = \sum_{j=1}^n (-1)^{1+j} a_{1,j} A_{1,j}.$$

Astfel, pentru matricea de ordinul 4:

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} \\ a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} \end{pmatrix},$$

$$\det(A) = a_{1,1}A_{1,1} - a_{1,2}A_{1,2} + a_{1,3}A_{1,3} - a_{1,4}A_{1,4}.$$

Fiecare dintre minorii $A_{1,j}$, $j = 1, \dots, 4$ este determinantul unei matrice de ordinul 3 și poate fi calculat direct.

Algoritmul de calcul al determinantilor numerici

Fie dată matricea (1).

Algoritmul de calcul al determinantului unei matrice de ordinul n se bazează direct pe definiția determinantului:

$$\det(A) = \sum_{j=1}^n (-1)^{1+j} a_{1,j} A_{1,j}.$$

În această formulă elementele necunoscute sînt minorii elementelor din prima linie. Fie un minor arbitrar $A_{1,j}$. El este determinantul unei matrice de ordinul $n-1$. Pentru a-l calcula, urmează să fie rezolvată o problemă echivalentă cu problema inițială, dar de dimensiune mai mică. Deoarece la un moment dat se ajunge la calculul unui determinant de ordinul 1, 2 sau 3, care se calculează direct, se respectă regula de consistență și poate fi aplicat un algoritm recursiv:

Fie matricea A are ordinul R .

ALGORITM CRD (A, R)

{Calcul recursiv determinant al matricei A de ordin R .}

Cazul elementar

Dacă ordinul matricei A este 1, ($R = 1$), atunci $\text{CRD} \leftarrow a_{1,1}$, altfel:

Ne amintim!

Pentru matricea A de ordinul 1, formată dintr-un singur element $a_{1,1}$, determinantul va fi chiar valoarea acestui element.

Exemplu: $A = (7)$; $\det(A) = 7$.

Pentru o matrice de ordinul 2,

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$$

determinantul va fi egal cu valoarea expresiei $a_{1,1}a_{2,2} - a_{1,2}a_{2,1}$.

Exemplu: $A = \begin{pmatrix} 2 & 5 \\ 1 & 7 \end{pmatrix}$; $\det(A) = 9$.

Pentru o matrice de ordinul 3,

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

determinantul poate fi calculat folosind regula lui Sarrus (regula diagonalelor și a triunghiurilor).

Astfel, pentru o matrice de ordinul 3 determinantul poate fi calculat direct după formula:

$$\det(A) = a_{1,1}a_{2,2}a_{3,3} + a_{1,3}a_{2,1}a_{3,2} + a_{1,2}a_{2,3}a_{3,1} - a_{1,3}a_{2,2}a_{3,1} - a_{2,1}a_{1,2}a_{3,3} - a_{1,1}a_{2,3}a_{3,2}.$$

Exemplu:

$$A = \begin{pmatrix} 1 & 3 & 7 \\ 0 & 0 & 2 \\ -1 & 2 & 1 \end{pmatrix}; \det(A) = -10.$$

Regula de consistență:

1. Există un caz elementar: matricea ce corespunde minorului curent are ordinul 1.

2. La nivelul k se fac k apeluri pentru calculul determinantilor de ordinul $k-1$. Prin urmare, procesul converge spre un caz elementar.

Cazul de reducere

1. Valoarea determinantului Δ se inițializează cu 0 ($\Delta \leftarrow 0$).
2. Pentru toți j de la 1 la R :
 - a) Se formează matricea $M_{1,j}$ prin excluderea din matricea curentă A a liniei 1 și a coloanei j . (Ordinul $M_{1,j}$ este $R-1$. Matricea dată corespunde minorului $A_{1,j}$.)
 - b) Se calculează determinantul $D_{1,j}$ al matricei $M_{1,j}$ prin apelul CRD ($M_{1,j}$, $R-1$).
 - c) Se actualizează valoarea $\Delta \leftarrow \Delta + (-1)^{1+j} \times D_{1,j}$.
3. CRD $\leftarrow \Delta$. SFÎRȘIT.

Implementarea algoritmului

Fie date declarațiile:

```
const nmax=10;  
type matrice=array[1..nmax,1..nmax] of real;
```

Matricea pentru care se calculează determinantul va fi stocată în tabloul X de tip `mat`. Ordinul maxim al matricei pentru care se calculează determinantul este limitat de constanta `nmax`. Minorul $M_{1,j}$, care se generează în cadrul apelului curent, este stocat în variabila `minor`, de tip `mat`. Semnul valorii $(-1)^{1+j}$ CRD ($M_{1,j}$, $R-1$) este determinat de paritatea variabilei j , prin urmare, valoarea calculată va fi adăugată pentru j impar și scăzută pentru j par.

O realizare posibilă în limbajul Pascal a funcției recursive de calcul al determinantilor este următoarea:

```
function cdet(var x:matrice; t:integer) : real;  
  var  
    i, j, k: integer;  
    s : real;  
    minor : matrice;  
begin  
  if t=1 then cdet:=x[1,1] {caz elementar}  
  else begin  
    s:=0;  
    for k:=1 to t do  
      begin  
        {Se exclude linia 1 si coloana k pentru a forma matricea,  
        care corespunde minorului elementului x[1,k]}  
        for i:=1 to t-1 do  
          for j:=1 to k-1 do  
            minor[i,j]:=x[i+1,j];  
          for i:=1 to t-1 do  
            for j:=k to t-1 do  
              minor[i,j]:=x[i+1,j+1];  
          {apelul recursiv}  
          if odd(k) then s:=s+x[1,k]*cdet(minor, t-1)  
          else s:=s-x[1,k]*cdet(minor, t-1);  
        end;  
        cdet:=s;  
      end;  
    end;  
end;
```

Numărul de operații necesare pentru calculul recursiv al determinantului unei matrice de ordinul n este determinat de numărul de apeluri recursive, precum și de numărul de operații în cadrul unui apel.

Dezvoltarea unei matrice de ordinul n după o linie presupune formarea a n minori de ordinul $n-1$. Consecutiv, la dezvoltarea fiecăruia dintre ei vor apare $n-1$ minori de ordinul $n-2$ și așa mai departe. Numărul total de apeluri va fi determinat de valoarea $n \times (n-1) \times (n-2) \times \dots \times 3 \times 2 \times 1 = n!$ Numărul de operații în cadrul fiecărui apel este proporțional cu n^2 . Prin urmare, complexitatea finală a subprogramului recursiv este $O(n^2n!)$, ceea ce face ca algoritmul să fie puțin eficient pentru valori mari ale lui n .

Există și un algoritm de complexitate polinomială, care permite calculul iterativ al determinantilor. Algoritmul folosește transformările elementare aplicate consecutiv asupra matricii inițiale pentru a o transforma într-o matrice triunghiulară. Determinantul acesteia din urmă este egal cu produsul valorilor elementelor de pe diagonala principală.

Pentru a obține elemente cu valoare nulă sub elementul diagonal din coloana j ($j = 1, \dots, n-1$), elementele liniei i ($i = j+1, \dots, n$) se vor aduna cu elementele respective ale liniei j înmulțite cu coeficientul $-\frac{a_{ij}}{a_{jj}}$. Dacă elementul $a_{j,j}$ are valoarea 0, se încearcă permutarea liniei j și a liniei k ($j < k$), astfel încât $a_{kj} \neq 0$. Dacă o asemenea linie nu există, se poate spune cu certitudine că determinantul matricii este zero.

Neajunsul metodei este numărul considerabil de împărțiri efectuate, care în cazul oscilațiilor mari ale valorilor elementelor matricii pot genera erori, ceea ce nu se întâmplă în cazul algoritmului recursiv.

O posibilă implementare a acestui algoritm este realizată în funcția CID:

```
function CID(x:matrice; r:integer): real;
  var i,j,k:integer;
      q:real;
begin {CID}
  for i:=1 to r-1 do
    begin
      {se verifica valoarea elementului diagonal din linia i} if x[i,i]=0
      then {daca e nula, se cauta o linie pentru inlocuire}
        begin
          k:=i;
          for j:=i+1 to r do
            if x[j,i]<>0 then k:=j;
          {daca nu exista linie pentru inlocuire}
          if k=i then begin CID:=0; exit; end
          {altfel are loc permutarea liniilor i, k}
          else
            for j:=1 to r do
              begin
                q:=x[i,j];
                x[i,j]:=x[k,j];
                x[k,j]:=-q;
              end;
            end;
          {modificarea liniilor in scopul obtinerii elementelor cu
          valoare nula in coloana i}
```

```

for j:=i+1 to r do
  begin
    q:=-x[j,i]/x[i,i];
    for k:=i to r do x[j,k]:=x[j,k]+x[i,k]*q
  end;
end;
{calculul valorii determinantului pentru matricea triunghiulara}
q:=1;
for i:=1 to r do q:=q*x[i,i];
CID:=q;
end;

```

Întrebări și exerciții

- ❶ Explicați legătura dintre matrice și determinantul ei.
- ❷ Elaborați o funcție pentru calculul determinantilor de ordinul 3, utilizând regula lui Sarrus.
- ❸ Descrieți algoritmul de calcul al determinantilor de ordinul n prin dezvoltare după elementele unei linii arbitrare.
- ❹ Elaborați un program pentru calculul determinantilor de ordinul n ($n \leq 10$), în care veți folosi funcția `cdet`, descrisă în paragraful curent. Ordinul matricei și valorile elementelor sale se vor introduce de la tastatură.
- ❺ Calculați, cu ajutorul programului elaborat, determinantii următoarelor matrice:

a) $\begin{pmatrix} 2 & 5 \\ 4 & -1 \end{pmatrix};$

b) $\begin{pmatrix} 1 & 0 & 3 \\ 5 & 8 & 2 \\ 6 & -1 & 4 \end{pmatrix};$

c) $\begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 0 & 5 \\ 3 & 6 & 9 & 6 \\ 4 & 5 & 8 & 7 \end{pmatrix};$

d) $\begin{pmatrix} 7 & 2 & 3 & 0 \\ 0 & 0 & 0 & 2 \\ 0 & 6 & 0 & 6 \\ 4 & 4 & 3 & 0 \end{pmatrix};$

e) $\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 2 & -3 & 4 & 1 \\ 3 & 4 & -3 & 9 \end{pmatrix};$

f) $\begin{pmatrix} -4 & 9 & -4 & 6 & -5 \\ -1 & 7 & 5 & 5 & 8 \\ 9 & -6 & 8 & -6 & 2 \\ 2 & 5 & 2 & 3 & 9 \\ 1 & 3 & 1 & 5 & -2 \end{pmatrix};$

g) $\begin{pmatrix} 1 & -1 & 3 & 2 \\ 1 & -1 & 6 & 5 \\ 1 & -1 & -9 & -10 \\ 1 & -1 & 2 & -1 \end{pmatrix};$

h) $\begin{pmatrix} 6 & 5 & 3 & -2 & -7 & 1 \\ 1 & 4 & -1 & 8 & -6 & -5 \\ 4 & -3 & 1 & 10 & 9 & -3 \\ 6 & 4 & -6 & 7 & 7 & 9 \\ -8 & 6 & 4 & -1 & -5 & 6 \\ 1 & 4 & -5 & 6 & -6 & 3 \end{pmatrix}.$

- ❻ În programul realizat în exercițiul 4, transformați parametrul variabilă x al funcției `cdet` în parametru valoare. Observați ce se va întâmpla în cazul calculului aceluiași determinant pentru diferite tipuri ale parametrului x . Explicați diferența observată.
- ❼ Încercați să aplicați programul realizat pentru calculul determinantilor matricelor cu dimensiuni n , $n > 20$, cu elemente de tip real. Ce se va întâmpla? Explicați.

4.2. Metoda Cramer de rezolvare a sistemelor de ecuații liniare

Formulele Cramer¹

Capacitatea de a calcula determinanții numerici de ordinul n permite rezolvarea unei game largi de probleme din diverse domenii. Una dintre aplicațiile practice ale calculului determinanților este rezolvarea sistemelor de ecuații liniare.

Fie dat sistemul din n ecuații liniare cu n necunoscute:

$$\left. \begin{aligned} a_{1,1}x_1 + a_{1,2}x_2 + \dots + a_{1,n}x_n &= b_1 \\ a_{2,1}x_1 + a_{2,2}x_2 + \dots + a_{2,n}x_n &= b_2 \\ \dots\dots\dots \\ a_{n,1}x_1 + a_{n,2}x_2 + \dots + a_{n,n}x_n &= b_n \end{aligned} \right\} \quad (1)$$

Pentru transcrierea sistemului în formă matriceală vor fi folosite următoarele notații:

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \dots & \dots & \dots & \dots \\ a_{n,1} & a_{n,2} & \dots & a_{n,n} \end{pmatrix} \text{ – matricea coeficienților,} \quad b = \begin{pmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{pmatrix} \text{ – vectorul termenilor liberi,} \quad x = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix} \text{ – vectorul soluțiilor.}$$

În formă matriceală sistemul (1) poate fi scris

$$Ax = b. \quad (2)$$

În cazul când determinantul A este diferit de 0, există matricea inversă A^{-1} . În urma înmulțirii ambelor părți ale egalității (2) la A^{-1} , se obține:

$$A^{-1}Ax = A^{-1}b,$$

ceea ce este echivalent cu

$$x = A^{-1}b. \quad (3)$$

Această formulă permite calculul soluțiilor sistemului (1) în cazul în care matricea A a sistemului este nesingulară. Formula detaliată pentru calcularea componentelor vectorului soluție rezultă nemijlocit din (3) și proprietățile matricei inverse:

$$x_i = \frac{\Delta_i}{\Delta} \quad i = 1, \dots, n$$

$$\Delta_i = \begin{vmatrix} a_{1,1} & \dots & a_{1,i-1} & b_1 & a_{1,i+1} & \dots & a_{1,n} \\ a_{2,1} & \dots & a_{2,i-1} & b_2 & a_{2,i+1} & \dots & a_{2,n} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ a_{n,1} & \dots & a_{n,i-1} & b_n & a_{n,i+1} & \dots & a_{n,n} \end{vmatrix}. \quad (4)$$

Ne amintim!

Matricea pătratică A se numește **nesingulară**, dacă determinantul ei este diferit de 0.

Matricea inversă matricei A – matricea care, fiind înmulțită la A (fie la stînga, fie la dreapta), dă ca rezultat matricea unitară E . Matricea inversă matricei A va fi notată A^{-1} :

$$AA^{-1} = A^{-1}A = E.$$

¹ Gabriel Cramer (1704–1752) – matematician elvețian. Născut în Geneva. Cele mai remarcabile rezultate în matematică le obține în domeniul algebrei (cercetarea curbelor algebrice, regula Cramer) și al probabilității (soluția „paradoxului din Sankt-Petersburg”).

Δ_i este determinantul matricei formate din matricea A , în care coloana i a fost înlocuită prin vectorul termenilor liberi.

Formulele (4) permit calculul direct al soluțiilor sistemului de ecuații liniare (1), deoarece conțin doar mărimi calculate prin utilizarea matricei coeficienților sistemului (1) și a vectorului termenilor liberi ai acestui sistem. Ele sînt numite *formulele Cramer*.

ALGORITMIZAREA METODEI

Fie dat sistemul (1) din n ecuații liniare cu n necunoscute. Pentru rezolvarea sistemului, folosind formulele Cramer, vor fi necesare următoarele structuri de date:

- Un tablou bidimensional ($n \times n$) cu elemente de tip real sau întreg (în funcție de valorile datelor de intrare) – matricea A a coeficienților sistemului;
- Un tablou unidimensional cu elemente de tip real sau întreg (în funcție de valorile datelor de intrare) – vectorul b al termenilor liberi ai sistemului;
- Un tablou unidimensional cu elemente de tip real – vectorul soluțiilor x .

Determinarea soluțiilor unui sistem de ecuații liniare, utilizînd formulele Cramer, se bazează pe calculul determinantilor. Pentru aceasta poate fi utilizată funcția recursivă de calcul al determinantilor, descrisă în paragraful precedent.

Fiind realizată funcția de calcul al determinantului unei matrice CRD (A, n), se va trece la algoritmul de rezolvare a sistemului, folosind formulele Cramer.

Pasul 1. Se calculează $\Delta = \text{CRD}(A, n)$ (A – matricea coeficienților sistemului).

Dacă $\Delta \neq 0$, se trece la *pasul 2*; în caz contrar, se afișează mesajul de imposibilitate de utilizare a metodei. SFÎRȘIT.

Pasul 2. $i \leftarrow 1$.

Pasul 3.

- Se creează o copie (C) a matricei coeficienților sistemului. $C \leftarrow A$.
- În tabloul C , coloana cu numărul de ordine i este înlocuită cu vectorul termenilor liberi b .
- Se calculează determinantul Δ_i al matricei (C), apoi componenta i a soluției:

$$x_i = \frac{\Delta_i}{\Delta}.$$

- Se trece la *pasul 4*.

Pasul 4. Dacă $i < n$, atunci $i \leftarrow i+1$, după care se revine la *pasul 3*; în caz contrar, este afișat vectorul x . SFÎRȘIT.

Notă. În procesul de realizare a programului, copia matricei coeficienților sistemului poate fi obținută prin transmiterea acesteia într-un subprogram în calitate de parametru valoare. Înlocuirea coloanei necesare și calculul determinantului pot fi efectuate în același subprogram.

Exemplul 1: Să se scrie programul care va calcula soluțiile sistemului:

$$\begin{cases} 3x_1 + 2x_2 - x_3 = 7 \\ -x_1 + x_2 + x_3 = 4 \\ x_1 - 6x_2 + 2x_3 = -13 \end{cases}$$

folosind formulele Cramer.

Intrare: Coeficienții și termenii liberi ai sistemului se introduc nemijlocit în program.

Ieșire: Rezultatele vor fi afișate pe ecran.

```

program cn11;
type mat=array[1..3,1..3] of integer;
      vec=array[1..3] of integer;
      vs=array[1..3] of real;
var   a      : mat;
      b      : vec;
      sol   : vs;
      p,n   : integer;
      de    : real;

function cdet( x:mat;t:integer):real;
var   i,j,k,l: integer;
      s      : real;
      minor  : mat;
begin
  if t=1 then cdet:=x[1,1]
  else
    begin   s:=0;
      for k:=1 to t do
        begin
          for i:=1 to t-1 do
            for j:=1 to k-1 do minor[i,j]:=x[i+1,j];
          for i:=1 to t-1 do
            for j:=k to t-1 do minor[i,j]:=x[i+1,j+1];
            if odd(k) then s:=s+x[1,k]*cdet(minor, t-1)
              else s:=s-x[1,k]*cdet(minor, t-1);
          end;
        end
      end
    end
  end;

function transforma(x:mat;t,l:integer):real;
var   i : integer;
begin   for i:=1 to t do x[i,l]:=b[i];
  end;
  transforma:=cdet(x,t);

end;
begin
  n:=3;
  a[1,1]:= 3; a[1,2]:= 2; a[1,3]:= -1; b[1]:= 7;
  a[2,1]:=-1; a[2,2]:= 1; a[2,3]:= 1; b[2]:= 4;
  a[3,1]:= 1; a[3,2]:=-6; a[3,3]:= 2; b[3]:=-13;
  de:=cdet(a,n);
  if de<>0 then
    begin   for p:=1 to n do sol[p]:= transforma(a,n,p)/de;
      for p:=1 to n do writeln('x[' ,p, ']=' ,sol[p]:0:3);
    end
  else writeln('Calcul imposibil');
end.

```

Rezultate:

x[1]=1.000
x[2]=3.000
x[3]=2.000

Întrebări și exerciții

- ❶ Pentru rezolvarea căror probleme pot fi folosite formulele Cramer? Care sînt condițiile cînd ele nu pot fi aplicate?
- ❷ Explicați sensul elementelor Δ și Δ_i din formulele Cramer.
- ❸ În baza exemplului **cn11** elaborați un program pentru calculul soluției unui sistem din n ecuații liniare cu n ($n \leq 10$) necunoscute, utilizînd formulele Cramer:
 - a) Numărul de ecuații n , valorile coeficienților ecuațiilor sistemului și ale termenilor liberi se introduc de la tastatură;
 - b) Datele inițiale vor fi citite din fișierul text **system.in** care va conține pe prima linie un număr întreg n – numărul de ecuații în sistem. Fiecare din următoarele n linii va conține cîte $n+1$ numere întregi, separate prin spațiu: coeficienții ecuațiilor în ordinea apariției lor și termenii liberi.
- ❹ Rezolvați următoarele sisteme de ecuații, utilizînd programul realizat în exercițiul 3:

$$\text{a) } \begin{cases} x_1 + x_2 + x_3 + x_4 = 10 \\ x_1 - x_2 + x_3 - x_4 = -2 \\ 2x_1 - 3x_2 + 4x_3 + x_4 = 12 \\ 3x_1 + 4x_2 - 3x_3 + 9x_4 = 38 \end{cases}$$

$$\text{b) } \begin{cases} -4x_1 + 9x_2 - 4x_3 + 6x_4 - 5x_5 = -8 \\ -x_1 + 7x_2 + 5x_3 + 5x_4 + 8x_5 = 199 \\ 9x_1 - 6x_2 + 8x_3 - 6x_4 + 2x_5 = 52 \\ 2x_1 + 5x_2 + 2x_3 + 3x_4 + 9x_5 = 198 \\ x_1 + 3x_2 + x_3 + 5x_4 - 2x_5 = 33 \end{cases}$$

$$\text{c) } \begin{cases} 6x_1 + 5x_2 + 3x_3 - 2x_4 - 7x_5 + x_6 = -48 \\ x_1 + 4x_2 - x_3 + 8x_4 - 6x_5 - 5x_6 = -106 \\ 4x_1 - 3x_2 + x_3 + 10x_4 + 9x_5 - 3x_6 = 70 \\ 6x_1 + 4x_2 - 6x_3 + 7x_4 + 7x_5 + 9x_6 = 64 \\ -8x_1 + 6x_2 + 4x_3 - 1x_4 - 5x_5 + 6x_6 = 25 \\ x_1 + 4x_2 - 5x_3 + 6x_4 - 6x_5 + 3x_6 = -82 \end{cases}$$

$$\text{d) } \begin{cases} x_1 - x_2 + 3x_3 + 2x_4 = 1 \\ x_1 - x_2 + 6x_3 + 5x_4 = 0 \\ x_1 - x_2 - 9x_3 - 10x_4 = 5 \\ x_1 - x_2 + 2x_3 - x_4 = \frac{4}{3} \end{cases}$$

$$\text{e) } \begin{cases} x_1 - 2x_2 - 8x_3 + 3x_4 - 4x_5 - 2x_6 + 2x_7 = -104 \\ 5x_1 + 8x_2 + x_3 - 5x_4 + 2x_5 - x_6 - 8x_7 = 94 \\ 2x_1 - 4x_2 + 7x_3 + 3x_4 + 7x_5 + 10x_6 + 9x_7 = 159 \\ -4x_1 + 2x_2 - x_4 + 3x_5 + 6x_6 - 8x_7 = -80 \\ -x_1 - x_2 + 8x_3 + 7x_4 + 2x_5 - 6x_6 + 2x_7 = 204 \\ 7x_1 + x_2 + 9x_3 - 9x_4 + 6x_5 - 6x_6 = 209 \\ -5x_1 + 9x_2 - 6x_3 - 5x_4 + 10x_5 - x_6 - 9x_7 = -29 \end{cases}$$

$$\text{f) } \begin{cases} -8x_1 + 5x_2 + 10x_3 - 8x_4 + 7x_5 = 106 \\ 4x_1 - 8x_2 - 4x_3 - 3x_4 + 3x_5 = 39 \\ 6x_1 - 4x_2 + 10x_3 + 6x_4 - 7x_5 = 97 \\ 8x_1 + 2x_2 + 4x_3 - 8x_4 - 7x_5 = 119 \\ -6x_1 - 8x_2 + 9x_3 - 6x_4 - 3x_5 = 10 \end{cases}$$

$$\text{g) } \begin{cases} -2x_1 + 6x_3 - 4x_4 = 68 \\ 4x_1 - 8x_2 - 2x_3 + 10x_4 = -78 \\ -9x_2 - 9x_3 + 5x_4 = -128 \\ -5x_1 + 7x_2 + 3x_3 + 4x_4 = 27 \end{cases}$$

4.3. Metoda Gauss de rezolvare a sistemelor de ecuații liniare

Una dintre metodele eficiente de determinare a soluției unui sistem de ecuații liniare este metoda excluderii consecutive a necunoscutelor. Această metodă este cunoscută și sub numele *metodei Gauss*². Ea poate fi utilizată în cazul când sistemul cercetat are o soluție unică sau o infinitate de soluții (în cel de-al doilea caz va fi stabilită o soluție particulară).

Fie dat sistemul (1) din n ecuații liniare cu n necunoscute cu matricea extinsă (1a):

$$\left\{ \begin{array}{l} a_{1,1}x_1 + \dots + a_{1,i}x_i + \dots + a_{1,n}x_n = b_1 \\ a_{2,1}x_1 + \dots + a_{2,i}x_i + \dots + a_{2,n}x_n = b_2 \\ \dots \\ a_{i,1}x_1 + \dots + a_{i,i}x_i + \dots + a_{i,n}x_n = b_i \\ \dots \\ a_{n,1}x_1 + \dots + a_{n,i}x_i + \dots + a_{n,n}x_n = b_n \end{array} \right. \quad (1)$$

$$\left(\begin{array}{cccccc} a_{1,1} & \dots & a_{1,i} & \dots & a_{1,n} & b_1 \\ a_{2,1} & \dots & a_{2,i} & \dots & a_{2,n} & b_2 \\ & & \dots & & & \\ a_{i,1} & \dots & a_{i,i} & \dots & a_{i,n} & b_i \\ & & \dots & & & \\ a_{n,1} & \dots & a_{n,i} & \dots & a_{n,n} & b_n \end{array} \right) \quad (1a)$$

Metoda Gauss presupune:

- excluderea variabilei x_1 din toate ecuațiile sistemului, începînd cu a doua;
- excluderea variabilei x_2 din toate ecuațiile sistemului, începînd cu a treia;
- ...
- excluderea variabilei x_i din toate ecuațiile sistemului, începînd cu ecuația $i+1$;
- ...
- excluderea variabilei x_{n-1} din ecuația cu numărul n .

$$\left\{ \begin{array}{l} a_{1,1}^*x_1 + a_{1,2}^*x_2 + \dots + a_{1,i}^*x_i + \dots + a_{1,n}^*x_n = b_1^* \\ a_{2,2}^*x_2 + \dots + a_{2,i}^*x_i + \dots + a_{2,n}^*x_n = b_2^* \\ \dots \\ a_{i,i}^*x_i + \dots + a_{i,n}^*x_n = b_i^* \\ \dots \\ a_{n,n}^*x_n = b_n^* \end{array} \right. \quad (2)$$

În termeni matematici, trebuie să se obțină un sistem de ecuații, echivalent cu sistemul (1), de forma (2).

Soluția unui sistem de tipul (2) poate fi determinată după componente, începînd cu x_n , care se determină nemijlocit din ultima ecuație a sistemului. În general, avînd calculate componentele $x_{i+1}, x_{i+2}, \dots, x_n$, din ecuația i a sistemului (2) se determină componenta x_i .

Astfel, procesul de calcul al soluției unui sistem de ecuații prin metoda Gauss se divide în două etape:

Etapa 1 (directă) – transformarea sistemului (1) în sistemul echivalent (2) prin transformări elementare a matricei (1a);

Etapa 2 (inversă) – calculul componentelor soluției sistemului (2).

În cazul când sistemul are o infinitate de soluții (conține variabile libere), metoda permite determinarea unei soluții particulare, prin setarea directă a valorilor particulare pentru aceste variabile.

² Carl Friedrich Gauss (1777–1855) – matematician și cercetător german. Născut în Braunschweig, a realizat pe parcursul vieții cercetări de valoare în domeniul teoriei numerelor, statisticii, analizei matematice, geometriei, geodeziei, geofizicii, electrostaticii, astronomiei și al opticii.

ALGORITMIZAREA METODEI

Fie dat sistemul (1) din n ecuații liniare cu n necunoscute. Pentru rezolvarea sistemului prin metoda Gauss vor fi necesare următoarele structuri de date:

- Un tablou bidimensional ($n \times n+1$) cu elemente de tip real – matricea extinsă A a coeficienților sistemului și termenilor liberi;
- Un tablou unidimensional cu n elemente de tip real – vectorul soluțiilor x .

Rezolvarea sistemului de ecuații liniare prin metoda Gauss presupune o serie de transformări consecutive ale elementelor matricei extinse a coeficienților.

Etapa directă

Pasul 1. $i \leftarrow 1$.

Pasul 2.

- Dacă $a_{ii} = 0$, linia i își schimbă locul cu o linie r ($r > i$) pentru care $a_{ri} \neq 0$. Dacă o astfel de linie nu există, se trece la *pasul 3*.
- Pentru fiecare din liniile j de la $i+1$ la n se repetă **procedura (P)**:

$$\text{I. se calculează } k = \frac{-a_{j,i}}{a_{i,i}};$$

$$\text{II. pentru toți } l \text{ de la } 1 \text{ la } n+1 \quad a_{j,l} \leftarrow a_{j,l} + k \times a_{i,l}.$$

Pasul 3. Dacă $i < n$, atunci $i \leftarrow i+1$ și se revine la *pasul 2*; în caz contrar, se trece la *pasul 4*. {Trecerea la *pasul 4* marchează sfârșitul etapei directe și începutul etapei inverse a metodei.}

Etapa inversă

Pasul 4. Pentru i de la n la 1 se calculează:

$$x_i = \begin{cases} \frac{b_i^* - \sum_{j=i+1}^n a_{i,j}^* x_j}{a_{i,i}^*} & a_{i,i} \neq 0 \\ 0 & a_{i,i} = 0 \end{cases}$$

Pasul 5. Pentru i de la 1 la n se afișează x_i . SFÎRȘIT.

Exemplu: Fie dat sistemul de n ($n < 20$) ecuații liniare cu coeficienți reali de forma:

$$\begin{cases} a_{1,1}x_1 + a_{1,2}x_2 + \dots + a_{1,i}x_i + \dots + a_{1,n}x_n = b_1 \\ \dots \\ a_{i,1}x_1 + a_{i,2}x_2 + \dots + a_{i,i}x_i + \dots + a_{i,n}x_n = b_i \\ \dots \\ a_{n,1}x_1 + a_{n,2}x_2 + \dots + a_{n,i}x_i + \dots + a_{n,n}x_n = b_n \end{cases}$$

Elaborați un program pentru a determina soluțiile acestui sistem prin metoda Gauss.

Intrare: Coeficienții sistemului și termenii liberi se citesc din fișierul text IN.TXT cu următoarea structură (valorile sînt separate prin spațiu) (vezi schema alăturată).

Ieșire: Soluția calculată se va afișa pe ecran.

	n		
$a_{1,1}$...	$a_{1,n}$	b_1
$a_{2,1}$...	$a_{2,n}$	b_2
	...		
$a_{n,1}$...	$a_{n,n}$	b_n

Rezolvare: Condițiile impun utilizarea algoritmului general al metodei. În cazul apariției unor variabile libere, li se va atribui valoarea 0.

```
program cnl2;
const nmax=10;
type matrice=array[1..nmax,1..nmax] of real;
      vect=array[1..nmax] of real;
var a:matrice;
    s :vect;
    i,n:integer;

procedure citeste(var x:matrice; var t:integer);
  var i, j: integer; f: text;
  begin {citeste}
    assign (f, 'in.txt'); reset(f);
    readln(f, t);
    for i:=1 to t do
      begin
        for j:=1 to t do read(f,x[i,j]);
        readln(f, x [i,t+1]);
      end;
    close(f);
  end; {citeste}

procedure direct(var x:matrice;t:integer);
  label linie_urmatoare;
  var i,j,k,l:integer; r:real;
  begin {direct}
    for i:=1 to t-1 do
      begin
        if x[i,i]=0 then
          begin
            k:=i;
            for j:=i+1 to n do if x[j,i]<>0 then k:=j;
            if k=i then goto linie_urmatoare
            else
              for j:=1 to t+1 do
                begin
                  r:=x[i,j]; x[i,j]:=x[k,j]; x[k,j]:=r;
                end;
              end;
            for j:=i+1 to t do
              begin
                r:=-x[j,i]/x[i,i];
                for k:=i to t+1 do x[j,k]:=x[j,k]+x[i,k]*r;
              end;
            linie_urmatoare: end;
          end;
      end;
  end; {direct}

procedure invers (var q:vect);
  var i,j: integer;
  s: real;
```

```

begin
  for i:=n downto 1 do
    begin
      s:=0;
      for j:=i+1 to n do s:=s+a[i,j]*q[j];
      if a[i,i]>0 then q[i]:=(a[i,n+1]-s)/a[i,i] else q[i]:=0;
    end;
  end;
begin
  citeste(a,n);
  direct(a,n);
  invers(s);
  for i:=1 to n do writeln('x[' ,i,' ]=' ,s[i]:0:3);
end.

```

Teste:

$$\begin{cases} 3x_1 + 2x_2 - x_3 = 7 \\ 6x_1 + 4x_2 - 2x_3 = 14 \\ x_1 - 6x_2 + 2x_3 = -13 \end{cases}$$

Rezultate:

$$\begin{cases} x[1]=0.800 \\ x[2]=2.300 \\ x[3]=0.000 \end{cases}$$

Note:

Se observă prezența a două ecuații cu coeficienți proporționali. Programul detectează variabila liberă $x[3]$, pe care o stabilește fiind egală cu 0.

$$\begin{cases} -8x_1 + 9x_2 - x_3 + 9x_4 - 5x_5 = -111 \\ -4x_1 - 9x_2 + 9x_3 + 2x_4 - 2x_5 = 88 \\ -4x_1 - 4x_2 + 7x_3 + 3x_4 + 4x_5 = 64 \\ -3x_1 + 6x_2 - 7x_3 - 6x_4 + 9x_5 = -110 \\ x_1 - 9x_2 - 9x_3 - 7x_4 - 6x_5 = -82 \end{cases}$$

$$\begin{cases} x[1]=9.000 \\ x[2]=-3.000 \\ x[3]=11.000 \\ x[4]=1.000 \\ x[5]=2.000 \end{cases}$$

Determinantul matricei coeficienților sistemului este diferit de 0. Prin urmare există o singură soluție a sistemului. Prin verificare directă se stabilește corectitudinea soluției date de program.

Întrebări și exerciții

- 1 Descrieți etapele algoritmului de calcul al soluției unui sistem din n ecuații liniare cu n necunoscute, utilizând metoda Gauss.
- 2 Indicați care sînt resursele de memorie necesare pentru calculul soluției unui sistem din n ecuații liniare cu n necunoscute prin metoda Gauss.
- 3 Elaborați un program pentru calculul soluției unui sistem din n ecuații liniare cu n ($n \leq 10$) necunoscute prin metoda Gauss.
- 4 Estimați complexitatea temporală a algoritmului de rezolvare a sistemelor de ecuații liniare prin metoda Gauss. Folosiți programul **cn12**.
- 5 Efectuați o modificare a programului **cn12**, care ar permite determinarea soluțiilor sistemelor de forma:

$$\text{a) } \begin{cases} a_{1,1}x_1 + a_{1,2}x_2 + a_{1,3}x_3 + \dots + a_{1,n}x_n = b_1 \\ a_{2,2}x_2 + a_{2,3}x_3 + \dots + a_{2,n}x_n = b_2 \\ \dots \\ a_{n-1,n-1}x_{n-1} + a_{n-1,n}x_n = b_{n-1} \\ a_{n,n}x_n = b_n \end{cases} \quad \text{b) } \begin{cases} a_{1,1}x_1 = b_1 \\ a_{2,1}x_1 + a_{2,2}x_2 = b_2 \\ \dots \\ a_{n-1,2}x_2 + a_{n-1,3}x_3 + \dots + a_{n-1,n}x_n = b_{n-1} \\ a_{n,1}x_1 + a_{n,2}x_2 + a_{n,3}x_3 + \dots + a_{n,n}x_n = b_n \end{cases}$$

- 6 Realizați o modificare a programului **cn12**, care ar permite atribuirea aleatorie a valorilor variabilelor libere.

Test de evaluare

1 Selectați răspunsul corect:

- Determinantul este o caracteristică numerică, pe care o posedă matricele:
 - numai de rangul 1;
 - de orice rang;
 - numai de rang mai mic sau egal cu 3.
- Pentru calculul determinantilor pot fi folosiți algoritmi recursivi:
 - adevărat;
 - fals.
- Pentru a stoca în memoria calculatorului o matrice de dimensiunea 10×10 , elementele căreia sînt numere reale, vor fi necesari:
 - 640 octeți;
 - 1 024 octeți;
 - 600 octeți;
 - 400 octeți;
 - 200 octeți.
- Metoda Cramer poate fi utilizată doar în cazul cînd determinantul matricei coeficienților sistemului este:
 - diferit de 0;
 - egal cu 0;
 - pozitiv;
 - un număr întreg.
- Fie Δ – determinantul matricei coeficienților unui sistem din n ecuații liniare cu n necunoscute, iar Δ_1 – determinantul matricei coeficienților sistemului, în care prima coloană a fost înlocuită cu vectorul termenilor liberi. Metoda Cramer utilizează următoarea formulă de calcul pentru componenta x_1 a vectorului soluție:
 - $x_1 = \frac{\Delta}{\Delta_1}$;
 - $x_1 = \frac{\Delta_1}{\Delta}$;
 - $x_1 = \frac{(\Delta_1 - \Delta)}{\Delta}$;
 - $x_1 = \frac{(\Delta_1 - \Delta)}{\Delta_1}$.
- Metoda Gauss de rezolvare a sistemelor de ecuații liniare conține:
 - o singură etapă;
 - două etape;
 - trei etape;
 - numărul etapelor nu poate fi estimat apriori.
- În cadrul metodei Gauss, la etapa directă, pentru un sistem cu n ecuații, necunoscuta x_i este exclusă din toate ecuațiile cu indicii:
 - de la 1 la i ;
 - de la 1 la $i + 1$;
 - de la i la n ;
 - de la $i + 1$ la n .
- Fie sistemul de ecuații liniare, unde $a_{i,i} \neq 0$ ($i = 1, \dots, n$).

$$\left\{ \begin{array}{l} a_{1,1}x_1 + a_{1,2}x_2 + a_{1,3}x_3 + \dots + a_{1,n-1}x_{n-1} + a_{1,n}x_n = b_1 \\ a_{2,2}x_2 + a_{2,3}x_3 + \dots + a_{2,n-1}x_{n-1} + a_{2,n}x_n = b_2 \\ \dots \\ a_{n-1,n-1}x_{n-1} + a_{n-1,n}x_n = b_{n-1} \\ a_{n,n}x_n = b_n \end{array} \right.$$

Metoda Gauss presupune determinarea componentei x_i a vectorului soluție prin formula:

$$\text{a) } x_i = \frac{b_i - \sum_{j=i+1}^n a_{ij}x_j}{a_{ii}};$$

$$\text{c) } x_i = \frac{a_{ii}}{b_i - \sum_{j=i+1}^n a_{ij}x_j};$$

$$\text{b) } x_i = \frac{b_i + \sum_{j=i+1}^n a_{ij}x_j}{a_{ii}};$$

$$\text{d) } x_i = \frac{-b_i + \sum_{j=i+1}^n a_{ij}x_j}{a_{ii}}.$$

II **Calculați folosind programele elaborate anterior. Pentru fiecare soluție calculată se va afișa o linie care va conține cuvîntul „Răspuns:”, urmat, după un spațiu, de soluția calculată:**

1. Determinanții matricelor:

$$\text{a) } \begin{bmatrix} 1 & 4 & 0 \\ 2 & 3 & 7 \\ -4 & 1 & -2 \end{bmatrix};$$

$$\text{b) } \begin{bmatrix} 4 & -1 & 2 & 5 \\ 2 & 2 & 1 & 4 \\ 0 & 3 & 5 & 6 \\ 2 & -1 & 4 & 1 \end{bmatrix}.$$

2. Soluțiile sistemelor (metoda Cramer):

$$\text{a) } \begin{cases} -6x_1 - 3x_2 - 3x_3 = -63 \\ -7x_1 + 8x_2 + 6x_3 = 10 \\ 2x_1 + 6x_2 - 7x_3 = -64 \end{cases}$$

$$\text{b) } \begin{cases} 9x_2 - 8x_3 + 2x_4 = -37 \\ 8x_1 - 7x_2 + 9x_3 + 8x_4 = 27 \\ -3x_1 - 7x_2 + 3x_3 + 4x_4 = 12 \\ -x_1 - 9x_3 + x_4 = -69 \end{cases}$$

3. Soluțiile sistemului (metoda Gauss):

$$\begin{cases} 5x_2 + x_4 + 5x_5 = 89 \\ -8x_1 + 4x_2 - x_4 + 4x_5 = 87 \\ 10x_1 + 3x_2 - 9x_3 - 3x_4 - 2x_5 = -13 \\ -7x_1 + 6x_2 - 6x_3 + 7x_4 + 7x_5 = 217 \\ -7x_1 + 6x_2 + 8x_3 - 7x_4 + x_5 = -1 \end{cases}$$

CAPITOLUL 5

INTEGRAREA NUMERICĂ

După studierea acestui capitol, veți fi capabili să:

- descrieți algoritmul de calcul al integralei definite prin metoda dreptunghiurilor (a trapezelor);
- elaborați subprograme pentru calculul numeric al integralelor definite prin metoda dreptunghiurilor (a trapezelor);
- identificați probleme din diverse domenii, care se rezolvă prin calculul numeric al integralelor definite.

5.1. Metoda dreptunghiurilor pentru calculul aproximativ al integralei definite

Una dintre cele mai des aplicate implementări ale calculului numeric este calcularea integralei definite prin metode aproximative. Metodele directe nu întotdeauna permit calculul analitic al integralei, de multe ori nici nu este cunoscută formula care definește funcția ce urmează să fie integrată. De obicei este dată doar o serie de puncte în care este cunoscută valoarea funcției. În aceste cazuri integrala poate fi calculată doar prin metode aproximative (în presupunerea că funcția integrată este continuă pe segmentul pe care se face integrarea).

Din cursul de matematică se știe că sensul geometric al integralei definite $\int_a^b f(x)dx$ este aria trapezului curbiliniu, determinat de axa Ox , dreptele $x = a$, $x = b$ și graficul funcției $f(x)$ pe segmentul $[a, b]$ (fig. 5.1).

Măsurarea exactă a ariei unei asemenea figuri nu este întotdeauna posibilă. În asemenea situații o soluție ar fi aproximarea figurii inițiale cu un set de figuri, a căror arie este ușor de calculat. Atunci valoarea integralei definite (aria figurii inițiale) va fi aproximată de suma ariilor calculate.

Fie f o funcție derivabilă pe $[a, b]$ și se cere să se calculeze $\int_a^b f(x)dx$.

Pentru rezolvarea numerică a problemei se consideră o diviziune a $[a, b]$ în forma $a=x_0 < x_1 < \dots < x_{n-1} < x_n = b$. Perechile consecutive de puncte $x_0x_1, x_1x_2, \dots, x_{i-1}x_i, \dots, x_{n-1}x_n$ determină n segmente distincte, reuniunea

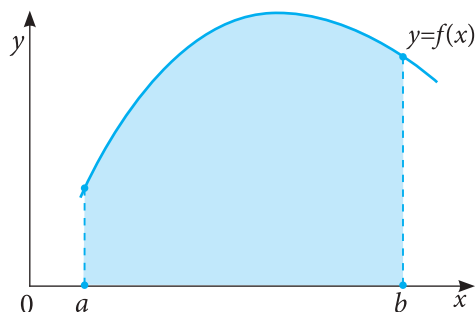


Fig. 5.1. Sensul geometric al integralei definite

căroră este $[a, b]$. Pentru simplitate punctele diviziunii vor fi echidistante (fig. 5.2). Atunci lungimea h a fiecărui segment elementar $[x_i, x_{i+1}]$ va fi determinată de formula

$$h = \frac{|b-a|}{n}.$$

Valorile x_i pot fi și ele exprimate prin mărimi cunoscute: $x_i = a + ih, i = 0, \dots, n$.

Fiind cunoscute valorile x_i și x_{i+1} , poate fi determinat și mijlocul z_i al fiecărui segment elementar $z_i = \frac{x_i + x_{i+1}}{2}$.

Pe fiecare dintre segmentele $[x_i, x_{i+1}]$ aria trapezului curbiliniu va fi aproximată prin aria dreptunghiului S_i , avînd lungimile laturilor h și $f(z_i)$ (fig. 5.3):

$$S_i = hf(z_i) = hf\left(a + ih + \frac{h}{2}\right).$$

În acest caz valoarea calculată I a integralei definite se calculează ca suma ariilor dreptunghiurilor și se determină nemijlocit din formula

$$I = h \sum_{i=0}^{n-1} f\left(a + ih + \frac{h}{2}\right),$$

unde

n – numărul de divizări ale segmentului inițial;

h – lungimea segmentului elementar;

I – valoarea calculată a integralei.

Astfel, calculul integralei se transformă în calculul valorii unei expresii aritmetice care depinde doar de numărul de diviziuni ale segmentului de integrare și de valoarea funcției în punctele de mijloc ale segmentelor elementare. Metoda care reduce calculul integralei la calculul unei sume de arii a dreptunghiurilor este numită *metoda dreptunghiurilor*.

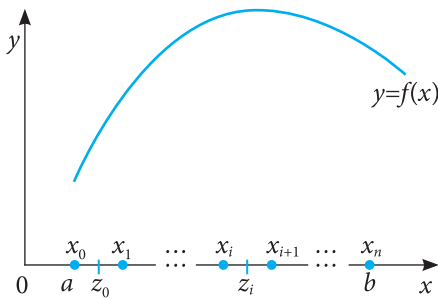


Fig. 5.2. Divizarea $[a, b]$ în segmente elementare. Mijlocul segmentului elementar $[x_i, x_{i+1}]$ este punctul $z_i = \frac{x_i + x_{i+1}}{2}$

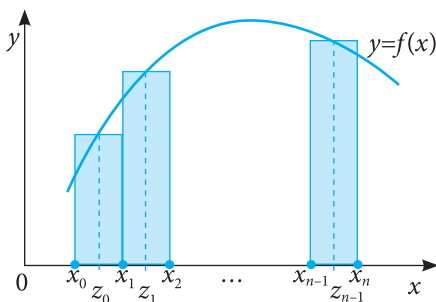


Fig. 5.3. Construcția consecutivă a dreptunghiurilor, care aproximează figura inițială. Pe fiecare segment elementar înălțimea dreptunghiului este determinată de valoarea funcției $f(x)$ în mijlocul segmentului

Eroarea metodei

Valoarea calculată a integralei este de cele mai multe ori diferită de valoarea exactă, calculată analitic. Eroarea apare din motivul aproximării pe fiecare segment elementar a funcției $f(x)$ cu o funcție constantă g , iar mărimea erorii ε este determinată de integrala erorii de aproximare și poate fi calculată după formula:

$$\varepsilon = \left| \int_a^b f(x) dx - I \right| \leq (b-a)M \frac{h}{4},$$

unde M – supremul $|f'(x)|$ pe $[a, b]$, I – valoarea calculată a integralei.

Din formula precedentă rezultă un fapt important: eroarea de calcul a metodei dreptunghiurilor

este proporțională cu numărul de divizări ale segmentului de integrare. Astfel, pentru a obține o eroare de calcul, ce nu depășește o valoare prestabilită ε , este suficient să se realizeze divizarea segmentului de integrare în n segmente elementare, valoarea lui n fiind determinată prin secvența de transformări

$$(b-a)M \frac{h}{4} \leq \varepsilon \Rightarrow \frac{(b-a)^2 M}{4n} \leq \varepsilon \Rightarrow n = \left\lceil \frac{(b-a)^2 M}{4\varepsilon} \right\rceil + 1.$$

ALGORITMIZAREA METODEI

Deoarece în cazul calculelor cu o eroare ce nu depășește valoarea prestabilită ε numărul necesar de divizări poate fi stabilit apriori, este suficient să se realizeze un singur algoritm – pentru un număr fixat de divizări n :

Pasul 1. Inițializare: Se introduc valorile extremităților segmentului de integrare a , b și numărul de divizări n .

Notă. În cazul în care este necesar calculul integralei cu o eroare, ce nu depășește o valoare prestabilită ε , numărul de divizări n se calculează cu ajutorul formulei $n = \left\lceil \frac{(b-a)^2 M}{4\varepsilon} \right\rceil + 1$.

Pasul 2. Se calculează lungimea segmentului elementar $h = \frac{|b-a|}{n}$, $S \leftarrow 0$.

Pasul 3. Pentru toți i de la 0 la $n-1$:

- Se calculează valorile $z_i \leftarrow a + ih + \frac{h}{2}$.
- Se calculează aria dreptunghiului elementar: $S_i \leftarrow f(z_i) \times h$.
- Aria calculată se sumează cu ariile precedente: $S \leftarrow S + S_i$.

Pasul 4. Se afișează aria totală calculată S . SFÎRȘIT.

Exemplul 1: Să se scrie un program care calculează integrala $\int_1^2 \frac{x^4}{\sqrt{1+x}} dx$ prin 20, 40,

80 și 160 de divizări ale segmentului de integrare, folosind metoda dreptunghiurilor. Pentru fiecare număr de divizări se va indica valoarea calculată cu șase semne după virgulă.

Rezolvare: Deoarece numărul de divizări este indicat în condițiile problemei, pre-procesarea matematică nu este necesară.

```

program cn13;
const r=4;
var S, a, b, h : real;
    j,i,n:integer;
function f(x:real): real;
begin
    f:=x*x*x*x/sqrt(1+x);
end;
begin
    a:=1; b:=2; n:=10;
    for j:=1 to r do
        begin
            S:=0; n:=n*2; h:=(b-a)/n;

```

```

for i:=0 to n-1 do
S:=S+ h*f(a + i*h + h/2);
writeln (' n=' ,n, ' I=' ,S:0:6);
end;
end.

```

Rezultate:

```

n=20 I=3.788513
n=40 I=3.789629
n=80 I=3.789908
n=160 I=3.789977

```

Exemplul 2: Să se scrie un program care calculează valoarea aproximativă a integralei

$\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \frac{\sin(\cos x^2)}{2} dx$ prin metoda dreptunghiurilor cu o eroare ce nu depășește valoarea

dată ε . Calculul va fi oprit, dacă se îndeplinește condiția: $\frac{\pi}{8} h \leq \varepsilon$; $h = \frac{|b-a|}{n}$, unde n – numărul curent de divizări.

Valorile a , b , ε se definesc nemijlocit în program.

Rezolvare: Din condițiile problemei se deduce numărul necesar de divizări:

$$\frac{\pi|b-a|}{8n} \leq \varepsilon; \Rightarrow n = \left\lceil \frac{\pi|b-a|}{8\varepsilon} \right\rceil + 1.$$

```

program1 cn14;
var S, a, b, e, h : real;
    j,i,n:integer;
function f(x:real): real;
begin
f:=sin(cos (x*x))/2;end;
begin
a:=-pi/2; b:=pi/2; e:=0.0001;
n:=round(pi*(abs(b-a))/(8*e))+1;
S:=0;
h:=(b-a)/n;
for i:=0 to n-1 do
S:=S+ h*f(a + i*h + h/2);
writeln (' n=' ,n, ' I=' ,S:0:6);
end.

```

Rezultate:

```

n=12338 I=0.729729

```

Întrebări și exerciții

- ❶ Motivați necesitatea calculului numeric al integralei definite.
- ❷ Ce procedeu stă la baza metodei dreptunghiurilor pentru calculul numeric al integralei?
- ❸ Motivați relația dintre numărul de divizări ale segmentului de integrare și eroarea de calcul a metodei dreptunghiurilor.

¹ Aici și în alte programe din prezenta ediție se folosește constanta predefinită π , utilizată în mediul Turbo Pascal. Utilizatorii altor limbaje de programare sau ai altor compilatoare Pascal vor defini constanta respectivă în cadrul programului. Ex.: `const pi=3.141.`

- 4 Scrieți un program care calculează integrala definită prin metoda dreptunghiurilor, pentru 200, 2 000, 20 000 de divizări ale segmentului de integrare. Pentru fiecare număr de divizări valoarea calculată se va afișa cu șase cifre zecimale.

$$a) \int_{-1}^1 \frac{(1+x^2)}{(1+x^4)} dx;$$

$$c) \int_0^{2\pi} \frac{dx}{(2+\cos x)(3+\cos x)};$$

$$e) \int_0^1 \frac{dx}{(x+1)\sqrt{x^2+1}};$$

$$b) \int_1^9 (x^3\sqrt{1-x}) dx;$$

$$d) \int_0^{2\pi} \frac{dx}{\sin^4 x + \cos^4 x};$$

$$f) \int_0^{\ln 2} \sqrt{e^x - 1} dx.$$

Explicați diferența dintre rezultatele obținute pentru valori diferite ale numărului de divizări. Verificați rezultatele obținute folosind calculatorul on-line pentru integrale definite: www.solveformath.com/online_math_calculator/calculus/definite_integral.

- 5 Scrieți un program care calculează integrala definită prin metoda dreptunghiurilor cu o eroare care nu va depăși valoarea prestabilită $\varepsilon = 0,001$; $\varepsilon = 0,0001$; $\varepsilon = 0,00001$. Valoarea M este cunoscută apriori:

$$a) \int_0^3 (x^2 + 3x) dx,$$

$$M = 9;$$

$$c) \int_1^2 (x^3 - 7x) dx,$$

$$M = 5;$$

$$b) \int_1^2 (x^3 - 2x^2 + 3x) dx,$$

$$M = 7;$$

$$d) \int_2^7 (x^2\sqrt{x-1}) dx,$$

$$M = 36.$$

Pentru fiecare valoare a erorii se va afișa valoarea calculată a integralei și numărul de divizări, pentru care se obține aceasta.

- 6 Conturul unui teren este determinat lateral de dreptele verticale $x = -2$ și $x = 0$, superior – de graficul funcției $f(x) = x^3 - 5x + 3 \cos x$, inferior – de graficul funcției $f(x) = x^3 - x^2$. Scrieți un program care va calcula aria terenului, folosind metoda dreptunghiurilor, pentru:

$$a) 10\,000 \text{ de divizări};$$

$$b) \varepsilon = 0,00001.$$

5.2. Variații ale metodei dreptunghiurilor

În unele cazuri punctul care determină înălțimea dreptunghiului ce aproximează integrala pe un segment elementar este selectat nefiind mijlocul segmentului elementar, ci una dintre extremitățile lui. În acest caz, dacă aproximarea este realizată prin extremitățile stîngi (fig. 5.4) ale segmentelor elementare, aria dreptunghiului elementar $S_i = hf(x_i)$, iar integrala definită se aproximează prin suma:

$$I_{st} = \sum_{i=0}^{n-1} hf(x_i) = h \sum_{i=0}^{n-1} f(a+ih),$$

unde $h = \frac{|b-a|}{n}$, n – numărul de divizări ale segmentului inițial, iar metoda este numită *metoda dreptunghiurilor de stînga*.

Dacă aproximarea este realizată prin extremitățile drepte (fig. 5.5) ale segmentelor elementare, aria dreptunghiului elementar $S_i = hf(x_{i+1})$, iar integrala definită se aproximează prin suma:

$$I_{dr} = \sum_{i=1}^n hf(x_i) = h \sum_{i=1}^n f(a+ih),$$

unde $h = \frac{|b-a|}{n}$, n – numărul de divizări ale segmentului inițial, iar metoda este numită *metoda dreptunghiurilor de dreapta*.

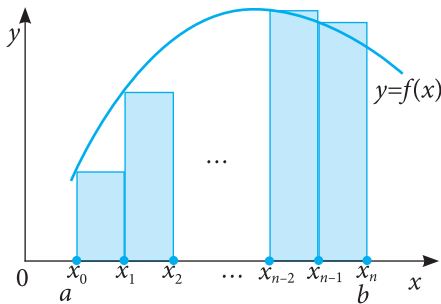


Fig. 5.4. Aproximarea integralei definite prin dreptunghiuri de stînga

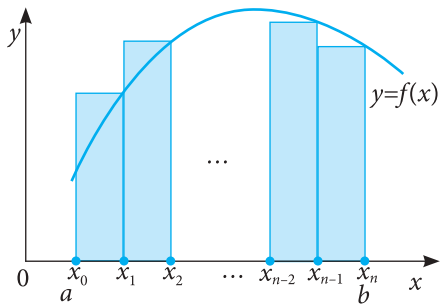


Fig. 5.5. Aproximarea integralei definite prin dreptunghiuri de dreapta

Eroarea de calcul pentru variațiile metodei dreptunghiurilor

În cazul aproximării integralei definite prin arii ale dreptunghiurilor de stînga (dreapta), formula de estimare a erorii se modifică neesențial în raport cu formula de bază:

$$\left| \int_a^b f(x) dx - I \right| \leq (b-a) M \frac{h}{2},$$

unde M – supremul $|f'(x)|$ pe $[a, b]$, I – valoarea calculată a integralei prin dreptunghiuri de stînga (dreapta).

La fel ca în cazul dreptunghiurilor de mijloc, numărul de divizări necesare pentru a obține o soluție calculată cu o eroare ce nu depășește o valoare prestabilită ε poate fi dedus direct din formula erorii:

$$n = \left\lceil M \frac{(b-a)^2}{2\varepsilon} \right\rceil + 1.$$

ALGORITMIZAREA METODEI (dreptunghiuri de stînga)²

Pasul 1. Inițializare: Se introduc valorile extremităților segmentului de integrare a, b și numărul de divizări n .

Notă. În cazul în care este necesar calculul integralei cu o eroare, care nu depășește o valoare prestabilită ε , numărul de divizări n se calculează cu ajutorul formulei $n = \left\lceil \frac{(b-a)^2 M}{2\varepsilon} \right\rceil + 1$.

Pasul 2. Se calculează lungimea segmentului elementar $h = \frac{|b-a|}{n}$. $S \leftarrow 0$.

Pasul 3. Pentru toți i de la 0 la $n-1$:

- Se calculează valorile $x_i \leftarrow a + ih$.
- Se calculează aria dreptunghiului elementar: $S_i \leftarrow f(x_i) \times h$.
- Aria calculată se sumează cu ariile precedente: $S \leftarrow S + S_i$.

Pasul 4. Se afișează aria totală calculată S . SFÎRȘIT.

Exemplul 3: Să se calculeze integrala definită $\int_0^{\pi} (x \sin(x))^2 dx$, utilizînd metoda dreptunghiurilor, variația dreptunghiurilor de stînga, pentru 10, 100, 1000 de divizări. Atribuirea valorilor inițiale se face direct în program. Pentru fiecare număr de divizări pe ecran se va afișa valoarea calculată și numărul de divizări, separate prin spațiu.

² Pentru dreptunghiurile de dreapta, algoritmul se deosebește doar în pasul 3 prin diapazonul valorilor pe care le primește indicele i : de la 1 la n .

```

program cn15;
const r=3;
var S, a, b, h : real;
    j,i,n:integer;
function f(x:real): real;
begin    f:=sqr((x* sin(x)));end;
begin    a:=0; b:=pi; n:=1;
        for j:=1 to r do
            begin S:=0; n:=n*10;
                h:=(b-a)/n;
                for i:=0 to n-1 do
                    S:=S+ h*f(a + i*h);
                writeln ('n=' ,n, ' I=' ,S:0:6);
            end;
        end.

```

Rezultate:

n=10	I=4.381796
n=100	I=4.382315
n=1000	I=4.382315

Exemplul 4: Să se calculeze integrala definită $\int_0^1 \sin x \sin 2x \sin 3x dx$, utilizând metoda dreptunghiurilor, variația dreptunghiurilor de dreapta, cu o eroare de calcul ce nu depășește valoarea $\varepsilon = 0,001$. Atribuirea valorilor inițiale se face direct în program. Pe ecran se va afișa valoarea calculată a integralei și numărul de divizări necesare pentru a obține precizia cerută, separate prin spațiu.

Rezolvare: Numărul necesar de divizări poate fi determinat analitic. Prin calcule elementare se determină valoarea $M = 6$:

$$n = \left\lceil M \frac{(b-a)^2}{2\varepsilon} \right\rceil + 1.$$

```

program cn16;
var S, a, b, e, h, M : real;
    j,i,n:integer;
function f(x:real): real;
begin f:=sin(x)* sin(2*x)* sin(3*x);end;
begin a:=0; b:=1; e:=0.001; M:=6;
        n:=trunc(M*(b-a)*(b-a)/(2*e))+ 1;
        S:=0; h:=(b-a)/n;
        for i:=1 to n do
            S:=S+ h*f(a + i*h);
        writeln ('n=' ,n, ' I=' ,S:0:6);
    end.

```

Rezultate:

n=3001	I=0.278729
--------	------------

Întrebări și exerciții

- 1 Care este diferența dintre metoda dreptunghiurilor de mijloc și variațiile sale?
- 2 Care este relația dintre numărul de divizări ale segmentului de integrare și eroarea de calcul a variațiilor metodei dreptunghiurilor?
- 3 Care dintre variațiile metodei dreptunghiurilor este mai exactă? De ce?
- 4 Scrieți un program care calculează integrala definită prin metoda dreptunghiurilor de stînga, de mijloc, de dreapta. Utilizați un număr variabil de divizări (de exemplu 10, 100, 1 000).

$$a) \int_1^3 (x \ln x)^2 dx;$$

$$b) \int_1^6 (x\sqrt{x-1}) dx;$$

$$c) \int_0^1 (x^2 \sqrt{1+3x^2}) dx;$$

$$d) \int_0^\pi (e^x \cos^2 x) dx;$$

$$e) \int_0^\pi (x \sin x)^2 dx;$$

$$f) \int_0^{2\pi} \frac{dx}{(2 + \cos x)(3 + \sin x)}.$$

Explicați diferența dintre rezultatele obținute pentru valori diferite ale numărului de divizări. Verificați rezultatele obținute folosind calculatorul on-line pentru integrale definite: www.solveymath.com/online_math_calculator/calculus/definite_integral.

- 5 Scrieți un program care calculează integrala definită prin metoda dreptunghiurilor cu o eroare care nu va depăși valoarea prestabilită $\epsilon = 0,005$; $\epsilon = 0,0005$; $\epsilon = 0,00005$. Valoarea M urmează să fie determinată prin metode analitice:

$$a) \int_1^3 (3x^2 + x + 2) dx;$$

$$b) \int_1^2 (2x^3 - x^2 + 5x) dx;$$

$$c) \int_{-3}^{-1} (2x^3 - 3x) dx.$$

Pentru fiecare valoare a erorii se va afișa valoarea calculată a integralei și numărul de divizări, pentru care se obține aceasta.

- 6 Scrieți un program ce va determina, folosind variațiile metodei dreptunghiurilor, care dintre integralele definite de mai jos este mai mare:

$$a) \int_0^\pi (e^{-x^2} \cos^2 x) dx \text{ sau } \int_\pi^{2\pi} (e^{-x^2} \cos^2 x) dx;$$

$$b) \int_0^1 (e^{-x}) dx \text{ sau } \int_0^1 (e^{-x^2}) dx.$$

5.3. Formula trapezelor

În paragrafele precedente, pentru aproximarea valorii integralei au fost folosite dreptunghiuri – figuri geometrice, suprafața cărora se calculează prin formule elementare.

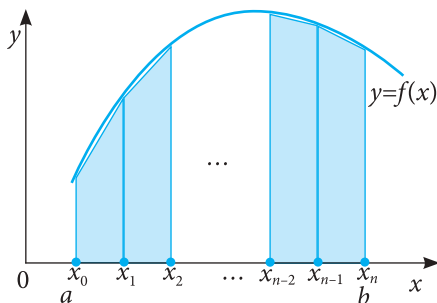


Fig. 5.6. Pe fiecare din segmentele elementare în care este divizat $[a, b]$, valoarea integralei definite este aproximată de aria unui trapez. Latura superioară a trapezului este segmentul cu extremitățile în punctele de coordonate $(x_i, f(x_i))$, $(x_{i+1}, f(x_{i+1}))$

Un neajuns al acestei metode este numărul mare de divizări, necesare pentru obținerea unor rezultate suficient de exacte. Este logică aproximarea mai exactă a integralei prin alte figuri geometrice, aria cărora poate fi calculată prin formule elementare. Una dintre figurile care permit acest lucru este trapezul (fig. 5.6).

Pentru calculul numeric al $\int_a^b f(x) dx$ se va folosi, la fel ca pentru metoda dreptunghiurilor, o divizare a segmentului de integrare în n segmente elementare de lungime egală. Lungimea h a unui segment elementar se determină conform formulei $h = \frac{|b-a|}{n}$, iar valorile x_i ale

punctelor care formează divizarea – prin formula:
 $x_i = a + ih, i = 0, \dots, n$.

Ca rezultat al divizării se obține un set de trapeze cu înălțimi egale (h). Bazele trapezului i sînt segmentele verticale care unesc punctele x_i și x_{i+1} cu graficul funcției $f(x)$. Lungimile bazelor sînt corespunzător $f(x_i)$ și $f(x_{i+1})$ (fig. 5.7).

Aria trapezului elementar determinat de punctele x_i și x_{i+1} va fi calculată după formula

$$S_i = h \frac{f(x_i) + f(x_{i+1})}{2}.$$

Prin urmare, pe $[a, b]$ valoarea calculată a integralei va fi:

$$I_{tr} = \frac{h}{2} \sum_{i=0}^{n-1} (f(x_i) + f(x_{i+1})).$$

Eroarea metodei

La fel ca în cazul formulei dreptunghiurilor, eroarea de calcul la integrare va fi cercetată ca suma integralelor erorilor de aproximare a funcției $f(x)$ printr-o funcție liniară pe fiecare segment elementar.

Eroarea de integrare pe segmentul $[a, b]$ este considerată ca fiind suma erorilor de integrare pe segmentele elementare și este dată de formula:

$$\left| \int_a^b f(x) dx - I_{tr} \right| \leq (b-a) \frac{M}{12} h^2,$$

unde M – supremul $f''(x)$ pe $[a, b]$, h – lungimea unui segment elementar.

În cazul cînd este necesar calculul integralei prin metoda trapezelor cu o eroare de calcul care nu depășește o valoare dată ε , numărul necesar de divizări ale segmentului de integrare, care asigură o eroare ce nu depășește valoarea predefinită,

poate fi determinat apriori din relația $(b-a) \frac{M}{12} h^2 \leq \varepsilon$ echivalentă cu $n \geq \sqrt{\frac{(b-a)^3 M}{12\varepsilon}}$,

$$\text{sau } n = \left\lceil \sqrt{\frac{(b-a)^3 M}{12\varepsilon}} \right\rceil + 1.$$

ALGORITMIZAREA METODEI

Pasul 1. Inițializare: Se introduc valorile extremităților segmentului de integrare a , b și numărul de divizări n .

Notă. În cazul în care este necesar calculul integralei cu o eroare ce nu depășește o valoare presta-

bilită ε , numărul de divizări n se calculează cu ajutorul formulei $n = \left\lceil \sqrt{\frac{(b-a)^3 M}{12\varepsilon}} \right\rceil + 1$.

Pasul 2. Se calculează lungimea segmentului elementar $h = \frac{|b-a|}{n}$. $S \leftarrow 0$.

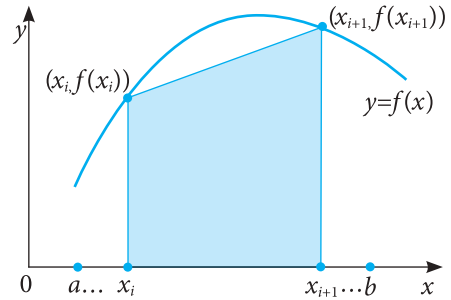


Fig. 5.7. Aproximarea valorii integralei definite pe un segment elementar prin aria trapezului determinat de axa Ox , dreptele $x = x_i$, $x = x_{i+1}$ și segmentul cu extremitățile $(x_i, f(x_i))$, $(x_{i+1}, f(x_{i+1}))$

Pasul 3. Pentru toți i de la 0 la $n-1$:

a) Se calculează valorile $x_i \leftarrow a+ih$; $x_{i+1} \leftarrow a + (i+1)h$.

b) Se calculează aria trapezului: $S_i \leftarrow h \frac{f(x_i) + f(x_{i+1})}{2}$.

c) Aria calculată se sumează cu ariile precedente: $S \leftarrow S + S_i$.

Pasul 4. Se afișează aria totală calculată S . SFÎRȘIT.

Exemplul 1: Să se calculeze integrala definită $\int_0^{2\pi} \frac{1}{\sin^4 x + \cos^4 x} dx$, utilizând metoda

trapezelor, pentru 20, 40, 80 de divizări ale segmentului de integrare. Atribuirea valorilor inițiale se face direct în program. Pentru fiecare număr de divizări pe ecran se va afișa valoarea calculată și numărul de divizări, separate prin spațiu.

Rezolvare: Numărul de divizări ale segmentului de integrare se modifică de la caz la caz prin formula $n \leftarrow 2n$.

```

program cn17;
const r=3;
var S, a, b, h : real;
    j,i,n:integer;
function f(x:real): real;
begin
    f:=1/(sqr(sqr(sin(x)))+sqr(sqr(cos(x)))));end;
begin
    a:=0; b:=2*pi; n:=10;
    for j:=1 to r do
        begin
            S:=0; n:=n*2;
            h:=(b-a)/n;
            for i:=0 to n-1 do
                S:=S+ h*(f(a + i*h)+f(a+(i+1)*h))/2;
                writeln ('n=',n, ' I=',S:0:11);
            end;
        end;
end.

```

Rezultate:

n=20	I=8.88312405500
n=40	I=8.88576626920
n=80	I=8.88576587630

Exemplul 2: Să se calculeze integrala definită $\int_1^3 (3x^3 - 7x^2 + 2) dx$, utilizând metoda trapezelor, cu o eroare de calcul ce nu depășește valoarea $\varepsilon = 0,001$. Atribuirea valorilor inițiale se realizează direct în program. Pe ecran se va afișa valoarea calculată a integralei și numărul de divizări necesar pentru a obține exactitatea cerută.

Rezolvare: Numărul de divizări ale segmentului de integrare se calculează direct cu

ajutorul formulei $n = \left\lceil \sqrt{\frac{(b-a)^3 M}{12\varepsilon}} \right\rceil + 1$. Prin calcule elementare se stabilește $M = 42$.


```

program cn18;
var S, a, b, e, h, M : real;
    j, i: integer;
    n: longint;
function f(x: real): real;
begin    f:=3*x*x*x-7*x*x+2;end;
begin    a:=1; b:=3; e:=0.001; M:=42;
        n:=trunc(sqrt(M*(b-a)*(b-a)*(b-a)/(12*e)))+ 1;
        S:=0;
        h:=(b-a)/n;
        for i:=0 to n-1 do
            S:=S+ h*(f(a + i*h)+ f(a + (i+1)*h))/2;
        writeln ('n=', n, ' I=', S:0:9);
end.

```

Rezultate:

n=168 I=3.333852986

Întrebări și exerciții

- ❶ Determinați condițiile în care poate fi aplicată metoda trapezelor.
- ❷ Care este deosebirea dintre metoda dreptunghiurilor și cea a trapezelor?
- ❸ Descrieți algoritmul metodei trapezelor pentru calculul numeric al integralei.
- ❹ Care dintre metodele studiate de calculul numeric al integralei definite este mai exactă? Motivați.
- ❺ Calculați integralele definite utilizând metoda trapezelor pentru 10, 20, 30 de divizări ale segmentului de integrare:

$$a) \int_1^4 e^{\sqrt{x+4}} dx;$$

$$c) \int_2^4 \left(x^3 + \frac{2x}{x^2+1} \right) dx;$$

$$e) \int_0^1 \sqrt{\sin x + \cos x} dx;$$

$$b) \int_1^2 \sqrt{3x^2 - 5x + 4} dx;$$

$$d) \int_0^{\pi} (x \sin x)^2 \cos x dx;$$

$$f) \int_{\frac{2\pi}{3}}^{\pi} (tgx^3 + \cos x \sin x)^2 dx.$$

- ❻ Scrieți un program care calculează integrala definită prin metoda trapezelor cu o eroare ce nu va depăși valoarea prestabilită $\varepsilon = 0,003$; $\varepsilon = 0,0003$; $\varepsilon = 0,00003$. Valoarea M urmează să fie determinată prin metode analitice:

$$a) \int_1^3 (e^x + 3) dx;$$

$$b) \int_1^2 (3x^2 - 5x + 4) dx;$$

$$c) \int_2^3 (x^3 + 3x - 8) dx.$$

Test de evaluare

I Selectați răspunsul corect:

- Metoda dreptunghiurilor permite calculul:
 - integralei definite;
 - integralei nedefinite;
 - ambelor tipuri de integrale.
- Pentru același număr de divizări ale segmentului de integrare, o eroare mai mică va fi obținută pentru:
 - metoda dreptunghiurilor (de mijloc);
 - metoda dreptunghiurilor (de stînga);
 - metoda dreptunghiurilor (de dreapta);
 - metoda trapezelor.
- Numărul de divizări ale segmentului de integrare, necesare pentru calculul integralei definite cu o eroare ce nu depășește valoarea prestabilită ε , poate fi determinat apriori:
 - numai pentru metoda trapezelor;
 - numai pentru metoda dreptunghiurilor;
 - pentru ambele metode;
 - pentru niciuna din metode.
- Fiind $\int_a^b f(x)dx$, n – numărul de divizări ale segmentului de integrare, iar $h = \frac{|b-a|}{n}$. Stabiliți

corelațiile dintre formulele de calcul ale integralei definite din coloana stîngă și denumirile metodelor respective din coloana dreaptă:

$$I = \frac{h}{2} \sum_{i=0}^{n-1} (f(x_i) + f(x_{i+1})) \quad \text{Dreptunghiuri de mijloc}$$

$$I = h \sum_{i=0}^{n-1} f\left(a + ih + \frac{h}{2}\right) \quad \text{Dreptunghiuri de stînga}$$

$$I = h \sum_{i=0}^{n-1} f(a + ih) \quad \text{Dreptunghiuri de dreapta}$$

$$I = h \sum_{i=1}^n f(a + ih) \quad \text{Trapeze}$$

- Fiind $\int_a^b f(x)dx$, ε – valoarea maximă admisibilă a erorii de calcul, M – supremul $|f'(x)|$ pe $[a, b]$.

Care dintre formulele de mai jos permite determinarea apriori a numărului de divizări ale segmentului de integrare, ce vor asigura o eroare de calcul fără a depăși valoarea ε pentru metoda dreptunghiurilor de mijloc:

$$\text{a) } n = \left\lceil \sqrt{\frac{(b-a)^3 M}{12\varepsilon}} \right\rceil + 1; \quad \text{b) } n = \left\lceil \frac{(b-a)^2 M}{4\varepsilon} \right\rceil + 1; \quad \text{c) } n = \left\lceil \frac{(b-a)^2 M}{2\varepsilon} \right\rceil + 1.$$

II Calculați integralele definite, folosind programele realizate anterior. Pentru fiecare valoare calculată se va afișa o linie care va conține cuvîntul „Răspuns:” urmat, după un spațiu, de soluția calculată cu șase cifre zecimale:

- Metoda dreptunghiurilor de stînga: $\int_2^4 \sqrt[3]{x^2 - 7x + 8} dx$ pentru 1 000 de divizări ale segmentului de integrare.
- Metoda dreptunghiurilor de mijloc (a trapezelor)³: $\int_0^{\frac{3\pi}{2}} \sqrt{\sin^2 x + x^3 - 8} dx$ pentru 5 000 de divizări ale segmentului de integrare.

³ Opțional.

CAPITOLUL 6

CONCEPTE GENERALE DESPRE BAZE DE DATE

După studierea acestui capitol, veți fi capabili să:

- explicați sensul termenilor *dată elementară*, *structură de date*, *bază de date*;
- recunoașteți date elementare, structuri de date și baze de date;
- descrieți structura bazelor de date ierarhice, în rețea și relaționale;
- recunoașteți modelul conceptual al unei baze de date.

6.1. Noțiuni și concepte despre date și despre baze de date

6.1.1. Date elementare și structuri de date

Data este un model de reprezentare a informației, accesibilă unui procesor (om, program etc.), model cu care se poate opera pentru a obține informații noi.

Data elementară este o entitate indivizibilă atât în raport cu informația pe care o reprezintă, cât și în raport cu procesorul logic (programul) sau cel fizic (unitatea centrală care o prelucrează).

Din punct de vedere logic, data elementară este caracterizată de:

- a) *identificatorul datei*;
- b) *valoarea datei*;
- c) *atributele datei*.

Identificatorul datei se folosește pentru a apela (a accesa) această dată.

Valoarea datei este conținutul zonei de memorie în care este stocată data. **Domeniul de definiție al datei** este mulțimea valorilor pe care le poate lua data în procesul prelucrării ei.

Atributele datei stabilesc caracteristici specifice datei. Unul dintre cele mai importante atribute ale datei este **tipul**, care definește apartenența datei la o anumită clasă de date. Fiecărui tip de date îi corespunde un model de reprezentare internă.

Exemplul 1: Declarația `var c : integer`, din Pascal, definește variabila `c` care în orice moment de timp al execuției programului identifică o dată elementară de tip *integer* ce va „ocupa” o zonă de memorie de doi octeți. Domeniul de definiție al tipului *integer* este mulțimea numerelor $\{-32\,768, -32\,767, \dots, 32\,767\}$. În urma instrucțiunii `c := 8`, în zona de memorie menționată se va stoca valoarea 8.

Exemplul 2: Valorile tipurilor simple din Pascal sînt date elementare.

Structura de date este un ansamblu de date, între elementele căruia sînt stabilite anumite relații care definesc metodele de identificare și prelucrare a acestor date. Structura

de date poate fi compusă din date elementare sau din alte structuri de date. Fiecare componentă a structurii de date poate fi localizată cu ajutorul identificatorului ei sau prin poziția ei în cadrul structurii.

Exemplul 3: Valorile tipului *record* din Pascal sînt structuri de date ale căror componente (cîmpurile) pot fi apelate folosind numele lor, adică identificatorii cîmpurilor. Un masiv este o structură de date, ale cărei componente (elementele masivului) pot fi apelate utilizînd poziția acestei componente (adică indicele ei).

Principalele caracteristici ale unei structurii de date sînt:

- a) *Omogenitatea*. O structură *omogenă* are toate componentele de același tip. Astfel, înregistrările (valori de tip *record*) sînt structuri neomogene, iar masivele – structuri omogene;
- b) *Modul de acces al componentelor*. Componentele unei structurii pot fi accesate direct sau secvențial. De exemplu, fișierele text sînt structuri de date cu acces secvențial;
- c) *Stabilitatea structurii*. Dacă pe parcursul execuției programului o structură nu-și schimbă numărul de elemente și relațiile dintre ele, atunci această structură este *statică*. În caz contrar, o astfel de structură se numește *structură dinamică*. De exemplu, listele înlanțuite sînt structuri dinamice, iar înregistrările – structuri statice;
- d) *Timpul de utilizare*. Structurile pot fi permanente sau temporare. De exemplu, fișierele externe sînt structuri de date permanente, iar masivele – structuri temporare.

6.1.2. Baze de date

Pînă în prezent am prelucrat diferite tipuri de date și structuri de date utilizînd un limbaj de programare sau o aplicație software (editor de texte, procesor tabelar etc.). Cea mai complexă structură de date prelucrată a fost fișierul.

Evident, informațiile despre o firmă, companie sau instituție pot fi păstrate în fișiere. De exemplu, în cazul unui liceu, se pot crea fișierele *Profesori*, *Elevi*, *Clase*, *Discipline* etc., în care se vor stoca datele, respectiv, despre profesori, elevi, clase, discipline studiate etc. Pentru a păstra informații despre repartizările profesorilor pe clase, se va crea alt fișier, *Profesor_Clasa*, pe baza conținuturilor fișierelor *Profesori* și *Clase*. Să presupunem că un șef de studii proaspăt angajat creează acest fișier. Dacă informația despre un profesor (din fișierul *Profesori*) este inexactă, atunci ea va apărea eronată și în tabelul *Profesor_Clasa*. Eventual, aceeași greșeală va fi și în fișierul *Orarul_lecțiilor*. Sesizînd-o, șeful de studii va fi nevoit să corecteze pe rînd toate cele trei fișiere. Mult mai simplu ar fi dacă actualizînd fișierul *Profesori*, automat vor fi actualizate și toate fișierele care „l-au utilizat”. În acest mod, s-ar asigura **integritatea datelor**.

În afară de *complexitatea actualizărilor*, metoda prelucrării datelor prin fișiere independente are și alte dezavantaje:

- *Prezența aceluiași date în cîteva fișiere independente* poate duce la duplicarea datelor, deci și la consumul mare de memorie.
- *Dificultatea obținerii informațiilor spontane* rezidă din independența fișierelor (de exemplu, pentru a obține lista profesorilor de sex feminin care predau științele exacte în clasa a 10-a, profil real, trebuie prelucrate cel puțin fișierele *Profesori*, *Clase*, *Discipline*).
- *Formatele incompatibile de fișiere* încetinesc prelucrarea informațiilor (de exemplu,

fișierele generate cu diferite limbaje de programare este posibil să aibă formate diferite, deci va fi necesară o aplicație care le va transforma în fișiere cu același format).

Acestea și alte probleme pot fi rezolvate dacă se folosește o structură de date de tip *bază de date*.

O **bază de date** este un ansamblu de informații (date), organizate într-un mod special, fapt care facilitează stocarea și extragerea lor. Informațiile dintr-o bază de date sînt păstrate sub formă de înregistrări sau de fișiere, între care există legături logice. În afară de depozitul de informații, baza de date conține descrierea lor: tipul, structura, modul de organizare și relațiile dintre ele.

Întrebări și exerciții

- 1 Numiți tipurile de date studiate. Care dintre ele sînt:
 - a) tipuri de date elementare;
 - b) tipuri de date structurate?
- 2 Selectați datele elementare: simbol, număr real, șir de caractere, masiv de numere, element al unei mulțimi, număr întreg, valoarea *false*, fișier.
- 3 Care este domeniul de definiție al tipului:
 - a) `integer`;
 - b) `char`;
 - c) `lit_M` declarat astfel `type lit_M = 'A'.. 'Z'`;
 - d) `boolean`?
- 4 Caracterizați structurile de date:
 - a) mulțime;
 - b) masiv;
 - c) înregistrare;
 - d) listă înlănțuită;
 - e) fișier.

Repere: omogenitate, mod de acces, stabilitate și timp de utilizare.
- 5 Explicați sensul noțiunii *bază de date*.
- 6 Argumentați de ce pentru păstrarea și prelucrarea informației unei companii este mai avantajos utilizarea unei baze de date comparativ cu utilizarea unui sistem de fișiere.
- 7 Care sînt deosebirile dintre o bază de date și o structură de date?

6.2. Tipuri de baze de date

După cum vom vedea în capitolul următor, un pas important în elaborarea bazelor de date este proiectarea *modelului conceptual* de date al bazei, care nu depinde de parametrii fizici ai mediului de păstrare a datelor și definește modul lor de organizare.

Modelul conceptual al unei baze de date reprezintă o descriere generală a bazei de date cu ajutorul limbajului natural, al celui matematic, al diagramelor și organigramelor, al graficelor și al altor mijloace înțelese de cei care elaborează baza de date.

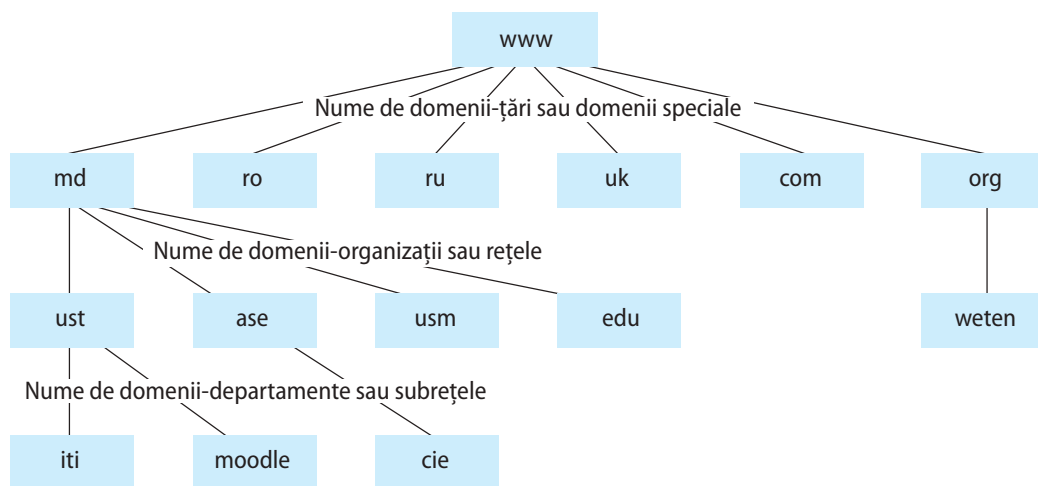
Cele mai răspîndite modele conceptuale sînt:

- a) *ierarhic* (sau arborescent);
- b) *rețea*;
- c) *relațional*.

În funcție de modelul ei conceptual, baza de date se numește *ierarhică*, *în rețea* sau *relațională*.

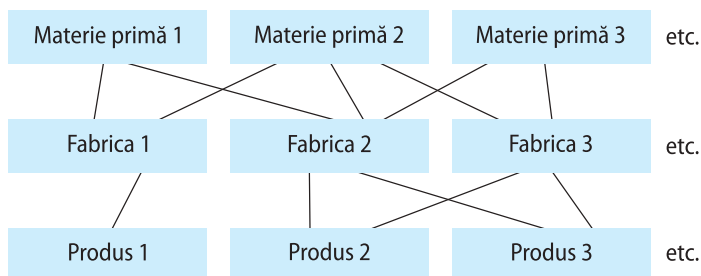
Entitățile (colecțiile de date) ale **bazei de date ierarhice** (*hierarchical database*) sînt organizate sub formă de *noduri*, la care sînt conectate ramurile unui *arbore*. Fiecare nod se subordonează cel mult unui nod de nivel ierarhic imediat superior (numit *nod-părinte*) și poate fi legat cu niciunul, unul sau cu cîteva noduri de nivel inferior (numite *noduri-copii*).

Exemplu: Baza de date a adreselor Internet are următorul model ierarhic:



În cazul **bazei de date în rețea** (*network database*), entitățile la fel sînt aranjate ierarhic, însă unui nod-copil poate să-i corespundă cîteva noduri-părinte.

Exemplu: Baza de date a unei ramuri industriale poate avea următorul model:



O bază de date relațională (*relational database*) are cel mai flexibil model conceptual de organizare a datelor. Parcurgerea entităților unei astfel de baze nu este ierarhică.

Modelul relațional este simplu, dar riguros din punct de vedere matematic. Pentru prima dată acest model a fost propus în 1970 de savantul Edgar Frank Codd*.

* *Edgar Frank Codd* (23.08.1923, Insula Portland, Anglia – 18.04.2003, Williams Island, Florida, SUA) – informatician american de origine engleză care, lucrînd pentru IBM, a inventat modelul relațional pentru gestiunea bazelor de date. Pentru contribuțiile sale în domeniul informaticii, în 1981 obține Premiul Turing, considerat „Premiul Nobel pentru Informatică”.

Baza de date relațională este formată din tabele, denumite *relații*. Fiecare tabel este format din rînduri și coloane. Rîndurile se numesc *înregistrări* de date, iar coloanele – *cîmpuri* de date. *Capul de tabel* (sau *antetul tabelului*) definește structura tabelului. Prin *crearea unui tabel* se subînțelege, de fapt, definirea capului de tabel.

În *figura 6.1* este reprezentată o parte din schema conceptuală a unei baze de date relaționale.

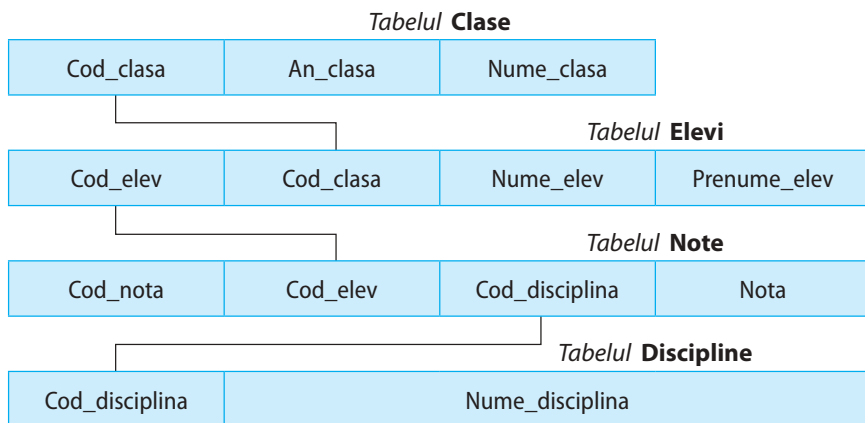


Fig. 6.1

Între tabelele unei baze de date relaționale există interdependențe (despre care vom afla mai detaliat în următoarele capitole).

Observație: În continuare vom examina doar baze de date relaționale.

Întrebări și exerciții

- ❶ Descrieți structura bazelor de date:
 - a) ierarhice;
 - b) în rețea;
 - c) relaționale.
- ❷ Stabiliți tipul bazei de date știind că:
 - a) între fiecare două entități ale bazei există o legătură;
 - b) entitățile bazei sînt noduri ale unui arbore binar;
 - c) o entitate este legată cu toate celelalte entități și alte legături nu există;
 - d) informațiile din bază sînt organizate în zece tabele.
- ❸ Reprezentați schematic un model ierarhic pentru o bază de date cu informații despre:
 - a) un liceu;
 - b) un magazin de produse alimentare;
 - c) o bibliotecă.
- ❹ Rezolvați exercițiul ❸ pentru cazul:
 - a) modelul rețea;
 - b) modelul relațional.
- ❺ Stabiliți tipul bazei de date a numerelor de telefoane fixe din țară.

CAPITOLUL 7

ELABORAREA ȘI GESTIUNEA BAZELOR DE DATE

După studierea acestui capitol, veți fi capabili să:

- descrieți etapele de elaborare a unei baze de date;
- explicați rolul persoanelor antrenate în elaborarea bazelor de date;
- descrieți structura și funcțiile sistemelor de gestiune a bazelor de date;
- recunoașteți structura tabelelor unei baze de date relaționale;
- identificați tipul relației dintre două tabele;
- alegeți sau să creați o cheie primară pentru un tabel;
- aplicați unele principii de proiectare a bazelor de date la elaborarea modelului conceptual al bazei.

7.1. Elaborarea unei baze de date

7.1.1. Aspecte generale

Un **sistem de bază de date** reprezintă un sistem de organizare și de prelucrare a bazei de date și este format din baza de date propriu-zisă și dintr-un ansamblu de programe prin care se asigură gestionarea și prelucrarea complexă a datelor.

Prin **proiectarea sau elaborarea unei baze de date** se subînțelege crearea sistemului acestei baze de date.

Proiectarea unei baze de date constă, de regulă, din următoarele **etape**:

1. Etapa de analiză. Se analizează domeniul de date și se depistează cerințele tuturor categoriilor de utilizatori către sistemul de bază de date, precum și operațiile pe care le va gestiona acest sistem. În urma acestei analize se proiectează modelul conceptual al bazei de date. De asemenea, se stabilește structura interfeței principale a sistemului de bază de date și a celor auxiliare.

2. Etapa de programare. Se creează componentele logice (programele): interfața principală, aplicațiile de introducere/actualizare/prelucrare a datelor, cele de interogare a bazei de date și de extragere a informațiilor.

3. Punerea în funcțiune și exploatarea sistemului de bază de date. Baza de date se completează cu date (înregistrări). În continuare se realizează operații de actualizare, de consultare și de întreținere ulterioară (inclusiv dezvoltare) a sistemului de bază de date.

Elaborarea și întreținerea unei baze de date se realizează, în genere, de un grup de persoane: *administratori* (care stabilesc structura inițială a bazei de date, modul de memorare a datelor la nivel fizic; acordă utilizatorilor drepturi de acces la baza de date, asigură securitatea datelor, modifică structura și dezvoltă baza de date), *programatori* (care scriu programele în limbaje de programare), *proiectanți* (care realizează designul interfețelor bazei de date) etc.

Observație: Există sisteme computerizate moderne, numite Sisteme de Gestiune a Bazelor de Date (SGBD), care au fost elaborate special pentru a crea sisteme de bază de date. Mai detaliat aceste SGBD-uri vor fi examinate în paragraful următor.

7.1.2. Proiectarea entităților unei baze de date relaționale

În capitolul precedent am menționat că entitățile unei baze de date relaționale sînt **tabelele** în care se păstrează *înregistrări* de date (rînduri de date).

Capul de tabel (sau *antetul tabelului*) definește structura tabelului. Prin *crearea unui tabel* se subînțelege, de fapt, definirea capului de tabel, adică descrierea cîmpurilor (a coloanelor) tabelului.

Exemplu: Considerăm o bază de date, denumită *Liceu*, în care sînt stocate informații despre un liceu. Tabelul *Elevi* al acestei baze de date păstrează informații despre elevii liceului:

Tabelul *Elevi*

Cod_elev	Nume_elev	Pren_elev	Cod_clasa	Data_elev	Foto_elev	Telefon	Gen_elev
e001	Bacinschi	Sabina	c01	28.09.1993	Package	29-82-54	F
e002	Belobrov	Andreea	c01	18.10.1993	Package	44-26-48	F
e003	Brîncă	Carmen	c01	20.03.1993	Package	67-46-64	F

La descrierea fiecărui cîmp al unui tabel se specifică *numele* și *tipul* cîmpului.

- Numele cîmpului* trebuie să fie unic în cadrul tabelului. Datele unui cîmp sînt omogene;
- Tipul cîmpului* definește tipul valorilor cîmpului (numeric, șir de caractere, text mare, dată calendaristică, logic, imagine etc.).

Astfel, primul cîmp are numele *Cod_elev* și tipul *Șir de caractere*, iar al cincilea cîmp – numele *Data_elev* și tipul *Dată calendaristică*.

Fiecare valoare din cîmpul *Cod_elev* al tabelului *Elevi* corespunde unei singure înregistrări.

Cîmpul ale cărui valori identifică fiecare înregistrare a tabelului se numește **cheie primară**. Evident, un cîmp este cheie primară doar dacă valorile lui nu sînt nule și nu se repetă.

Se recomandă ca fiecare tabel al unei baze de date să conțină un cîmp-cheie primară. Uneori, rolul cheii primare îi poate reveni cîtorva cîmpuri (dacă acestea împreună identifică univoc rîndurile tabelului).

Dacă valorile unui cîmp se repetă, atunci acest cîmp se numește **cheie secundară**. Astfel, cîmpul *Cod_elev* este cheie primară, iar *Cod_clasa* – cheie secundară ale tabelului *Elevi*.

De regulă, fiecare tabel al unei baze de date relaționale se află într-o legătură cu cel puțin un alt tabel al aceleiași baze. Există trei **tipuri de relații între tabele**:

- unu la unu*;
- unu la mulți*;
- mulți la mulți*.

Într-o relație de tipul **unu la unu** (se notează 1:1), fiecărei înregistrări a unui tabel îi poate corespunde cel mult o înregistrare a celuilalt tabel și invers. Acest tip de relație se obține atunci cînd un tabel, avînd prea multe cîmpuri, este divizat în două tabele. În acest caz cîmpurile primare ale tabelelor vor avea date identice.

De exemplu, între tabelele *Elevi* și *Adrese_elevi* a fost creată o relație de tipul *unu la unu*. Observăm că ambele tabele au același cîmp-cheie primară *Cod_elev* cu date identice.

Tabelul Adrese_Elevi

Cod_elev	Loc_elev	Strada_elev	Nr_casa_elev	Ap_elev
e001	Chișinău	Mihail Sadoveanu	40	23
e002	Stăuceni	Viei	9	

Schematic această relație este reprezentată în figura 7.1.

În cazul relației **unu la mulți** (se notează 1:∞ sau 1:M), fiecărei înregistrări a unui tabel îi pot corespunde câteva înregistrări ale celui alt tabel, iar fiecărei înregistrări din tabelul al doilea îi poate corespunde cel mult o înregistrare a primului tabel. Pentru a stabili o astfel de relație, primul tabel trebuie să conțină o cheie primară, iar cel de-al doilea – o cheie secundară de tip compatibil cu tipul cheii primare a primului tabel.

De exemplu, tabelul *Clase* se află în relația **unu la mulți** cu tabelul *Elevi*, deoarece fiecărei clase „îi corespund câțiva” elevi, iar fiecare elev „aparține” unei singure clase.

Tabelul Clase

Cod_clasa	Anul_de_studii	Nume_clasa	Cod_prof
c01	10	A	p1
c02	10	B	p1
c03	10	C	p2
c04	10	D	p2

Cheia primară *Cod_clasa* a tabelului *Clase* este „compatibilă” cu cheia secundară *Cod_clasa* a tabelului *Elevi*. Schematic această relație este reprezentată în figura 7.2.

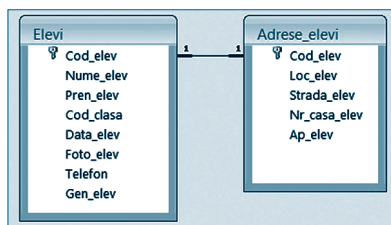


Fig. 7.1

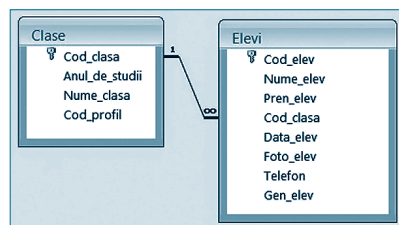


Fig. 7.2

Observație: Cheia secundară în cazul relației unu la mulți se mai numește *cheie străină*.

Într-o relație **mulți la mulți** (se notează ∞:∞ sau M:M), fiecărei înregistrări a unui tabel îi pot corespunde câteva înregistrări ale celui alt tabel și invers. De exemplu, tabelele *Clase* și *Discipline* se află în relația **mulți la mulți**, deoarece în fiecare clasă se predau mai multe discipline și fiecare disciplină poate fi predată în câteva clase.

Tabelul Discipline

Cod_disciplina	Nume_disciplina
d01	Matematica
d02	Informatica
d03	Chimia
d04	Fizica

Tabelul Prof_dis_clasa

Cod_repart	Cod_clasa	Cod_dis	Cod_prof
r001	c01	d01	prof01
r002	c01	d02	prof01
r003	c01	d03	prof02
r004	c01	d05	prof04

Relația *mulți la mulți* dintre două tabele se poate stabili prin două relații de tipul *unu la mulți*. Fiecare dintre cele două tabele se leagă într-o relație *unu la mulți* cu un al treilea tabel.

De exemplu, relația *mulți la mulți* dintre tabelele *Clase* și *Discipline* se realizează prin tabelul *Prof_dis_clasa* (fig. 7.3).

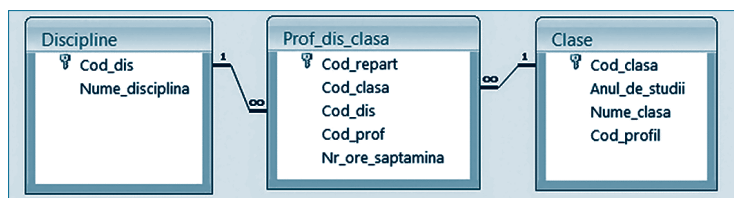


Fig. 7.3

Observație: Prima înregistrare din tabelul *Prof_dis_clasa* informează că în clasa c01 (adică în clasa a 10-a A) disciplina d01 (adică *Matematica*) este predată de profesorul prof01 (datele despre acest profesor se păstrează în tabelul *Profesori*, p. 74).

7.1.3. Principii de proiectare

Proiectînd o bază de date relațională, programatorii definesc tabelele astfel, încît ele să respecte anumite principii (de exemplu, să fie aduse la așa-numitele *forme normale*).

- Unul dintre aceste principii presupune că datele cîmpurilor tabelelor trebuie să fie **date elementare** (vezi paragraful 6.1 al capitolului 6). Din aceste considerente, adresa unui elev sau a unui profesor se păstrează în cîteva cîmpuri. Este evident că prelucrarea înregistrărilor ar fi fost mai dificilă dacă adresa era stocată într-un singur cîmp.

- Pentru ca actualizarea înregistrărilor din tabele să fie optimală (de exemplu, să fie mai rapidă), se recomandă excluderea multiplă a repetărilor valorilor fiecărui cîmp (excepție făcînd cazul cîmpurilor legate cu chei primare ale altui tabel) prin **descompunerea tabelului** respectiv în cîteva tabele.

De exemplu, datele despre profesori și despre disciplinele pe care ei le predau în clase sînt stocate în patru tabele: *Profesori*, *Discipline*, *Clase* și *Prof_dis_clase*. Așa cum între aceste tabele există relații, pentru a modifica în toate înregistrările numele unui profesor, este suficient să efectuăm schimbarea numelui doar în tabelul *Profesori*.

Întrebări și exerciții

- 1 Numiți etapele de proiectare a unei baze de date.
- 2 Care sînt entitățile unei baze de date relaționale?
- 3 Explicați principiile de proiectare a unei baze de date relaționale.
- 4 Prin ce se deosebește relația M : M de relația 1 : M?
- 5 Examinați tabelul *Elevi* al bazei de date *Liceu* și stabiliți tipul valorilor fiecărui cîmp al acestui tabel.
- 6 Alegeți o cheie primară pentru tabelul:
 - a) *Clasa_12* cu cîmpurile *Număr_de_ordine*, *Nume_elev*, *Prenume_elev*, *Telefon_elev*;
 - b) *Angajați* cu cîmpurile *Nume_angajat*, *Prenume_angajat*, *Virsta_angajat*, *Gen_angajat*, *Număr_buletin_de_identitate_angajat*;
 - c) *Parcare_auto* cu cîmpurile *Marca_auto*, *Număr_de_înmatriculare_auto*, *Foto_auto*, *Nume_proprietar_auto*;
 - d) *Țări* cu cîmpurile *Nume_țară*, *Suprafață_țară*, *Capitală_țară*, *Număr_locuitori_țară*.

- 7) Ce tip de relație se poate stabili între tabelele:
- Vinuri și Producători_de_vinuri;*
 - Cărți și Autori;*
 - Poezii și Autori;*
 - Specialități și Universități?*
- 8) Proiectați tabelele bazei de date care va conține informații:
- dintr-o agendă de telefoane;
 - despre cărțile bibliotecii personale;
 - despre automobile;
 - despre marile personalități ale țării noastre.

7.2. Sisteme de gestiune a bazelor de date (SGBD)

7.2.1. Concepte generale despre sisteme de gestiune a bazelor de date

Un **sistem de gestiune a bazelor de date** este un ansamblu de programe care permit construirea bazelor de date, introducerea și actualizarea datelor, asigurarea controlului de acces al acestor date, gestionarea informațiilor din baza de date, precum și crearea aplicațiilor cu baze de date. Un SGBD exercită următoarele funcții:

- definește baza de date*, în sensul că descrie tipurile și structurile de date, relațiile dintre ele și modalitățile de accesare a informațiilor din bază;
- actualizează baza de date*, adică permite inserarea, redactarea și ștergerea datelor;
- execută interogări către baza de date*, în urma cărora informațiile se sortează sau se filtrează după anumite criterii formulate de utilizatori;
- crează date noi*, care se obțin în baza efectuării unor calcule, inclusiv a totalizărilor;
- crează rapoarte* simple și complexe sub formă de tabele, grafice, diagrame etc.;
- asigură întreținerea bazei de date*, care presupune repararea bazei în cazul unor defecțiuni, compactarea (sau defragmentarea) ei, precum și crearea copiilor de siguranță atât pentru date, cât și pentru obiectele bazei;
- protejează baza de date* de accesul care nu este autorizat și stabilește această autorizare pentru acces complet sau partajat.

• Software-ul SGBD este format din:

- Dicționarul de date (Data Dictionary)*, care conține o descriere a structurii datelor folosite în baza de date;
- Limbajul de interogare (Query Language)*, care asigură accesul la informațiile din baza de date. Cel mai răspândit limbaj de interogare (folosit în diferite SGBD) este limbajul SQL (*Structured Query Language*).

• Hardware-ul SGBD poate fi format:

- dintr-un singur calculator* sau
- dintr-o rețea de calculatoare*, în care pe calculatorul principal (*serverul*) se află programele-componente din SGBD care administrează și controlează accesul către baza de date, iar pe celelalte calculatoare – programele destinate utilizatorilor.

Cele mai răspândite SGBD sînt Microsoft Office Access, Oracle, FoxPro, Paradox, dBase, Microsoft SQL Server etc.

7.2.2. Sistemul de gestiune a bazelor de date Microsoft Office Access

Microsoft Office Access (în continuare vom spune doar Access) este un sistem de gestiune a bazelor de date relaționale care funcționează în mediul Windows. Prima sa versiune a fost lansată în anul 1993. Cu ajutorul lui pot fi create baze de date complexe, avînd pentru acest scop tot instrumentarul necesar. Comparativ cu alte SGBD-uri, este comod și simplu în utilizare. Pentru crearea interogărilor, formularelor, rapoartelor și a altor produse de baze de date, Access pune la îndemîna programatorilor și utilizatorilor un set de programe de asistență, care automatizează diferite etape de lucru.

Sistemul Access poate fi lansat în cîteva moduri:

- executînd un dublu-clic pe referința Access de pe suprafața de lucru Windows;
- selectînd opțiunea Microsoft Office Access cu ajutorul meniului Start;
- executînd un dublu-clic pe pictograma oricărui fișier cu extensia .mdb.

Fiind lansat Access, putem **crea o bază de date**, selectînd *File* → *New* → *Blank database...*

Apare fereastra *File New Database*. În caseta *File name*: scriem numele bazei de date pe care dorim să o creăm. Cu ajutorul listei derulante *Save in* selectăm catalogul în care vom păstra această bază de date.

În interiorul ferestrei Access apare **fereastra bazei de date** cu numele ei.

Suprafețele de lucru ale variatelor versiuni ale sistemului Access nu sînt total diferite. Fiecare dintre aceste suprafețe are următoarele elemente de bază:

- bara de titlu Access;
- bara cu meniuri (*File, Edit, View, Insert, Tools, Window, Help*) a ferestrei Access;
- bara cu instrumente Access;
- fereastra bazei de date cu opțiuni (sau file) pentru crearea și gestionarea celor șapte clase de obiecte Access (*Tables* – tabele, *Queries* – interogări, *Forms* – formulare, *Reports* – rapoarte, *Pages* – pagini, *Macros* – macroinstrucțiuni, *Modules* – module cu instrucțiuni în limbajul *Visual Basic for Applications*). Obiectele bazei de date se pot accesa prin intermediul pictogramelor corespunzătoare lor, amplasate pe bara *Objects* a ferestrei bazei de date;
- bara cu meniuri a ferestrei de date (*Open, Design, New*).

În continuare vom explica cum se elaborează și se gestionează cu Access o bază de date. Vom crea și vom prelucra baza de date *Liceu*, care va păstra informații despre elevii și profesorii unui liceu.

7.2.3. Structura bazei de date *Liceu*

Baza de date *Liceu* va fi formată din 8 tabele legate între ele (fig. 7.4).

Prezentăm structura celor 8 tabele. Atragem atenția că aici sînt arătate doar cîteva înregistrări ale fiecăruia din *tabelele 2-8*.

Tabelul 1. **Profiluri**

Cod_profil	Nume_profil
p1	Real
p2	Umanist

Tabelul 2. **Discipline**

Cod_dis	Nume_disciplina
d01	Matematica
d02	Informatica

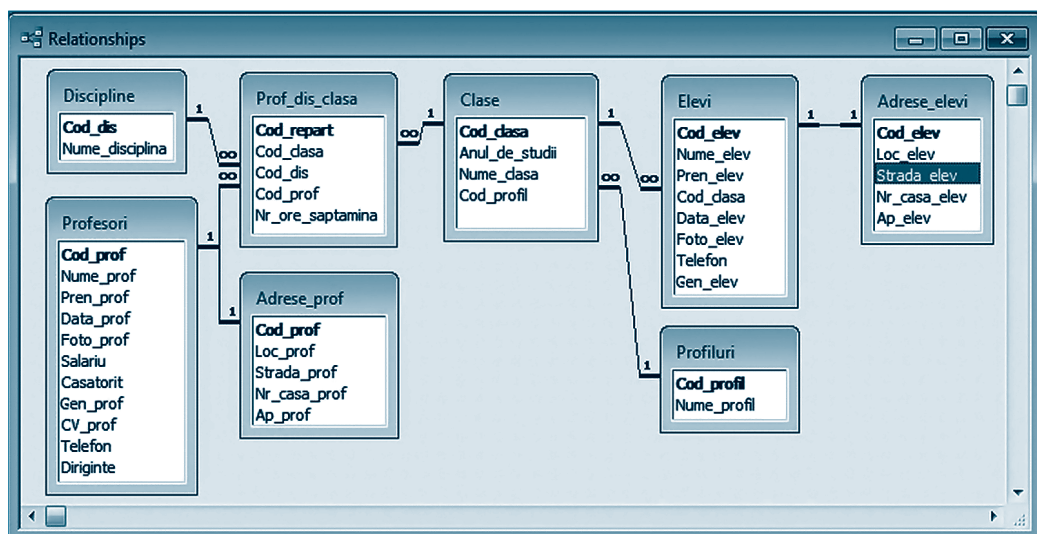


Fig. 7.4

Tabelul 3. Profesori

Cod_prof	Nume_prof	Pren_prof	Data_prof	Foto_prof	Salariu	CV_prof	Casatorit	Gen_prof	Telefon	Diriginte
prof01	Albu	Ion	01.03.1968	Package	3500	Grad...	Yes	M	64-41-72	C02
prof02	Moraru	Vasile	11.05.1973	Package	3000	S-a năs...	No	M	29-10-16	

Tabelul 4. Adrese_prof

Cod_prof	Loc_prof	Strada_prof	Nr_casa_prof	Ap_prof
prof01	Cricova	Vinului	89	
prof02	Chişinău	Grenoble	81	77

Tabelul 5. Clase

Cod_clasa	Anul_de_studii	Nume_clasa	Cod_prof
c01	10	A	p1
c02	10	B	p1

Tabelul 6. Elevi

Cod_elev	Nume_elev	Pren_elev	Cod_clasa	Data_elev	Foto_elev	Telefon	Gen_elev
e001	Bacinschi	Sabina	c01	28.09.1998	Package	29-82-54	F
e002	Belobrov	Andreea	c01	18.10.1998	Package	44-26-48	F

Tabelul 7. Adrese_elevi

Cod_elev	Loc_elev	Strada_elev	Nr_casa_elev	Ap_elev
e001	Chişinău	Mihail Sadoveanu	40	23
e002	Săuceni	Viei	9	

Tabelul 8. Prof_dis_clasa

Cod_repart	Cod_clasa	Cod_dis	Cod_prof
r001	c01	d01	prof01
r002	c01	d02	prof01

Observație: Tabelele bazei de date *Liceu* pot fi descărcate de pe adresa www.bd.ust.md.

Întrebări și exerciții

- 1 Descrieți structura unui sistem de gestiune a bazelor de date.
- 2 Explicați funcțiile sistemelor de gestiune a bazelor de date.
- 3 Cum poate fi lansat SGBD Access?
- 4 Examinați suprafața de lucru a SGBD Access și caracterizați clasele de obiecte Access.
- 5 Observați înregistrările din tabelele bazei de date *Liceu* și stabiliți:
 - a) profilul clasei în care învață eleva Sabina Bacinschi;
 - b) numele profesorului care predă Informatica în clasa a 10-a A;
 - c) denumirea unei discipline predate de profesorul Ion Albu;
 - d) adresa dirigintelui clasei a 10-a A.

Test de evaluare

1. Care dintre următoarele valori sînt date elementare în limbajul Pascal:
 - a) șirul de caractere 'Informatica';
 - b) constanta caracterială 'E';
 - c) valoarea logică *false*;
 - d) vectorul *V* cu elemente de tip *byte*;
 - e) constanta *Pi*;
 - f) numărul 32 109?
2. Stabiliți tipul bazei de date, ale cărei entități sînt legate între ele ca-n imagine:



Argumentați răspunsul.

3. Descrieți structura tabelului în care pot fi păstrate informații despre notele obținute la Matematică de elevii clasei a 12-a pe parcursul unui semestru.
4. Alegeți o cheie primară pentru tabelul:
 - a) *Lista clasei a 12-a* cu câmpurile *Număr_de ordine*, *Nume*, *Prenume*, *Data_nașterii*, *Gen*;
 - b) *Pagini_Web* cu câmpurile *Nume_pagină*, *Adresa_pagină*, *Limba_pagină*, *Autor_pagină*;
 - c) *Filme* cu câmpurile *Nume_film*, *Autor_film*, *Gen_film*, *Durata_film*.
5. Determinați tipul de relație între tabelele:
 - a) *Mărci_auto* și *Proprietari_auto*;
 - b) *Profesori* și *Elevi*;
 - c) *Copii* și *Mame*;
 - d) *Țări* și *Capitale*;
 - e) *Canale_TV* și *Filme*.
6. a) Proiectați tabelele bazei de date *Agenda elevului* în care se vor stoca informații din agenda unui elev.
b) Ce tipuri de relații există între tabelele proiectate?

CAPITOLUL 8

TABELELE – PRINCIPALELE OBIECTE ALE UNEI BAZE DE DATE

După studierea acestui capitol, veți fi capabili să:

- creați tabele;
- modificați structura unui tabel;
- stabiliți proprietăți pentru câmpurile tabelului;
- creați reguli de validare a datelor care urmează a fi introduse în tabel;
- elaborați șabloane de restricționare a caracterelor ce urmează a fi introduse într-un câmp al tabelului;
- stabiliți relații între două tabele;
- introduceți date într-un tabel;
- editați înregistrările unui tabel;
- modificați aspectul unui tabel;
- utilizați expresii pentru crearea regulilor de validare a datelor.

8.1. Crearea tabelelor

Am menționat deja că entitățile unei baze de date Access sînt tabelele. Cu ajutorul lor se pot crea celelalte obiecte ale bazei de date: interogări, formulare, rapoarte etc. Prin urmare, o bază de date relațională nu are sens fără existența tabelelor.

8.1.1. Crearea structurii unui tabel

Prin **structura unui tabel** se subînțelege informația care descrie câmpurile tabelului: numărul lor, tipul și proprietățile fiecărui câmp, câmpurile care constituie cheia primară etc.

Pentru a defini structura unui tabel:

1. Selectăm pictograma *Tables* de pe bara de obiecte *Objects* a ferestrei bazei de date (de fapt, la crearea sau deschiderea unei baze de date această pictogramă este automat selectată). În zona de conținut a ferestrei apar opțiunile:

- *Create table in Design view* (crearea tabelului în regim de proiectare);
- *Create table by using wizard* (crearea tabelului cu ajutorul unui program de asistență);
- *Create table by entering data* (crearea tabelului prin introducerea directă a datelor).

Executăm un dublu-clic pe prima opțiune.

Observație: *Pasul 1* este echivalent cu selectarea *New* → *Design view*.

2. Apare fereastra de dialog de tip *Table* cu numele implicit *Table1* (fig. 8.1). Ea este formată din două zone:

- a) zona pentru descrierea structurii documentului;
- b) zona descrierii proprietăților câmpului selectat din prima zonă (*Field Properties*).

Zona pentru descrierea structurii documentului este divizată în trei coloane:

- *Field Name* (identificatorul câmpului);
- *Data Type* (tipul câmpului, adică al valorilor lui);
- *Description* (descrierea câmpului).

Astfel, pentru fiecare câmp al tabelului ce urmează a fi creat se precizează identificatorul, tipul și descrierea câmpului.

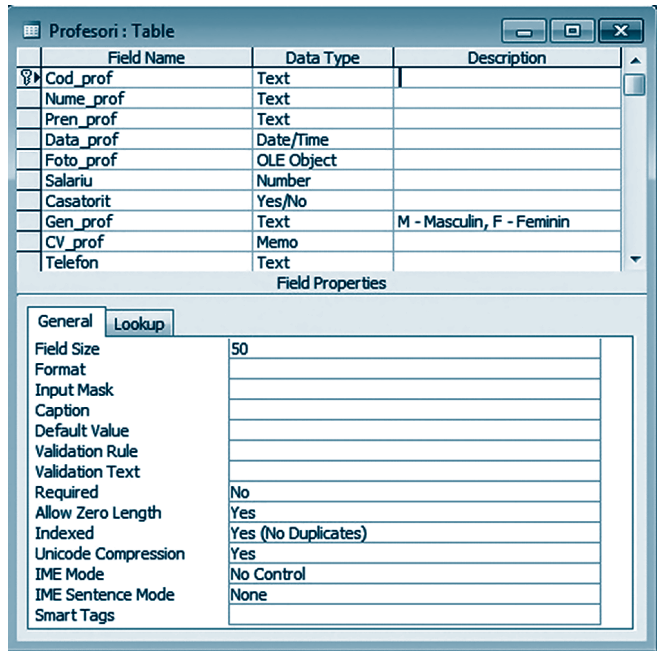



Fig. 8.1

Identificatorul câmpului poate conține diferite caractere, în afară de semnele „, ”, „, ”, „[”, „]”, spațiul de debut și caracterele care nu sînt vizibile (de exemplu, returul de car). Lungimea identificatorului câmpului nu poate fi mai mare decît 64 de caractere.

Tipurile câmpurilor acceptate de Access sînt prezentate în următorul tabel:

Denumirea tipului	Descrierea valorilor tipului
<i>Text</i>	Șiruri de caractere alfanumerice. Pînă la 255 de caractere
<i>Memo</i>	Texte mari. Pînă la 64 Ko
<i>Number</i>	Numere
<i>Date/time</i>	Date calendaristice
<i>Currency</i>	Valori monetare
<i>AutoNumber</i>	Numerele naturale 1, 2, 3, ... inserate în ordine și în mod automat la adăugarea unei înregistrări
<i>Yes/No</i>	Valorile <i>Yes</i> sau <i>No</i>
<i>OLE Object</i>	Imagini sau sunete
<i>Hiperlink</i>	Adrese Web

Descrierea câmpului (nu este obligatorie) poate conține note explicative referitor la câmp. Proprietățile câmpurilor vor fi descrise mai tîrziu.

3. După ce am definit câmpurile tabelului, **stabilim o cheie primară** pentru tabel. Cu acest scop selectăm câmpul necesar (ale cărui valori nu se vor repeta), apoi alegem comanda *Primary Key* din meniul flotant al câmpului (sau executăm un clic pe instrumentul  de pe bara cu instrumente Access). În tabelul din *figura 8.1*, câmpul *Cod_prof* a fost

ales drept cheie primară. Atunci când nu definim o cheie primară, sistemul Access va sugera stabilirea unei astfel de chei imediat după salvarea tabelului.

Dacă utilizatorul va accepta sugestia, atunci sistemul va stabili drept cheie primară primul câmp de tip *AutoNumber* sau (dacă tabelul nu conține un astfel de câmp) va crea un astfel de câmp (cu numele implicit *ID*).

4. Salvăm tabelul selectând comanda *Save* din meniul *File*. Apare fereastra *Save As* în care scriem numele tabelului.

Exercițiu: Observați denumirile și tipurile câmpurilor tabelului din *figura 8.1*. Creați similar tabelul *Elevi* al bazei de date *Liceu*, descrisă în capitolul precedent.

8.1.2. Proprietățile câmpurilor tabelului

Proprietățile câmpului sînt caracteristici care stabilesc un control suplimentar asupra modului în care sînt memorate, introduse sau afișate datele lui. Aceste proprietăți depind de tipul câmpului și se precizează în zona inferioară a ferestrei tabelului (*fig. 8.1*).

1. Proprietatea *Field Size* determină formatul mărimii datelor câmpului și există doar pentru tipurile *Text* și *Number*.

- Pentru tipul *Text* se acceptă valori de la 0 la 255, stabilind astfel limita de lungime a șirului de caractere ce va fi memorat în câmp. Valoarea 50 este implicită.
- Pentru tipul *Number* se poate alege una din valorile *Byte*, *Integer*, *Long integer* (valoare implicită), *Single*, *Double*, *Replication ID*, *Decimal*.

2. Proprietatea *Format* particularizează printr-un șablon modul în care vor fi afișate datele câmpului și există pentru toate tipurile cu excepția tipului *OLE Object*.

- În cazul tipurilor *Text* și *Memo*, șablonul poate fi creat cu ajutorul următoarelor simboluri:

Simbol	Descriere
@	Caracter de text sau spațiu.
&	Caracterul de text nu este obligatoriu.
<	Toate caracterele vor fi minuscule.
>	Toate caracterele vor fi majuscule.
-	Afișează caracterul -.

Exemplu: Șablonul @-@@-@@> va afișa în loc de textul *abcdef* textul *AB-CD-EF*.

- Pentru tipurile *Number* și *Currency* se poate selecta una din valorile *General number*, *Currency*, *Euro*, *Fixed*, *Standard*, *Percent* sau *Scientific*.
- Formatul tipului *Date/Time* poate fi *General Date*, *Long Date*, *Medium Date*, *Short Date*, *Long Time*, *Medium Time*, *Short Time*. Precizăm că dacă anul este scris cu două cifre, atunci pentru valori din intervalul [00, 29] Access subînțelege anii 2000–2029, iar pentru valori din intervalul [30, 99] – anii 1930–1999.

Exemplu: Formatul *Long Date* va afișa data 28.11.89 astfel: 28 noiembrie 1989.


- Tipul *Yes/No* acceptă una din formatele *Yes/No*, *On/Off*, *True/False*. În ultimul caz utilizatorii vor putea scrie în câmp și valorile 1, respectiv, 0.

3. Proprietatea **Input Mask** se folosește la elaborarea unui șablon de restricționare a caracterelor (se mai spune *mască de intrare*) ce urmează a fi introduse în câmp. Pentru crearea șablonului sînt admise următoarele simboluri:

Simbol	Descriere
0	Cifră (nu poate fi precedată de + ori -). Introducere obligatorie.
9	Cifră (poate fi precedată de + ori -) sau spațiu. Introducere opțională.
#	Cifră (poate fi precedată de + ori -) sau spațiu (la salvare este eliminat). Introducere opțională.
L	Literă. Introducere obligatorie.
?	Literă. Introducere opțională.
A	Literă sau cifră. Introducere obligatorie.
a	Literă sau cifră. Introducere opțională.
&	Orice caracter sau spațiu. Introducere obligatorie.
C	Orice caracter sau spațiu. Introducere opțională.
, ; : - /	Separatori pentru date calendaristice sau pentru clasele numărului (unități, mii, milioane, miliarde etc.). Separatorul predefinit depinde de setările din fereastra <i>Regional Settings</i> (poate fi afișată din Panoul de control <i>Windows</i>).
<	Transformă simbolurile-litere din dreapta în minuscule.
>	Transformă simbolurile-litere din dreapta în majuscule.
!	Forțează introducerea datelor de la dreapta spre stînga.
\	Afișează doar caracterul care urmează după \ (de exemplu, șablonul \M afișează litera M).
"Șir de caractere"	Afișează doar șirul de caractere (fără ghilimele).
Password	În loc de simbolurile introduse, se vor afișa caractere *.

Exemple:

1. Șablonul >L<L0\S acceptă șiruri de caractere din 4 simboluri, dintre care primul este o literă majusculă, al doilea – o literă minusculă, al treilea – o cifră, ultimul caracter fiind litera S.
2. Pentru scrierea numerelor de telefon doar de forma (+373 22) XX-XX-XX se va folosi șablonul "(+373 22)" 00\00\00.

Pentru crearea șabloanelor se poate folosi programul *Input Mask Wizard* (se lansează prin executarea unui clic pe butonul  din caseta proprietății *Input Mask*), care oferă 10 formate uzuale ale măștilor de intrare.

4. Proprietatea **Caption** specifică textul care va fi afișat în calitate de denumire a câmpului în interogări, formulare sau rapoarte. Dacă această proprietate nu se completează, atunci Access va folosi în calitate de nume identificatorului câmpului.

5. Proprietatea **Default Value** stabilește valoarea implicită a câmpului. Apare automat la inserarea unei noi înregistrări.

6. Cu ajutorul proprietății **Validation Rule** se pot forma condiții de validare a datelor care urmează a fi introduse în câmp. Condițiile de validare reprezintă expresii care se scriu utilizînd operatori și funcții Access sau VBA (*Visual Basic for Applications*). De exemplu, condiția >= 100 permite utilizatorului să introducă într-un câmp numeric doar numere mai mari sau egale cu 100.

7. În proprietatea **Validation Text** se scrie textul care va apărea într-o fereastră de avertizare, dacă valoarea introdusă în câmp nu respectă condițiile de validare din *Validation Rule*.

8. Proprietatea **Required** acceptă doar valorile *Yes* și *No*. Valoarea *Yes* obligă utilizatorul să completeze câmpul. Este inutil de completat această proprietate pentru cheile primare (deoarece un astfel de câmp nu acceptă valori vide) sau dacă condiția de validare este *Is Not Null* (nu este nulă).

9. În proprietatea **Allow Zero Length** de asemenea se poate scrie doar una din valorile *Yes* sau *No*. Ea există doar pentru câmpurile de tip *Text* și *Memo*. Pentru *Yes* câmpul va accepta valori de lungime 0, adică șiruri vide de caractere, chiar dacă proprietatea *Required* va fi *Yes*.

10. Proprietatea **Indexed** permite (pentru indexul *Yes (Duplicates Ok)*) sau interzice (pentru indexul *Yes (No Duplicates)*) repetarea valorilor în câmp. Indexul existent poate fi eliminat dacă din lista derulantă a proprietății se alege valoarea *No*. Pentru o cheie primară indexul *Yes (No Duplicates)* va apărea automat (fig. 8.1) și nu va putea fi modificat.

11. Proprietatea **New Values** se aplică doar câmpurilor de tip *AutoNumber*. Pentru valoarea *Increment* sistemul Access va genera valori noi în câmp, adăugând 1 la cea mai mare valoare existentă. Dacă atribuim proprietății *New Values* valoarea *Random*, atunci câmpul va fi completat cu valori generate aleator.

Întrebări și exerciții

- 1 Examinați tabelul *Profesori* al bazei de date *Liceu* și descrieți structura lui.
- 2 Care sînt tipurile de date ce pot fi păstrate într-o bază de date Access?
- 3 Care trebuie să fie tipul unui câmp pentru a putea stoca în el fotografii? Dar adrese de pagini Web? Biografia unei personalități?
- 4 Ce trebuie să facem pentru ca fiecare înregistrare nouă să nu fie acceptată în tabel, dacă utilizatorul nu a completat ultimele două câmpuri ale lui?
- 5 Un câmp care nu este cheie primară nu acceptă repetări de valori în diferite înregistrări. Care este cauza?
- 6 Cum putem verifica dacă în baza de date *Liceu* există informații despre elevi cu aceeași zi de naștere?
- 7 Cărui an îi aparține data:
a) 01.01.01; b) 30.12.30; c) 17.12.89; d) 15.04.28?
- 8 Creați un tabel Access care va conține informații despre o colecție de muzică. Includeți cel puțin câmpurile pentru memorarea numelui interpretului, a firmei producătoare de înregistrări, a anului apariției, a formatului (disc, casetă, CD etc.) și a valorii de evaluare atribuite piesei muzicale, potrivit preferințelor (de exemplu, de la 5 la 10).
- 9 Creați un tabel Access care va conține date despre o colecție de rețete culinare. Includeți cel puțin câmpurile în care se vor stoca denumirile bucatelor, tipurile lor (felurile întii, garnituri, fripturi, copturi, deserturi etc.), timpul de pregătire, originea bucatelor (moldovenești, franțuzești, japoneze etc.).
- 10 Alcătuiți un șablon care va obliga introducerea într-un câmp numeric doar a numerelor întregi:
a) din intervalul [10... 99]; b) de trei cifre; c) negative de două cifre.
- 11 Creați un șablon care va permite introducerea într-un câmp a numerelor de identificare a buletinelor moldovenești. Orice buletin moldovenesc are un număr de identificare format dintr-o literă majusculă, urmată de 8 cifre.


8.2. Stabilirea relațiilor dintre tabele


În capitolul 7 am aflat că între două tabele ale unei baze de date relaționale poate exista una din următoarele tipuri de relație: 1 : 1, 1 : M, M : M.

Sistemul Access folosește pentru afișarea și crearea relațiilor dintre tabele fereastra *Relationships*.

Vom examina crearea relațiilor dintre două tabele printr-un exemplu.

Să stabilim relația 1 : M dintre tabelele *Clase* și *Elevi* ale bazei de date *Liceu*.

1. Executăm un clic pe butonul  de pe bara de instrumente sau alegem *Tools* → *Relationships*. Apare fereastra *Relationships*.

2. Executăm un clic pe butonul  de pe bara de instrumente sau selectăm *Relationships* → *Show Table*. Apare fereastra *Show Table* (fig. 8.2), din care selectăm pe rând tabelele *Elevi* și *Clase*, confirmând de fiecare dată alegerea prin apăsarea butonului *Add*. În fereastra *Relationships* apar identificatorii câmpurilor tabelului selectat (fig. 8.3).

3. Cheia primară a fiecărui tabel este mai pronunțată față de celelalte câmpuri. Selectăm câmpul *Cod_clasa* al tabelului *Elevi*, apoi, ținând apăsat butonul mouse-ului, tragem cursorul spre câmpul *Cod_clasa* al tabelului *Clase*. Apare fereastra *Edit Relationships*, în care automat a fost determinat tipul relației 1 : M (*Relationship Type: One-To-Many*). De asemenea, putem activa următoarele caracteristici ale relației (fig. 8.4):

- Asigurarea integrității referențiale (*Enforce Referential Integrity*);
- Actualizarea în cascadă a înregistrărilor (*Cascade Update Related Fields*);
- Excluderea în cascadă a înregistrărilor (*Cascade Delete Related Records*).

Dacă este activată caracteristica **Asigurarea integrității referențiale**, atunci câmpul secundar al tabelului subordonat va accepta doar valori ale câmpului cheie primară al

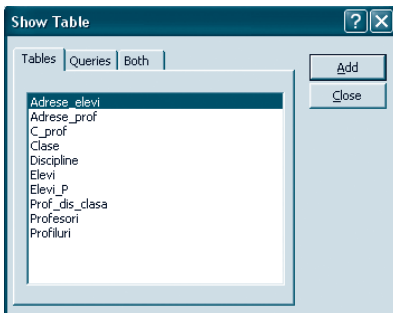


Fig. 8.2

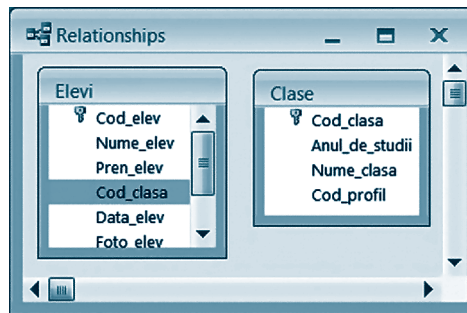


Fig. 8.3

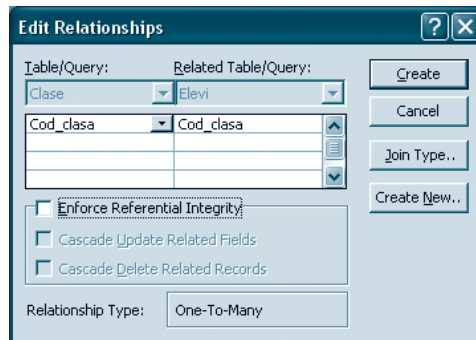


Fig. 8.4

tabelului principal. Astfel, în câmpul *Cod_clasa* al tabelului *Elevi* vom putea scrie doar „codurile” claselor înregistrate în tabelul *Clase*.

Actualizarea în cascadă a înregistrărilor înseamnă că la modificarea unei valori *V* din câmpul cheie primară al tabelului principal, automat se vor modifica corespunzător toate valorile *V* din câmpul cheie străină al tabelului subordonat. De exemplu, dacă vom schimba în tabelul *Clase* „codul” clasei a 10-a A din c01 în c001, atunci fiecare valoare c01 din câmpul *Cod_clasa* al tabelului *Elevi* va fi substituită cu valoarea c001.


Dacă este activată caracteristica **Excluderea în cascadă a înregistrărilor**, atunci la eliminarea unei înregistrări *X* din tabelul principal se vor elimina toate înregistrările din tabelul subordonat, care conțin în câmpul cheie străină valoarea din câmpul cheie primară a înregistrării *X*. De exemplu, dacă vom exclude din tabelul *Clase* ultima înregistrare (are „codul” c12 și corespunde clasei a 12-a D), atunci din tabelul *Elevi* vor fi „eliminați” toți elevii clasei a 12-a D.


Întrebări și exerciții

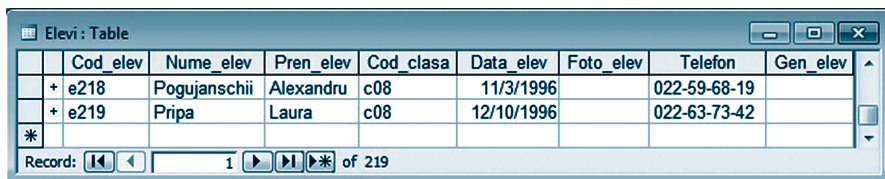
- ❶ Explicați algoritmul de stabilire a relației dintre două tabele Access.
- ❷ Ce înseamnă *integritatea referențială a datelor*?
- ❸ Disciplinele școlare aparțin următoarelor arii curriculare:
 - a) limbă și comunicare;
 - b) educație socioumanistică (istorie, geografie, educație civică);
 - c) matematică și științe (matematică, fizică, chimie, biologie);
 - d) tehnologii (informatică);
 - e) sport (educație fizică).
 Deschideți baza de date *Liceu*. Creați și completați în ea tabelul *Arii curriculare*, apoi stabiliți o relație dintre acest tabel și tabelul *Discipline*.
- ❹ Care este rolul caracteristicii *Cascade Update Related Fields* a unei relații?
- ❺ Cu ce scop se utilizează caracteristica *Cascade Delete Related Records* a unei relații?

8.3. Modificarea tabelelor

8.3.1. Introducerea și editarea datelor

Pentru a introduce sau a edita date într-un tabel, deschidem acest tabel în regimul *Datasheet View* prin executarea unui clic pe butonul  al ferestrei bazei de date.

Comutarea între regimurile *Datasheet View* (regim de editare, fig. 8.5) și *Design View* (regim de proiectare) se realizează prin intermediul butonului *View*  de pe bara de instrumente Access.








	Cod_elev	Nume_elev	Pren_elev	Cod_clasa	Data_elev	Foto_elev	Telefon	Gen_elev
+	e218	Pogujanschii	Alexandru	c08	11/3/1996		022-59-68-19	
+	e219	Pripa	Laura	c08	12/10/1996		022-63-73-42	
*								

Record: 1 of 219

Fig. 8.5

La crearea sau modificarea conținutului unei înregistrări, la stînga ei apare un **selector de înregistrare (SI)**, al cărui aspect depinde de starea înregistrării (vezi *tabelul* care urmează).

SI	Starea înregistrării
	Înregistrarea curentă este selectată.
	Înregistrare nouă, în care se pot introduce date.
	Utilizatorul editează înregistrarea, iar modificările nu sînt încă salvate.
	Înregistrarea este blocată de alt utilizator și nu se poate edita (cazul mediului multi-utilizator – mediu în care mai multe persoane pot utiliza simultan baza de date).

Pentru gestionarea rapidă a înregistrărilor se pot utiliza instrumentele din partea de jos a ferestrei tabelului: , avînd respectiv următoarele funcții (de la stînga spre dreapta):

- activarea primei înregistrări;
- activarea înregistrării predecesoare celei curente;
- afișarea numărului de ordine al înregistrării curente sau activarea înregistrării cu numărul de ordine din casetă;
- activarea înregistrării succesoare celei curente;
- activarea ultimei înregistrări;
- adăugarea unei înregistrări noi.

Introducerea și editarea datelor unui tabel se fac prin metode caracteristice lucrului cu texte. De exemplu, se pot copia date utilizînd memoria *Clipboard*, iar ștergerea lor se poate realiza folosind tastele *Backspace* și *Delete*.

O singură apăsare pe tasta *Esc* anulează acțiunile de modificare a informației din cîmpul curent, iar o dublă apăsare pe ea – acțiunile de modificare a înregistrării curente.

Nu este obligatoriu de completat toate cîmpurile unei înregistrări, cu excepția cîmpului cheie primară (care nu poate conține valori vide) și a celor care au setată cu *Yes* proprietatea *Required*.

Un cîmp poate fi selectat prin executarea unui clic pe *antetul* lui (celula care afișează identificatorul cîmpului), iar o înregistrare – prin executarea unui clic pe *antetul* ei (celula în care apare selectorul de înregistrare). Dacă după selectarea unui cîmp (sau a unei înregistrări) nu eliberăm butonul mouse-ului și-l poziționăm pe următorul cîmp (respectiv pe următoarea înregistrare), vor deveni selectate ambele cîmpuri (respectiv ambele înregistrări).

Menționăm că **asupra datelor sau înregistrărilor se pot efectua următoarele operații:**

- *adăugarea unei înregistrări* noi înaintea celei curente (comanda *New Record* din meniul flotant al antetului înregistrării sau din meniul *Insert* al meniului principal *Access*);
- *ștergerea înregistrării curente* (comanda *Delete Record* din meniul flotant al antetului înregistrării sau din meniul *Edit* al meniului principal *Access*);
- *completarea cîmpurilor*;
- *copierea conținutului unei celule* (comenzile *Copy* și *Paste* din meniul flotant al celulei);
- *copierea unei înregistrări* (comenzile *Copy* și *Paste* din meniul flotant al antetului înregistrării sau din meniul *Edit* al meniului principal *Access*).

Observații:

1. În urma ultimei operații, utilizatorul va fi obligat să modifice valoarea celulei din cîmpul cheie primară.
2. Unele dintre operațiile menționate se pot efectua și cu ajutorul instrumentelor de pe bara de instrumente *Access*.

3. Pentru introducerea rapidă a datelor se pot folosi comenzile rapide lansate cu ajutorul combinațiilor de taste.
4. Modificările unei înregistrări sînt automat salvate de Access atunci cînd se trece la o altă înregistrare sau la închiderea tabelului.
5. La completarea tabelului cu înregistrări, mai întîi se vor introduce date în tabelele principale, apoi în cele subordonate.



8.3.2. Modificarea aspectului tabelului

Sistemul Access afișează în mod implicit datele tabelului respectînd anumite caracteristici. De exemplu, înregistrările tabelului apar ordonate crescător automat după valorile cîmpului cheie primară, iar cîmpurile lui se succed în ordinea în care au fost create.

Utilizatorul poate schimba modul de prezentare a informației dintr-un tabel. Mai exact, sînt permise următoarele **acțiuni de modificare a aspectului tabelului**:

- a) schimbarea ordinii de afișare a cîmpurilor;
- b) modificarea ordinii de afișare a înregistrărilor;
- c) schimbarea înălțimii înregistrărilor sau lățimii cîmpurilor;
- d) ascunderea coloanelor;
- e) filtrarea înregistrărilor.

Schimbarea ordinii de afișare a cîmpurilor se face prin deplasarea lor. Pentru a deplasa unul sau cîteva cîmpuri consecutive, selectăm aceste cîmpuri, apoi, ținînd apăsat butonul stîng al mouse-ului, poziționăm indicatorul lui pe cîmpul în fața căruia dorim să mutăm cîmpurile selectate.

Pentru ca înregistrările unui tabel să apară ordonate crescător (respectiv descrescător) după valorile unui cîmp, selectăm acest cîmp, apoi executăm un clic pe butonul  de pe bara de instrumente (respectiv pe butonul ) sau alegem *Records* → *Sort* → *Sort Ascending* (respectiv *Records* → *Sort* → *Sort Descending*). Similar se face ordonarea după valorile cîtorva cîmpuri consecutive. Menționăm că, în acest caz, ordonarea se va face de la stînga spre dreapta, adică valorile din cîmpul din dreapta se vor lua în considerare doar în cazul în care vor coincide valorile cîmpului din stînga.

Acțiunile de modificare a înălțimii înregistrărilor sau a lățimii cîmpurilor sînt similare cu acțiunile de schimbare a înălțimii rîndurilor sau a lățimii coloanelor unui tabel într-un redactor de texte ori într-un procesor tabelar.

Pentru a ascunde o coloană, o selectăm, apoi alegem comanda *Hide Columns* din meniul ei flotant sau din meniul *Format*. Pentru a reafișa coloanele ascunse, selectăm comanda *Unhide Columns* din meniul *Format*. Apare fereastra *Unhide Columns* în care alegem coloanele necesare.

Filtrarea înregistrărilor, adică selectarea acelor înregistrări care respectă anumite condiții, se poate face prin crearea unui **filtru** (se alege *Records* → *Filter* → *Advanced Filter/Sort*) și aplicarea lui (se execută *Filter* → *Apply Filter/Sort*).

Un filtru poate fi înlăturat executînd *Records* → *Remove Filter/Sort*.


Observații:

1. Modificările aspectului tabelului nu sînt automat salvate de Access la închiderea tabelului. Utilizatorul le poate salva executînd un clic pe butonul *Save* de pe bara de instrumente sau lansînd comanda *Save* din meniul *Edit*.
2. Modificările aspectului tabelului nu afectează structura tabelului.

8.3.3. Modificarea structurii tabelului

Este firesc ca pe parcursul proiectării (sau chiar al gestionării) unei baze de date să apară necesitatea schimbării structurii unor tabele.

Atenție! Aceste schimbări pot afecta integritatea informațiilor din baza de date. De exemplu, micșorarea dimensiunii unui câmp de tip text poate atrage după sine trunchierea valorilor acestui câmp, iar eliminarea unui câmp cheie primară poate duce la ștergerea înregistrărilor tabelului subordonat. De fapt, în funcție de caracteristicile relațiilor dintre tabele, sistemul Access poate să nu accepte unele schimbări ale structurii tabelului.

Astfel, procesul de modificare a câmpurilor cheie primară sau a câmpurilor cheie străină se va face doar după ce vor fi distruse relațiile dintre tabele, urmînd a fi restabilite ulterior. Pentru a modifica structura unui tabel, acesta trebuie deschis în regimul *Design View* prin executarea unui clic pe butonul  al ferestrei bazei de date.

În acest regim sînt posibile următoarele acțiuni:

- a) adăugarea unui câmp nou;
- b) eliminarea unui câmp;
- c) schimbarea identicatorului câmpului;
- d) modificarea proprietăților câmpurilor;
- e) stabilirea unei chei primare;
- f) eliminarea unei chei primare.

Acțiunile de modificare a structurii unui tabel sînt similare cu acțiunile de creare a tabelului.

8.3.4. Caracteristica *Lookup* a câmpurilor

Caracteristica *Lookup* permite înlocuirea casetelor de text ale câmpurilor cu liste derulante. Astfel, utilizatorul va putea completa un câmp selectînd o valoare dintr-o listă de valori acceptabile. Conținutul listei poate fi încărcat dintr-un câmp cheie primară al unui tabel asociat celui curent sau poate fi creat la stabilirea caracteristicii *Lookup*.

Să alcătuiam o listă derulantă cu ajutorul unui program de asistență pentru câmpul *Cod_clasa* al tabelului *Elevi*.

1. Deschidem tabelul *Elevi* în regim de proiectare. Alegem opțiunea *Lookup Wizard* din lista derulantă a câmpului *Cod_clasa*.

2. Apare prima casetă de dialog *Lookup Wizard*. Selectăm prima opțiune, stabilind astfel că valorile listei care urmează a fi creată vor fi luate dintr-un tabel asociat. Opțiunea a doua precizează că valorile listei vor fi create la ceilalți pași.

3. În următoarele două ferestre alegem tabelul *Clase*, apoi câmpurile *Cod_clasa*, *An_de_studii* și *Nume_clasa*. Valorile câmpurilor alese vor apărea în listă.

4. În fereastra a patra ajustăm lățimea coloanelor listei și atribuim un nume acestei liste.

Întrebări și exerciții

- ❶ Ce operații pot fi efectuate asupra datelor unui tabel?
- ❷ Care modificări ale tabelului pot afecta structura lui?
- ❸ Deschideți baza de date *Liceu*. Adăugați tabelului *Adrese_elevi* câmpul *Adresa_Web*. Completați acest câmp pentru primele 10 înregistrări.

- ④ Deschideți tabelul *Elevi* în regim de editare:
 - a) adăugați 5 înregistrări în tabel;
 - b) ștergeți penultima înregistrare;
 - c) copiați datele primei înregistrări în câmpurile ultimei înregistrări.
- ⑤ Deschideți tabelul *Profesori* în regim de editare:
 - a) afișați datele tabelului în ordinea: numele, prenumele, sexul, telefonul, data nașterii a profesorilor;
 - b) ascundeți câmpurile *Salariu*, *CV_prof* și *Foto_prof*;
 - c) reafișați câmpurile ascunse.
- ⑥ Afișați în ordine alfabetică lista elevilor bazei de date *Liceu*.
- ⑦ Afișați numele și prenumele elevilor în ordinea descrescătoare a vârstei lor.
- ⑧ Afișați lista profesorilor bazei de date *Liceu* în ordine crescătoare a salariilor lor.

8.4. Crearea expresiilor Access

Examinînd proprietatea *Validation Rule* a câmpurilor, am menționat că regulile de validare a datelor care urmează a fi introduse în câmp se scriu folosind expresii Access. După cum vom vedea ulterior, expresiile se utilizează de asemenea la formularea cererilor de căutare a datelor și la elaborarea rapoartelor.

O expresie Access este o declarație de intenție, care conține cel puțin un operator și un operand: constantă, identificator sau funcție.

Expresia returnează o valoare. Un **identificator** Access este numele unui obiect al bazei de date. Tabelul, câmpul, interogarea, formularul, raportul și însăși baza de date sînt obiecte. De regulă, în cadrul expresiilor identificatorii se scriu între simbolurile [și] .

Identificatorul unui „subobiect” este format din numele clasei de obiecte și numele „subobiectului” delimitate printr-un punct sau printr-un semn de exclamare. În așa mod, fiecărui obiect îi corespunde un singur identificator în interiorul bazei de date.

Totuși, uneori, dacă nu sînt confuzii, în calitate de identificator al „subobiectului” se poate folosi doar numele lui.

Exemple: [Elevi].[Nume_elev], [Profesori].[Salariu], [Cod_elev].

8.4.1. Operatori Access

Vom examina 6 categorii de operatori Access.

- **Operatorii aritmetici** (+, -, *, /, \, Mod, ^) se aplică asupra valorilor numerice.

Exemple: $15 \setminus 6$ returnează 2, iar $15 \text{ Mod } 6$ returnează 3, deoarece $15 = 6 \cdot 2 + 3$.

- **Operatorii de comparare** compară valorile a doi operanzi și returnează una din valorile logice *True* sau *False*. În Access se utilizează aceiași operatori de comparare și cu aceeași semnificație ca și a operatorilor relaționali din limbajul de programare Pascal: <, <=, =, >=, >, <>.

- **Operatorii logici** Access se aplică asupra operanzilor logici și de asemenea coincid cu cei din Pascal: **And**, **Or**, **Not**, **Xor**.

- **Operatorul de atribuire** = atribuie valoare unui obiect, unei variabile sau unei constante.

- **Operatorul de concatenare** + unește două șiruri de caractere în unul.

- **Alți operatori.** Următorii operatori nu fac parte din categoriile precedente. Expresiile care îi conțin returnează una din valorile *True* (sau -1), *False* (sau 0).

Operator	Descriere
Is	Se aplică asupra valorii Null (valoarea vidă) și verifică dacă o valoare este sau nu este vidă.
Like	Stabilește dacă un șir de caractere respectă șablonul specificat de Like. Șablonul se scrie între simbolurile "și". Șablonul poate să conțină caractere de înlocuire (? pentru un caracter, # pentru o cifră, * pentru orice număr de caractere, inclusiv lipsa lor) și liste de valori. Lista de valori se scrie între simbolurile [și]. Dacă valorile listei sînt precedate de simbolul !, atunci se consideră toate valorile cu excepția celor precedate de !.
In	Stabilește dacă o valoare aparține unei liste de valori. Valorile listei se delimitează prin simbolul ;.
Between	Stabilește dacă o valoare numerică aparține unui interval.

Exemple:

1. Expresia `[Elevi].[Telefon] Is Null` returnează valoarea *True* doar în cazul în care câmpul *Telefon* al tabelului *Elevi* nu conține nicio valoare (se are în vedere înregistrarea curentă).
2. `Like "B*ov"` specifică șiruri de caractere care încep cu litera *B* și se termină cu combinația de litere *ov*. Prin urmare, expresia `"Belousov" Like "B*ov"` returnează valoarea *True*.
3. `Like "[CK]#?"` specifică șiruri din 3 caractere: primul este una din literele *C* sau *K*, al doilea – o cifră, al treilea – orice simbol.
4. `Like "*[5ad-g]"` specifică șiruri de caractere care se termină cu cifra 5 sau cu una din literele *a*, *d*, *e*, *f*, *g*.
5. `Like "*[!ae]"` specifică șiruri de caractere care nu se termină cu litera *a* sau cu litera *e*.
6. Expresia `"Duminica" In ("Luni"; "Marti"; "Miercuri"; "Joi"; "Vineri")` returnează valoarea *False*, iar expresia `4 In (2; 4; 7; 8)` – valoarea *True*.
7. Expresia `Between 2 And 10` este echivalentă cu expresia `>=2 And <=10`.

Observație: La scrierea expresiei condiției de validare în caseta proprietății *Validation Rule* nu se scrie identificatorul primului operand, acesta fiind considerat implicit identificatorul câmpului respectiv. Astfel, regula de validare `In ("Luni"; "Marti"; "Miercuri"; "Joi"; "Vineri")` va permite utilizatorului să scrie în câmp doar unul din cuvintele *Luni*, *Marti*, *Miercuri*, *Joi*, *Vineri*.

8.4.2. Funcții Access

O funcție returnează o valoare prin numele ei. Access oferă peste 100 de funcții standard pentru prelucrarea diferitor tipuri de date: numerice, șiruri de caractere, calendaristice etc. Cele mai uzuale sînt prezentate în următoarele tabele:

Unele funcții pentru prelucrarea datelor calendaristice

Funcția	Rezultatul returnat
Date()	Data curentă
DateAdd(<i>T</i> ; <i>N</i> ; <i>D</i>)	Data calendaristică care se obține adunînd la data <i>D</i> sau scăzînd din ea <i>N</i> (în cazul cînd <i>N</i> este negativ) unități calendaristice de tip <i>T</i> , unde <i>T</i> poate fi "yyyy", "q", "m", "ww", "d", "h", semnificînd respectiv ani, trimestre, luni, săptămîni, zile, ore
DateDiff(<i>T</i> ; <i>D</i> ₁ ; <i>D</i> ₂)	Diferența exprimată în unități calendaristice de tip <i>T</i> dintre datele <i>D</i> ₁ și <i>D</i> ₂
Time()	Ora curentă

Now()	Data și ora curentă
Year(<i>D</i>)	Anul scris cu 4 cifre corespunzător datei calendarisice <i>D</i>
Month(<i>D</i>)	Numărul de ordine în an al lunii corespunzătoare datei calendarisice <i>D</i>
Day(<i>D</i>)	Numărul de ordine în lună al zilei corespunzătoare datei calendarisice <i>D</i>
WeekDay(<i>D</i>)	Numărul de ordine în săptămână al datei calendarisice <i>D</i> (1 corespunde Duminicii, 2 – zilei de Luni etc.)
Hour(<i>D</i>)	Ora (număr întreg de la 0 la 23) corespunzătoare valorii calendarisice <i>D</i>

Unele funcții de manipulare a textului

Funcția	Rezultatul returnat
Asc(<i>C</i>)	Codul ANSI al caracterului <i>C</i>
Chr(<i>N</i>)	Caracterul al cărui cod ANSI este numărul <i>N</i>
InStr(<i>S</i> ₁ ; <i>S</i> ₂)	Poziția, începînd cu care șirul <i>S</i> ₂ se conține în șirul <i>S</i> ₁
Mid(<i>S</i> ; <i>N</i> ₁ ; <i>N</i> ₂)	Subșirul șirului <i>S</i> de lungime <i>N</i> ₂ începînd cu poziția <i>N</i> ₁ . Parametrul <i>N</i> ₂ poate să lipsească, ceea ce înseamnă că se va returna subșirul obținut prin înlăturarea primelor <i>N</i> ₁ -1 caractere ale șirului <i>S</i>
LCase(<i>S</i>)	Șirul de caractere obținut din șirul <i>S</i> prin transformarea literelor majuscule în litere mici
UCase(<i>S</i>)	Șirul de caractere obținut din șirul <i>S</i> prin transformarea literelor mici în majuscule
Len(<i>S</i>)	Numărul de caractere ale șirului <i>S</i>
LTrim(<i>S</i>)	Șirul obținut din șirul <i>S</i> după lichidarea spațiilor de debut
RTrim(<i>S</i>)	Șirul obținut din șirul <i>S</i> după lichidarea spațiilor de sfîrșit
Trim(<i>S</i>)	Șirul obținut din șirul <i>S</i> după lichidarea spațiilor de debut și a celor de sfîrșit
Left(<i>S</i> ; <i>N</i>)	Șirul format din primele <i>N</i> caractere ale șirului <i>S</i>
Right(<i>S</i> ; <i>N</i>)	Șirul format din ultimele <i>N</i> caractere ale șirului <i>S</i>
Str(<i>X</i>)	Șirul format din simbolurile valorii <i>X</i> în aceeași ordine
Val(<i>S</i>)	Numărul obținut din simbolurile șirului <i>S</i> în aceeași ordine (dacă acesta are formatul potrivit)

Unele funcții matematice

Funcția	Rezultatul returnat
Abs(<i>X</i>)	Valoarea absolută a numărului <i>X</i>
Atn(<i>X</i>)	Arctangenta (în radiani) a numărului <i>X</i>
Avg(<i>C</i>)	Media aritmetică a valorilor cîmpului <i>C</i>
Count(<i>C</i>)	Numărul valorilor nevide ale cîmpului <i>C</i>
Max(<i>C</i>)	Valoarea maximală din cîmpul <i>C</i>
Cos(<i>X</i>)	Cosinusul numărului <i>X</i>
Exp(<i>X</i>)	Valoarea e^x
Int(<i>X</i>)	Partea întregă a numărului <i>X</i>
Log(<i>X</i>)	Logaritmul zecimal al numărului <i>X</i>

Rnd()	Un număr aleator cuprins între 0 și 1
Sgn(X)	0 dacă numărul X este pozitiv, -1 dacă numărul X este negativ
Sin(X)	Sinusul numărului X
Sqr(X)	Rădăcina pătrată a numărului X
Tan(X)	Tangenta numărului X

Observații:

- În calitate de parametri ai funcțiilor pot fi identificatorii câmpurilor (evident, scriși între simbolurile [și]).
- Dacă parametrul funcției este o constantă calendaristică, atunci ea se scrie între simbolurile " și ".

Exemple:

- `DateAdd("d",-50; Date())` returnează data calendaristică care a fost cu 50 de zile în urmă.
- `Weekday("27.09.1993")` returnează 2, deoarece pe 28 septembrie 1993 a fost luni. (Luni se consideră a doua zi din săptămână.)
- `InStr("Informatica"; "forma")` returnează 3.
- `LCase("INFORMATICA")` returnează textul "informatica".
- `LTrim(" forma")` returnează textul "forma".
- `Sgn(- 20)` returnează valoarea -1.

Întrebări și exerciții

- Pentru ce se utilizează expresiile în Access?
- Deschideți baza de date *Liceu* și scrieți o regulă de validare pentru câmpul *Nr_ore_saptamina* al tabelului *Prof_dis_clasa* care să admită doar valori întregi strict mai mari ca 0.
- Ce valoare va returna expresia:
 - `Mid("Propoziție"; 3);`
 - `Mid("Calculator"; 1; 3);`
 - `Int(Rnd()*50);`
 - `Month("15.11.1990");`
 - `Left("Tractor"; 5);`
 - `5 In ("4"; "5"; "6"; "7"; "8");`
 - `Val("25") - 25;`
 - `"R" Like "[TR]*";`
 - `"R" + Str(Date());`
- Scrieți o expresie care va returna:
 - salariul mediu al profesorilor din tabelul *Profesori*;
 - data calendaristică ce va fi peste 5 săptămîni după data curentă;
 - a cîta zi în an este data curentă;
 - a cîta zi din săptămîna a fost 1 ianuarie 2000;
 - diferența de zile dintre 5 martie 2005 și 5 decembrie 2005.
- Scrieți pentru câmpul *Gen_prof* al tabelului *Profesori* o condiție de validare care ar admite în câmp doar valorile *M* sau *F*.

Test de evaluare

1. Stabiliți valoarea de adevăr a propoziției: „Valorile tabelului fac parte din structura lui”.
2. Completați adecvat:
 - a) Proprietatea stabilește valoarea implicită a câmpului unui tabel.
 - b) Dacă valoarea proprietății a unui câmp este, atunci Access nu va accepta o înregistrare nouă pînă cînd utilizatorul nu va completa acest câmp al înregistrării.
3. Descrieți acțiunea de restricționare a șablonului >L0L 0L09.
4. Creați un șablon ce va obliga introducerea într-un câmp de text a șirurilor formate:
 - a) din 3 caractere, dintre care primul va fi o literă, iar celelalte două – cifre;
 - b) din cel puțin 4 caractere și cel mult 6 caractere, dintre care ultimul va fi cifră.
5. Care dintre caracteristicile:
Asigurarea integrității referențiale (Enforce Referential Integrity),
Actualizarea în cascadă a înregistrărilor (Cascade Update Related Fields),
Excluderea în cascadă a înregistrărilor (Cascade Delete Related Records)
trebuie activată pentru ca modificarea unei valori din câmpul cheie primară al tabelului principal să atragă după sine modificări corespunzătoare în tabelul subordonat?
6. Fie tabelul *Lista* cu următoarea structură:

Nume	Prenume	Data nașterii	Raionul	Profesia	Genul
Popa	Ion	12.08.1979	Briceni	profesor	M
...

Descrieți acțiunile care trebuie realizate pentru a afișa datele următoarelor câmpuri ale tabelului *Lista* în ordinea: *Raionul, Data nașterii, Prenume, Nume*.

7. Completați adecvat:
 - a) Expresia specifică șiruri de caractere care încep cu litera *T* și se termină cu litera *R*.
 - b) Expresia specifică șiruri de caractere care nu încep cu niciuna dintre literele *C, D, E, T*.
 - c) Expresia specifică numere întregi cu modulul mai mare decît 15 și mai mic decît 40.
8. Care va fi valoarea expresiei:
 - a) `DateAdd("ww";1; "01.09.2010");`
 - b) `Year("01.09.30");`
 - c) `"PAR" Like "[A-D]*"`?
9. Scrieți pentru câmpul *Adrese_email* al unui tabel o condiție de validare care ar admite în câmp doar șiruri de caractere ce conțin în interiorul lor caracterul @.

CAPITOLUL 9

INTEROGĂRI

După studierea acestui capitol, veți fi capabili să:

- descrieți tipurile de interogări;
- creați interogări (prin proiectare independentă sau cu ajutorul sistemului de asistență) care vor selecta date din unul sau mai multe tabele;
- executați interogări;
- actualizați datele din tabele prin intermediul interogărilor;
- obțineți cîmpuri de date noi în baza informațiilor existente;
- creați totalizări ale datelor din unul sau mai multe tabele;
- reprezentați compact subseturi de date din unul sau mai multe tabele.

9.1. Noțiuni generale despre interogări

Sistemele de gestiune a bazelor de date au fost concepute pentru a păstra informații și pentru a automatiza procesul de prelucrare a acestor informații. Chiar și pentru o bază de date cu câteva sute de înregistrări, căutarea manuală a datelor care satisfac anumite condiții este anevoioasă. Menționăm că există baze de date care conțin milioane de înregistrări. De exemplu, unii operatori de telefonie mobilă din Republica Moldova au peste 1 000 000 de abonați! La ultimele alegeri parlamentare din țară au participat peste 1,5 milioane de alegători. Deci, în cazul votării electronice, se vor prelucra peste 1,5 milioane de înregistrări ale unei baze de date!

Căutarea unor date poate implica consultarea cîtorva tabele. De exemplu, pentru a afla ce discipline studiază un elev anume din baza de date *Liceu*, trebuie să examinăm tabelele *Elevi*, *Clase*, *Prof_dis_clasa* și *Discipline*. Să ne convingem!

- Consultați baza de date *Liceu* și determinați ce discipline studiază elevul *Dan Moraru*.

Exercițiul propus este doar un mic argument menit să demonstreze necesitatea cercetării și prelucrării automatizate a informațiilor unei baze de date.

Pentru selectarea rapidă din unul sau din mai multe tabele a seturilor de date care corespund unor condiții, dar și pentru actualizarea accelerată a înregistrărilor, sistemele de gestiune a bazelor de date utilizează *interogări* – cereri de căutare și/sau de acțiune în conformitate cu aceste condiții.

Interogările sînt obiecte ale sistemelor de gestiune a bazelor de date care reprezintă adresări de căutare, analizare și/sau de modificare a datelor bazei.

Menționăm că în calitate de surse de date, în afară de înregistrările tabelor bazei, o interogare poate folosi rezultatele altei interogări, executate anterior.

De exemplu, pentru a afișa lista profesorilor cu salariul maximal, se vor crea două interogări: prima interogare va găsi valoarea *Max* a salariului maximal, iar a doua – va selecta din tabelul *Profesori* înregistrările cu valorile din câmpul *Salariu*, egale cu *Max* (adică egale cu rezultatul primei interogări).

Interogările îndeosebi se utilizează pentru:

- vizualizarea unui subset de înregistrări dintr-un tabel, fără a-l deschide;
- afișarea într-un singur tabel a informațiilor din câteva tabele;
- actualizarea datelor tabelelor (modificări, adăugări, excluderi de date);
- efectuarea calculului asupra valorilor câmpurilor și obținerea informațiilor noi;
- crearea totalurilor, valorilor medii etc.

În funcție de tipul acțiunii și de rezultate, interogările se clasifică în cele:

- a) *de selecție*;
- b) *de excludere a unor înregistrări*;
- c) *de modificare a unor înregistrări*;
- d) *de creare a câmpurilor rezultante*;
- e) *de grupare și totalizare*;
- f) *încrucișate*.

Interogările de selecție sînt cereri formulate în baza unor condiții logice. Ele selectează un subset de date din unul sau mai multe tabele legate între ele. De exemplu, căutarea elevilor născuți pînă la 10 ianuarie 1992, afișarea elevilor clasei a 10-a B sînt interogări de selecție.

Interogările de excludere a unor înregistrări reprezintă cereri de eliminare dintr-un tabel a tuturor înregistrărilor care satisfac criteriile logice specificate. De exemplu, cererea de ștergere din tabelul *Elevi* al bazei de date *Liceu* a informațiilor despre elevii claselor a 12-a (în legătură cu absolvirea liceului) reprezintă o astfel de interogare.

Interogările de modificare a unor înregistrări schimbă valorile unui câmp al tabelului după același algoritm. De exemplu, mărirea cu 50% a valorilor din câmpul *Salariu* al tabelului *Profesori* poate fi realizată cu ajutorul unei interogări de modificare.

Uneori se pot solicita informații care trebuie afișate în câmpuri noi. De exemplu, vârsta elevilor se va prezenta într-un câmp aparte. Cu acest scop se va utiliza o **interogare de creare a câmpurilor rezultante**.

Interogările de grupare și totalizare se folosesc pentru sumarea datelor câmpurilor, obținerea valorilor medii, a celor minimale sau maxime etc. De exemplu, calcularea numărului total de ore pe săptămînă realizate în fiecare clasă a bazei de date *Liceu* se va face cu ajutorul unei interogări de grupare și totalizare.

Interogările încrucișate sînt destinate prezentării compacte a informațiilor sub formă de tabel. De exemplu, informația despre numărul săptămînal de ore rezervat fiecărei discipline în fiecare clasă poate fi afișată cu ajutorul unei interogări încrucișate sub forma următorului tabel:

Anul_de_studii	Nume_clasa	Biologia	Chimia	...
10	A	2	3	...
10	D	1	1	...
11	A	3	2	...
12	D	1	1	...
...

Așa cum rezultatele interogărilor depind de informațiile din tabele, modificările realizate asupra datelor din tabele vor atrage după sine modificarea rezultatelor interogărilor (evident după executarea lor repetată).

La rîndul său, în cazul interogărilor a) – d), modificarea de către utilizator a rezultatelor interogării poate duce la schimbarea conținutului tabelului. Din aceste considerente rezultatul interogărilor a) – d) se numește **set dinamic de date**.

Setul dinamic de date nu este constant, deci nu se memorează. El există doar pe parcursul execuției interogării.

Pentru a crea o interogare nouă:

1. Selectăm clasa de obiecte *Queries* de pe bara de obiecte *Objects*. În zona de conținut a ferestrei bazei de date apar opțiunile:

- *Create query in Design view* (crearea interogării în regim de proiectare);
- *Create query by using wizard* (crearea interogării cu ajutorul unui program de asistență).

Executăm un dublu-clic pe prima opțiune.

Observație: *Pasul 1* este echivalent cu selectarea *New* → *Design View*.

2. Apare fereastra interogării și fereastra de dialog *Show Table* (fig. 9.1).

Selectăm pe rînd tabelele necesare apăsînd de fiecare dată butonul *Add*. Tabelele alese (mai exact cîmpurile lor) apar în interiorul ferestrei interogării împreună cu reprezentarea grafică a relațiilor dintre ele (fig. 9.2).

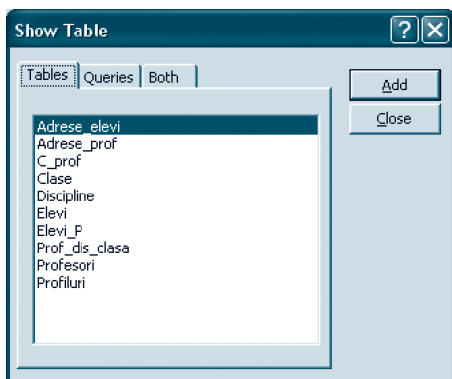


Fig. 9.1

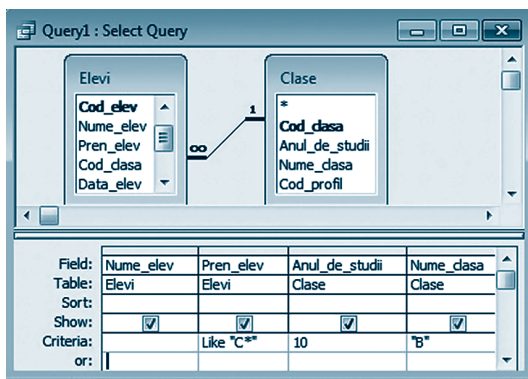


Fig. 9.2


3. Fereastra unei interogări este divizată în două zone. Zona superioară afișează denumirile cîmpurilor tabelului. Zona inferioară afișează un **formular QBE*** în care programatorul poate folosi exemple de declarații parțiale sau expresii pentru a crea o interogare. Alegem cîmpurile care vor fi utilizate pentru scrierea criteriului de selecție și cîmpurile ale căror valori vor fi afișate. Adăugarea unui cîmp se face prin executarea unui dublu-clic pe denumirea lui sau prin selectarea și „tragerea” acestui cîmp cu ajutorul mouse-ului.

* Caracteristica QBE (*Query by Example*) a fost creată inițial pentru a permite utilizatorilor aplicațiilor de bază de date să găsească și să afișeze porțiuni de date fără a fi nevoiți să știe un limbaj de programare. Pe parcursul timpului, aplicațiile QBE au ajuns instrumente preferate pentru scrierea interogărilor. Majoritatea sistemelor de gestiune a bazelor de date și-au dezvoltat propriile aplicații QBE, care permit formularea unui spectru larg de cereri de interogare.

Ordinea câmpurilor afișate în formular poate fi schimbată în același mod ca și ordinea coloanelor unui tabel afișate în regimul *Datasheet View*.

4. Pentru fiecare câmp adăugat în formularul QBE, în afară de câmpul și tabelul care conține acest câmp (afișate în rîndurile *Field* și *Table*), se pot preciza:

- modul de sortare a înregistrărilor după acest câmp (rîndul *Sort*) alegînd una din valorile *Ascending* (crescător) sau *Descending* (descrescător);
- afișarea sau ascunderea valorilor câmpului la executarea interogării (rîndul *Show*);
- o condiție de selectare pe care o vor respecta valorile afișate (rîndurile *Criteria* și *or*).

5. Putem **afișa datele specificate** de interogare pînă la salvarea ei prin selectarea comenzii *Datasheet View* din meniul *View* sau din lista derulantă  de pe bara de instrumente *Access*.

6. **Salvăm interogarea** (instrumentul *Save* sau comanda cu același nume din meniul *File*).

Exemplu: În figura 9.2 este prezentată în regim de proiectare o interogare care cere afișarea listei elevilor clasei a 10-a B, al căror prenume începe cu litera C. În figura 9.3 este arătată lista pînă la salvarea interogării.



	Nume_elev	Pren_elev	Anul_de_studii	Nume_clasa
▶	Bujor	Călin	10	B
	Cozariuc	Cătălina	10	B
	Mihalachi	Cristina	10	B
*				

Fig. 9.3

Întrebări și exerciții

- Cu ce scop se utilizează interogările?
- Caracterizați principalele tipuri de interogări.
- Descrieți algoritmul de creare a unei interogări.
- De ce rezultatul unor interogări se numește *set dinamic de date*?
- Stabiliți tipul următoarelor interogări:
 - aflarea numărului de profesori de gen masculin și a numărului celor de gen feminin;
 - determinarea salariului mediu al profesorilor din baza de date *Liceu*;
 - afișarea listei fetelor din clasele a 11-a;
 - determinarea numărului de ore realizate săptămînal de fiecare profesor;
 - căutarea profesorilor de gen feminin care predau în clasele a 10-a.
- Examinați baza de date *Liceu* și formulați două cereri de interogări:
 - de selecție;
 - de excludere a unor înregistrări;
 - de modificare a unor înregistrări;
 - de creare a câmpurilor rezultante;
 - de grupare și totalizare;
 - încrucișate.

9.2. Interogări de selecție

Pornind de la faptul că interogările de selecție sînt cele mai des utilizabile, *Access* stabilește implicit acest tip pentru toate interogările nou-create. După cum vom vedea ulterior, utilizatorul trebuie să întreprindă acțiuni suplimentare pentru a schimba tipul interogării noi.

Deci, pentru a crea o interogare de selecție, vom alege tabelele și câmpurile necesare conform algoritmului descris în tema precedentă.

9.2.1. Criterii de selecție

Un moment important în procesul de elaborare a unei astfel de interogări este scrierea **criteriului de selecție**.

Dacă criteriul se formulează pentru un singur câmp, atunci expresia logică, care va controla afișarea datelor, se scrie în celula *Criteria* a acestui câmp. Menționăm că operatorul *Like* este inserat automat de Access în cazul în care utilizatorul scrie un șablon de restricționare a datelor. În particular, în celula *Criteria* se poate scrie o constantă de tip compatibil cu tipul valorilor câmpului respectiv.

Exemplu: Pentru a afișa lista elevilor cu prenumele Ion, este suficient să se scrie în celula *Criteria* a câmpului *Pren_elev* (din tabelul *Elevi*) șirul de caractere "Ion".

Deoarece criteriile de selecție sînt expresii logice Access, pentru scrierea lor pot fi utilizate funcțiile și operatorii studiați în capitolul precedent, inclusiv cei *logici*.

În același timp, formularul QBE oferă ajutor în crearea criteriilor compuse, formate din cîteva condiții și din operatorii AND și/sau OR. Astfel:

- pentru un câmp pot fi definite mai multe condiții de selecție legate între ele cu operatorul OR: prima se scrie în rîndul *Criteria*, iar celelalte – mai jos, cîte una în fiecare celulă;
- două sau mai multe condiții din rîndul *Criteria* se consideră legate între ele cu operatorul AND.

Exemple:

- Dacă în formularul interogării anterioare mai jos de valoarea "Ion" (în celula *or*) se va scrie "Vasile", atunci interogarea va afișa lista elevilor cu prenumele Ion sau Vasile. În cazul completării celulei *Criteria* din câmpul *Nume_elev* cu condiția "B*", atunci interogarea va afișa lista elevilor cu prenumele Ion și al căror nume începe cu litera "B".
- Interogarea din *figura 9.4* va afișa lista elevilor clasei a 11-a B născuți în luna ianuarie.
- Interogarea din *figura 9.5* va afișa lista elevilor din clasele 10–11 cu profil real. Evident, rezultatul va fi același dacă vom completa doar rîndul *Criteria*, substituind valoarea 10 cu expresia 10 OR 11.
- Interogarea din *figura 9.6* va afișa lista profesorilor cu salariul lunar mai mare decît 2 500 lei și mai mic sau egal cu 3 000 lei în ordinea crescătoare a acestor salarii.

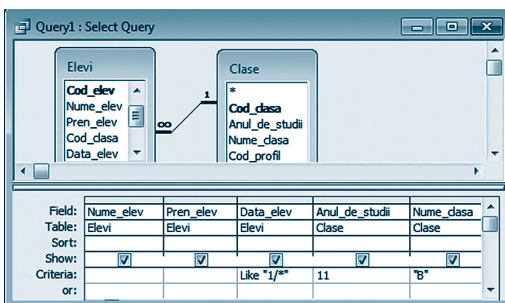


Fig. 9.4

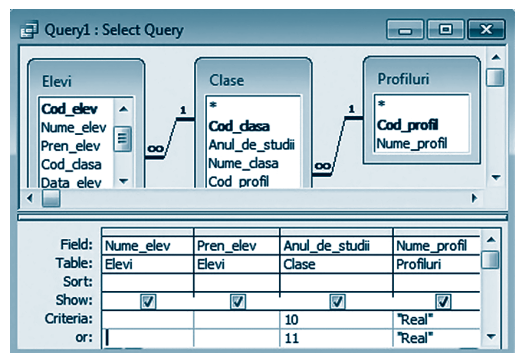


Fig. 9.5

9.2.2. Interogări cu parametri

Evident, este imposibil de a prezice toate cererile de selecție de care vor avea nevoie utilizatorii. Mai mult chiar, unele interogări se pot deosebi între ele doar prin unele valori din formularul QBE. De exemplu, interogările care vor afișa lista disciplinelor studiate în clasa a 10-a B, respectiv, în clasa a 11-a B, se vor deosebi doar prin valorile câmpului *Anul_de_studii*.

În astfel de situații se poate crea o singură interogare, unde în loc de valoare, în câmpul corespunzător, se va scrie *un parametru*.

Un șir de caractere încadrat între simbolurile [și], scris într-o celulă a rîndului *Criteria* este interpretat de Access drept **parametru**.

Pentru fiecare parametru, la executarea interogării, mai întâi va apărea o fereastră de dialog în care utilizatorul va scrie valoarea parametrului (deci o valoare a câmpului pentru care a fost creat parametru), apoi se vor afișa înregistrările ale căror valori din câmpul parametrului coincid cu valoarea scrisă de utilizator.

De regulă, șirul de caractere ce definește parametru este un text explicativ, care sugerează utilizatorului ce tip de valoare trebuie să scrie în fereastra de dialog.

În mod implicit parametru se consideră de tip *Text*. Pentru a modifica tipul parametrului, se va selecta *Query* → *Parameters*. Va apărea fereastra *Query Parameters*, în care se va scrie fiecare parametru și tipul lui.

Exemplu: În figura 9.7 este reprezentată fereastra unei interogări de selecție cu trei parametri: primul pentru câmpul *Anul_de_studii*, al doilea – pentru câmpul *Nume_clasa*, iar al treilea (de tip *Number*) – pentru câmpul *Nr_ore_saptamina*. Observăm că prin intermediul acestei interogări se poate afla lista disciplinelor studiate de elevii clasei indicate de utilizator, al căror număr de ore pe săptămînă este de asemenea specificat de utilizator.

La executarea interogării din figura 9.7 vor apărea pe rînd trei ferestre în care utilizatorul va preciza anul de studii, numele clasei (A, B, C sau D) și, respectiv, numărul săptămînal de ore.

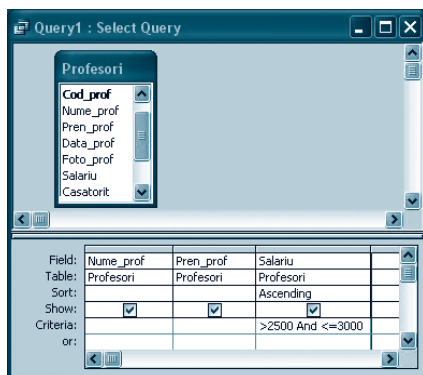


Fig. 9.6

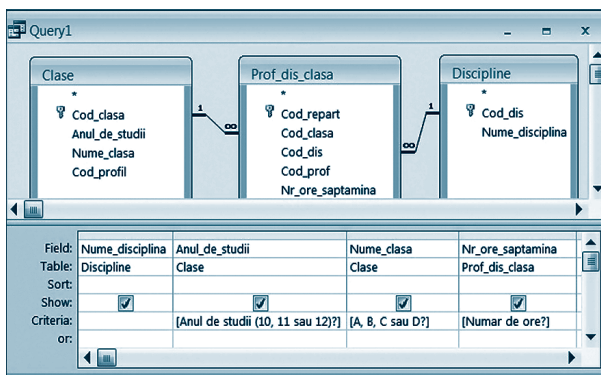


Fig. 9.7

Întrebări și exerciții

- ❶ Ce constituie interogările de selecție?
- ❷ Care este rolul criteriului de selecție într-o interogare?
- ❸ Cu ce scop se folosesc parametrii într-o interogare?
- ❹ Deschideți baza de date *Liceu*. Creați o interogare de selecție ce va afișa lista:
 - a) fetelor bazei;
 - b) diriginților;
 - c) profesorilor născuți pînă la 12 februarie 1970;
 - d) profesoarelor născute în luna martie;
 - e) băieților în ordine crescătoare a vârstei lor;
 - f) elevilor care nu au telefon;
 - g) profesorilor de gen masculin care nu au împlinit 50 de ani;
 - h) elevilor născuți vara;
 - i) profesorilor al căror număr de telefon începe cu 022 48;
 - j) elevilor care nu locuiesc în Chișinău;
 - k) profesorilor al căror prenume începe cu litera A sau cu litera E;
 - l) elevilor care nu învață Filosofia;
 - m) elevilor care studiază Matematica 5 ore pe săptămîină.
- ❺ Deschideți baza de date *Liceu*. Creați o interogare de selecție cu parametru ce va afișa lista:
 - a) disciplinelor studiate de elevul specificat de utilizator;
 - b) profesorilor care predau în clasa specificată de utilizator;
 - c) profesorilor care locuiesc în localitatea specificată de utilizator;
 - d) elevilor care studiază disciplina specificată de utilizator;
 - e) profesorilor care predau disciplina specificată de utilizator.
- ❻ Formulați și creați 5 interogări de selecție pentru baza de date *Liceu*.
- ❼ Pentru baza de date *Liceu* formulați și creați 5 interogări de selecție cu parametru.

9.3. Interogări de acțiune

Interogările de acțiune se utilizează pentru a crea tabele noi în baza informațiilor din tabelele existente și/sau pentru a realiza modificări în aceste tabele. În fereastra bazei de date numele acestor interogări este precedat de semnul exclamării.

Atenție! Interogările de acțiune (cu excepția celor care generează tabele) modifică conținutul tabelelor.

9.3.1. Interogări care generează tabele

Interogările de selecție extrag date din tabele și le afișează doar la executarea interogării. Rezultatul unei astfel de interogări nu se păstrează (eventual într-un tabel). Cu acest scop, interogarea de selecție poate fi transformată într-o interogare care va păstra rezultatul într-un tabel nou, deci va crea un tabel cu setul dinamic al interogării.

Să alcătuim o interogare care va genera un tabel cu date despre elevii claselor a 10-a.

1. Elaborăm interogarea de selecție corespunzătoare (fig. 9.8).

2. Din meniul *Query* alegem *Make-Table Query...* Apare fereastra *Make Table* unde precizăm numele tabelului nou (de exemplu, *Clasele_10*) și numele bazei de date în care va fi păstrat tabelul. Implicit, tabelul va fi creat în baza de date curentă.

3. Scriem numele tabelului, confirmăm prin apăsarea butonului OK, apoi salvăm interogarea.

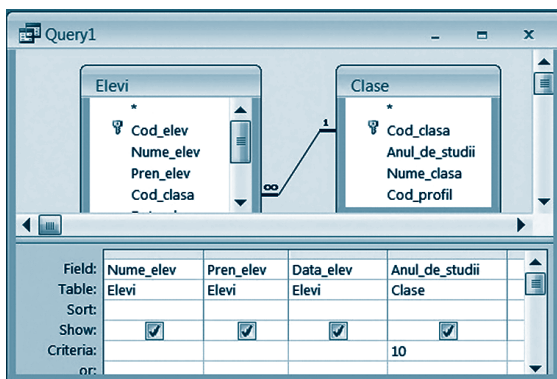


Fig. 9.8

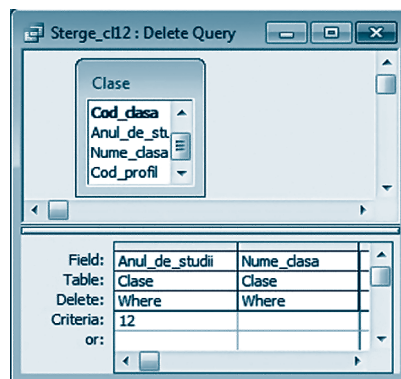


Fig. 9.9

4. Pentru a obține tabelul *Clasele_10*, executăm interogarea recentă. Apare o fereastră cu mesaj de avertizare, în care confirmăm intenția de creare a tabelului prin apăsarea butonului *Yes*.

9.3.2. Interogări de excludere a unor înregistrări

Deseori apare necesitatea eliminării unor înregistrări din table. De exemplu, în cazul bazei de date *Liceu* să admitem că trebuie să excludem datele despre elevii claselor a 12-a (în legătură cu absolvirea liceului).

1. Așa cum între tablele *Clase* și *Elevi* există relația **unu la mulți**, având caracteristica *Excluderea în cascadă a înregistrărilor* (*Cascade Delete Related Records*), este suficient să ștergem înregistrările din tabelul *Clase* care au valoarea câmpului *Anul_de_studii* egală cu 12. Invers nu este corect, deoarece tabelul *Elevi* este subordonat tabelului *Clase*.

2. Elaborăm o interogare care afișează lista claselor. Din meniul *Query* alegem *Delete Query*. În formularul QBE, în locul rîndurilor *Show* și *Sort*, apare rîndul *Delete*. În celula *Criteria* a câmpului *Anul_de_studii* scriem 12 (fig. 9.9).

3. Salvăm interogarea. Ștergerea înregistrărilor se va realiza după executarea interogării.

Atenție! Așa cum înregistrările șterse nu pot fi restabilite, se recomandă de afișat rezultatul interogării de selecție înainte ca aceasta să fie transformată în una de excludere a înregistrărilor. Astfel, se va verifica corectitudinea acțiunii interogării.

9.3.3. Interogări de modificare a unor înregistrări

Dacă valorile mai multor înregistrări ale unui câmp pot fi transformate după același algoritm, atunci pentru automatizarea actualizărilor se va folosi o interogare de modificare.

Să formulăm o interogare care va mări cu 50% valorile din câmpul *Salariu* al tabelului *Profesori*.

1. Creăm o interogare de selecție care afișează valorile câmpului *Salariu*.

2. Din meniul *Query* alegem *Update Query*. În formularul QBE în locul rîndurilor *Sort* și *Show* apare rîndul *Update To*. Scriem expresia $[Salariu]*1,5$ în celula obținută

la intersecția rîndului *Update To* și a cîmpului *Salariu* (fig. 9.10).

3. Salvăm interogarea. Modificarea înregistrărilor se va realiza după executarea interogării.

Atenție! Interogările de modificare se execută o singură dată. La repetarea lor valorile se vor modifica din nou. Astfel, dacă vom executa de 2 ori ultima interogare, salariul profesorilor „se va mări” cu 125% în loc de 50%.

9.3.4. Interogări care adaugă înregistrări în table existente

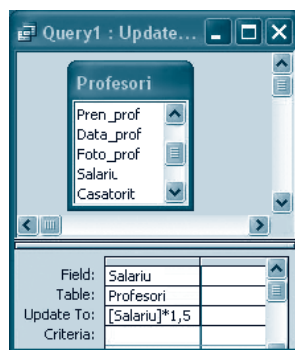


Fig. 9.10

Uneori trebuie să adăugăm într-un tabel înregistrări din alt tabel. De exemplu, fie tabelul *Elevi1* care conține date despre elevii născuți în luna mai. Dacă dorim să adăugăm în acest tabel înregistrările elevilor din tabelul *Elevi* născuți în luna iunie, vom proceda astfel:

1. Creăm o interogare de selecție care va afișa lista elevilor născuți în luna iunie.

2. Din meniul *Query* alegem *Append Query*. Apare fereastra *Make Table*. Din lista derulantă *Table Name* selectăm *Elevi1*.

3. Salvăm, apoi executăm interogarea.

Întrebări și exerciții

- 1 Cu ce scop se utilizează interogările de acțiune?
- 2 Descrieți algoritmul de creare a unei interogări:
 - a) care generează tablele;
 - b) de modificare a unor înregistrări;
 - c) de excludere a unor înregistrări;
 - d) care adaugă înregistrări în table existente.
- 3 Examinați baza de date *Liceu*. Creați o interogare ce va genera tabelul:
 - a) T1 cu date despre profesorii de gen feminin;
 - b) T2 cu date despre elevii născuți în luna mai;
 - c) T3 cu date despre diriginții de clase;
 - d) T4 cu date despre profesorii care predau matematica sau chimia;
 - e) T5 cu date despre elevii care nu locuiesc în Chișinău.
- 4 Examinați baza de date *Liceu*. Elaborați o interogare ce șterge din tabelul:
 - a) T1 datele despre profesorii născuți vara;
 - b) T2 datele despre elevii clasei a 11-a;
 - c) T3 datele despre diriginții claselor cu profil real;
 - d) T4 datele despre profesorii care nu predau chimia;
 - e) T5 datele despre elevii care locuiesc în Cricova.
- 5 Examinați baza de date *Liceu*. Alcătuiți o interogare ce va:
 - a) mări salariul profesorilor cu 500 de lei;
 - b) micșora cu 300 de lei salariul profesorilor care predau doar o disciplină;
 - c) mări cu 25% salariul profesorilor născuți pînă la 01.01.1960.
- 6 Examinați baza de date *Liceu*. Creați o interogare ce va adăuga în tabelul:
 - a) T1 date despre profesorii de gen masculin care predau matematica;
 - b) T2 date despre elevii născuți iarna;
 - c) T3 date despre profesorii care nu sînt diriginți și locuiesc în Chișinău;
 - d) T4 date despre profesorii care predau limba străină;
 - e) T5 date despre elevii din Chișinău care învață în clasa a 10-a.

9.4. Interogări de totalizare

9.4.1. Interogări de creare a câmpurilor rezultante (calculate)

În capitolele precedente am menționat că la proiectarea entităților unei baze de date relaționale se vor exclude câmpurile ale căror valori pot fi obținute din câmpurile rămase.

Din aceste considerente în tabelul *Elevi* nu a fost inclus câmpul *Virsta*. Valorile acestui câmp depind de valorile câmpului *Data_elev*. Să elaborăm o interogare care va afișa într-un câmp nou vârstele elevilor bazei de date *Liceu*. Interogarea nu va afecta structura și datele tabelului *Elevi*.

1. Creăm o interogare de selecție pe baza tabelului *Elevi*. În primele două coloane ale formularului QBE includem câmpurile *Nume_elev*, *Pren_elev*, iar în locul denumirii coloanei a treia scriem expresia *Virsta: DateDiff("yyyy";[Data_elev];Date())*. Menționăm că *Virsta* este identificatorul câmpului nou, iar funcția *DateDiff(T; D₁; D₂)* returnează diferența exprimată în unități calendaristice *T* dintre datele *D₁* și *D₂* (a se vedea tema *Funcții Access*). Observăm că:

- *T* este egal cu "yyyy", deci exprimă ani;
- *D₁* este [Data_elev], adică data nașterii elevului;
- *D₂* este Date(), adică data curentă.

2. Salvăm interogarea. Câmpul *Virsta* este dinamic. El există atît timp cît sînt afișate rezultatele interogării.

9.4.2. Interogări de grupare și totalizare

Pentru a obține rezultate bazate pe înregistrările unui sau ale mai multor tabele, se vor utiliza *interogări de grupare și totalizare*.

Să definim o interogare care va afișa numărul total de elevi ai fiecărei clase din baza de date *Liceu*.

1. Creăm o interogare de selecție pe baza tabelelor *Clase* și *Elevi* în care includem câmpurile *Anul_de_studii*, *Nume_clasa* și *Cod_elev*.

2. Executăm un clic pe butonul *Totals* (Σ) de pe bara de instrumente. În formularul QBE apare rîndul *Total* (fig. 9.11). Completăm celulele acestui rînd:

- în primele două coloane din listele derulante ale celulelor selectăm valoarea *Group By* (deoarece grupăm înregistrările după anul de studii și numele clasei), iar

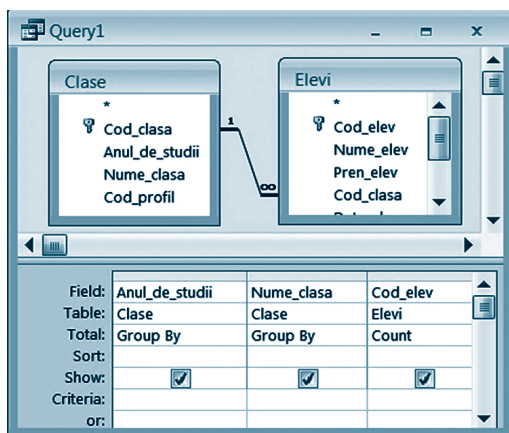


Fig. 9.11

Anul_de_studii	Nume_clasa	CountOfCod_elev
10	A	29
10	B	18
10	C	37
10	D	31
11	A	33
11	B	26
11	C	22
11	D	23
12	A	22
12	B	24
12	C	20

Fig. 9.12

– în coloana a treia selectăm funcția *Count* (deoarece calculăm numărul de înregistrări ale câmpului *Cod_elev*).

3. Salvăm și executăm interogarea. Rezultatul interogării este prezentat în figura 9.12.

Observații:

1. Listele derulante ale celulelor rîndului *Total* oferă diferite funcții globale (funcții aplicabile asupra grupurilor de celule de date) pentru obținerea totalizărilor: *Sum*, *Max*, *Min*, *Avg*, *First* etc.
2. În interogări de grupare și totalizare, de asemenea, se pot formula criterii de selecție. De exemplu, dacă pentru câmpul *Cod_elev* al interogării precedente se va scrie condiția >25 , atunci interogarea va afișa datele doar despre clasele al căror număr total de elevi este mai mare decât 25.

9.4.3. Interogări încrucișate

Interogările încrucișate sînt interogări de totalizare care permit utilizatorului să stabilească exact modul în care vor fi afișate rezultatele sub formă tabelară. Astfel de interogări sînt recomandate în cazul unei cantități mari de date totalizatoare. La crearea unei interogări încrucișate se va ține cont de următoarele restricții:

- a) denumirile rîndurilor tabelului-rezultat pot fi valori din unul sau mai multe câmpuri;
- b) denumirile coloanelor tabelului-rezultat pot fi valori doar ale unui singur câmp;
- c) valorile celorlalte celule ale tabelului-rezultat sînt calculate cu ajutorul unei funcții globale;
- d) înregistrările din rezultat nu pot fi ordonate după câmpurile celulelor calculate.

Să alcătuim o interogare care va afișa numărul total de ore rezervat fiecărei discipline în fiecare clasă a bazei de date *Liceu*.

1. Creăm o interogare de selecție pe baza tabelelor *Clase*, *Discipline* și *Prof_dis_clasa* în care includem câmpurile *Anul_de_studii*, *Nume_clasa*, *Nume_disciplina* și *Nr_ore_saptamina*.

2. Din meniul *Query* alegem *Crosstab Query*. În formularul QBE apar rîndurile *Total* și *Crosstab* (fig. 9.13).

3. Completăm celulele rîndurilor *Total* și *Crosstab* ca în figura 9.13. Astfel, valorile câmpurilor *Anul_de_studii* și *Nume_clasa* vor fi denumiri ale rîndurilor tabelului-rezultat, valorile câmpului *Nume_disciplina* – denumiri ale coloanelor tabelului-rezultat, iar valorile câmpului *Nr_ore_saptamina* vor fi sumate pentru a completa celelalte celule ale tabelului-rezultat (fig. 9.14).

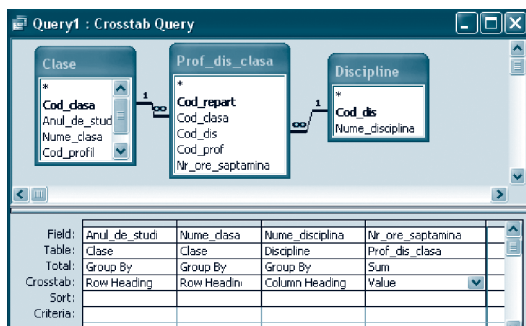


Fig. 9.13

Anul_de_studii	Nume_clasa	Biologia	Chimia	Educația civică
10	A	2	3	1
10	B	2	3	1
10	C	1	1	1
10	D	1	1	1
11	A	3	2	1
11	B	3	2	1
11	C	1	1	1
11	D	1	1	1
12	A	3	3	1
12	B	3	3	1
12	C	1	1	1
12	D	1	1	1

Fig. 9.14

Întrebări și exerciții

- 1 Ce reprezintă interogările:
a) de creare a câmpurilor rezultante; b) de grupare și totalizare; c) încrucișate?
- 2 Descrieți algoritmul de elaborare a unei interogări:
a) de creare a câmpurilor rezultante; b) de grupare și totalizare; c) încrucișate.
- 3 Examinați baza de date *Liceu*. Aflați cu ajutorul unei interogări:
a) vârsta elevilor exprimată în zile;
b) numărul de profesori de gen masculin și numărul celor de gen feminin;
c) numărul de clase la fiecare din profilurile *real* și *umanist*;
d) salariul mediu lunar al profesorilor;
e) numărul de elevi din fiecare localitate;
f) numărul de ore realizate săptămînal de fiecare profesor;
g) numărul de ore realizate săptămînal de fiecare clasă;
h) valoarea salariului minimal;
i) profesorii care au salariul maximal;
j) numărul de profesori care au salariul mai mic decît cel mediu.
- 4 Examinați baza de date *Liceu*. Formulați și realizați două cereri de interogări:
a) de creare a câmpurilor rezultante; b) de grupare și totalizare; c) încrucișate.

Test de evaluare

1. Examinați baza de date *Liceu*. Creați o interogare ce va afișa lista:
a) elevilor al căror prenume începe cu litera A;
b) elevilor născuți joi;
c) profesorilor care predau disciplina specificată de utilizator.
2. Examinați baza de date *Liceu*. Alcătuiți o interogare ce va:
a) adăuga în tabelul *Elevi* câmpul *Adresa_Web*, ale cărui valori vor fi formate în această ordine din numele, prenumele elevului și secvența „@liceu.md”. De exemplu, adresa Web a elevului Ion Popa va fi *Popalon@liceu.md*;
b) genera tabelul *Test* cu date despre elevii care nu învață în clasele a 10-a A, a 11-a B, a 12-a C (inclusiv adresa Web descrisă în a));
c) elimina din tabelul *Test* datele despre elevii care învață în clasele a 10-a B, a 11-a C;
d) modifica adresa Web a fiecărui elev din tabelul *Test*, substituind subșirul „liceu.md” cu secvența „elev.md”;
e) adăuga în tabelul *Test* date despre elevii de gen feminin care învață în clasa a 10-a A;
f) mări cu 40% salariul profesorilor cu vârsta mai mare decît 40 de ani.
3. Examinați baza de date *Liceu*. Elaborați o interogare ce va afișa numărul:
a) de elevi la fiecare din profilurile *real* și *umanist*;
b) elevilor de gen masculin și numărul celor de gen feminin.
4. Creați o interogare încrucișată ce va afișa numărul elevilor în fiecare clasă a bazei de date *Liceu* (ca în figura 9.15).

Anul de studii	Nume_clasa	Real	Umanist
	10 A	29	
	10 B	18	
	10 C		37
	10 D		31
	11 A	33	
	11 B	26	
	11 C		22
	11 D		23

Fig. 9.15

FORMULARE ȘI RAPOARTE

După studierea acestui capitol, veți fi capabili să:

- identificați formulare și elementele care alcătuiesc aceste formulare;
- elaborați formulare și subformulare cu ajutorul sistemului de asistență;
- utilizați formulare pentru vizualizarea, modificarea, formatarea și validarea datelor;
- alcătuiți formulare pe baza tabelelor corelate;
- identificați rapoarte și elementele care formează aceste rapoarte;
- elaborați rapoarte și subrapoarte cu ajutorul sistemului de asistență;
- creați rapoarte pe baza tabelelor corelate;
- grupați și să totalizați date într-un raport.

10.1. Formulare

Formularele sînt obiecte ale bazei de date Access, proiectate special pentru a crea o interfață pentru utilizator cu scopul facilitării introducerii, editării și afișării înregistrărilor din tabele sau interogări.

Formularele măresc viteza și minimizează erorile de introducere a datelor. În același timp, ele permit vizualizarea datelor într-un format mai atractiv decît cel al regimului *Datasheet View*.

Cu ajutorul formularelor pot fi validate intrări de date pe baza informațiilor din alte tabele (decît cel la care se lucrează), pot fi create *subformulare* (formulare conținute în alte formulare), *casete de căutare* (pentru accesarea rapidă a înregistrării), *liste de opțiuni* etc. Componentele unui formular se numesc **elemente de control** sau **obiecte de control**.

10.1.1. Crearea unui formular cu ajutorul programului de asistență


Să creăm un formular care va permite editarea datelor din tabelul *Elevi* al bazei de date *Liceu*.

1. Selectăm clasa de obiecte **Forms** de pe bara de obiecte *Objects*. În zona de conținut a ferestrei bazei de date apar opțiunile:

- *Create form in Design view* (crearea formularului în regim de proiectare);
 - *Create form by using wizard* (crearea formularului cu ajutorul programului de asistență).
- Executăm un dublu-clic pe a doua opțiune.

Observație: *Pasul 1* este echivalent cu selectarea *New* → *Form wizard*.

2. Apare fereastra *Form Wizard* (fig. 10.1). Din lista combinată *Table/Queries* selectăm valoarea *Table: Elevi*. Astfel, în lista *Available Fields* apar identificatorii cîmpurilor tabelului *Elevi*.

3. Selectăm cîmpurile care vor fi incluse în formular, confirmînd alegerea prin apăsarea butonului . Pentru a include toate cîmpurile din listă, vom executa un clic pe

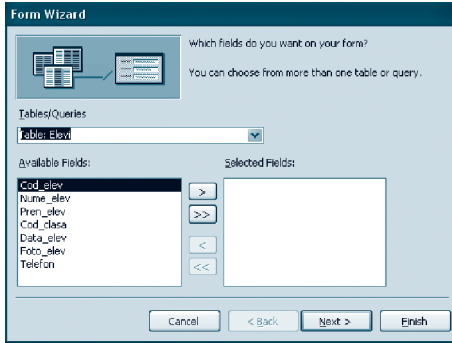


Fig. 10.1

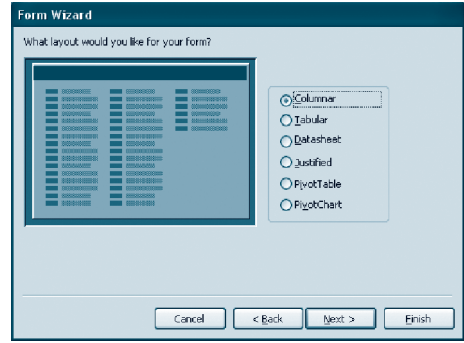


Fig. 10.2

butonul **>>**. Identificatorii câmpurilor selectate sînt mutați în lista *Selected Fields*. Butoanele **<** și **<<** se folosesc pentru a deplasa înapoi câmpurile greșit alese. Selectăm toate câmpurile și apăsăm butonul *Next*.

4. Următoarea fereastră *Form Wizard* oferă posibilitatea alegerii modului de afișare a câmpurilor în formular (fig. 10.2). Selectăm opțiunea *Columnar* (câmpurile alese la pasul precedent vor fi afișate în formular una sub alta în coloane).

5. La pasul următor alegem unul din stilurile predefinite de formular (fig. 10.3).

6. În ultima fereastră precizăm numele formularului (*Formular_Elevi*). Dacă alegem opțiunea *Modify the Form's design* (fig. 10.4), atunci după apăsarea butonului *Finish*, formularul va fi deschis în regim de proiectare. În caz contrar, formularul va apărea în regim de editare a datelor.

7. Deschidem formularul pentru a vedea și a introduce informații (fig. 10.5).

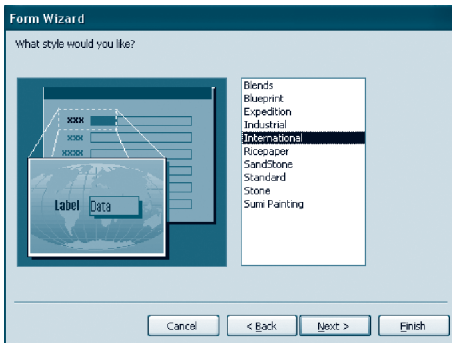


Fig. 10.3

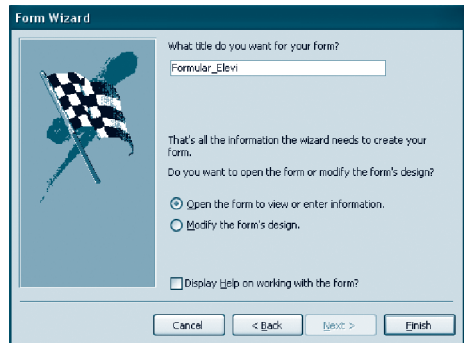






Fig. 10.4



Fig. 10.5

Butoanele din partea de jos a formularului se folosesc pentru a lista înregistrările. Astfel, butoanele ,  afișează prima și, respectiv, ultima înregistrare. Pentru a vedea precedentă sau următoarea înregistrare față de cea curentă, se va acționa butonul  sau, respectiv, .

La inserarea datelor noi se va apăsa butonul  sau se va executa *New Record* din meniul *Insert*.

Observații:

1. Un formular poate fi creat în baza mai multor obiecte-sursă (tabele și interogări).
2. Modificarea datelor în formular atrage după sine modificarea datelor din obiectul-sursă.
3. Dacă dorim să adăugăm înregistrări noi cu ajutorul unui formular, atunci acesta trebuie să conțină câmpurile cheie primară și câmpurile care au proprietatea *Required* setată *Yes*. Dacă aceste câmpuri nu vor fi completate, atunci înregistrarea nouă nu va fi acceptată.

10.1.2. Crearea sau modificarea formularelor în regimul *Design View* (regim de proiectare)

Regimul *Design View* este cel mai complet mod de creare sau modificare a formularelor, deoarece în el se poate edita orice element de control al formularului.

Suprafața formularului este formată din câteva secțiuni (*fig. 10.6*). Redimensionarea fiecărei secțiuni se realizează cu ajutorul mouse-ului.

Linia orizontală (de jos) și linia verticală (la dreapta) stabilesc marginea de jos și, respectiv, marginea dreaptă a formularului.

- Secțiunea **Detail** apare automat la crearea sau editarea unui formular nou. În ea sînt afișate înregistrările din sursa de date a formularului.

- Secțiunile **Form Header** și **Form Footer** nu apar în mod implicit la editare, afișarea lor fiind posibilă prin selectarea opțiunii *Form Header/Footer* a meniului *View*. Ele reprezintă *zona de antet* și, respectiv, cea de *subsol* ale formularului, în care, de regulă, se scriu informații explicative sau utile. Aceste informații nu se schimbă de la o înregistrare la alta și pot conține, de exemplu, sugestii de utilizare a formularului, totaluri, date despre autori, timpul curent, data creării formularului etc.

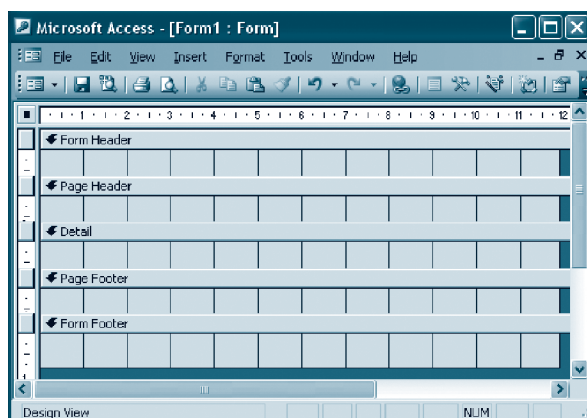


Fig. 10.6


• Secțiunile **Page Header** (*antetul de pagină*) și **Page Footer** (*subsolul de pagină*) de asemenea trebuie afișate explicit activând opțiunea *Page Header/Footer* a meniului *View*.


Page Header afișează informații în partea de sus, iar *Page Footer* – în partea de jos a fiecărei pagini a formularului.


Observație: Un antet și un subsol de formular se elimină sau se deplasează doar în pereche. În afară de suprafața formularului, fereastra *Form Design* conține barele de instrumente *Form Design* și *Toolbox*.

• Bara *Form Design* se utilizează pentru a executa rapid unele acțiuni, fără a apela la meniul principal al ferestrei de proiectare a formularului.


• Bara *Toolbox* se utilizează pentru a plasa elemente de control pe suprafața formularului. Să examinăm unele instrumente ale ei:

 *Select Objects* nu plasează un oarecare element de control, el se folosește pentru a selecta (atunci când este activ) elemente de control din formular.


 *Control Wizard* (atunci când este activ) permite intervenția programelor de asistență la crearea unor elemente de control.

 *Label* creează o casetă care poate afișa un text static (ce nu va putea fi modificat de utilizator).


 *Text Box* creează o casetă care va afișa texte ce vor putea fi modificate de utilizator.

 *Option Group* creează o casetă dreptunghiulară în care pot fi inserate butoane de opțiune sau casete de validare. În cazul butoanelor, utilizatorul va putea selecta în orice moment de timp un singur buton.

 *Toggle Button* creează un buton de comutare pentru activarea sau dezactivarea unei stări.

 *Option Button* creează un buton (se mai spune buton radio) utilizat de asemenea pentru activarea sau dezactivarea unei stări (ori opțiuni). De regulă, se folosește în grup cu mai multe butoane de acest tip pentru organizarea opțiunilor eliminatorii.


 *Check Box* creează o casetă de validare pentru validarea sau interzicerea unei stări.

 *Combo Box* creează o casetă combinată (se mai spune listă ascunsă) formată dintr-o casetă de text editabilă și o listă derulantă din care se poate alege o valoare.


 *List Box* creează o listă derulantă din care se poate alege o valoare.


 *Command Button* creează un buton de comandă pentru lansarea la execuție a unei comenzi.





 *Image* afișează o imagine statică (care fiind plasată în formular nu va putea fi modificată).

 *Unbound Object* afișează un obiect OLE neasociat creat cu o aplicație OLE (de exemplu, Microsoft Draw). Nu este o valoare dintr-un câmp OLE a unei înregistrări.

 *Bound Object* afișează un obiect OLE – conținut al unui câmp OLE (de exemplu, o imagine).

 *Page Break* creează un semn pentru imprimantă, determinînd-o să treacă la o pagină nouă.

 *Tab Control* creează un element de control cu mai multe pagini. Fiecare pagină poate afișa un formular. Cu acest element ușor se realizează comutarea între formulare.

-  *Subform/Subreport* atașează formularului un subformular sau un subraport.
-  *Line* creează o linie utilizată pentru design (culoarea și grosimea liniei pot fi modificate).
-  *Rectangle* creează un dreptunghi utilizat pentru design.
-  *More Controls* afișează instrumente pentru crearea altor elemente de control.

• Din punctul de vedere al funcționalității, deosebim următoarele **categorii ale elementelor de control**:

- a) *elemente de control legate*;
- b) *elemente de control calculate*;
- c) *elemente de control independente*.

Elementele de control legate servesc pentru afișarea și editarea datelor câmpurilor tabelor și interogărilor în baza cărora a fost creat formularul. Un element de control legat este format dintr-un câmp (în care apar datele) și o etichetă asociată acestui câmp (care afișează un text explicativ, de exemplu, denumirea câmpului). În mod implicit, eticheta asociată afișează identificatorul câmpului ale cărui valori se vor afișa de elementul de control (fig. 10.7). În regimul *Datasheet View* în câmpul din figura 10.7 al formularului creat anterior va apărea un număr de telefon (fig. 10.5).

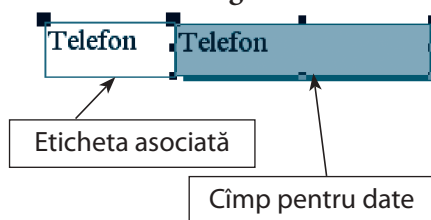






Fig. 10.7

Componentele elementului de control pot fi redimensionate și deplasate. Dacă la plasarea mouse-ului pe element apare simbolul mâinii , atunci prin metoda *Drag & Drop* putem deplasa simultan ambele componente ale elementului de control. Plasând mouse-ul pe unul din cele două puncte de contact din colțul stînga-sus al componentelor, va apărea simbolul degetului arătător . În acest caz putem deplasa în parte fiecare componentă a elementului de control.

Adăugarea unui element de control legat se poate face plasînd în formular (prin metoda *Drag & Drop*) câmpul necesar din submeniul *Field List* al meniului *View* sau inserînd un element *Text Box*  (de pe bara *ToolBox*) și indicînd în caseta *Unbound* numele câmpului necesar.

În pofida denumirii sale, o casetă de text poate afișa, în afară de texte, date numerice, calendaristice etc.

În mod implicit, Access aliniaza textul la stînga și numerele la dreapta.

Dacă caseta de text creată va afișa texte mari (de exemplu, conținuturi ale câmpurilor de tip *Memo*), atunci ea poate fi redimensionată la mărimea dorită. Pentru a atașa acestei casete o bară de defilare, se execută un clic pe butonul *Properties* , apoi se atribuie valoarea *Vertical* proprietății *Scroll Bars* din fila *Format* (a ferestrei apărute, fig. 10.9).

Elementele de control calculate se folosesc pentru afișarea valorilor expresiilor și pot fi modificate ca și elementele de control legate. Adăugarea unui astfel de element se face de asemenea prin inserarea unui element *Text Box*, doar că în caseta *Unbound* se scrie expresia (precedată de semnul =) a cărei valoare va apărea.

De exemplu, elementul de control din figura 10.8, fiind inserat în formularul creat anterior, va afișa vârsta elevului.



Fig. 10.8

Elementele de control independente nu depind de informațiile din tabele sau interogări, deci ele nu se modifică de la o înregistrare la alta. Se folosesc pentru a crea efecte de design.

- **Proprietățile unui element** de control stabilesc diferite caracteristici ale acestora: culoarea, dimensiunile, sursa de date, poziția în cadrul formularului, comportamentul elementului la unele acțiuni ale utilizatorului etc.

Proprietățile unui element pot fi modificate cu ajutorul ferestrei, care apare la efectuarea unui dublu-clic pe acest element (fig. 10.9). Ele sînt grupate în patru categorii: *Format*, *Data*, *Event*, *Other*. Grupul *All* include toate proprietățile, pe care le afișează în ordine alfabetică.

- Grupul **Format** înglobează proprietățile de prezentare a obiectului (dimensiuni, culoare, format etc.).
- Grupul **Data** conține proprietățile referitoare la sursa înregistrărilor care vor fi gestionate de elementul de control.
- Grupul **Event** include lista evenimentelor (acțiuni provocate de utilizator sau de aplicație) la care poate reacționa elementul.
- Grupul **Other** păstrează celelalte proprietăți. Ele se referă la ferestrele Windows.

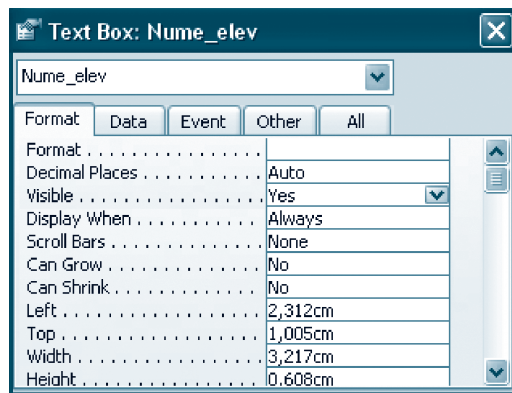


Fig. 10.9

- Un formular poate fi imprimat ca oricare alt obiect al bazei de date: executînd un clic pe butonul *Print* de pe bara de instrumente sau selectînd comanda *Print* din meniul *File*.

Pentru a vedea cum arată formularul înainte de imprimare, se execută un clic pe butonul *Preview* de pe bara de instrumente sau se alege comanda *Preview* din meniul *File*.

10.1.3. Subformulare

Un **subformular** este un formular care se conține în alt formular. Putem include subformulare într-un formular cu ajutorul unui program de asistență sau prin proiectare individuală. Vom examina doar cazul programului de asistență, acesta fiind disponibil pentru apelare doar dacă este activat butonul *Control Wizard* din caseta *Toolbox*.

Fie formularul *Clase* din figura 10.10 pentru afișarea și modificarea datelor despre clasele bazei de date *Liceu*. Vom adăuga un subformular în care vor apărea informații despre elevii clasei selectate în formularul principal.

1. Deschidem formularul *Clase* în regimul *Design*. Activăm mai întîi butonul *Control Wizard*, apoi butonul *Subform/Subreport* din caseta *Toolbox* și executăm un clic pe formularul de bază în poziția în care va apărea subformularul. Apare fereastra *SubForm Wizard* din care selectăm tabelul *Elevi*.

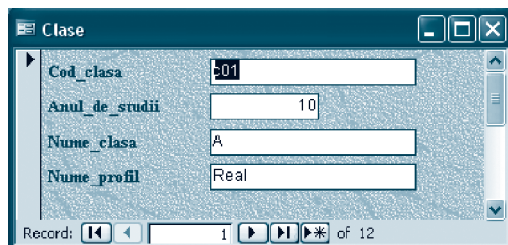


Fig. 10.10

2. În următoarea fereastră de dialog alegem cîmpurile tabelului *Elevi*, iar în fereastra a treia confirmăm denumirea cîmpului de legătură (*Cod_clasa*) dintre formular și subformular.

3. În ultima fereastră *SubForm Wizard* scriem numele subformularului (*Lista_elevi*). Activînd regimul *Form View*, putem vedea lista elevilor clasei selectate în formularul principal.

Observație: Subformularul creat *Lista_elevi* automat se va păstra pe disc la salvarea formularului de bază.

Întrebări și exerciții

- ❶ Cu ce scop se utilizează formularele?
- ❷ Ce obiecte ale bazei de date pot fi surse de date pentru formulare?
- ❸ Caracterizați secțiunile formularului în regim de proiectare.
- ❹ Care este rolul elementelor de control în cadrul unui formular?
- ❺ Descrieți categoriile elementelor de control.
- ❻ Care sînt modalitățile de modificare a proprietăților unui element de control?
- ❼ Examinați baza de date *Liceu*. Creați cu ajutorul unui program de asistență un formular pentru editarea datelor despre:
a) profesorii bazei, inclusiv adresa lor; b) disciplinele bazei.
- ❽ Adăugați într-un formular creat anterior o casetă de text care va afișa data curentă și timpul curent.
- ❾ Adăugați într-un formular creat anterior o casetă de text care va afișa numărul de zile rămase pînă la sfîrșitul: a) anului curent; b) anului școlar curent.
- ❿ Examinați baza de date *Liceu*. Creați un formular care va afișa datele despre fiecare clasă și va conține un subformular cu lista profesorilor ce predau în această clasă.
- ⓫ Examinați baza de date *Liceu*. Creați un formular care va afișa datele despre fiecare clasă și va conține un subformular cu lista disciplinelor studiate în această clasă.
- ⓬ Examinați baza de date *Liceu*. Creați un formular care va afișa datele despre fiecare profesor și va conține un subformular cu lista disciplinelor predate de acest profesor.

10.2. Rapoarte

Rapoartele reprezintă produsul final al unei baze de date. Ele combină date din tabele, interogări și formulare pentru a fi tipărite sau pentru a fi salvate pe disc într-un format atractiv și ușor de citit. Spre deosebire de formulare, rapoartele:

- a) sînt special destinate tipăririi;
- b) nu pot modifica datele din tabelele sau din interogările care stau la baza lor;
- c) nu afișează datele în formă tabelară (nu are un regim de tip *Datasheet View*).

Ca și în cazul formularelor, rapoartele pot fi elaborate cu ajutorul unui program de asistență sau prin proiectare independentă.

10.2.1. Crearea unui raport cu ajutorul programului de asistență

1. Selectăm clasa de obiecte **Reports** de pe bara de obiecte *Objects*. Alegem *New* → *Report Wizard*.

2. Apare fereastra *Report Wizard* (fig. 10.11). Din lista combinată *Tables/Queries* alegem pe rînd tabelele *Clase*, *Elevi* și interogarea *Virsta* (vezi tema 9.4.1), iar din caseta *Available Fields* selectăm cîmpurile *Anul_de_sudii*, *Nume_clasa* (ale tabelului *Clase*), *Nume_elev*, *Pren_elev* (ale tabelului *Elevi*) și *Virsta* (al interogării *Virsta*).

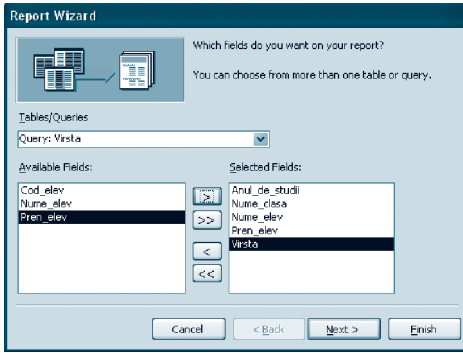


Fig. 10.11

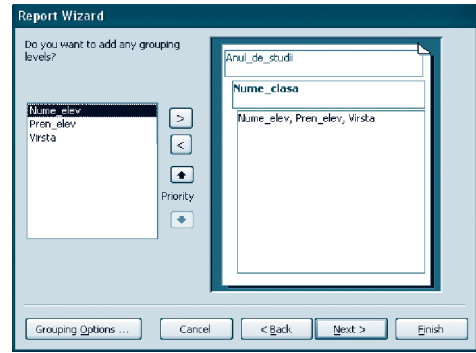


Fig. 10.12

3. În următoarea fereastră *Report Wizard* selectăm câmpurile, după care se va face gruparea înregistrărilor (fig. 10.12).

4. În fereastra a treia *Report Wizard* indicăm câmpurile în conformitate cu care se va face sortarea înregistrărilor (fig. 10.13).

Cu ajutorul butonului *Summary Options* putem specifica pentru fiecare câmp numeric una sau mai multe opțiuni de calcul: suma, media aritmetică, valoarea minimală sau cea maximală. Pentru a calcula vârsta medie a elevilor (pentru fiecare clasă și pe liceu), selectăm opțiunea *Avg* (fig. 10.14). Activarea opțiunii *Summary Only* va afișa doar vârsta medie pe liceu.

5. În celelalte trei ferestre *Report Wizard* specificăm modul de aranjare a înregistrărilor pe pagină, stilul și, respectiv, titlul raportului. În figura 10.15 este reprezentată o parte din raportul creat.

Pentru a tipări raportul, se execută un clic pe butonul *Print* de pe bara de instrumente.

Observație: Crearea subrapoartelor se realizează ca și crearea subformularelor.

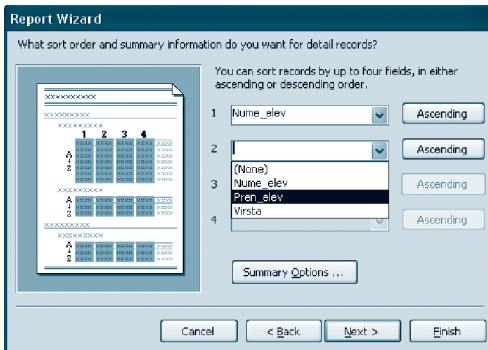


Fig. 10.13

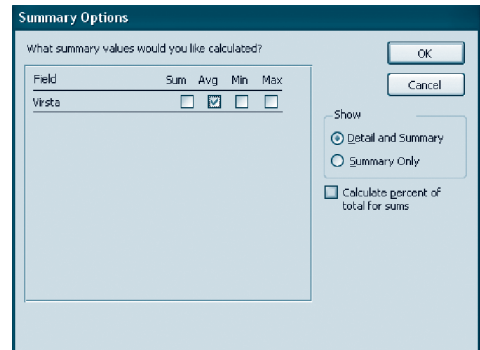


Fig. 10.14



Fig. 10.15

10.2.2. Modificarea rapoartelor în regimul *Design View*

Regimul *Design View* pentru crearea sau modificarea rapoartelor este asemănător cu regimul *Design View* al formularelor. Ca și formularele, rapoartele conțin 5 secțiuni principale: *Report Header*, *Page Header*, *Detail*, *Page Footer* și *Report Footer*. De asemenea, raportul poate conține și secțiuni pentru grupurile de date. De exemplu, în figura 10.16 este reprezentat raportul creat anterior, deschis în regim de proiectare. Observăm că, în afară de cele 5 secțiuni principale, mai apar secțiunile *Anul_de_studii Header*, *Nume_clasa Header*, *Anul_de_studii Footer* și *Nume_clasa Footer*.

Operațiile cu elementele de control ale rapoartelor se realizează similar operațiilor cu elementele de control din formulare.

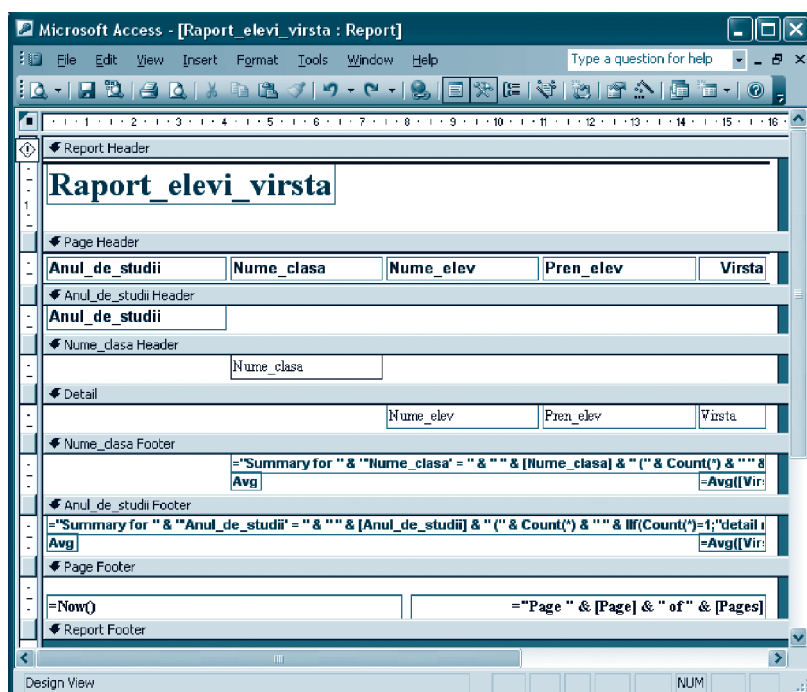


Fig. 10.16

10.2.3. Crearea diagramelor în cadrul rapoartelor

Se știe că reprezentarea grafică a datelor este mai ușor înțeleasă, mai ales atunci când se fac comparații între diferite valori numerice.

Într-un sistem de gestiune a bazelor de date diagramele sînt considerate un tip special de rapoarte. Putem crea o diagramă în baza unui tabel sau a unei interogări.

Exemplu: Să creăm o diagramă care va reprezenta repartizarea elevilor bazei de date *Liceu* pe clase în baza interogării *Nr_elevi_clasa* creată în tema 9.4.2.

1. Selectăm clasa de obiecte **Reports** de pe bara de obiecte *Objects*. Executăm un clic pe butonul *New*. Apare fereastra *New Report*, din care alegem tipul de raport *Chart Wizard* și interogarea *Nr_elevi_clasa* din lista derulantă. Confirmăm alegerile prin apăsarea butonului *Ok*.

2. Apare prima fereastră *Chart Wizard* în care selectăm câmpurile *Anul_de_studii*, *Nume_clasa* și *CountOfCod_elev*.

3. În următoarea fereastră *Chart Wizard* specificăm tipul diagramei: *Column Chart* (diagramă cu coloane).

4. În fereastra a treia *Chart Wizard* (fig. 10.17) stabilim sursa categoriilor de date (*Anul_de_studii*), sursa seriilor de date (*CountOfCod_elev*) și legenda (*Nume_clasa*).

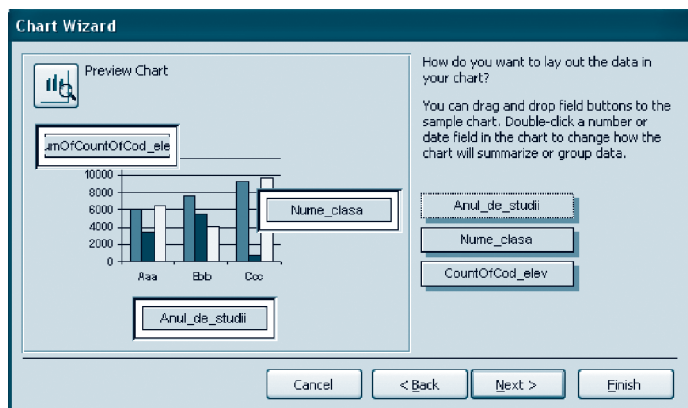


Fig. 10.17

5. În ultima fereastră *Chart Wizard* atribuim nume diagramei și precizăm dacă va fi sau nu afișată legenda. Obținem diagrama din figura 10.18.

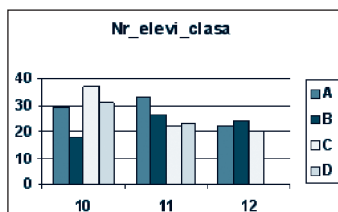


Fig. 10.18

Întrebări și exerciții

- 1 Cu ce scop se utilizează rapoartele?
- 2 Ce obiecte ale bazei de date pot fi surse de date pentru rapoarte?
- 3 Care este rolul diagramelor într-o bază de date?
- 4 Examinați baza de date *Liceu*. Creați un raport în care să fie arătată:
 - a) media aritmetică a orelor realizate săptămînal în fiecare an de studii;
 - b) media aritmetică a orelor realizate săptămînal de fiecare profil;
 - c) suma totală de ore realizate săptămînal în fiecare an de studii și în ansamblu pe liceu.
- 5 Examinați baza de date *Liceu*. Creați o diagramă care ar reprezenta grafic numărul:
 - a) de elevi la fiecare profil;
 - b) de elevi din fiecare localitate;
 - c) profesorilor de fiecare gen;
 - d) de ore săptămînal realizat de fiecare clasă.
- 6 Examinați baza de date *Liceu*. Creați o diagramă care ar reprezenta grafic numărul de elevi de fiecare vîrstă.

10.3. Întreținerea bazelor de date (opțional)

Pentru ca o bază de date să funcționeze normal, ea trebuie dezvoltată continuu și întreținută. De regulă, aceste operații sînt realizate de administratorul bazei de date. În acest paragraf vom examina unele acțiuni care trebuie efectuate pe măsura gestionării unei baze de date.

10.3.1. Compactarea și repararea unei baze de date

- Pe parcursul timpului, în urma adăugării și actualizării datelor, a modificărilor obiectelor bazei, *crește volumul fișierului bazei de date și scade viteza de gestionare* a bazei.

Aceste consecințe pot fi cauzate nu doar de volumul datelor, dar și de:

- prezența în baza de date a unor obiecte temporare (ascunse de utilizator), chiar dacă Access nu mai are nevoie de ele;
- existența în fișerul bazei de date a unor spații libere, rezultate din ștergerea obiectelor.

În acest caz baza de date are nevoie de *compactare* – operație de înlăturare a fragmentării fișierului bazei –, adică de eliminare a spațiilor și obiectelor neutilizate.

- Uneori fișierul bazei de date se poate deteriora. Acest lucru se poate întîmpla din cauza unor situații extreme (de exemplu, deconectarea sursei de energie, dereglări ale rețelei) sau atunci cînd mai mulți utilizatori lucrează în mod simultan cu fișierul dat.

În acest caz baza de date are nevoie de *reparație*.

Pentru a **compacta și a repara o bază de date**:

1. Închidem baza de date, lăsînd deschis sistemul Access.
2. Executăm *Tools* → *Database Utilities* → *Compact and Repair Database*.
3. Apare caseta de dialog *Database to Compact From*, în care precizăm numele bazei de date ce urmează a fi compactată și reparată. Confirmăm prin apăsarea butonului *Compact*.
4. Apare caseta de dialog *Compact Database Into* unde scriem numele sub care se va memora baza de date compactată și reparată. Confirmăm prin apăsarea butonului *Save*.

10.3.2. Crearea cópiilor de rezervă

Access poate repara complet sau parțial o bază de date deteriorată. În ultimul caz, datele care nu au fost restabilite pot fi recuperate dintr-o copie de rezervă a bazei de date.

Prin urmare, administratorul bazei de date va avea grijă să realizeze periodic astfel de copii.

Această acțiune poate fi făcută în mod tradițional (prin copierea fișierului bazei într-un alt dosar) sau executînd *Tools* → *Database Utilities* → *Back Up Database*.

10.3.3. Asigurarea securității datelor

O bază de date poate conține date confidențiale și poate fi destinată mai multor utilizatori, care vor avea acces la ea prin rețea (posibil prin Internet). În această situație se impun măsuri de prevenire a accesului neautorizat. Mai mult chiar, diferiți utilizatori pot

avea diferite drepturi de acces. De exemplu, unii pot avea dreptul să adauge informații, iar alții – doar să vadă unele informații.

Cu acest scop, unui sau mai multor utilizatori li se va atribui:

- un nume unic de acces;
- o parolă secretă;
- drepturi de acces sau de proprietar.

Cea mai simplă metodă de protejare a bazei de date de accesul neautorizat este stabilirea unei parole, în lipsa căreia baza de date nu va putea fi deschisă.

Pentru **crearea parolei**:

1. Închidem baza de date, lăsând deschis sistemul Access.
2. Executăm *File* → *Open*. Selectăm fișierul bazei de date, apoi opțiunea *Open Exclusive* din lista derulantă *Open*.
3. Executăm *Tools* → *Security* → *Set Database Password*. Apare caseta de dialog *Set Database Password* în care stabilim și confirmăm parola.

Observații:

1. Există și alte acțiuni care pot fi realizate de administratorul bazei de date pentru a asigura protejarea datelor. De exemplu, se poate face **criptarea** bazei, pentru ca datele să nu fie citite cu un editor de texte sau cu utilitare de disc fără a fi decriptate.
2. Crearea grupurilor, parolelor și a drepturilor de acces de asemenea se face apelând submeniul *Security* al meniului *Tools*.

Creăm, cercetăm și exersăm

În imagine (fig. 10.19) sînt reprezentate relațiile dintre tabelele bazei de date „Club sportiv” (cîmpul *Platit* este de tip logic).

1. Creați baza de date conform imaginii și completați-o cu înregistrări.
2. Explicați de ce cîmpurile **Cod_client**, **Cod_abonament** și **Luna** ale tabelului *Evidenta* formează împreună cheia primară a acestui tabel.
3. Creați o interogare care va afișa:
 - a) lista clienților al căror prenume începe cu litera A sau C;
 - b) lista antrenorilor al căror nume este format din cel puțin 5 litere;
 - c) salariul mediu al antrenorilor;
 - d) lista antrenorilor cu salariul mai mare decît cel mediu;
 - e) lista codurilor abonamentelor cu preț maxim;
 - f) numărul de clienți al fiecărui antrenor;

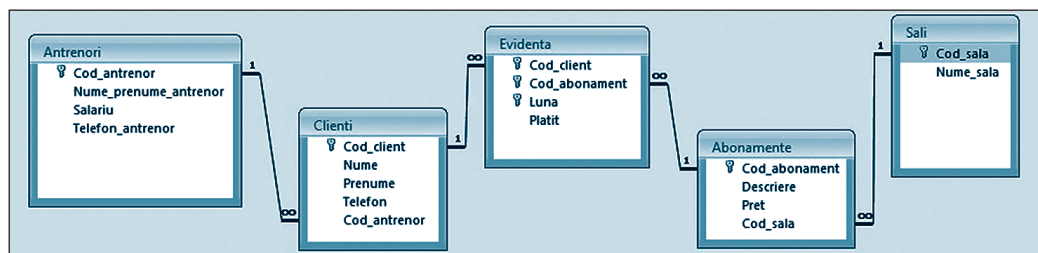


Fig. 10.19

- g) numele antrenorilor cu cei mai mulți clienți;
 - h) suma totală de bani, plătită de fiecare client;
 - i) numele clienților care au plătit cel mai mult (în total);
 - j) sălile frecventate cel mai des de clienți.
4. Formulați și creați 5 interogări.
 5. Elaborați formulare de introducere și editare a datelor în baza de date.

Creăm în echipă

1. Proiectați un model conceptual relațional al unei baze de date cu informații despre:
 - a) agenda elevului;
 - b) țările Europei;
 - c) automobile;
 - d) titluri de cărți;
 - e) pagini web.
2. Creați tabelele bazei de date proiectate.
3. Stabiliți cheile primare și relațiile dintre tabelele bazei de date.
4. Completați fiecare tabel cu cel puțin 5 înregistrări.
5. Formulați și realizați cel puțin:
 - a) 4 interogări de selecție, dintre care una cu parametru;
 - b) 3 interogări de modificare a unor înregistrări;
 - c) 2 interogări de grupare și totalizare;
 - d) o interogare încrucișată;
 - e) o interogare de excludere a unor înregistrări.
6. Creați două formulare pentru introducerea/modificarea înregistrărilor în tabelele bazei de date.
7. Creați cel puțin două rapoarte cu informații din baza de date.
8. Creați o diagramă care va reflecta datele din rapoarte.

Test de evaluare

1. Stabiliți valoarea de adevăr a propoziției:
 - a) Odată cu modificarea înregistrărilor în formular, datele se modifică și în tabelul subordonat.
 - b) Access salvează modificările imediat după completarea înregistrării chiar dacă formularul este închis fără salvare.
 - c) Un antet și un subsol de formular se elimină sau se deplasează doar în pereche.
2. Cum se poate modifica dimensiunea textului dintr-un element de control cu ajutorul barei de instrumente?
3. Creați cu ajutorul programului de asistență un formular pentru a edita adresele profesorilor bazei de date *Liceu*.
4. Creați prin proiectare individuală un formular cu numele *Test* și fundalul verde, care va conține trei casete de editare, respectiv, a numelui, prenumelui și a datei de naștere a elevilor din baza de date *Liceu*.
5. Elaborați un raport pentru baza de date *Liceu* care va afișa lista și numărul total de profesori de gen feminin la fiecare disciplină.
6. Creați o diagramă care ar reprezenta grafic datele din raportul creat în itemul 5.

Capitolul 11

DOCUMENTE WEB

După studierea acestui capitol, veți fi capabili să:

- recunoașteți documente WEB;
- proiectați un document Web;
- creați un document Web cu ajutorul aplicațiilor de oficiu.

11.1. Noțiuni și concepte

În cursul de Informatică pentru clasa a 10-a ați studiat deja mai multe noțiuni fundamentale din domeniul rețelelor de calculatoare. Cunoașteți elemente de tipologie și de arhitectură ale rețelelor de calculatoare, tehnologii de cooperare în rețea, rețeaua Internet, componentele și serviciile ei. Știți că informația și operațiile de gestionare a ei constituie cele mai importante elemente ale rețelei globale, iar cel mai modern serviciu care ajută să operați cu informația este serviciul WWW (World Wide Web – păienjenișul global).

Vă este cunoscută și noțiunea de pagină Web, precum și cea de program de răsfoire (explorare). Puteți localiza paginile Web folosind adresele URL (Uniform Resource Locator), vizualiza elementele lor informaționale, copia informațiile necesare, prezentate în diverse formate. Ați observat că paginile Web pot conține mai multe elemente, natura cărora este distinctă: text, imagini, secvențe sonore și video, elemente de control (butoane) și referințe către alte pagini Web. Toate acestea pot fi asamblate împreună datorită hipertextului și limbajului HTML (Hyper Text Markup Language – limbajul de marcare a hipertextului).

Un document Web este o colecție de pagini Web interreferite, împreună cu elementele lor informaționale și de control încorporate.

Documentul Web publicat în rețeaua globală este cunoscut sub denumirea de site Web.

11.2. Tipurile documentelor Web

Cele mai cunoscute și des utilizate documente Web sînt **documentele HTML** – documentele în care informațiile, legăturile, elementele de control și metadatele sînt descrise în limbajul HTML. Atunci cînd un program de explorare deschide un fișier HTML, el interpretează marcasele întîlnite în text în conformitate cu regulile limbajului. Documentele

HTML au extensia HTML (în unele cazuri HTM, dacă provin din sisteme de operare care acceptă doar extensii ale numelor de fișier din nu mai mult de trei simboluri). Un document HTML poate fi creat cu ajutorul oricărui editor de texte și vizualizat, folosind orice program de explorare (fig. 11.1).

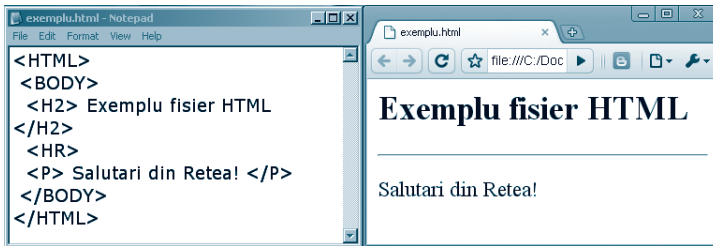


Fig. 11.1. Conținutul unui fișier HTML (în stînga) și interpretarea acestuia de programul de explorare (în dreapta)

Mulți dintre utilizatorii Web denumesc documentele descrise cu ajutorul limbajului HTML *documente în format HTML*.

Documentele PHP sînt paginile Web, care conțin elemente descrise cu ajutorul limbajului PHP (Hypertext Preprocessor). Utilizarea PHP permite crearea unor pagini cu un conținut dinamic: conținutul paginii PHP este prelucrat pe un server Web, după care se generează o pagină HTML, ce este transmisă pe calculatorul client, unde este vizualizată cu ajutorul programului de explorare. Documentele PHP se folosesc frecvent pentru afișarea dinamică a datelor stocate în bazele de date de rețea.

Documentele ASP sînt pagini Web elaborate cu ajutorul tehnologiei Active Server Page. La fel ca documentele PHP, acestea se bazează pe componente programate într-un limbaj de nivel înalt, numite obiecte care pot fi transmise de la o pagină la alta.

Documentele grafice SVG (Scalable Vector Graphics) sînt realizate cu ajutorul limbajului XML* pentru reproducerea calitativă a elementelor grafice bidimensionale. Instrumentele grafice încorporate în acest limbaj permit utilizarea graficii vectoriale în documente Web (fig. 11.2).

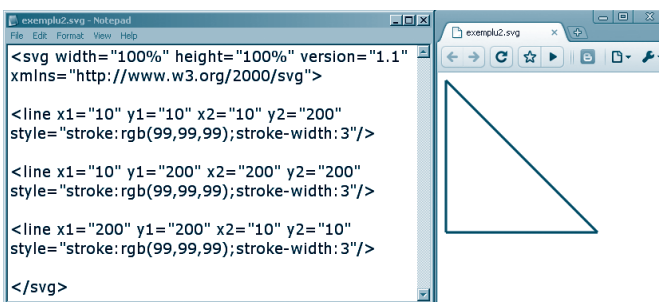


Fig. 11.2. Conținutul unui document Web grafic (în stînga) și interpretarea acestuia de programul de explorare (în dreapta)

* Meta-limbajul XML (*eXtensible Meta Language*) este un limbaj asemănător cu HTML, care a fost proiectat în scopul transferului de date între aplicații, inclusiv în rețea.

11.3. Proiectarea și realizarea unui document Web

Tradițional asociem termenul *document* unui fișier separat, care conține informații de diversă natură, dar din același domeniu. Documentul Web, sau *site-ul* este în majoritatea absolută a cazurilor o colecție de fișiere (numite și pagini) cu referințe interne și externe. În activitatea cotidiană veți avea necesitatea nu doar să utilizați documente Web, dar și să elaborați unele documente specifice. Instrumentele Web moderne permit automatizarea în mare măsură a elaborării unor pagini standard, cum ar fi paginile de *profil personal* în rețelele sociale, *blogurile*, paginile *wiki*. În cazurile când urmează să elaborați un document mai puțin standardizat, aceste instrumente vă pot ajuta doar în procesul de construcție a documentului. Fiind destinat unui public larg, documentul Web urmează să fie elaborat astfel, încât să acorde utilizatorilor săi un acces cât mai comod și rapid la informațiile structurate după criterii asociate tematicii documentului. Aceasta ne duce la concluzia că este necesară o *proiectare prealabilă* a documentului Web.

Indiferent de tipul documentului Web, proiectarea acestuia începe de la *stabilirea scopului* de utilizare. Având fixat scopul, putem selecta mai simplu tipul de document și instrumentarul necesar pentru elaborare. De exemplu, dacă doriți să faceți publică o colecție de eseuri proprii și ale colegilor veți alege în calitate de pagini documente HTML. Același tip veți utiliza pentru a crea o galerie nu prea mare de imagini. Dacă însă doriți să realizați un document cu descrieri ale figurilor geometrice plane, va fi oportună selectarea documentelor SVG, care va acorda o calitate și flexibilitate sporită a imaginilor din pagini.

Odată fiind stabilit scopul de utilizare, puteți trece la următoarea etapă – *stabilirea cerințelor* față de document. Există mai multe categorii de cerințe: față de conținut și dozarea lui în pagini, față de culori și caractere, față de navigare între pagini și elementele paginii. Cerințele sînt specifice pentru fiecare document în parte și se stabilesc în funcție de grupul-țintă pentru document și de conținuturile care urmează să fie plasate.

Web Design este procesul de creare a unei pagini sau a unui document Web, care îmbină modul de realizare a cerințelor etice, estetice, de conținut și operare mecanică cu aspectul vizual al documentului elaborat.

La următoarea etapă este definită *structura documentului Web*: structura paginii principale, conținutul ei, amplasarea elementelor informaționale și de navigație, informațiile despre autor; structura paginilor subordonate, conținuturile și modul de referire între paginile documentului. Pentru a evita amplasarea nereușită a elementelor în pagini și apariția referințelor incorecte, este utilă crearea unei scheme grafice a documentului. Schema va conține în calitate de noduri viitoarele pagini ale documentului, iar legăturile direcționate între noduri vor indica referințele între paginile documentului.

Exemplu: Fie că doriți să realizați un document Web cu informații de interes personal, care va conține: date personale, o galerie de imagini, pagini separate dedicate activității școlare, prietenilor, activităților de interes (hobby) și o colecție de referințe la site-uri cu conținuturi educaționale.

Pagina principală va conține un antet (titlu) al documentului, referințe către paginile subordonate, date de contact despre autor și un bloc de informații personale. Fiecare

dintre paginile subordonate va fi una tematică: liceu, prieteni, galerie foto, interese. Pagina cu lista de referințe este asociată procesului de studii și va fi subordonată paginii cu date despre liceu (fig. 11.3).

Se consideră a fi o bună practică posibilitatea de referire directă de pe orice pagină a documentului către pagina principală. Nu este necesar de specificat aceste referințe pentru a nu supraîncărca schema de structură a documentului Web.

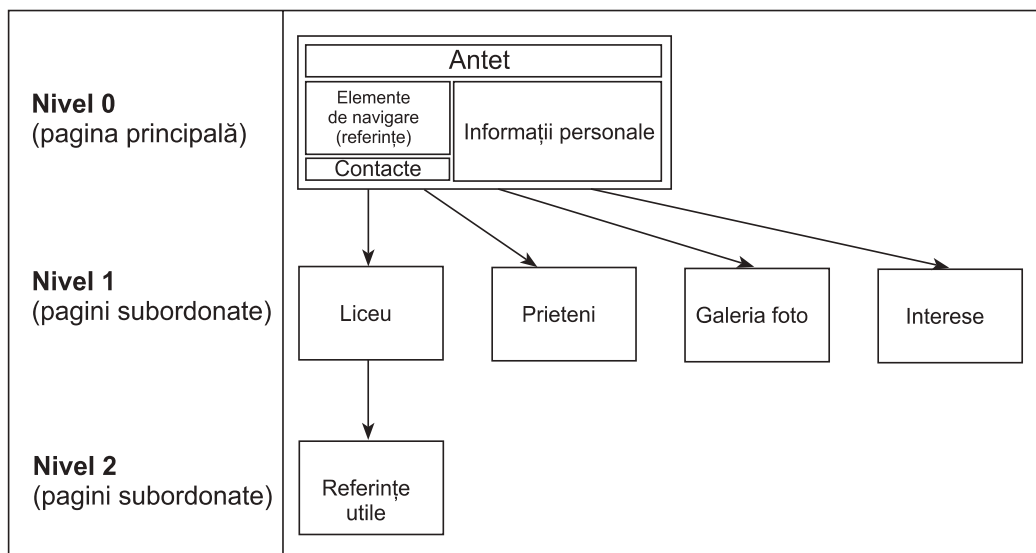


Fig. 11.3. Schema de structură a unui site cu date de interes personal. Este detaliată structura paginii principale

După proiectarea structurii referințelor în document, poate fi detaliată structura paginilor subordonate. Se recomandă utilizarea unui model comun pentru toate paginile, cu modificări minime, dictate de specificul conținutului.

Programarea paginilor documentului Web. Este următoarea etapă de elaborare, și în funcție de tehnica abordată de autorul documentului, poate să conțină nu doar procesul de programare a paginilor cu ajutorul unui limbaj de programare Web, dar și introducerea informației în pagini. Dacă se folosesc sisteme de producere și asistență Web (cum ar fi sites.google.com, wordpress.com), elementele de programare devin mai puțin vizibile, dar se pierde și o parte din libertatea de organizare a structurii și conținuturilor.

Testarea documentului Web. Se vizualizează documentul local, cu ajutorul diferitor aplicații de explorare, se stabilește corectitudinea afișării structurii, a conținuturilor, a culorilor și a referințelor. În cazul depistării erorilor, se revine la etapa de proiectare. După corectarea lor, documentul este gata de a fi publicat. Pentru aceasta el urmează să fie plasat pe un server Web. Dacă pentru elaborarea documentului au fost folosite sisteme on-line de producere și asistență Web, deplasarea pe serverul Web este opțională.

Menținerea și dezvoltarea documentului. Periodic, informația de pe paginile site-ului urmează să fie reînnoită. Pot să apară și modificări de structură, design etc. Pentru site-urile elaborate nemijlocit în mediul Web operațiile de editare se realizează direct, înlocuind dinamic versiunile vechi ale paginilor. Dacă însă site-ul a fost creat local și mai apoi publicat, versiunea nouă se testează, apoi se publică pe serverul Web, înlocuind site-ul vechi.

11.4. Crearea documentelor Web cu ajutorul aplicațiilor de oficiu

În activitatea cotidiană folosim diverse programe de aplicații pentru a prelucra informații: editoare de texte, editoare grafice, sisteme de prezentări electronice și calcul tabelar, alte aplicații specializate.

Fișierele create de aceste sisteme pentru stocarea informațiilor au formate specifice, care nu pot fi prelucrate și afișate direct de aplicațiile de explorare. Dar, de multe ori, informația din aceste fișiere urmează să fie plasată într-un document Web sau într-o pagină solitară. O soluție simplă, pe care o pun la dispoziție mai multe aplicații de uz general, este crearea automată a unui document Web, structura și conținutul căruia repetă fidel documentul original din aplicație. Aceasta se creează în urma selectării opțiunilor speciale de salvare a documentului, sau de exportare.

Tipurile standard ale documentelor Web care pot fi create în acest mod sînt **Document Web (HTML)**. Pentru a obține automat un document Web, este suficient să se selecteze în meniul aplicației opțiunea de salvare a documentului: *Web Page* (fig. 11.4).



Fig. 11.4. Salvarea fișierelor aplicațiilor de oficiu ca document Web

De remarcat că în cazul cînd documentul inițial conține informații neomogene sau elemente care nu pot fi reproduse direct cu ajutorul instrumentelor HTML, aplicațiile creează tot setul de fișiere subordonate, amplasate într-un director, numele căruia este format din numele documentului salvat, însoțit de sufixul **files** (fig. 11.5).

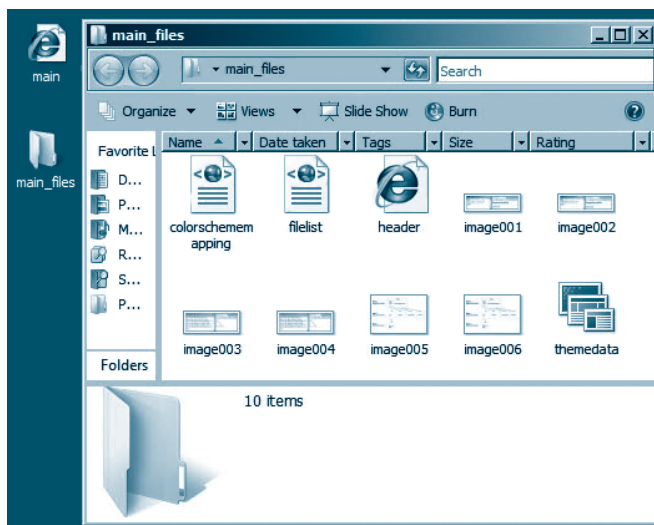


Fig. 11.5. La salvarea fișierului cu numele **main** în formă de pagină Web, s-a creat și directorul **main_files**, care conține mai multe fișiere cu imagini și informații de serviciu, necesare pentru reprezentarea corectă a paginii Web **main**

Mai multe aplicații de oficiu propun o opțiune suplimentară de generare a documentului Web: *Web Page, Filtered* (pagina Web filtrată). Deosebirea de pagina Web obișnuită este excluderea codurilor excesive și a informațiilor de serviciu, care ocupă un volum semnificativ. Această opțiune permite micșorarea volumului documentului Web, dar creează riscul de reprezentare incorectă în unele aplicații de explorare.

Extensia fișierelor care conțin documentul Web este în ambele cazuri HTML. Extensiile fișierelor auxiliare și de imagini pot varia în funcție de aplicația de oficiu, din care este generat documentul Web.

Un alt tip de documente Web care poate fi produs în mod automat cu ajutorul aplicațiilor de oficiu este **arhiva Web**. Pentru a crea o arhivă Web din fișierul cu format specific al aplicației, în meniu este selectată opțiunea de salvare a documentului: *Single File Web Page* (fig. 11.6).



Fig. 11.6

Acest tip de documente încorporează toate elementele documentului original, păstrează relațiile între componentele referite intern și poate fi deschis cu un program de explorare local, pe calculatorul pe care se află.

Extensia fișierelor care conțin arhive Web este MHT.

Întrebări și exerciții

- 1 Ce numim document Web? Care documente Web se numesc site-uri Web?
- 2 Care dintre următoarele tipuri de documente Web sînt realizate cu ajutorul unui limbaj de programare de nivel înalt:
 - a) documentele HTML;
 - b) documentele ASP;
 - c) documentele PHP;
 - d) documentele SVG?
- 3 Care este tipul de documente Web, ce permite vizualizarea textului formatat cu ajutorul unui limbaj de marcaje?
- 4 Care este tipul de documente Web, ce permite crearea directă a elementelor grafice în aplicația de explorare?
- 5 Enumerați etapele de proiectare a unui document Web.
- 6 Proiectați schema de structură a unui site – o microbibliotecă virtuală pentru cel mult trei autori. Pentru fiecare autor urmează să fie plasate în pagini aparte cel mult două opere. Pagina principală va conține lista de referințe către paginile autorilor distincți. Fiecare pagină a autorului distinct va conține referințe către paginile operelor lui literare.
- 7 Selectați un domeniu pentru care să proiectați un site Web. Determinați scopul de utilizare. Proiectați schema de structură.
- 8 Cu ajutorul unui editor de texte, creați un document care să conțină: text cu elemente de formatare, tabele, diagrame, imagini. Folosind opțiunile de salvare a documentului, generați automat un document Web, echivalent cu cel original. Care este diferența, dacă generați un document arhivă Web?
- 9 Creați o prezentare electronică, care să conțină text cu elemente de formatare, tabele, diagrame, imagini. Folosind opțiunile de salvare a documentului, generați automat un document Web, echivalent cu prezentarea originală. Cum credeți, care elemente ale unei prezentări nu pot fi păstrate în documentul Web?

Capitolul 12

LIMBAJUL HTML

După studierea acestui capitol, veți fi capabili să:

- elaborați pagini Web utilizând sintaxa HTML;
- formatați conținutul documentelor Web cu ajutorul instrumentelor HTML;
- alcătuiți și să organizați liste cu ajutorul instrumentelor HTML;
- creați și să utilizați în interiorul documentelor HTML referințe interne și externe;
- inserați în documente HTML imagini;
- creați și să editați tabelele în documente HTML;
- utilizați tabelele pentru amplasarea în pagină a elementelor HTML.

12.1. Structura generală a unui document HTML

12.1.1. Despre documente HTML

Am menționat deja că cele mai utilizate documente Web sînt documentele HTML. Cele mai apreciate caracteristici ale formatului HTML sînt: independența față de platformă (același document HTML este afișat asemănător de calculatoare diferite), referințele, formatarea structurată. Referințele simplifică esențial navigarea printr-un document mare și/sau prin mulțimea documentelor unui site. În calitate de referință poate servi un cuvînt, o frază, o imagine sau orice alt element al paginii Web.

Prima versiune HTML a fost lansată în 1989 (proiectată de Tim Berners-Lee*). Chiar dacă unele programe de explorare nu recunosc toate cuvintele-cheie ale limbajului HTML, acesta rămîne a fi unul dintre cele mai utilizate și mai moderne limbaje pentru elaborarea paginilor Web. Fișierele HTML sînt fișiere în format ASCII, deci pot fi create cu orice editor de texte. Fără a exagera, menționăm că proiectarea unei pagini Web simple se realizează (după cum veți vedea ulterior) foarte ușor.

Prin **element** al unui document HTML vom înțelege orice componentă a structurii documentului: tabel, paragraf, listă, titlu, buton, casetă de text, imagine etc.

Pentru a marca elementele unui fișier HTML, se utilizează diferite cuvinte-cheie, numite **etichete** sau **marcaje** (tags). Orice etichetă se încadrează între simbolurile < și >.

Etichetele, de regulă, sînt pereche pentru încadrarea unui element. Eticheta de sfîrșit arată ca și cea de început, mai avînd în fața ei caracterul / (slash). O etichetă de început poate avea unul sau mai multe atribute care se pot specifica în forma următoare:

<Etichetă $a_1 = v_1$, $a_2 = v_2$, ... $a_i = v_i$ >, unde v_i este valoarea atributului a_i .

De reținut că limbajul HTML nu face diferență dintre literele mari și mici.

* Sir Timothy Jon (Tim) Berners-Lee (n. 08.06.55) – programator englez, inventator al limbajului HTML și al WWW-ului, director al Consorțiului WWW (W3C), organizație care tutează standardele Webului.

12.1.2. Structura generală a unui document HTML

Un document HTML are, de regulă, următoarea structură:

<HTML>	Începutul documentului
<HEAD> <TITLE> Titlul documentului </TITLE> </HEAD>	Secțiunea de antet a documentului
<BODY> Conținutul documentului </BODY>	Secțiunea de conținut a documentului
</HTML>	Sfârșitul documentului

• Textul scris între marcasele <TITLE> și </TITLE> este afișat pe bara de titlu a ferestrei programului de explorare.

• Etichetele <BODY> și </BODY> cuprind conținutul propriu-zis al documentului HTML, care va fi afișat în fereastra programului de explorare și va fi văzut de utilizator.

Menționăm câteva dintre atributele etichetei <BODY>:

- $bgcolor = \#n_1n_2n_3n_4n_5n_6$, unde $\bar{n}_1\bar{n}_2, \bar{n}_3\bar{n}_4, \bar{n}_5\bar{n}_6$ sînt numere hexazecimale care determină intensitatea culorilor roșu, verde, respectiv, albastru ale fundalului paginii. Culoarea poate fi precizată și prin numele ei (red, blue, black, yellow, green, cyan, purple, white, gray etc.);
- $text = \#n_1n_2n_3n_4n_5n_6$ specifică culoarea textului;
- $background = \text{URL}$, unde URL este adresa și numele fișierului grafic a cărui imagine va fi utilizată ca fundal al documentului;
- $leftmargin = n$, unde n este număr natural, stabilește distanța în pixeli dintre marginea stîngă a ferestrei programului de explorare și marginea stîngă a conținutului paginii;
- $topmargin = n$, unde n este număr natural, stabilește distanța în pixeli dintre marginea de sus a ferestrei programului de explorare și marginea de sus a conținutului paginii.

Exemplu: Următorul fișier HTML interpretat de un program de explorare va afișa o pagină Web cu fundalul de culoare *aqua* și textul de culoarea #FF2233.

```
<HTML>  
<Head> <Title> Exemplu </Title> </Head>  
<Body leftmargin="140" topmargin="200" bgcolor="aqua" text="#FF2233">  
Salut! Acesta este un document .html  
</Body>  
</HTML>
```

Întrebări și exerciții

- 1 Explicați semnificația referințelor într-un document HTML.
- 2 Ce editor de texte poate fi folosit pentru crearea unui fișier HTML?
- 3 Ce înțelegeți prin element al unui document HTML?
- 4 Ce semnificație au etichetele într-un document HTML?
- 5 Ce simboluri se utilizează pentru a pune în evidență o etichetă?

- ⑥ Care este deosebirea dintre eticheta de început și cea de sfârșit?
- ⑦ Care este structura generală a unui document HTML?
- ⑧ Cum se utilizează etichetele <TITLE> și </TITLE> ?
- ⑨ La ce folosesc etichetele <BODY> și </BODY> și care sînt atributele lor?
- ⑩ Creați o pagină Web cu fundalul galben și textul verde, conținutul paginii fiind afișat la 110 pixeli de la marginea stîngă și 105 pixeli de la marginea de sus ale ferestrei programului de explorare.
- ⑪ Creați o pagină Web cu fundalul albastru și textul alb, conținutul paginii fiind afișat la 10 pixeli de la marginea stîngă și 15 pixeli de la marginea de sus ale ferestrei programului de explorare.
- ⑫ Ce va afișa documentul HTML cu următorul conținut?

```
<HTML>
<Head> Ce este un fisier .html? <Title> PROBLEMA </Title> </Head>
<Body>
Care este structura unui document .html?
</Body>
</HTML>
```

- ⑬ Creați o pagină Web care va afișa pe bara de titlu a ferestrei programului de explorare textul UN DOCUMENT HTML și va avea în calitate de fundal o imagine.

12.2. Formatarea textului

- În HTML se utilizează 6 **niveluri de titlu**, care pot fi specificate cu ajutorul etichetelor <Hn> și </Hn>, unde n poate lua valori de la 1 la 6. Fiecare dintre aceste perechi de etichete stabilesc un anumit nivel de titlu. Nivelul 1 este cel mai proeminent, iar nivelului 6 îi corespund cele mai mici și mai subțiri caractere.

Etichetele de titlu pot avea atributul *align* = "tip", unde *tip* este unul dintre cuvintele *center*, *left*, *right*, pentru stabilirea modului de aliniere a titlului.

- Spre deosebire de un editor de texte obișnuit, HTML nu ține cont de lungimea textului, de returnul de car (Enter) și de numărul consecutiv de spații. Acestea din urmă vor fi afișate ca un singur spațiu. Programul de explorare afișează din rînd nou un text, dacă acesta este precedat de unul din marcajele <P>,
, <PRE>, <DIV> sau <CENTER>.

- a) Astfel, textul cuprins între etichetele <P> și </P> se consideră **paragraf**. Eticheta <P> de asemenea acceptă atributul *align*.
- b) Eticheta
 realizează **trecerea la o linie nouă** în paragraful curent.
- c) Textul încadrat între etichetele <PRE>, </PRE> este afișat exact așa cum este scris, deci este un **text preformatat**.
- d) Deseori un **bloc de text** se formează cu ajutorul etichetelor <DIV>, </DIV>. Eticheta <DIV> poate avea atributele *align* și *nowrap* (care interzice întreruperea rîndurilor de programul de explorare). Un bloc <DIV>...</DIV> poate include alte subblocuri. În acest caz, alinierea precizată de atributul *align* al blocului are efect asupra tuturor subblocurilor incluse în blocul <DIV>.
- e) Elementele cuprinse între etichetele <CENTER> și </CENTER> sînt automat centrate.

- Blocul de text cuprins între etichetele <NOBR> și </NOBR> va fi afișat într-un rînd. Dacă fereastra programului de explorare va fi micșorată de utilizator, se poate întîmpla ca textul să fie trunchiat.

• **Linile orizontale** sînt folosite pentru divizarea unei pagini în secțiuni logice. În funcție de necesitate, eticheta <HR>, care inserează o linie orizontală, poate include câteva atribute: *align* (stabilește alinierea liniei); *size* (specifică înălțimea liniei în pixeli); *width* (specifică lățimea liniei în pixeli sau în procente față de lățimea ferestrei programului de explorare); *color* (specifică culoarea liniei); *noshade* (specifică o linie fără nicio evidențiere).

• **Fontul** unui text poate fi precizat utilizînd marcajele și . El poate fi caracterizat de următoarele atribute: *face* (precizează familia de caractere și poate avea ca valoare una sau mai multe constante delimitate prin virgule, cum ar fi *arial*, *serif*, *cursive*, *monospace*, *fantasy* etc.); *color* (precizează culoarea fontului); *size* (specifică mărimea caracterelor și poate avea ca valoare constantele 1, 2, ..., 7 sau +1, +2, ... (mărind astfel dimensiunea caracterelor față de dimensiunea curentă), sau -1, -2, ... (micșorînd astfel dimensiunile caracterelor față de dimensiunea curentă)); *weight* (determină grosimea caracterelor, avînd valori posibile 100, 200, ..., 900).

Exemplul 1 (fig. 12.1):

```
<HTML>
<Head>
  <Title> Text preformatat </Title>
  <H1 align="center"> Chisinau</H1>
</Head>
<Body leftmargin="20" topmargin="20" bgcolor="lightgreen" text="blue">
  <HR size="5" width="70%" color="black">
  <Font color="purple">
  <H2>Sectoarele Chisinaului:</H2>
  <H3>
  <Pre>
  Centru
    Buiucani
      Riscani
        Botanica
          Ciocana
  </Pre>
  </H3>
  </Font>
</Body>
</HTML>
```

• Pentru a pune în evidență un fragment de text, se utilizează **stilurile**. Acestea se împart în două categorii:

- stilurile fizice, care stabilesc înfățișarea exactă a textului;
- stilurile logice, care marchează textul în funcție de semnificația lui.

Stilurile fizice afișează același text la fel în orice program de navigare. Pentru stabilirea stilului fizic se utilizează etichetele:

- , – text cu caractere îngroșate (bold);
- <I>, </I> – text cu caractere înclinate (italic);
- <U>, </U> – text cu caractere subliniate;

<TT>, </TT> – text cu caractere monospațiate;
 <BIG>, </BIG> – text cu caractere mai mari cu o unitate decît cele curente;
 <SMALL>, </SMALL> – text cu caractere mai mici cu o unitate decît cele curente;
 _, – text cu caractere aliniate ca indice;
 [,] – text cu caractere aliniate ca exponent;
 <S>, </S> – text cu caractere tăiate de o linie.

Etichetele pentru stilurile fizice pot fi imbricate.

Exemplul 2 (fig. 12.2):

```
<HTML>
<Head>
<Title> Exemplul 2 </Title>
<H2 align="center"> Stiluri HTML</H2>
</Head>
<Body>
<H3>Stiluri fizice:</H3>
<Pre>
  Text obisnuit
  <B>Text cu caractere ingrosate</B>
  <I>Text cu caractere inclinate</I>
  <U>Text cu caractere subliniate</U>
  <TT>Text cu caractere monospațiate</TT>
  <big>Text cu caractere marite</big>
  <small>Text cu caractere micorate</small>
  Y = X<sup>2</sup>
  M = (m<sub>1</sub>, m<sub>2</sub>)
  <s>Text taiat</s>
  <B><I>Text cu caractere ingrosate si inclinate</I></B>
</Pre>
</Body>
</HTML>
```



Fig. 12.1

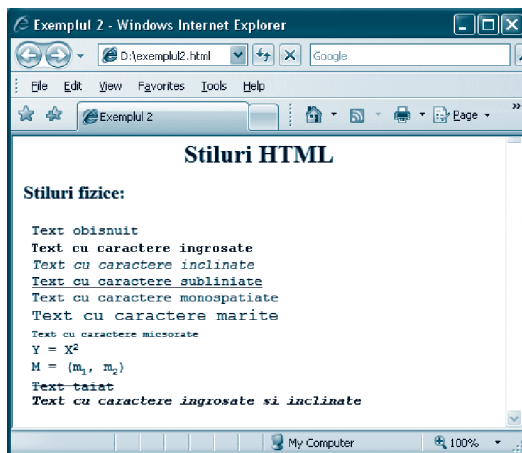


Fig. 12.2

Stilurile logice depind de modul de configurare a programului de navigare și țin cont de rolul textului în cadrul paginii Web. Pentru ele se utilizează următoarele etichete:

- , – text evidențiat (de regulă, italic);
- , – text important (de regulă, bold);
- <KBD>, </KBD> – text cules de la tastatură (de regulă, font cu grosime fixă);
- <CODE>, </CODE> – text de program pentru calculator (de regulă, font cu grosime fixă);
- <DFN>, </DFN> – text-definiție (de regulă, italic);
- <VAR>, </VAR> – identificator de variabilă;
- <ADDRESS>, </ADDRESS> – text-adresă;
- <CITE>, </CITE> – text-citată.

• Una dintre modalitățile de **incluere a semnelor diacritice române** într-un document HTML este codificarea directă a lor prin &-consecutivități (escape-consecutivități). Unele programe de explorare recunosc o &-consecutivitate dacă sfârșitul ei este marcat cu simbolul ;

Litera	Ă	ă	Â	â	Î	î	Ș	ș	Ț	ț
Codul	Ă	ă	Â	â	Î	î	Ş	ş	Ţ	ţ

De asemenea, &-consecutivitățile se folosesc pentru reprezentarea diferitor simboluri speciale:

Simbolul	&	<	>	“	*	©	spațiu	§
Codul	&	<	>	"	®	©	 	§

Simbolul	±	×		1/2	1/4	3/4	°
Codul	±	×	¦	½	¼	¾	°

Exemplul 3 (fig. 12.3):

```
<HTML>
<Head><Title> Exemplul 3 </Title><Body>
<H4>
Acest capitol v&#259 &icirc;nva&#355&#259 s&#259 crea&#355i pagini Web
</H4>
</Body>
</HTML>
```

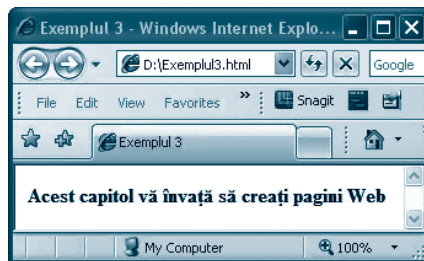


Fig. 12.3

Întrebări și exerciții

- 1 Care etichetă indică un nivel de titlu?
- 2 Câte niveluri de titlu se pot utiliza în HTML?
- 3 Care sînt atributele etichetei de titlu?
- 4 Care etichete se utilizează pentru a delimita un paragraf?
- 5 Ce etichete se folosesc pentru trecerea în rînd nou?
- 6 Cum se inserează o linie orizontală?
- 7 Ce etichete se folosesc pentru specificarea fontului?
- 8 Numiți atributele etichetei .
- 9 Ce sînt stilurile și la ce folosesc?
- 10 Care este deosebirea dintre stilurile logice și cele fizice?
- 11 Numiți etichetele pentru punerea în evidență a stilurilor: a) logice; b) fizice.
- 12 Creați o pagină Web pentru un magazin, astfel încît denumirea magazinului să apară ca un titlu de nivelul 1 avînd culoare neagră. Mai jos se vor afișa produsele unui stoc. Denumirea stocului va avea nivelul 3, iar produsele lui vor fi prezentate ca un text preformatat. Denumirea magazinului va fi delimitată de stoc printr-o o linie orizontală de grosimea 4 și de culoare albastră, aliniată la stînga și ocupînd 80% din lățimea ferestrei programului de explorare.
- 13 Creați o pagină Web care va afișa două teoreme de congruență a triunghiurilor. Cuvîntul *Teorema* va fi scris ca un titlu de nivelul 3, avînd culoarea neagră, caractere îngroșate și subliniate. Cele două enunțuri vor fi înclinate și delimitate unul de altul cu o linie roșie orizontală de grosimea 2, aliniată la centru, avînd lungimea 70% din lățimea ferestrei programului de explorare.
- 14 Creați o pagină Web care va afișa în centrul paginii formula:
 - a) $(a + b)^2 = a^2 + 2ab + b^2$;
 - b) $(a_1 + a_2 + a_3)^2 = a_1^2 + a_2^2 + a_3^2 + 2a_1a_2 + 2a_1a_3 + 2a_2a_3$;
 - c) $|a + b| < |a| + |b|$, pentru orice numere nenule a și b .
- 15 Creați un document HTML care va afișa următoarea pagină Web (fig. 12.4 a, b):

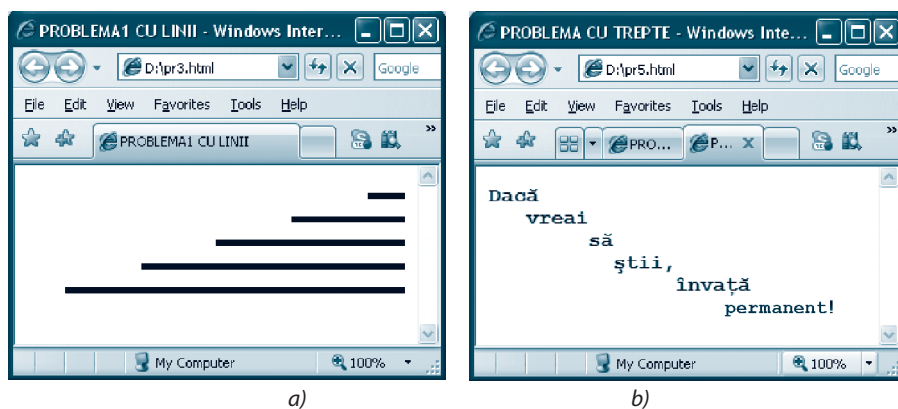


Fig. 12.4

- 16 Creați o pagină care va afișa o poezie cu titlul de culoare verde, centrat, pe fundal galben, cu nivelul de titlu 3, despărțit de restul textului cu o linie de grosimea 5 și lungimea 50% din lățimea ferestrei programului de explorare. Caracterele textului vor fi îngroșate și înclinate.

12.3. Liste

Pentru organizarea structurată a informației unui document HTML se pot folosi listele. Deosebim:

- a) *liste neordonate*;
- b) *liste ordonate*;
- c) *liste de definiții*.

Listele neordonate se utilizează pentru a indica o succesiune fără o subordonare ierarhică.

- Pentru a specifica începutul și sfârșitul unei astfel de liste, se utilizează etichetele `` și `` (*unordered list* – „listă neordonată”).
- Orice element al listei este precedat de eticheta `` (*list item* – „element al listei”).

Fiecărei dintre etichetele ``, `` i se poate atașa atributul *type*, care poate lua una dintre următoarele trei valori: *Circle*, *Square*, *Disc*. Acest atribut specifică marcajul care precede fiecare element al listei.

Listele ordonate (se mai numesc *liste numerotate*) se încadrează între etichetele `` și `` (*ordered list* – „listă ordonată”).

Ca și în cazul listei neordonate, orice element al listei ordonate este precedat de eticheta ``. Atributul *type* al etichetei `` poate lua una din valorile: *A*, *a*, *I*, *i*, *1*.

Dacă atributul *type* nu este specificat, elementele listei vor fi numerotate cu cifre arabe.

Listele de definiții se utilizează, de regulă, pentru organizarea glosarelor. Termenii unui glosar sînt listați în ordine alfabetică, fiecare fiind urmat de definiția lui.

- Listele de definiții se încadrează între etichetele `<DL>` și `</DL>` (*definition list* – „listă de definiții”).
- Fiecare termen al listei este precedat de eticheta `<DT>` (*definition term* – „termen definit”), iar definiția (descrierea) lui – de eticheta `<DD>` (*definition description* – „descrierea definiției”).

Observații:

1. Listele pot fi imbricate.
2. În interiorul unei liste sînt permise utilizarea etichetelor `<P>` și `
` și a elementelor de formatare a textului.
3. O listă ordonată poate fi întreruptă pentru un text. În acest caz se folosesc două seturi de etichete ``: una pentru prima parte a listei (pînă la textul inserat) și alta pentru partea a doua (după text). În acest caz, etichetei `` din partea a doua a listei i se atașează atributul *start* cu valoarea întregă, care indică începutul numerotării.

Exemplul 1 (fig. 12.5):

```
<HTML>
<Head> <Title> Liste neordonate</Title> </Head>
<Body>
  <H4> Unele domenii ale matematicii</H4>
  <UL>
    <LI> Algebra
    <LI> Geometria
    <LI> Teoria ecuatiilor diferentiale
```

```

<LI> Analiza matematica
</UL>
<H4> Unele domenii ale informaticii</H4>
<UL TYPE=square>
<LI> Teoria algoritmilor
<LI> Programarea Web
<LI> Teoria bazelor de date
<LI> Criptografia
</UL>
</Body>
</HTML>

```

Exemplul 2 (fig. 12.6):

```

<HTML>
<Head> <Title> Liste ordonate</Title> </Head>
<Body>
  <H4> Unele discipline studiate in clasa a X-a </H4>
  <OL>
  <LI> Limba romana
  <LI> Matematica
  <LI> Informatica
  </OL>
  <H4> Unele discipline studiate
  in clasa a XII-a</H4>
  <OL TYPE=A>
  <LI> Informatica
  <LI> Biologia
  <LI> Matematica
  </OL>
</Body>
</HTML>

```

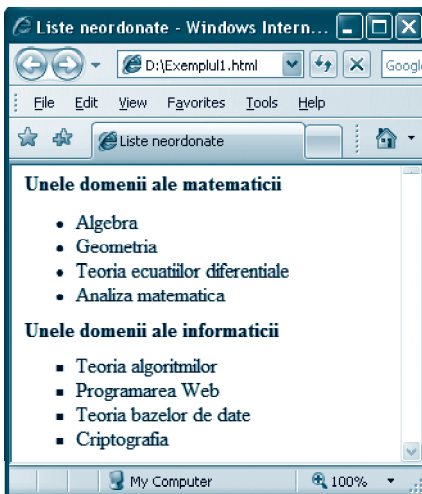


Fig. 12.5

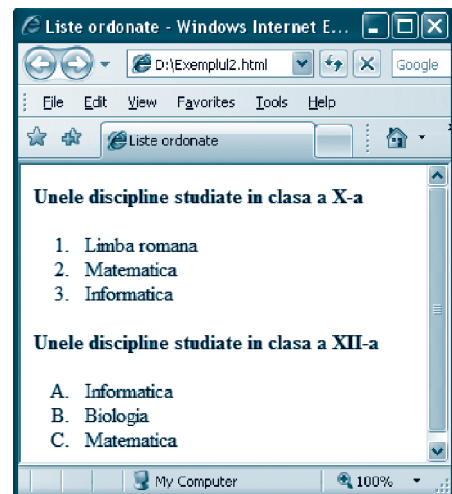


Fig. 12.6

Exemplul 3 (fig. 12.7):

```
<HTML>
<Head> <Title> Lista de definitii</Title> </Head>
<Body>
  <H4> Citeva notiuni din informatica </H4>
  <DL>
    <DT> Constanta <DD> - marime a carei valoare nu se poate
    modifica pe parcursul executiei algoritmului
    <DT> Variabila <DD> - marime a carei valoare se poate modifica pe
    parcursul executiei algoritmului
    <DT> Identificator <DD> - secventa de caractere care incepe (de regula) cu
    o litera, ce poate fi urmata de nici una sau de mai multe litere sau cifre,
    utilizata pentru a denumi constante, variabile, tipuri, subprograme.
  </DL>
</Body>
</HTML>
```

Întrebări și exerciții

- 1 Cu ce scop se folosesc listele?
- 2 Ce tipuri de liste cunoașteți?
- 3 Numiți etichetele utilizate pentru fiecare tip de listă.
- 4 Cum se specifică un element al unei liste?
- 5 Care sînt atributele acestor etichete și ce valori pot primi acestea?
- 6 Creați o pagină Web care va afișa titlul „Disciplinele preferate” pe fundal *bleu*, de nivelul 4, după care va urma o listă neordonată cu disciplinele respective. În calitate de marcaj se va folosi *pătratul*.
- 7 Creați o pagină Web care va afișa titlul „Disciplinele studiate în primul semestru al clasei a 12-a” pe fundal *bleu*, de nivelul 3, după care va urma o listă neordonată cu aceste discipline. În calitate de marcaj se va folosi *discul*. După listă vor fi 2 rînduri goale, apoi va apărea o altă listă neordonată cu disciplinele la care urmează să susțineți examenul de bacalaureat.
- 8 Creați o pagină Web care va afișa titlul „Lista elevilor clasei a 12-a” pe fundal de culoare *aqua*, de nivelul 5, aliniat la centru, după care va urma o listă neordonată cu numele elevilor clasei voastre. După listă vor fi 2 rînduri goale și va apărea altă listă neordonată cu numele elevilor care au absolvit clasa a 11-a cu media generală mai mare decît 8.
- 9 Creați o pagină Web care va afișa titlul „CUPRINS” de nivelul 3, după care vor urma capitolele unei cărți și temele cuprinse în aceste capitole. Folosiți liste ordonate și liste imbricate.

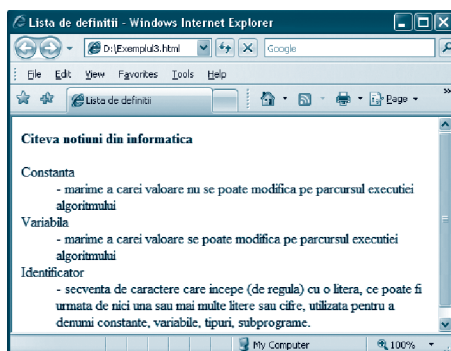


Fig. 12.7

- 10) Creați o pagină Web care va afișa 2 liste ordonate: prima cu titlul „Produse alimentare” de nivelul 2, în care elementele vor fi numerotate, iar cealaltă cu titlul „Produse industriale” de nivelul 3, în care elementele vor fi marcate cu litere.
- 11) Creați o pagină Web care va afișa o listă ordonată a unor produse culinare. Lista va fi întreruptă de următorul text cursiv: „Aceste produse culinare sînt pe placul meu, acum urmează produsele culinare care nu-mi plac:”. Titlul listei va avea nivelul 4 și culoarea albastră.
- 12) Creați un document HTML care va afișa pagina Web din figura 12.8.
- 13) Creați o pagină Web care va afișa definițiile a 5 noțiuni geometrice precedate de titlul „Noțiuni geometrice” de nivelul 3 avînd culoare gri, aliniat la dreapta și despărțit de noțiuni printr-o linie orizontală neagră cu înălțimea 3, aliniată la centru.

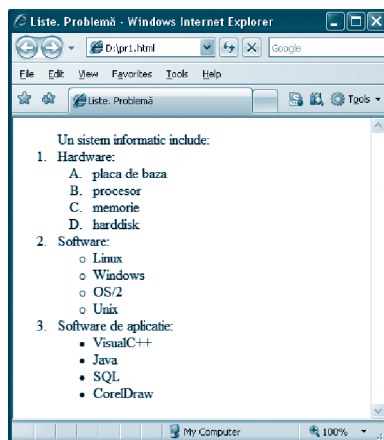


Fig. 12.8

12.4. Referințe

În paragraful 12.1 am menționat că referințele (se mai numesc *linkuri*, *legături*, *hiperlegături*) reprezintă, probabil, cea mai importantă caracteristică a limbajului HTML. Prin intermediul unei referințe se poate ajunge rapid către altă secvență a documentului curent sau către alt document (pe același sau pe alt server). Referințele sînt *zone active* ale paginii Web, în sensul că la executarea unui clic pe ele, programul de explorare va actualiza pagina.

• Pentru a pune în evidență o referință, se folosesc etichetele `<A>` (de la „ancoră”) și ``. Atributul *href* al etichetei `<A>` este obligatoriu. Valoarea acestui atribut este numele fișierului HTML cu care se face legătura. Numele acestui fișier se scrie între simbolurile „ ” și „ ”. Textul sau imaginea cuprinsă între etichetele `<A>` și `` devine zonă activă.

Astfel, o referință se declară în modul următor:

` text sau imagine`,

unde URL (*Uniform Resource Locator* – identificator unic al resursei) este numele fișierului-destinație.

Observație: În acest paragraf vom examina doar referințe, în care zona activă este un text. În general o referință poate:

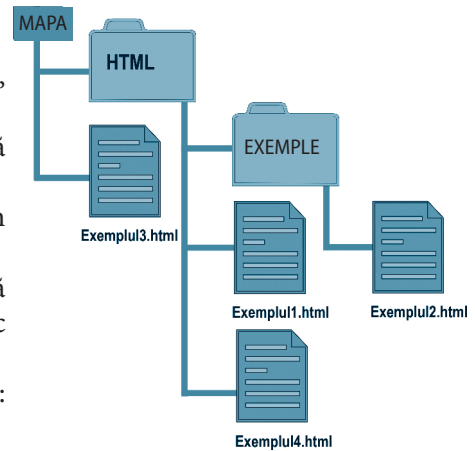
- lega documentul curent cu un document aflat în același director;
 - lega documentul curent cu un document aflat pe același disc local;
 - lega documentul curent cu un alt site;
 - lega documentul curent cu o altă secvență a aceluiași document;
 - lega documentul curent cu o secvență a altui document;
 - lansa la execuție o aplicație de expediere a mesajelor electronice;
 - lega documentul curent cu un fișier de format arbitrar cu scopul creării unei copii pe disc a acestui fișier.
- a) Dacă documentul referit se află în același director, atunci URL-ul este chiar numele acestui document.
- b) Dacă documentul referit se află pe același disc local, dar într-un alt director, atunci se utilizează adresarea relativă.

c) Dacă documentul referit reprezintă un alt site, atunci se indică adresa acestui site, scrisă între simbolurile " și " .

Exemplul 1 (fig. 12.9):

Prezentăm conținutul fișierului *exemplul4.html*, care conține legături către:

- fișierul *exemplul1.html* (fișierul referit se află în același director);
- fișierul *exemplul2.html* (fișierul referit se află în subdirectorul EXEMPLE al directorului curent);
- fișierul *exemplul3.html* (fișierul referit se află în directorul MAPA, deci pe un nivel ierarhic imediat superior);
- site-ul Universității de Stat din Tiraspol: <http://www.ust.md>.



```
<HTML>
<Head> <Title> Referinte</Title> </Head>
<Body>
  <H4> <a href = "exemplul1.html"> Treci la exemplul 1 </a>
  <p> <a href = "EXEMPLE/exemplul2.html"> Treci la exemplul 2 </a> </p>
  <p> <a href = "../exemplul3.html"> Treci la exemplul 3 </a> </p>
  <p> <a href = "http://www.ust.md"> Universitatea de Stat din
  Tiraspol </a> </p>
</H4>
</Body>
</HTML>
```

d) O referință către o altă secvență a aceluiași document se numește *ancoră cu nume*. În acest caz URL-ul este numele atribuit secvenței, precedat de simbolul #. Pentru a atribui unei secvențe un nume, se va scrie o ancoră de genul:

` `, unde numele secvenței se scrie între simbolurile " și " .

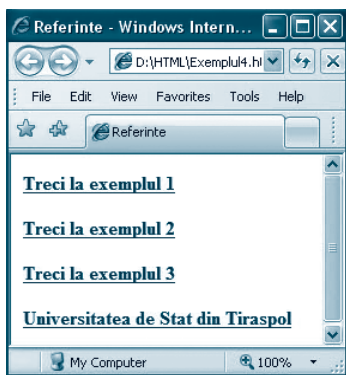


Fig. 12.9

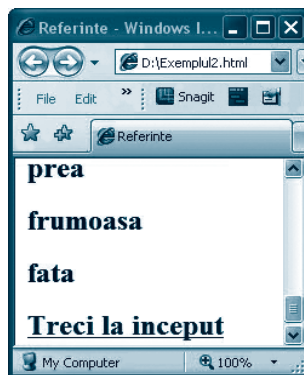


Fig. 12.10

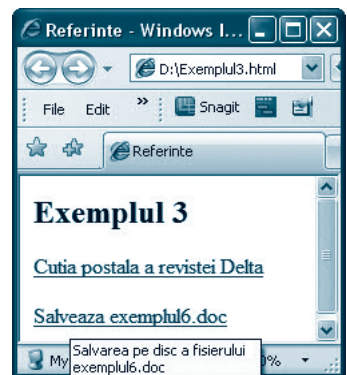


Fig. 12.11

e) URL-ul către o secvență a altui document este format din numele fișierului și numele secvenței, ultimul fiind precedat de simbolul #.

De exemplu, URL-ul secvenței cu numele *capitolul2* al fișierului *exemplul1.html* din același catalog este "exemplul1.html#capitolul2".

Exemplul 2 (fig. 12.10, vezi p. 133):

```
<HTML>
<Head> <Title> Referinte</Title> </Head>
<Body>
<H2>
<a name="inceput"></a>
  <P>A</P> <P>fost</P> <P>odata</P> <P>ca-n</P> <P>povesti</P>
  <P>A</P> <P>fost</P> <P>ca</P> <P>niciodata</P>
  <P>Din</P> <P>rude</P> <P>mari</P> <P>imparatesti</P>
  <P>O</P> <P>prea</P> <P>frumoasa</P> <P>fata</P>
  <a href="#inceput"> Treci la inceput </a>
</H2>
</Body>
</HTML>
```

f) Pentru lansarea la execuție a unei aplicații de expediere a mesajelor electronice (se presupune că această aplicație este instalată pe calculatorul utilizatorului) se va scrie o ancoră de forma:

``.

g) Pentru a crea o referință către un fișier de format arbitrar cu scopul creării unei copii pe disc a acestui fișier sau cu scopul lansării la execuție a aplicației care va interpreta astfel de fișiere, se va utiliza o ancoră de forma:

` text `,

unde URL este numele fișierului, iar *text* este zona activă.

Observație: Etichetei `<A>` i se poate atașa atributul *title*, care se folosește pentru a furniza informații suplimentare despre semnificația legăturii, atunci când mouse-ul este plasat asupra ei. Valoarea atributului *title* se scrie între simbolurile " și " .

Exemplul 3 (fig. 12.11, vezi p. 133):

```
<HTML>
<Head> <Title> Referinte</Title>
  <H2>Exemplul 3</H2>
</Head>
<Body>
<a href=mailto: delta_mi@mail.md> Cutia postala a revistei Delta </a>
<p><a href="exemplul6.doc" title="Salvarea pe disc a fisierului
exemplul6.doc " > Salveaza exemplul.doc </p> </a>
</Body>
</HTML>
```

Executând un clic pe referința Salveaza exemplul6.doc, va apărea fereastra din figura 12.12, care ne permite să creăm pe disc o copie a fișierului exemplul6.doc sau să deschidem acest fișier în aplicația Microsoft Word.

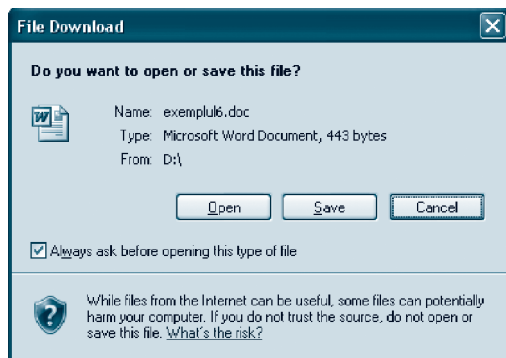


Fig. 12.12

Întrebări și exerciții

- 1 Ce semnificație au referințele pentru o pagină Web?
- 2 Care sînt etichetele folosite pentru declararea unei referințe?
- 3 Numiți atributul obligatoriu pentru eticheta <A>. Ce valoare poate avea acest atribut?
- 4 Ce este o zonă activă?
- 5 Care este forma generală de declarare a unei referințe?
- 6 Care elemente ale unei pagini Web pot fi referințe?
- 7 Unde se poate afla documentul referit?
- 8 Ce este o ancoră cu nume?
- 9 Cum se atribuie nume unei secvențe de document HTML?
- 10 Explicați în ce caz simbolul # se folosește la scrierea URL-ului.
- 11 Cum se creează o legătură către o secvență din același document?
- 12 Din ce este format URL-ul unei secvențe din alt document?
- 13 Cum dintr-un document HTML se poate lansa la execuție o aplicație de expediere a mesajelor electronice?
- 14 Creați o pagină Web care va afișa o listă cu disciplinele studiate. Fiecare element al listei va fi o referință către un fișier (aflat în același director ca și pagina Web) cu informația despre disciplina respectivă.
- 15 Creați o pagină Web care va afișa o listă ordonată cu denumirea unor universități din Republica Moldova. Fiecare element al listei va fi o referință către site-ul universității respective.
- 16 Creați o pagină Web care va afișa o listă de licee. Fiecare element al listei va fi o referință către fișierul (site-ul) liceului respectiv. Pagina creată se va afla în directorul LICEE, iar fiecare fișier referit – în subdirectorul cu numele liceului. Toate subdirectoarele se vor afla în directorul LICEE.
- 17 Examinați desenul exemplului 1. Care este URL-ul referinței către fișierul:
 - a) exemplul1.html, dacă referința se află în fișierul exemplul3.html;
 - b) exemplul2.html, dacă referința se află în fișierul exemplul1.html;
 - c) exemplul4.html, dacă referința se află în fișierul exemplul2.html;
 - d) exemplul3.html, dacă referința se află în fișierul exemplul2.html?
- 18 Creați o pagină Web care va afișa o listă cu câteva titluri de cărți. Fiecare titlu va fi o referință către o scurtă prezentare în același document HTML a cărții cu acest titlu.

19 Examinați desenul. Care este URL-ul referinței către fișierul:

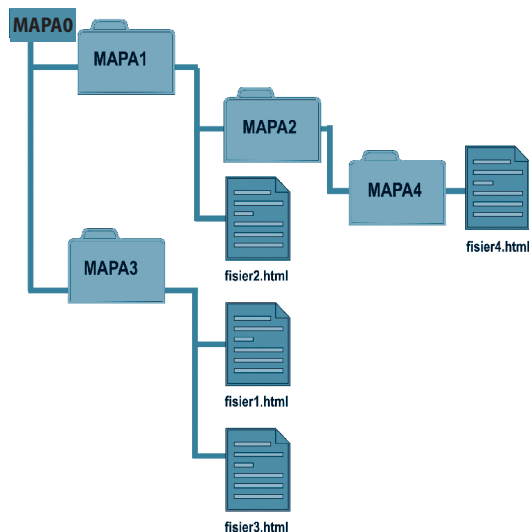
- a) f1.html, dacă referința se află în f4.html;
- b) f2.html, dacă referința se află în f3.html;
- c) f4.html, dacă referința se află în f2.html;
- d) f4.html, dacă referința se află în f3.html?

20 Examinați desenul. În ce director se află documentul HTML care conține referința dacă URL-ul ei este:

- a) "Mapa4/fisier4.html";
- b) "Mapa1/Mapa2/Mapa4/fisier4.html";
- c) "../Mapa1/Mapa2/Mapa4/fisier4.html";
- d) "../ ../Mapa3/fisier3.html"?

21 Creați două pagini Web, astfel încât prima va conține câteva referințe către diferite secvențe ale paginii a doua, iar pagina a doua – câteva referințe către diferite secvențe ale primei pagini.

22 Creați o pagină Web care va afișa o listă cu referințe către fișiere de diferite tipuri pentru a fi deschise spre vizionare sau stocare.



12.5. Imagini

În afară de faptul că imaginile furnizează informații, ele pot conferi unei paginii Web atractivitate. Totuși utilizarea lor într-un document HTML se va face rațional, deoarece fiecare imagine „consumă” timp pentru descărcarea, decodificarea și afișarea ei.

Fișierele-imagini au diferite formate, cele mai utilizate în Web design fiind:

- GIF (*Graphics Interchange Format*) – fișiere cu extensia .gif;
- JPEG (*Join Photographic Expert Group*) – fișiere cu extensia .jpeg sau .jpg;
- BMP (*BitMap*) – fișiere cu extensia .bmp.

Pentru a insera o imagine, se utilizează eticheta , cu atributul obligator SRC, a cărui valoare este adresa URL a imaginii:

Sintaxa pentru URL este identică cu cea utilizată pentru referințe (vezi paragraful 12.4).

În afară de atributul SRC, eticheta poate avea următoarele atribute:

- ALIGN, utilizat pentru alinierea imaginii și care are una din valorile *top* (aliniere deasupra – partea de sus a imaginii se aliniază cu partea de sus a textului ce precede imaginea), *bottom* (aliniere la bază – partea de jos a imaginii se aliniază cu linia de bază a textului), *middle* (aliniere la mijloc – mijlocul imaginii se aliniază cu linia de bază a textului ce precede imaginea), *left* (aliniere la stînga – textul și celelalte elemente sînt plasate în dreapta), *right* (aliniere la dreapta – textul și celelalte elemente sînt plasate în stînga);

- ALT, utilizat pentru afișarea unui *text explicativ* în locul imaginii (în cazul în care programul de explorare nu încarcă imaginea – opțiune stabilită de utilizator sau o altă cauză) și care are ca valoare însuși textul explicativ (scris între "și");

- WIDTH și HEIGHT, folosite pentru specificarea mărimii în pixeli a imaginii;

- BORDER, folosit pentru crearea în jurul imaginii a unui chenar de grosime (în pixeli) egală cu valoarea atributului;

- HSPACE și VSPACE, utilizate pentru precizarea distanței în pixeli pe orizontală, respectiv pe verticală, dintre imagine și restul elementelor din pagină.

Exemplul 1 (fig. 12.13):

```

<HTML>
<Head>
  <Title>Exemplul 1. Imagine</Title>
</Head>
<Body>
  <H3 align=center>Universitatea de Stat din Tiraspol</H3>
  <P>
  Universitatea de Stat din Tiraspol (UST) este prima institutie de
  invatamint superior din Republica Moldova, fondata la 1 octombrie 1930.

  <IMG SRC="ust.jpg" align=right hspace=10 vspace=10>

  Din anul 1992 UST se afla in Chisinau. UST a pregatit peste 80 000 de
  specialisti cu studii superioare. Multi dintre ei au devenit personalitati
  marcante in diverse domenii - invatamint preuniversitar si universitar,
  savanti in pedagogia si economia nationala. Majoritatea cadrelor de
  instruire detin titluri stiintifice si didactice. UST colaboreaza cu institutii
  din tara si de peste hotare - Romania, Rusia, Ucraina, SUA, Germania,
  Italia, Portugalia s.a.

</Body>
</HTML>

```

Imaginile pot fi utilizate și în calitate de referințe.

Exemplul 2 (fig. 12.14):

Următoarea pagină Web afișează 4 imagini-referință. La executarea unui clic pe una dintre referințe se va afișa imaginea corespunzătoare avînd dimensiuni reale.

Această tehnică este deseori folosită la crearea așa-numitelor „galerii-foto” (referințele pot fi fișiere de dimensiuni mai mici decît fișierele-imagini referite). Fișierele-imagini se află în subdirectorul DESENE al directorului curent.



Fig. 12.13

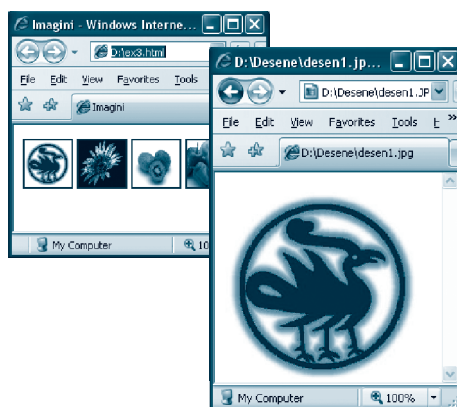


Fig. 12.14

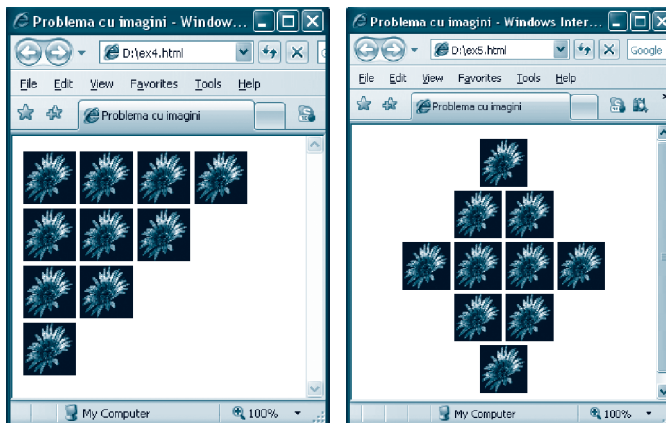
```

<HTML>
<head> <Title>Imagini</Title> </head>
<Body>
<A HREF="Desene/desen1.jpg"><img src=Desene/desen1.jpg width=
=50 height=50></A>
<A HREF="Desene/desen2.jpg"><img src=Desene/desen2.jpg width=
=50 height=50></A>
<A HREF="Desene/desen3.jpg"><img src=Desene/desen3.jpg width=
=50 height=50></A>
<A HREF="Desene/desen4.jpg"><img src=Desene/desen4.jpg width=5
=0 height=50></A>
</Body>
</HTML>

```

Întrebări și exerciții

- ❶ Care sînt formatele fișierelor-imagini acceptate de programele de explorare Web?
- ❷ Ce etichetă se utilizează pentru inserarea unei imagini într-un document HTML? Care este atributul obligatoriu al acestei etichete?
- ❸ Ce atribute neobligatorii poate avea eticheta pentru inserarea imaginilor într-un document HTML și care sînt valorile acestor atribute?
- ❹ Cum poate fi creată o referință-imagine?
- ❺ Creați o pagină Web care va afișa un CV cu fotografia autorului acestui CV, aliniată la centrul paginii.
- ❻ Creați o pagină Web care va afișa cîteva imagini-referințe cu dimensiunile 5 cm x 5 cm. La executarea unui clic pe fiecare imagine, programul de explorare va afișa această imagine cu dimensiunile 15 cm x 15 cm.
- ❼ Creați o pagină Web care va afișa un text despre localitatea natală cu titlul de nivelul 5 avînd culoarea neagră și fundalul gri. O imagine a localității va apărea în chenar și va fi aliniată la dreapta textului.
- ❽ Creați o pagină Web care va afișa graficul unei funcții.
Indicație: Acest grafic poate fi construit cu programul Microsoft Excel.
- ❾ Creați o pagină Web care va afișa o organigramă.
- ❿ Creați un document HTML care va afișa următoarea pagină Web (fig. 12.15 a, b) (se pot folosi alte imagini):



a)

b)

Fig. 12.15

12.6. Tabele

În cele mai dese cazuri tabelele se utilizează pentru organizarea structurată a datelor. De asemenea, așa cum fiecare celulă a tabelului poate avea propriile culori, stiluri, efecte pentru text și fundal, tabelele se folosesc și pentru soluții de design.

Un tabel, de regulă, are:

- a) *titlu* pentru specificarea numelui tabelului;
- b) *rînduri și coloane*;
- c) *cap de tabel* („table header”) pentru descrierea conținutului coloanelor sau al rîndurilor;
- d) *celule de date*, care formează rîndurile și coloanele tabelului.

Etichete pentru definirea unui tabel într-un document .html

Etichete	Descrierea etichetelor
<TABLE> </TABLE>	Definesc un tabel. Eticheta <TABLE> poate avea atributele: <ul style="list-style-type: none">– BORDER, care adaugă un chenar tabelului și a cărui valoare întregă pozitivă este grosimea în pixeli a chenarului (valoarea 1 este prestabilită, iar valoarea 0 semnifică absența chenarului);– ALIGN, care stabilește modul de aliniere a tabelului în document și poate avea una din valorile <i>left, right, center</i>;– BGCOLOR, care specifică culoarea de fond pentru tot tabelul;– CELSPACING, care stabilește distanța în pixeli dintre două celule vecine ale tabelului (valoarea prestabilită este 2);– CELLPADDING, care stabilește distanța în pixeli dintre marginea celei și conținutul ei (valoarea prestabilită este 1);– WIDTH și HEIGHT, care specifică lățimea și înălțimea tabelului (valori ale atributului pot fi numere întregi pozitive (număr de pixeli) sau numere întregi pozitive mai mici decât 101 și urmate de simbolul % (specificînd astfel fracțiuni din lățimea și înălțimea paginii)).
<CAPTION> </CAPTION>	Definesc titlul tabelului și pot fi plasate între etichetele <TABLE> și </TABLE>, dar nu în interiorul etichetelor care definesc celulele sau rîndurile tabelului. Eticheta <CAPTION> poate avea atributul ALIGN, care stabilește poziția titlului față de tabel și care va avea una din valorile <i>bottom, top, left, right</i> .
<TR> </TR>	Specifică un rînd în tabel. Eticheta <TR> poate avea atributele: <ul style="list-style-type: none">– ALIGN pentru alinierea orizontală (cu una din valorile <i>left, center, right</i>);– VALIGN pentru alinierea verticală (cu una din valorile <i>top, middle, bottom</i>).
<TH> </TH>	Definesc o celulă-antet. În mod prestabilit, textul unei astfel de celule este centrat și îngroșat. Eticheta <TH> poate avea atributele: <ul style="list-style-type: none">– WIDTH și HEIGHT pentru stabilirea lățimii și înălțimii celei;– ALIGN și VALIGN;– ROWSPAN, care unifică celula cu cîteva celule de desubtul ei (valoarea atributului este numărul total de celule care se unifică);– COLSPAN, care unifică celula cu cîteva celule din dreapta ei (valoarea atributului este numărul total de celule care se unifică);– NOWRAP, care interzice întreruperea liniei de text din celulă (deci coloana care conține celula poate deveni oricît de mare) și care se scrie fără valori;– BGCOLOR, care specifică culoarea de fond pentru celulă (culoarea textului celei poate fi stabilită cu ajutorul atributului <i>Color</i> al etichetei).
<TD> </TD>	Definesc o celulă de date obișnuită. Eticheta <TD> poate avea oricare dintre atributele etichetei <TH>.

Dacă o celulă a tabelului este vidă, atunci în mod normal ea este afișată fără chenar. Pentru a evita acest efect, în locul conținutului celulei se va scrie textul sau
.

Exemplul (fig. 12.16):

```
<HTML>
<Head> <Title>Exemplul 1. Tabele</Title> </Head>
<Body>
<Table border=2 width="90%" height="80%" align="center" bgcolor="lightcyan">
<Caption align="left"><Font color="Red" Size=6> Orarul lectiilor </font>
</Caption>
<Tr><Th Colspan=2>Luni<Th>Marti<Th>Miercuri<Th>Joi<Th>Vineri </Tr>
<Tr align="center"><Td>1<Td>Romana<Td>Chimia<Td>Fizica
  <Td>Istoria<Td>Algebra </Tr>
<Tr align="center"><Td>2<Td>Engleza<Td>Fizica<Td>Informatica
  <Td>Rusa<Td>Biologia </Tr>
<Tr align="center"><Td>3<Td>Franceza<Td>Algebra<Td>Geometria
  <Td>Muzica<Td> Romana </Tr>
</Table>
</Body>
</HTML>
```



Fig. 12.16

Întrebări și exerciții

- 1 Pentru ce sînt utilizate tabelele?
- 2 Numiți atributele etichetei <TABLE> și valorile posibile ale acestor atribute.
- 3 Care sînt atributele etichetei <CAPTION> și valorile posibile ale acestor atribute?
- 4 Cum se specifică un rînd de tabel?
- 5 Care etichete definesc o celulă antet?
- 6 Care sînt atributele acestor etichete și valorile lor posibile?

7 Care etichete definesc o celulă de date? Care sînt atributele acestor etichete?

8 Creați o pagină Web care va afișa următorul tabel:

a)

A	C	E
B	D	F

b)

A	B	C
	D	E

c)

A		D
B	C	E

d)

A	C	E
B		

e)

A	B			C
	D	E	F	
G	H			I

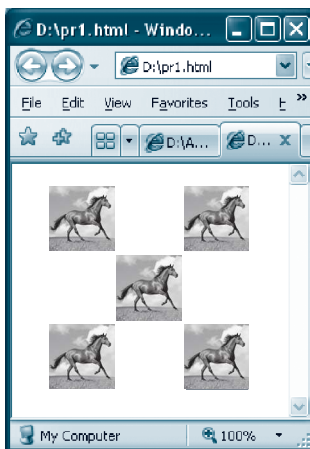
f)

A	C	E
B	D	F
	G	H

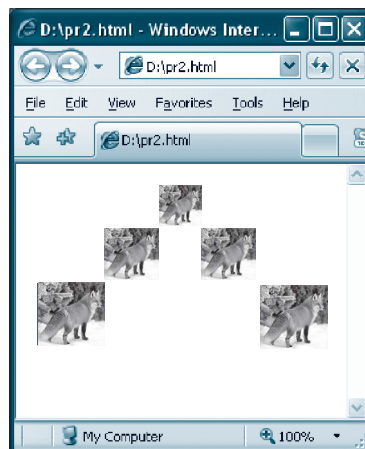
g)

A	B	D	E	F	G
	C		H		

9 Creați un document HTML care va afișa următoarea pagină Web (fig. 12.17 a, b) (se pot folosi alte imagini):



a)



b)

Fig. 12.17

10 Ce va afișa pagina Web cu următorul cod HTML?

```
a) <HTML>
<Head><Title>Ce afisam?</Title> </Head>
<Body>
<Table align=center border=2 width=100>
  <Tr align=center>
    <Td colspan=2>B</Td><Td rowspan=2>D</Td>
  </Tr>
  <Tr align=center>
    <Td>C</Td><Td>A</Td>
  </Tr>
</Table>
</Body>
</HTML>
```









```
b) <HTML>
<Head><Title>Ce afisam?</Title> </Head>
<Body>
<Table align=center width=100 border=2>
  <Tr align=center>
    <Td rowspan=2>B</Td><Td>C</Td><Td>A</Td>
  </Tr>
  <Tr align=center>
    <Td colspan=2>D</Td>
  </Tr>
</Table>
</body>
</HTML>
```









11 Creați un document HTML care va afișa următorul tabel:

Tabel. **Extragerea pietrelor prețioase**

	2010	2012	2014
Rubin	1 483	4 532	6 743
Safir		3 835	7 482
Diamant	683	238	

12 Creați un document HTML care va afișa un tabel de forma:

Animale domestice		Animale sălbatice		Se întîlnesc pe teritoriul țării noastre.
Pot zbura	Nu pot zbura	Pot zbura	Nu pot zbura	
 Găina	 Iepurele	 Cioara	 Lupul	
 Gîsca	 Capra	 Bufnița	 Vulpea	

 Rața	 Calul	 Flamingo	 Elefantul	Nu se întîlnesc pe teritoriul țării noastre.
 Prepelița	 Măgarul	 Colibri	 Leul	

Creăm în echipă

- Proiectați structura unui site cu titlul:
 - Republica Moldova – țară europeană;
 - Localitatea noastră de baștină;
 - Liceul nostru;
 - Republica Moldova – atracție turistică;
 - Mănaștirile Republicii Moldova;
 - Personalități notorii;
 - Ani de liceu;
 - Fauna Republicii Moldova;
 - Bucătăria moldovenească;
 - Cărțile mele preferate;
 - Profesii de prestigiu;
 - Enciclopedia modei;
 - Umorul – sursă de bună dispoziție;
 - Absolvenți de succes ai liceului nostru;
 - Produse autohtone de calitate;
 - Sportul este sănătate.
- Utilizînd limbajul HTML și consultînd diferite surse de informație, creați paginile web – componente ale site-ului. Stabiliți între ele legături de navigare. Toate paginile vor avea același meniu principal. Testați site-ul cu diferite programe de navigare.
- Creați și atașați site-ului:
 - o pagină web cu referințe spre surse informaționale suplimentare la tema site-ului;
 - o pagină web cu galerii de imagini corespunzătoare temei.

Test de evaluare

- Stabiliți valoarea de adevăr a propoziției:
 - Textul scris între etichetele <HEAD> și </HEAD> ale unui document HTML va fi afișat pe bara de titlu a ferestrei programului de explorare în care a fost interpretat acest document.
 - O etichetă poate avea cîteva atribute.
 - Programul de explorare va afișa din rînd nou un text, dacă acesta va fi precedat de oricare din etichetele <P>,
, <HR>.
 - Un text cuprins între etichetele <DIV> și </DIV> este un text preformatat.
- Selecțați etichetele care se folosesc pentru a stabili stiluri fizice pentru un text: , , <DFN>, </DFN>, <VAR>, </VAR>, <BIG>, </BIG>, , , _,.

3. Scrieți un cod HTML care, fiind interpretat de programul de explorare, va afișa următoarele liste imbricate:
- Fructe
 - a) Mere
 - b) Pere
 - c) Prune
 - d) Caise
 - Legume
 1. Cartofi
 2. Morcovi
 3. Ceapă
 - Lactate
 - o Lapte
 - o Brînză
 - o Iaurt
4. Fișierul f1.html se află în mapa A, iar fișierul f2.html se află în mapa B. Care este URL-ul referinței din fișierul f1.html către fișierul f2.html, dacă:
- a) mapele A și B se află în același catalog;
 - b) mapa A se conține în mapa B;
 - c) mapa B se conține în mapa A;
 - d) mapa A se conține în submapa C a mapei B?
5. Comentați semnificația următorului cod HTML:
``.
6. Scrieți un cod HTML care, fiind interpretat de programul de explorare, va afișa următorul tabel:

A		B	C	D
E		F		G
H	I			J