

ACEST MANUAL A FOST FOLOSIT DE:

Mariana Miloșescu

# Informatică

## Profilul real

Specializarea:  
matematică-informatică, științe ale naturii

Manual pentru clasa a IX - a



EDITURA DIDACTICĂ ȘI PEDAGOGICĂ S.A.

|                                                                    |    |
|--------------------------------------------------------------------|----|
| <b>1. Informatica și societatea</b> .....                          | 3  |
| 1.1. Prelucrarea informațiilor .....                               | 3  |
| 1.2. Informatica .....                                             | 4  |
| 1.3. Etapele rezolvării unei probleme .....                        | 6  |
| 1.4. Algoritmul .....                                              | 8  |
| <b>Evaluare</b> .....                                              | 10 |
| <b>2. Datele</b> .....                                             | 11 |
| 2.1. Definiția datelor .....                                       | 11 |
| 2.1.1. Clasificarea datelor .....                                  | 12 |
| 2.1.2. Tipul datei .....                                           | 17 |
| 2.2. Operatorii .....                                              | 19 |
| 2.3. Expresiile .....                                              | 24 |
| <b>Evaluare</b> .....                                              | 28 |
| <b>3. Algoritmii</b> .....                                         | 34 |
| 3.1. Reprezentarea algoritmilor .....                              | 34 |
| 3.2. Principiile programării structurate .....                     | 36 |
| 3.2.1. Structura liniară .....                                     | 36 |
| 3.2.2. Structura alternativă .....                                 | 37 |
| 3.2.3. Structura repetitivă .....                                  | 41 |
| 3.3. Algoritmi elementari .....                                    | 47 |
| 3.3.1. Algoritmi pentru interschimbare .....                       | 47 |
| 3.3.2. Algoritmi pentru determinarea maximului (minimului) .....   | 49 |
| 3.3.3. Algoritmi pentru prelucrarea cifrelor unui număr .....      | 51 |
| 3.3.4. Algoritmi pentru calcularea c.m.m.d.c. ....                 | 56 |
| 3.3.5. Algoritmi pentru testarea unui număr prim .....             | 58 |
| 3.3.6. Algoritmi pentru prelucrarea divizorilor unui număr .....   | 61 |
| 3.3.7. Algoritmi pentru conversii între sisteme de numerație ..... | 64 |
| 3.3.8. Algoritmi pentru generarea șirurilor recurente .....        | 66 |
| 3.4. Eficiența algoritmilor .....                                  | 67 |
| <b>Evaluare</b> .....                                              | 72 |
| <b>4. Aplicarea algoritmilor</b> .....                             | 76 |
| 4.1. Rezolvarea problemelor de matematică .....                    | 76 |
| 4.2. Rezolvarea problemelor de fizică .....                        | 80 |
| <b>5. Implementarea algoritmilor</b> .....                         | 83 |
| 5.1. Caracteristicile limbajului de programare .....               | 83 |
| 5.2. Structura programului .....                                   | 84 |
| 5.3. Structurile de control .....                                  | 86 |

# 1. Informatica și societatea

## 1.1. Prelucrarea informațiilor

**Calculatorul a fost inventat de om pentru a prelucra informația.** El îl ajută să prelucraze *foarte ușor*, într-un timp *extrem de scurt*, cu *foarte mare acuratețe*, o *mare cantitate* de informație *foarte complexă*.

Prelucrarea informației este veche de când lumea. **Prelucrarea voluntară a informației** s-a făcut însă abia atunci când babilonienii au scris primele semne cuneiforme pe tăblițele de lut. Așadar, prima manifestare a prelucrării informației a fost *scrisul*. Încă din antichitate, pot fi puse în evidență două tipuri de prelucrări de informații:

- ✓ **prelucrarea textelor** = **scrisul**;
- ✓ **prelucrarea numerelor** = **calculul numeric**.

**Prelucrarea automată a informației** a fost posibilă o dată cu apariția calculatoarelor electronice. Așadar, scopul utilizării unui calculator este de a prelucra **informația**. Informația prelucrată poate fi formată din texte, numere, imagini sau sunete și este păstrată pe diferite medii de memorare, în diferite formate, sub formă de **date**.

Transformarea datelor în informații nu este un atribut exclusiv al calculatorului. Acest fenomen a apărut o dată cu omul. De la primele reprezentări ale unor cantități cu ajutorul degetelor, al pietricelelor sau al bețișoarelor, și de la manipularea manuală a acestor obiecte pentru a afla câte zile mai sunt până la un anumit eveniment sau câte animale au fost vâdate sau câți războinici are tribul vecin, putem spune că are loc un proces de transformare a datelor în informații. Degetele, pietricelele și bețele reprezintă datele, iar ceea ce se obține prin manipularea lor (numărul de animale vâdate, numărul de zile, numărul de războinici) reprezintă informația. Cele ce deosebesc un astfel de proces de o prelucrare cu ajutorul calculatorului sunt viteza de obținere a informațiilor și modul de reprezentare a informațiilor sub formă de date.

Calculatorul nu știe să prelucraze decât șiruri de cifre binare, care pot fi modelate fizic prin impulsuri de curent, cu două niveluri de tensiune, ce corespund celor două cifre binare: 0 și 1. Prin urmare, datele vor fi codificări binare ale informației existente în exteriorul calculatorului. Dacă, într-o prelucrare manuală, datele sunt reprezentate de obiecte care pot fi manipulate de om (bețișoare, pietricele sau degete), în cazul unei prelucrări automate datele vor fi reprezentate prin obiecte pe care le poate manipula calculatorul, adică șiruri de biți.

Așadar, din punct de vedere al unei prelucrări automate a informației, diferența dintre dată și informație este:

- ✓ **Informația** este un mesaj care înlătură necunoașterea unui anumit eveniment și are caracter de noutate. Informațiile sunt interpretate de oameni.
- ✓ **Data** este reprezentarea informației în interiorul calculatorului. Calculatorul nu înțelege conținutul acestor date, el numai le prelucraază, prin operații specifice fiecărui tip de dată. În urma prelucrării datelor, calculatorul poate furniza omului informații.

Pentru a rezolva o anumită sarcină, trebuie să cunoaștem modul în care se poate face acest lucru. Așadar, trebuie să găsim o anumită **metodă**, adică un set de pași pe care trebuie să-i executăm ca să realizăm sarcina. Acest set de pași formează **algoritmul** pentru rezolvarea problemei respective. Inițial, studiul algoritmilor a fost o disciplină a matematicii, prin care se căuta să se găsească un set unic de instrucțiuni prin care să se descrie rezolvarea oricărei probleme dintr-o anumită categorie. Ați învățat deja o parte din acești algoritmi: algoritmul lui Euclid pentru găsirea celui mai mare divizor comun dintre două numere, algoritmul împărțirii unui număr, algoritmul extragerii rădăcinii pătrate dintr-un număr, algoritmul conversiei unui număr reprezentat în baza zece într-un număr reprezentat într-o altă bază de numerație etc. Cu timpul, descrierea metodei de rezolvare a unei probleme cu ajutorul algoritmilor s-a extins și în alte domenii de activitate.

La fel ca și omul, și un calculator, pentru a putea rezolva o anumită sarcină, trebuie „să aibă cunoștințe” despre modul în care se poate face aceasta, adică să cunoască algoritmul de rezolvare a problemei. Această informație i se transmite calculatorului prin intermediul unui **program**. Deoarece **limbajul natural** (limbajul prin care comunică oamenii) nu este înțeles de calculator, care este construit astfel încât să poată prelucra numai cifre binare, programul prin care i se comunică algoritmul este scris într-un limbaj de programare. **Limbajul de programare** este un **limbaj artificial** care, prin exprimări simbolice (**instrucțiuni**), descrie operațiile de prelucrare pe care trebuie să le execute calculatorul. El îi permite omului să comunice cu calculatorul (să îi dea comenzi pe care să le execute), deoarece fiecare instrucțiune din limbajul de programare va fi tradusă într-un grup de **instrucțiuni mașină** (instrucțiuni în **limbaj mașină**), adică un șir de biți care au o anumită semnificație pentru calculator. Acest limbaj se numește limbaj mașină deoarece este propriu fiecărui tip de mașină (calculator), fiind implementat – cu ajutorul circuitelor electronice – în procesor.

Așadar, o sarcină se poate rezolva cu ajutorul calculatorului numai dacă modul în care se rezolvă poate fi descompus în pași, pentru a putea fi descris cu ajutorul unui algoritm, deoarece calculatorul este o **mașină algoritmică**.

Dezvoltarea prelucrării automate a informațiilor cu ajutorul calculatorului s-a făcut în două direcții:

- ✓ **dezvoltarea echipamentelor**, astfel încât acestea să fie capabile să stocheze cât mai multă informație, pe care să o prelucreze cu viteză cât mai mare, folosind algoritmi cât mai complecși;
- ✓ **găsirea de noi algoritmi**, cât mai performanți, pentru rezolvarea problemelor complexe și **îmbunătățirea tehnicilor de reprezentare și comunicare** a lor.

## 1.2. Informatica

Folosirea calculatorului a dus la apariția unei noi științe și a unui nou domeniu de activitate: **informatica**.

**Informatica** reprezintă un complex de discipline prin care se asigură prelucrarea rațională a informațiilor prin intermediul mașinilor automate.

Primul calculator electronic a apărut în anul 1946, ca urmare a unei cereri precise din partea armatei americane, care a fost capabilă să finanțeze un proiect atât de costisitor. Apoi, administrația americană a cumpărat primul calculator non-militar în 1951, pentru recensământul populației. Doi ani mai târziu a fost construit primul calculator destinat unei firme particulare, când General Electric a cumpărat un calculator pentru uzina sa din Louisville. Începând din 1953, firma IBM a început să pătrundă și ea pe piața de calculatoare, prezentându-și în mediile științifice propriile calculatoare. Astfel, calculatoarele au început să pătrundă și în mediile universitare. Dezvoltarea continuă a echipamentelor electronice de calcul a făcut ca, **din 1965, informatica să nu mai fie doar o activitate anexă, ci să se transforme ea însăși într-o industrie.** În contextul unei creșteri puternice a pieței, calculatorul a devenit o unealtă folosită în toate domeniile de activitate.

La primele calculatoare electronice, programele erau scrise în cod mașină (binar) sau erau cablate sub formă de circuite electronice. Modificarea unui program și introducerea unuia nou erau foarte complicate, deoarece însemna introducerea programului bit cu bit. Din necesitatea rezolvării acestei probleme, au apărut primele sisteme de operare și primele limbaje de programare numite limbaje de nivel înalt: în 1956, limbajul **Fortran** orientat pe calcule tehnico-științifice, și în 1960 limbajul **Cobol** orientat pe aplicații economice care folosesc puține operații de calcul, dar care manipulează un volum mare de date. Limbajele de programare s-au dezvoltat continuu, pentru a se adapta la noile echipamente hardware, la noile sisteme de operare și la noile cerințe ale utilizatorilor – care însemnau de fapt noi sarcini pe care trebuia să le rezolve calculatorul, adică noi algoritmi, orientați pe rezolvarea anumitor probleme. În 1971 a fost creat în universitățile elvețiene limbajul **Pascal**, primul **limbaj structurat** (fiecare prelucrare elementară este considerată ca un bloc, iar blocurile pot fi închise – încapsulate – unele în altele). O dată cu apariția microcalculatoarelor, acest limbaj s-a răspândit foarte mult. Limbajul **Basic** a fost creat în Statele Unite, în 1975, ca un **limbaj interactiv**, și nu putea fi folosit decât pe microcalculatoare. El permitea abordarea programării și de către persoane care nu erau specialiste în informatică. În 1971 a fost creat de firma Bell-Telephone limbajul **C**, pentru a permite realizarea sistemului de operare Unix. Este un limbaj foarte performant, care posedă atât conceptele limbajelor structurate de nivel înalt, cât și conceptele limbajelor de nivel scăzut, care îi permit accesul la hardware. Programele scrise în limbajele apărute recent au crescut productivitatea programatorilor. Limbajele de nivel înalt au pus bazele **ingineriei programării**.

La începutul anilor '60, în mediile universitare au început să se formeze departamente pentru cercetarea și studierea calculatoarelor. Cu timpul, a apărut o bogată literatură de specialitate, iar cursurile din domeniul informaticii au început să fie orientate pe subdomenii și să fie gradate pe niveluri de dificultate. Astăzi, informatica este divizată în următoarele nouă **subdomenii**.

1. **Algoritmi și structuri de date.** Studiază metodele prin care se pot obține aplicații care să prelucereze diferite clase de informații, modul în care vor fi reprezentate informațiile care vor fi prelucrate și metodele de optimizare a pașilor necesari pentru realizarea aplicațiilor. Scopul acestui subdomeniu este de a identifica problemele care pot fi descrise cu ajutorul algoritmilor, de a găsi modul în care trebuie procedat pentru a descoperi algoritmul și metodele de analiză și comparare a caracteristicilor algoritmilor pentru a obține algoritmi cât mai eficienți.

2. **Limbaje de programare.** Studiază notațiile (limbajele) prin care vor fi reprezentăți algoritmi și structurile de date, astfel încât aplicația să poată fi prelucrată. Aceste limbaje sunt apropiate de limbajul natural și pot fi ușor traduse în secvențe de comenzi pe care să le înțeleagă calculatorul. Scopul acestui subdomeniu este de a găsi noi tehnici de reprezentare și comunicare a algoritmilor.
3. **Arhitectura calculatoarelor.** Studiază modul în care sunt organizate diferite componente hardware ale calculatorului și modul în care sunt conectate, pentru a putea obține un sistem eficient, sigur și util. Scopul acestui subdomeniu este de a realiza mașini algoritmice cât mai bune folosind cunoștințele despre algoritmi deja dobândite și tehnologia existentă.
4. **Sisteme de operare.** Studiază felul în care trebuie să fie organizate programele care controlează și coordonează toate operațiile din sistemul de calcul. Scopul acestui subdomeniu este de a face un calculator să rezolve în același timp mai multe sarcini, fără ca pașii algoritmilor care descriu rezolvarea acestor sarcini să interfereze unii cu alții, iar atunci când este cazul să se poată realiza comunicarea între diverși algoritmi.
5. **Ingineria programării.** Studiază metodele prin care poate fi automatizată activitatea de proiectare a aplicațiilor și de prelucrare a informațiilor, astfel încât să se obțină programe corecte, eficiente, fără erori și ușor de exploatat.
6. **Calcul numeric și simbolic.** Studiază descrierea fenomenelor din lumea reală prin intermediul formulelor matematice, care pot fi manipulate algebric astfel încât să se obțină modele matematice ușor de descris prin algoritmi. Scopul acestui subdomeniu este de a găsi modele matematice care să permită descrierea și reprezentarea în calculator a fenomenelor complexe, cum sunt: zborul avioanelor, curenții marini, traiectoria sateliților și a planetelor, mișcarea particulelor etc.
7. **Sisteme de gestiune a bazelor de date.** Studiază modul în care pot fi organizate cantități mari de date ce nu necesită, în prelucrare, calcule matematice complexe. Este cazul informațiilor prelucrate în procesele economico-sociale, în întreprinderi și în administrație. Prelucrarea acestor date trebuie să se facă eficient, fără erori, cu asigurarea securității lor.
8. **Inteligența artificială.** Studiază modul în care percepe și raționează mintea umană, cu scopul de a putea fi automatizate aplicații pe care omul le realizează prin metode „inteligente”, care sunt dificil de descris cu ajutorul algoritmilor, ca de exemplu înțelegerea unui limbaj, crearea de noi teorii matematice, compunerea muzicii, crearea operelor de artă, luarea deciziilor în urma evaluării unor situații complexe (stabilirea unui diagnostic în medicină, mutarea pieselor la jocul de șah etc.).
9. **Animație și robotică.** Studiază metodele prin care pot fi generate și prelucrate imaginile și modul în care se poate răspunde unei situații din exterior prin acționarea unui robot.

### 1.3. Etapele rezolvării unei probleme

Orice prelucrare automată a informațiilor presupune definirea următorului lanț:



Din această cauză, pentru orice rezolvare a unei probleme cu ajutorul calculatorului, trebuie parcurse următoarele **etape**:

1. **analiza problemei;**
2. **elaborarea modului de rezolvare a problemei;**
3. **codificarea într-un limbaj de programare a modului de rezolvare a problemei;**
4. **testarea programului și corectarea erorilor.**

**1. Analiza problemei.** Această etapă constă în formularea enunțului problemei din care vor rezulta **specificatiile** complete și precise ale programului care va rezolva problema. Aceste specificații trebuie să țină cont de condițiile concrete de realizare a programului. Specificațiile sunt:

- ✓ **Funcția programului.** Prin ea, se determină ceea ce urmează să realizeze programul.
- ✓ **Identificarea fluxului de informații.** Aceasta presupune identificarea **informațiilor de intrare** și, respectiv, a **informațiilor de ieșire** care vor fi descrise cu ajutorul **datelor**: **date de intrare** și, respectiv, **date de ieșire**.

Fiecărui tip de informație îi corespunde un anumit mod de stocare în mediul de memorare, adică un anumit tip de dată. Între datele prelucrate de un program există diferite relații. Modul în care vor fi aranjate aceste date în mediul de memorare depinde de legătura dintre ele.

**2. Elaborarea modului de rezolvare a problemei.** Această etapă constă în găsirea metodei prin care să se poată rezolva problema. Ea presupune identificarea **prelucrărilor** care se fac asupra datelor de intrare pentru a obține datele de ieșire. Descrierea acestor prelucrări se face cu ajutorul **algoritmului** de rezolvare a problemei. Această fază este cea mai importantă și cea mai grea, deoarece presupune definirea logică a unei secvențe de operații pe care să le poată executa calculatorul, astfel încât să se obțină rezultatele dorite.

**3. Codificarea într-un limbaj de programare a modului de rezolvare a problemei.** Algoritmul de rezolvare a problemei este transpus într-un limbaj de programare ales în conformitate cu specificul problemei care trebuie să fie rezolvată, pentru a fi comunicat calculatorului.

**4. Testarea programului și corectarea erorilor.** Pentru testarea programului se va folosi o mulțime de seturi de date de intrare, care trebuie să prevadă toate situațiile care pot să apară în exploatarea curentă a programului. Testarea constă în executarea repetată a programului, pentru fiecare set de date de intrare. Dacă această mulțime de seturi de date nu este aleasă corect, programul nu va fi testat pe toate traseele algoritmului și în etapa de exploatare pot apărea erori. În această etapă se pun în evidență erorile de sintaxă, erorile de logică și dacă reprezentarea externă a rezultatelor are aspectul grafic dorit. Erorile de sintaxă apar din scrierea incorectă a instrucțiunilor și ele vor fi corectate în program. Erorile de logică apar din cauza metodei de rezolvare alese și ele vor trebui identificate în cadrul algoritmului și corectate în program.

Așadar, pentru ca un calculator **să poată produce informații**, trebuie ca, la rândul său, **să primească două categorii de informații**:

- ✓ Descrierea modului în care realizează sarcina, adică **algoritmul**, care i se comunică sub forma unui **program**.

- ✓ Informațiile de care are nevoie algoritmul ca să realizeze acea sarcină care i se comunică sub formă de **date de intrare**.

## Studiu de caz

**Scop:** exemplificarea etapelor de rezolvare a unei probleme.

**Enunțul problemei:** Fiind date două numere reale  $a$  și  $b$ , să se rezolve ecuația de gradul întâi cu acești coeficienți:  $ax + b = 0$ .

În urma analizei problemei, se obține **specificația programului**:

- ✓ **Funcția programului.** Dacă pentru ecuația de gradul întâi  $ax + b = 0$  există o soluție reală, se calculează; în caz contrar, se afișează un mesaj.
- ✓ **Informațiile de intrare** sunt coeficienții ecuației, iar suportul extern prin care se vor introduce este tastatura. Reprezentarea internă a informației se va face prin **datele de intrare**  $a$  și  $b$ .
- ✓ **Informația de ieșire** va fi soluția ecuației, dacă există, iar dacă nu există, un mesaj. Suportul extern pe care va fi reprezentată informația de ieșire este ecranul monitorului. Reprezentarea internă a soluției ecuației se va face prin **data de ieșire**  $x$ .

Metoda folosită pentru rezolvarea problemei va fi algoritmul matematic de rezolvare a ecuației de gradul întâi.

Pentru testarea programului, se va considera că un set de date de intrare este format de perechea de coeficienți  $(a; b)$ , iar o mulțime completă de seturi de date de intrare poate fi  $\{(0; 0), (0; 1.5), (2.5; 1.5)\}$ .



## 1.4. Algoritmul

Datele de intrare sunt supuse unui proces de prelucrare, pentru a se obține datele de ieșire. În funcție de rezultatele care se doresc, prelucrarea datelor este realizată după un anumit algoritm.

**Algoritmul** reprezintă o **mulțime ordonată și finită de pași executabili** prin care se definește fără echivoc modul în care se poate realiza o anumită sarcină.

Între datele de intrare și datele de ieșire ale algoritmului există o relație bine determinată de însăși construcția algoritmului.

În activitățile zilnice întâlnim la tot pasul algoritmi: algoritmul de utilizare a mașinii de spălat rufe sau vase (exprimat prin setul de instrucțiuni din cartea tehnică a mașinii sau de pe capacul mașinii de spălat), algoritmul de înregistrare pe o casetă video (exprimat prin setul de instrucțiuni din cartea tehnică a videorecorderului), algoritmul de interpretare a muzicii (exprimat prin partitură), algoritmul de construire a unui model de avion sau de navă (exprimat prin setul de instrucțiuni care însoțesc piesele care compun modelul), algoritmul de rezolvare a unei probleme matematice (exprimat printr-un set unic de operații prin care se descrie modul de rezolvare a oricărei probleme dintr-o categorie de probleme). De fapt, aproape toa-

te acțiunile noastre se desfășoară după un algoritm bine definit. Un exemplu de algoritm al activităților zilnice este o convorbire telefonică:

- Pasul 1.** Început.
- Pasul 2.** Mergi la telefon.
- Pasul 3.** Ridică microreceptorul telefonului.
- Pasul 4.** Dacă **are ton**, formează numărul de telefon; altfel, pleacă la vecin și mergi la **Pasul 10**.
- Pasul 5.** Dacă **telefonul este ocupat**, închide telefonul și mergi la **Pasul 11**; altfel, așteaptă să răspundă.
- Pasul 6.** Dacă **nu răspunde**, pune microreceptorul în furcă și mergi la **Pasul 12**; altfel, începi discuția cu persoana care a răspuns.
- Pasul 7.** Dacă **a răspuns persoana căutată**, mergi la **Pasul 9**; altfel, cere să vină la telefon persoana căutată.
- Pasul 8.** Dacă **persoana căutată nu poate să vină la telefon**, mergi la **Pasul 13**; altfel, așteaptă să vină la telefon.
- Pasul 9.** Discută la telefon cu persoana căutată și mergi la **Pasul 13**.
- Pasul 10.** Anunță la serviciul „Deranjamente telefoane” că ai telefonul defect și mergi la **Pasul 14**.
- Pasul 11.** Așteaptă 15 minute și mergi la **Pasul 2**.
- Pasul 12.** Așteaptă 1 oră și mergi la **Pasul 2**.
- Pasul 13.** Închide telefonul.
- Pasul 14.** Terminat.

Un exemplu de algoritm matematic este rezolvarea ecuației de gradul întâi:

$$a \times z + b = 0$$

unde **a** și **b** sunt coeficienții ecuației și pot lua orice valori din domeniul numerelor reale, iar **z** reprezintă un număr care se calculează și care poate lua și el orice valoare reală, astfel încât să fie îndeplinită relația definită prin ecuație. Algoritmul de rezolvare a ecuației va prezenta un set unic de operații, prin care se calculează valoarea lui **z**, oricare ar fi valorile pentru **a** și **b**:

- Pasul 1.** Început.
- Pasul 2.** **Comunică** valorile pentru **a** și **b**.
- Pasul 3.** **Compară** **a = 0**. Dacă este adevărat, execută **Pasul 4**; altfel, execută **Pasul 7**.
- Pasul 4.** **Compară** **b = 0**. Dacă este adevărat, execută **Pasul 5**; altfel, execută **Pasul 6**.
- Pasul 5.** **Comunică** mesajul "Ecuația are o infinitate de soluții". Mergi la **Pasul 9**.
- Pasul 6.** **Comunică** mesajul "Ecuația nu are soluții". Mergi la **Pasul 9**.
- Pasul 7.** **Calculează** **z = -b/a**.
- Pasul 8.** **Comunică** valoarea lui **z**.
- Pasul 9.** Terminat.

Numărul de pași este finit (9 pași). Toți pașii reprezintă acțiuni care se pot executa: compară, calculează, comunică. O dată definit acest algoritm, pașii lui se vor executa pentru orice valori ale lui **a** și **b**, deci algoritmul descrie rezolvarea unei probleme generale. La fiecare executare a algoritmului care descrie o problemă generală va fi tratat un caz particular, adică se rezolvă ecuația de gradul întâi pentru va-

lori precizate ale lui  $a$  și  $b$ , ca de exemplu  $2x - 4 = 0$  ( $a = 2$ ,  $b = -4$ ) sau  $0x - 4 = 0$  ( $a = 0$ ,  $b = -4$ ) sau  $0x - 0 = 0$  ( $a = 0$ ,  $b = 0$ ).

Așadar, algoritmi au următoarele **proprietăți**:

- ✓ **Claritatea.** Orice algoritm trebuie să fie precis definit, să prezinte clar toate etapele care trebuie parcurse până la obținerea soluției, fără să formuleze nimic ambiguu.
- ✓ **Finitatea.** Algoritmul trebuie să fie format dintr-un număr finit de pași, prin executarea cărora să se ajungă la rezolvarea problemei și obținerea rezultatelor.
- ✓ **Sucesiunea determinată a pașilor.** Pașii care compun algoritmul trebuie executeți într-o ordine bine determinată. De obicei, ei se execută în ordine secvențială (ordinea în care au fost scriși). În cazul în care apare necesitatea schimbării acestei ordini, trebuie să se precizeze clar pasul care urmează să fie executat.
- ✓ **Universalitatea.** Algoritmul trebuie să permită rezolvarea unei clase de probleme, care sunt de același tip și care diferă între ele numai prin datele de intrare. El trebuie să ofere posibilitatea de a rezolva orice problemă din acea clasă de probleme.
- ✓ **Realizabilitatea.** Pașii care compun algoritmul trebuie să reprezinte operații care se pot executa cu resursele disponibile.
- ✓ **Eficiența.** Operațiile care compun algoritmul trebuie alese astfel încât soluția problemei să fie obținută după un număr minim de pași, cu precizia prestabilită sau cu o precizie satisfăcătoare.

## Evaluare

### Răspundeți:

1. Ce este un algoritm? Ce sunt pașii algoritmului?
2. Determinați algoritmul pentru prepararea unui ceai. Identificați proprietățile unui algoritm, în cazul acestui algoritm.
3. Citiți o rețetă din cartea de bucate. Determinați algoritmul pentru prepararea respectivului produs culinar.
4. Dați patru exemple de probleme a căror rezolvare nu poate fi descrisă cu ajutorul algoritmului și patru exemple de probleme a căror rezolvare poate fi descrisă cu ajutorul algoritmului.