

Ministerul Educației și Cercetării

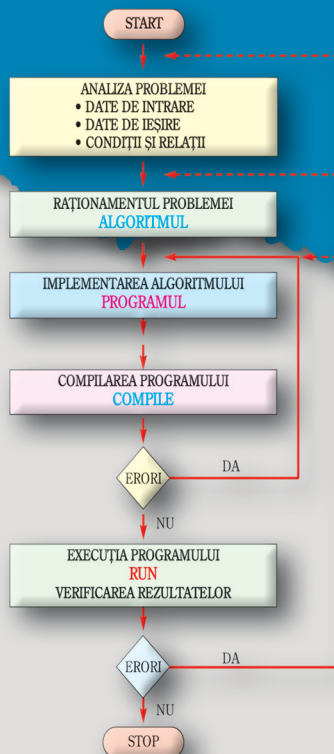
Informatică

Profil real

Manual pentru clasa a IX - a

Mioara Gheorghe
(coordonator)

Mariana Kisch
Monica Tătărâm



CORINT

În lumea informaticii

1

1.1. Informatica – știință interdisciplinară

Lumea în care trăim, cu toată diversitatea sa, este un ansamblu unitar format din elemente care comunică între ele. Un astfel de ansamblu se numește *sistem*.

Sistem = ansamblu de elemente a căror interacțiune conduce spre o finalitate comună numită *scop* sau *obiectiv*.

Exemple: sistemul planetar, corpul omenesc, societatea, familia, sistemul sanitar, orice aparat electrocasnic (frigider, televizor), calculatorul electronic și multe altele.

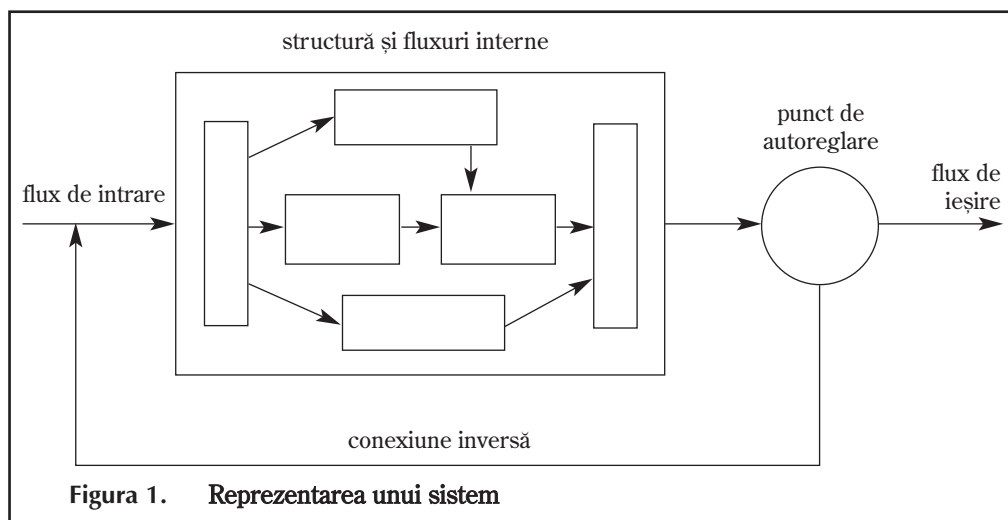


Figura 1. Reprezentarea unui sistem

Indiferent de scara la care privim lumea, aspectul sistemic poate fi demonstrat punând în evidență:

- elementele componente (structura);
- legăturile dintre elemente (fluxurile interne);
- legăturile cu exteriorul (fluxurile externe sau interfețele de intrare-ieșire);
- scopul sau rostul sistemului (finalitatea);
- capacitatea de autoreglare (conexiune inversă sau *feedback*).

În Tabelul 1 sunt prezentate exemple de sisteme.

Tabelul 1. Exemple de sisteme

Sistem	Structură	Fluxuri interne	Fluxuri externe	Finalitate	Conexiune inversă
Organismele vii	– celule – țesuturi – organe	Toate procesele care asigură integralitatea organismului	– substanțe nutritive – substanțe reziduale – energie	Transformarea factorilor de mediu în factori proprii, printr-un flux continuu de materie și energie	Consumul de energie reglează cantitatea de substanțe nutritive
Calculatorul electronic	– echipamente periferice – memorie – procesor	– semnale – comenzi – instrucțiuni	– comenzi – mesaje – date de intrare/ieșire	Memorarea și prelucrarea automată a datelor	Mesajele de eroare determină reintroducerea datelor sau a comenzilor

Unitatea lumii ca sistem este rezultatul comunicării și prelucrării de informații.

Informația este un concept modern ce caracterizează materia în toate formele ei de existență și de manifestare. Reacțiile chimice, conexiunile sistemului nervos, legăturile interumane sunt doar câteva exemple.

Informația are un conținut măsurabil — *mesajul* — și o formă-cod specifică mediului prin care este transmisă. Același mesaj poate primi mai multe forme-coduri. Spre exemplu, două persoane își pot trimite un mesaj în scris, prin vorbire directă, prin gesturi.

Informația semantică = conținutul măsurabil al informației (*mesajul*).

Informația sintactică = forma de reprezentare și codificare a informației (*codul*).

Transmiterea și prelucrarea informației se fac la nivel de cod, mesajul fiind accesibil doar cunoscătorilor. Prin codificare se conservă semantica într-o nouă sintaxă. Culorile, mirosurile, semnalele acustice sau luminoase sunt exemple de coduri prin care organismele lumii vii comunică. Și o valoare numerică poate fi reprezentată în diverse forme, în funcție de sistemul de numerație sau de baza de numerație folosite: spre exemplu, valoarea treizeci și cinci se scrie 35 în baza 10, sau 10011 în baza 2, sau 120 în baza 5.

Informația este rezultatul acțiunii unui element din sistem; ea este prelucrată și transformată printr-o altă acțiune într-o nouă informație care se transmite către alt element din sistem. Astfel are loc interacțiunea și comunicarea dintre elemente.

Scopul comunicării este atins atunci când informația generată de un element sursă ajunge la elementul destinat în timp util luării unei decizii generatoare, la rândul ei, de acțiune și informație. Spre exemplu, în timpul funcționării unui frigider cu termostat, comanda termostatului trebuie să ajungă la dispozitivul de

răcire exact în momentul în care temperatura internă a ajuns la limita fixată, altfel producându-se dezghețarea frigiderului.

Indiferent de specificul sistemului, elementele care generează, păstrează, prelucrează sau transmit informația formează un *sistem informațional*.

Sistemele informaționale controlate de om sunt supuse unor erori sau întârzieri care pot altera atât calitatea, cât și oportunitatea informației. Ce s-ar întâmpla, spre exemplu, dacă, în lipsa termostatului, ar trebui să reglăm manual temperatura frigiderului?

Înlocuirea sau asistarea muncii umane cu echipamente și programe de memorare, prelucrare și transmitere a unor cantități mari de informații, într-un timp cât mai scurt, au dus la informatizarea sistemelor informaționale și transformarea lor în sisteme informatice.

Informatizarea instituțiilor financiar-bancare, a învățământului, a asistenței medicale sunt doar câteva obiective ale societății moderne în care trăim. Pentru a face distincție între informația transmisă și prelucrată convențional (fără calculator) și informațiile transmise și prelucrate automat, acestea din urmă au primit numele de *date*, iar prelucrarea lor se numește *prelucrarea automată a datelor*.

Informatica este știința prelucrării automate a datelor.

Apariția și dezvoltarea informaticii ca știință sunt strâns legate de evoluția calculatoarelor electronice.

Ca știință și domeniu de activitate, informatica este un complex de discipline și specializări orientate spre dezvoltarea arhitecturii calculatoarelor și a sistemelor de operare, ingineria programării, inteligența artificială, automatizări și roboți industriali, sisteme pentru gestiunea bazelor de date, rețele de calculatoare și aplicații hipertext pentru Internet, grafică, animație, jocuri și multe altele.

Întrucât informația există în orice sistem, informatica nu are un domeniu propriu de aplicație. *Informatica este o știință interdisciplinară.*

Ca disciplină școlară, informatica oferă metode de rezolvare a problemelor cu calculatorul și formează abilitățile de gândire, abstractizare și orientare virtuală necesare integrării în lumea informatizată a viitorului.



TEME

- 1** Exemplifică noțiunea de sistem, folosind:
 - a) un exemplu inspirat din lumea înconjurătoare (natură, societate etc.);
 - b) un exemplu dintr-un domeniu de activitate umană.
- 2** Mulțimea cărților din bibliotecă formează un sistem? Dar Internetul? Justifică răspunsul.
- 3** Realizează o prezentare cât mai concisă, în viziune sistemică, a disciplinelor școlare studiate. Urmărește legăturile dintre ele. Există discipline izolate? Ce semnificație ar putea avea acest aspect?

1.2. Impactul informaticii asupra societății

Înlocuirea muncii umane de către calculator a produs o nouă revoluție tehnică: revoluția informatizării.

De la calculatorul industrial, care mobila la începutul anilor 1960 o întreagă încăpere, și până la miniaturizarea care face astăzi posibilă implantarea cipurilor inteligente, capacitatea calculatorului de a modela raționamente umane face posibilă utilizarea sa ca mașină rapidă de calcul, ca instrument de decizie și control în prelucrări sau transmisii de date, sau ca mediu virtual de simulare a unor fenomene ori procese, în scopul învățării sau cercetării.

Calculatorul electronic a devenit un echipament nelipsit de pe masa de lucru a oricărui funcționar bancar, proiectant, agent de turism sau bibliotecar. Distanțele geografice își pierd semnificația, când oamenii comunică prin poșta electronică. Dintr-o mașină de calcul foarte complicată și accesibilă unui număr restrâns de specialiști — așa cum era primul calculator electronic ENIAC (1940) —, calculatorul a devenit astăzi accesibil oricărui copil.

Posibilitățile și limitele unui calculator depind, dincolo de parametrii săi tehnici, de cerințele celui ce folosește calculatorul — *utilizatorul*.

Pentru un copil, un calculator cu platformă de jocuri poate fi suficient. Pentru un elev sau un student, e necesar accesul la Internet și un mediu de programare pentru testarea propriilor raționamente. Pentru fiecare domeniu de aplicație al calculatoarelor există pachete de programe (*software*) specializate în rezolvarea problemelor specifice domeniului respectiv.

Producerea, distribuirea, instalarea și utilizarea pe scară largă a calculatoarelor personale sau interconectate în rețele au condus la apariția unor profesii și specializări noi, cuprinse în categoria generală de *informatician*. Dintre aceste **specializări** enumerăm doar câteva:

- *inginer de sistem* — specialist cu solide cunoștințe tehnice pentru instalarea sau dezvoltarea unor configurații de calculatoare atât la nivel de echipamente (*hardware*), cât și la nivel de programe (*software*);

- *analist* — specialist în analiza sistemelor, în scopul descompunerii la nivel de detaliu și al identificării tuturor datelor și evenimentelor de care depinde rezolvarea unei probleme;

- *programator* — specialist în programarea calculatorului și integrarea programelor în pachete cu interfața cât mai accesibilă utilizatorului;

- *administrator de rețea* — specialist în instalarea și gestionarea resurselor dintr-o rețea de calculatoare;

- *web-designer* — specialist în proiectarea și realizarea paginilor de prezentare pe Internet;

- *operator calculator* — lucrător care introduce datele specifice unui domeniu de aplicație și asigură obținerea la timp a rezultatelor specifice.

Domenii de aplicație:

Activități economice: prelucrarea unor cantități mari de valori financiare, contabile, prețuri, vânzări, salarii, costuri; gruparea și selectarea rapidă, la termene

impuse de specificul activității; prezentarea sugestivă în tabele, rapoarte sau grafice comparative.

Activități industriale: utilizarea mașinilor numerice, a roboților industriali sau a calculatoarelor de proces care măsoară, interpretează și analizează procese industriale.

Proiectare: de la componente electronice la navele spațiale, calculatorul oferă un mediu flexibil pentru realizarea de modele și simularea comportamentului în condiții greu de asigurat în realitate.

Cartografie, geologie, topografie, meteorologie: realizarea hărților și interpretarea comparativă a datelor fizice, economice, demografice, meteorologice, realizarea de previziuni, prognoze, anticiparea unor evenimente cu consecințe sociale.

Medicină, biologie: prelucrarea rapidă a datelor despre pacienți, resurse, medicamente, prelucrarea imaginilor (radiografii, ecografii), simularea comportamentului unor organe sau țesuturi.

Tipografie: introducerea textului, procesarea imaginilor, așezarea în pagină (tehnoredactarea), pretipărirea și păstrarea informațiilor în format electronic (*e-book*).

Cinematografie, artă: de la animație la SF, grafica interactivă a devenit un instrument indispensabil pentru graficienii și scenariștii a căror imaginație este întreținută de capacitatea calculatorului de a inventa forme și efecte scenice.

Lista domeniilor poate continua cu cercetarea, educația și învățământul, producția de jocuri recreative sau dedicate dezvoltării creativității ș.a. Lista nu poate fi practic închisă, căci, pentru aceasta, ar trebui găsit răspuns la întrebarea: Care este domeniul de activitate umană în care calculatorul nu are aplicație?



TEMĂ

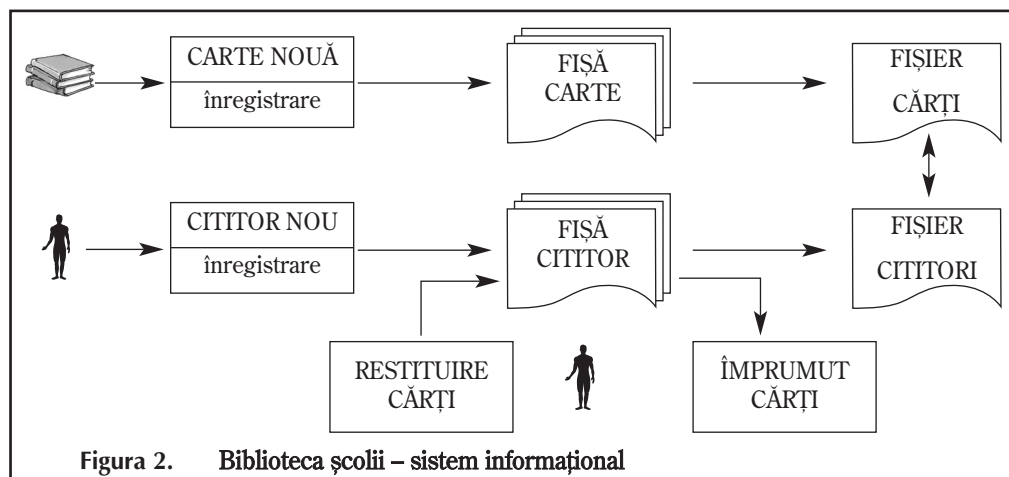
Se dorește informatizarea activității specifice bibliotecii din școala noastră. În acest scop este necesară o echipă de informaticieni.

Cerințe:

- 1** Propune o echipă de lucru, precizând numărul de membri și rolul (specialitatea) fiecăruia.
- 2** Ce rol ți-ar plăcea să primești în această echipă? Justifică răspunsul.

1.3. Studiu de caz. Biblioteca școlii

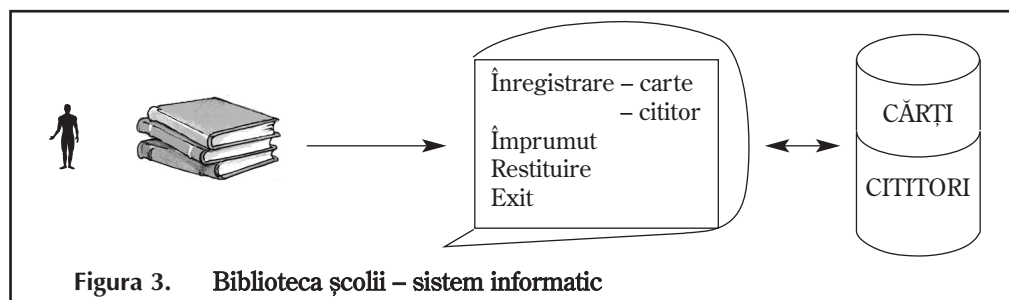
Din mulțimea problemelor specifice activității de bibliotecă (Figura 2), alegem înregistrarea unei cărți noi. Când primește o carte nouă, bibliotecarul completează o fișă în care înregistrează datele specifice cărții: autor, titlu, editura, anul apariției, prețul. Pentru a fi mai ușor de identificat, cartea primește și un *cod*.



Rezolvarea acestei probleme cu ajutorul calculatorului elimină fișa cărții în formatul ei convențional (hârtie).

Când primește o carte nouă, bibliotecarul introduce de la tastatură datele despre carte; acestea sunt memorate pe suport magnetic (disc), de unde pot fi oricând selectate și folosite în alte prelucrări.

Folosirea calculatorului reduce timpul în care bibliotecarul înregistrează date sau caută fișele pentru cărți sau cititori.



Privită ca sistem, biblioteca școlii poate fi reprezentată prin elementele de structură și fluxurile din Tabelul 2.

Tabelul 2. Biblioteca școlii ca sistem

Structura	Fluxuri	Interfețe
<ul style="list-style-type: none"> – cărți – bibliotecar – cititori 	<ul style="list-style-type: none"> – o carte este la dispoziția unui cititor – un cititor poate împrumuta mai multe cărți – bibliotecarul înregistrează cărțile și cititorii 	<ul style="list-style-type: none"> – cataloage cărți – fișier cititori

1.4. Informatica în România – repere istorice

Încă din prima jumătate a secolului al XX-lea au existat preocupări și realizări de seamă ale profesorilor și cercetătorilor români în domenii moderne precum electronica și microelectronica, matematica, cibernetica și tehnica de calcul.

Medicul Ștefan Odobleja (1902–1978), numit și „precursorul român al ciberneticii”, a evidențiat pentru prima dată în lume principiile cibernetiche de autoreglare ale sistemelor biologice, psihologice și sociale. Lucrarea sa *Psihologia consonantistă*, publicată în limba franceză în anii 1938–1939, este un reper în literatura științifică de specialitate.

Matematicianul Grigore Moisil (1906–1973) s-a preocupat de aplicarea logicii matematice în tehnică, dezvoltând teoria algebrică a automatelor. În anul 1957, profesorul Grigore Moisil a inițiat primele cursuri de calculatoare electronice în cadrul Facultății de Matematică din București, susținând școala românească de informatică prin întreaga sa carieră de profesor și cercetător.

În țara noastră, primele calculatoare (CIFA) au fost construite în perioada anilor 1953–1964, de către specialiști de la Institutul de Fizică Atomică din București. Au urmat calculatoarele din generația a treia FELIX 256/512/1024, realizate în perioada 1970–1980, și minicalculatoarele din generația 3½ INDEPENDENT/CORAL, realizate în anii 1981–1989, la Întreprinderea de Calculatoare din București.

În ultimii ani, pătrunderea calculatoarelor personale de ultimă generație pe piața românească de profil a deschis posibilitatea integrării calculatorului în activități cotidiene, prin programe flexibile și accesibile unor largi categorii de utilizatori.

Informatizarea societății românești este un proces dinamic, în continuă evoluție. Pentru a face față diversității și ritmului de evoluție a tehnologiei informatice, trebuie să fim pregătiți să învățăm continuu.

„Utilizarea unui calculator pune în evidență toată capacitatea științifică a celui ce îl întrebuințează. Un calculator utilizat de un om nepriceput nu se strică, e mai rău: lucrează cu randament scăzut.” (Gr. Moisil)



TEMĂ

Miniproiect „pe echipe”. Se dorește informatizarea următoarelor activități: disperatul de cazare într-o tabără școlară; evidența jucătorilor la un club sportiv; rezervări de bilete (concerte, agenție de voiaj); secretariatul școlii.

Cerințe:

- 1** Se formează echipe astfel încât fiecare elev să participe la o singură temă.
- 2** Fiecare echipă întocmește un scurt proiect care să cuprindă: prezentarea activității alese; argumente pentru justificarea informatizării; reprezentarea schematică a sistemului informatic; contribuția fiecărui membru din echipă la realizarea proiectului.
- 3** Fiecare proiect va fi prezentat în fața clasei și va fi „avizat” de celelalte echipe pe baza unui punctaj stabilit de comun acord (se poate puncta conținutul proiectului, realizarea, aspectul, prezentarea, munca în echipă, participarea la dialog).

Rezolvarea algoritmică a problemelor

2

2.1. Descompunerea unui raționament în operații elementare

Rezolvarea problemelor cu calculatorul necesită un mod specific de abordare și soluționare. Orice raționament este descompus în operații elementare de gândire — acele operații care pot fi executate de calculator.

În general, orice calculator „știe” să execute: (1) operații aritmetice: adunare, scădere, înmulțire, împărțire întreagă (determinarea câtului și restului dintre două numere întregi), împărțire zecimală (cu virgulă) și (2) operația logică de a compara între ele două valori. Valorile pe care le prelucrează calculatorul sunt reținute în zone de memorie.

Fiecare operație este transmisă calculatorului printr-o comandă (instrucțiune) specifică limbajului de programare — limbajul accesibil calculatorului. Executând comenzile una câte una, exact în ordinea în care le-a primit, calculatorul rezolvă problema. Timpul în care calculatorul rezolvă o problemă depinde de numărul comenzilor pe care acesta le execută și de durata fiecărei comenzi.

O problemă este rezolvată corect doar dacă secvența comenzilor transmise corespunde unui raționament corect. De aceea, înainte de a transpune raționamentul în instrucțiuni accesibile calculatorului, verificăm pas cu pas raționamentul, analizând și executând noi înșine fiecare operație elementară. Avem astfel posibilitatea să îmbunătățim raționamentul, înainte de a-l proba cu calculatorul.

Alcătuirea secvenței de operații elementare se face pas cu pas, în ordinea logică ce conduce la determinarea rezultatelor cerute în enunț.

La alcătuirea secvenței se urmărește folosirea eficientă a resurselor calculatorului (memorie și timp de lucru).

Pentru exemplificare și exercițiu prezentăm următoarele aplicații:

Aplicația 1 CASĂ DE MARCAT

Ne aflăm într-un magazin unde se aplică reduceri la rechizite școlare. Prețul unui caiet este redus cu 20%, dacă un client cumpără între 5 și 7 caiete, sau cu 50%, dacă el cumpără cel puțin 8 caiete. Cum determină mașina automată de marcat valoarea pe care trebuie s-o achite cumpărătorul?

La prima vedere, răspunsul poate fi dat cu ușurință: la casa de marcat este înregistrat prețul fiecărui tip de caiet și numărul de bucăți (cantitatea) cumpărate. Dacă este cazul, se aplică reducerea, apoi se calculează valoarea caietelor.

Pentru ca mașina să determine corect cât trebuie să plătim, ea trebuie învățată să aplice procentul de reducere corespunzător numărului de caiete cumpărate, să calculeze prețul de vânzare și valoarea pentru fiecare tip de caiet, să calculeze valoarea totală a caietelor cumpărate de fiecare client în parte.

Fiecare operație de care depinde rezultatul final trebuie executată corect și la timp. Descompusă pas cu pas, secvența de operații pe care le execută mașina automată de marcat este următoarea:

Început secvență

Pas 1. Mașina anulează valoarea totală a cumpărăturilor făcute de clientul precedent. Pentru un nou client, totalul pornește de la zero.

total ← 0

Pas 2. Mașina înregistrează prețul pentru un tip de caiet.

preț ← preț caiet

Pas 3. Mașina înregistrează numărul de caiete cumpărate (cantitate).

cantitate ← numărul de caiete

Pas 4. Mașina determină procentul de reducere.

Pas 4.1. Dacă avem (cantitate ≤ 4), atunci **procent ← 0**

Pas 4.2. Dacă avem (cantitate ≥ 5) și (cantitate ≤ 7), atunci **procent ← 20**

Pas 4.3. Dacă avem (cantitate ≥ 8), atunci **procent ← 50**

Pas 5. Mașina calculează reducerea pe care o va aplica prețului.

reducere ← preț * procent / 100

Pas 6. Mașina calculează prețul de vânzare.

preț vânzare ← preț - reducere

Pas 7. Mașina calculează valoarea caietelor.

valoare ← cantitate * preț vânzare

Pas 8. Mașina adaugă valoarea caietelor la totalul cumpărăturilor.

total ← total + valoare

Mașina repetă secvența de operații de la **Pas2** la **Pas8** până când au fost înregistrate toate caietele de diverse tipuri care au fost cumpărate.

Pas 9. Mașina afișează totalul cumpărăturilor.

afișează (total)

Sfârșit secvență

Raționamentul propus folosește șapte zone de memorie. Timpul de lucru al calculatorului depinde de numărul operațiilor elementare efectuate, de durata fiecărei operații și de numărul de tipuri de caiete cumpărate.



TEME

- 1** Analizează secvența operațiilor pas cu pas și răspunde la următoarele întrebări:
 - a) Ce se întâmplă dacă mașina nu execută Pasul 1?
 - b) Ce se întâmplă dacă mașina repetă secvența de operații Pas 1 – Pas 8?

- c) Ce se întâmplă dacă mașina nu execută Pasul 2?
 d) Ce se întâmplă dacă mașina repetă secvența Pas 2 – Pas 7?

2 Execută pas cu pas secvența de operații și determină valoarea afișată de mașină pentru următorul exemplu:

Un cumpărător a pus în coșul de cumpărături 6 caiete tip 1, 10 caiete tip 2 și 3 caiete studentești.

Pentru ușurința calculului, construiește pe caiet și completează următorul tabel de variație (coloanele tabelului reprezintă valorile reținute la un moment dat în fiecare zonă de memorie):

Tip caiet	Preț	Cantitate	Procent	Reducere	Preț vânzare	Valoare	Total
tip 1	10.000	6 buc.					
tip 2	15.000	10 buc.					
studentesc	30.000	3 buc.					

Aplicația 2 ROBOT 1

Firma Robot pregătește o mașină automată care să decupeze șabloane rotunde din bucăți dreptunghiulare de material (pânză, carton). Mașina trebuie să decupeze șabloane cât mai mari, întrucât resturile nu se pot folosi.

Informaticienii firmei trebuie să compună secvența pas cu pas a operațiilor prin care, înainte de croire, mașina determină și afișează numărul de șabloane decupate și cantitatea de material rămasă (restul). Ei au primit următoarele modele:

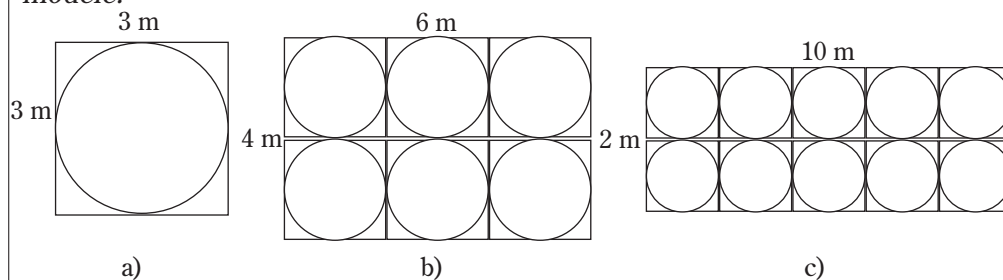


Figura 1. Modele pentru decupaj

Iată cum au procedat informaticienii pentru rezolvarea acestei probleme. După ce au analizat toate modelele prezentate, ei au formulat următoarele întrebări:

1. Poate croi mașina orice suprafață de material? Care sunt dimensiunile minime, respectiv, maxime?
2. Suprafețele decupate dintr-o bucată de material sunt egale între ele?

Răspunsurile primite:

1. Dimensiunile bucăților de material vor fi mai mari de 20 cm, dar nu vor depăși 100 m.

2. Toate suprafețele decupate din aceeași bucată de material sunt egale între ele.

Ținând seama de modelele prezentate și de răspunsurile primite la întrebări, informaticienii au formulat următorul raționament de rezolvare:

a) Se verifică dimensiunile materialului — să se încadreze în limitele precizate. Pentru ușurința calculelor se lucrează în metri.

b) Se calculează suprafața materialului înainte de croire (S).

c) Întrucât mașina trebuie să execute cât mai puține operații de croire, se va determina numărul minim de suprafețe rotunde egale între ele ce pot fi decupate.

d) Dacă bucata de material are laturile egale (Figura 1a), soluția este chiar cercul înscris în pătratul respectiv.

e) Pentru bucățile dreptunghiulare, se va căuta descompunerea dreptunghiului într-un număr cât mai mic de pătrate egale între ele și cu latura cât mai mare.

Din exemplele prezentate rezultă:

— pentru cazul b) bucata de material poate fi descompusă în 6 pătrate cu latura de 2 m;

— pentru cazul c) bucata de material poate fi descompusă în 10 pătrate cu latura de 1 m.

Un dreptunghi cu laturile de 24 m și 18 m va fi descompus în 12 pătrate cu latura (L) de 6 m. Se poate demonstra că latura pătratului (L) este egală cu cel mai mare divizor comun (cmmdc) dintre lungimea și lățimea dreptunghiului.

Pentru determinarea restului, se procedează astfel: cunoscând latura unui pătrat, se determină raza cercului înscris și aria acestuia; aria se înmulțește cu numărul de pătrate; valoarea rezultată se scade din suprafața dreptunghiului inițial.

Iată și secvența pas cu pas prin care mașina determină, înainte de croire, numărul de forme decupate și restul de material.

Început secvență

Pas 1. Mașina primește dimensiunile bucății dreptunghiulare:

înregistrează (Lățime, Lungime).

Pas 2. Mașina verifică dacă bucata de material poate fi croită:

Lățime \in [0.2,100] Lungime \in [0.2,100].

Pas 3. Dacă se poate croi,
atunci

Pas 4. Mașina calculează suprafața inițială:

S \leftarrow Lățime x Lungime.

Pas 5. Mașina determină cmmdc (Lățime, Lungime):

D \leftarrow cmmdc (Lățime, Lungime).

Pas 6. Mașina calculează numărul de pătrate:

N \leftarrow (Lățime/D) x (Lungime/D).

Pas 7. Mașina calculează raza cercului înscris în pătrat:

$$R \leftarrow D/2.$$

Pas 8. Mașina calculează aria cercului:

$$A \leftarrow \pi \times R \times R.$$

Pas 9. Mașina calculează suprafața totală ce va fi decupată:

$$T \leftarrow N \times A.$$

Pas 10. Mașina calculează restul de material:

$$\text{Rest} = S - T.$$

Pas 11. Mașina afișează numărul de forme ce vor fi decupate și restul de material:

afișează (N, Rest).

altfel

Pas 12. Mașina afișează un mesaj de eroare:

afișează ('Nu se poate croi').

Sfârșit secvență

Raționamentul propus folosește nouă zone de memorie. Timpul de lucru al calculatorului depinde de numărul operațiilor elementare efectuate și de durata fiecărei operații.

Expertul firmei Robot a apreciat munca echipei de informaticieni, dar a făcut următoarea observație: întrucât în secvența pas cu pas nu se precizează cum se determină cel mai mare divizor comun, secvența nu poate fi transmisă calculatorului. Echipa de informaticieni va rezolva și această problemă.



TEME

1 Analizează secvența pas cu pas și precizează ce operații matematice trebuie să cunoască calculatorul pentru a executa comenzile din secvență.

2 Execută secvența de operații pas cu pas și completează pe caiet tabelul de variație:

lungime	lățime	S	D	N	R	A	T	Rest
24	18							

3 Cum poate fi redus numărul zonelor de memorie pe care le folosește calculatorul?

4 Prezintă un raționament pentru determinarea celui mai mare divizor comun a două numere.

5 Căută și alte soluții de rezolvare a cerințelor acestei probleme.

2.2. Rezolvarea problemelor cu calculatorul. Etape

Vrem să rezolvăm o problemă cu calculatorul. Ce avem de făcut?

La început, vom lucra *singuri, fără calculator*. Parcurgem următorii pași:

Pas 1. Citim cu atenție enunțul problemei — trebuie să stabilim și să înțelegem cerința.

Pas 2. Determinăm elementele cunoscute și elementele necunoscute din problemă — ceea ce „se dă” și ceea ce „se cere”, sau datele de intrare și datele de ieșire.

Pas 3. Notăm elementele problemei cu litere sau cuvinte sugestive pentru a stabili mai ușor condițiile sau relațiile dintre ele.

Pas 4. Formulăm exemple cu valori concrete și rezolvăm problema pentru fiecare caz în parte. Reținem situațiile pentru care problema are o rezolvare specială sau nu are soluție.

Pas 5. Stabilim un raționament general de rezolvare a problemei, valabil pentru toate cazurile — *algoritmul* problemei.

Pas 6. Reprezentăm algoritmul într-o formă cât mai simplă și mai clară.

Pas 7. Verificăm raționamentul pentru valori concrete.

Pas 8. Implementăm algoritmul în instrucțiunile limbajului de programare și obținem un *program*.

Din acest moment, avem nevoie de *calculator*.

Ce facem noi

Pas 9. Transmitem calculatorului programul prin editare de text și comenzi pentru lucrul cu fișiere.

Pas 10. Transmitem comanda de compilare (COMPILE).

Pas 11. Corectăm și recompilăm programul până când compilatorul nu mai găsește nici o eroare.

Pas 12. Transmitem comanda de execuție (RUN) și introducem valori pentru datele de intrare.

Ce face calculatorul

Înregistrează programul într-un *fișier sursă*.

Compilează programul: verifică corectitudinea sintactică a programului și semnalează erorile.

Execută programul instrucțiune cu instrucțiune și afișează pe ecran rezultatele (*datele de ieșire*).

Pas 13. Verificăm soluțiile problemei pentru exemple concrete, cât mai variate.

Pas 14. Comparăm rezultatele obținute de calculator cu cele obținute prin rezolvarea „de mână”. Când rezultatele diferă, e semn că s-a strecurat o greșală în program sau chiar în raționament. În acest caz, o luăm de la capăt, nu înainte de a *demonstra matematic* corectitudinea algoritmului.

Schema generală a etapelor de rezolvare a problemelor cu calculatorul este prezentată în Figura 2.

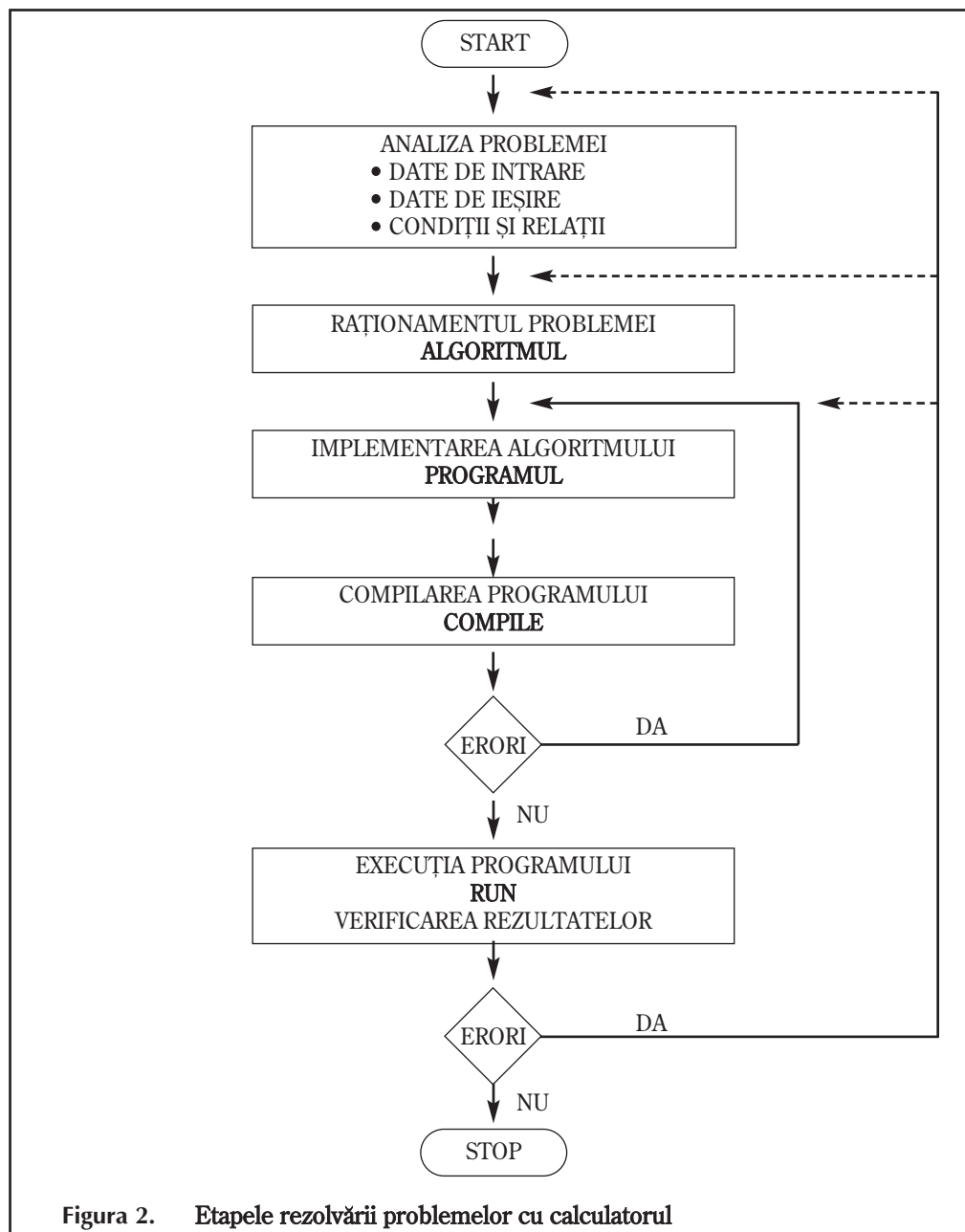


Figura 2. Etapele rezolvării problemelor cu calculatorul

2.3. Etapa de analiză

Oricât de simplă sau complicată este o problemă, rezolvarea ei cu calculatorul necesită o analiză până la detaliu atât a datelor, cât și a evenimentelor și restricțiilor ce pot apărea pe parcursul rezolvării.

Analiza și clasificarea datelor

Din enunțul problemei sunt extrase **datele explicite** (clar formulate), spre exemplu *cantitatea, prețul, lungimea și lățimea* dreptunghiului, precum și **datele implicite** (nu totdeauna precizate) necesare în rezolvare: *prețul de vânzare sau latura pătratului*.

Se grupează datele după *gradul de cunoaștere*:

— *date de intrare* (ceea ce se cunoaște) — spre exemplu: *prețul*;

— *date de ieșire* (ceea ce se determină — rezultatele) — spre exemplu: *restul de material*;

— *date intermediare* sau de *manevră* (valori temporare necesare pentru determinarea rezultatelor) — spre exemplu: *numărul de pătrate* în care se descompune bucata dreptunghiulară de material.

Se urmărește *stabilitatea datelor*.

Datele care-și schimbă valoarea pe parcursul rezolvării se numesc *variabile* (spre exemplu, *Total*), iar cele care-și păstrează aceeași valoare se numesc *constante* (constanta matematică π).

Se determină *natura* sau *tipul* datelor.

Tipuri de date

— *numerice* (*valori întregi, naturale sau reale*);

— *caracter* (orice simbol de pe tastatură);

— *șir de caractere*;

— *logice* cu semnificația **adevărat** sau **fals** (în general, datele cu două valori posibile = masculin/feminin, promovat/nepromovat, achitat/neachitat).

Ținând seama de criteriile de grupare ce pot fi folosite în analiză, datele unei probleme pot fi clasificate ca în Figura 3.

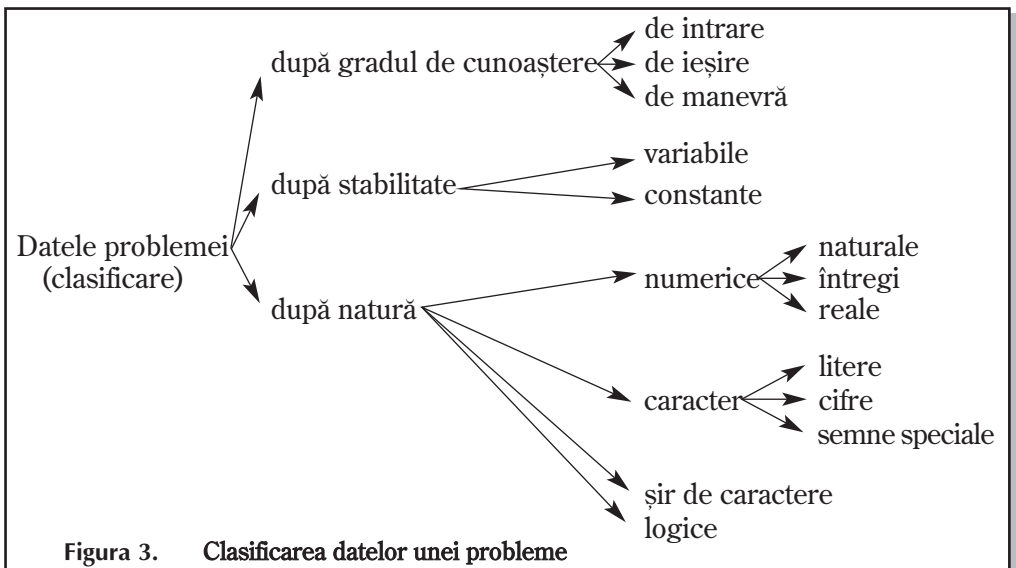


Figura 3. Clasificarea datelor unei probleme

Analiza evenimentelor și a restricțiilor

Evenimentele sunt situații previzibile de care depinde execuția unor operații pe parcursul rezolvării unei probleme.

Evenimentele se desprind din enunțul problemei sau se deduc din analiza unor exemple.

În aplicația *Casa de marcat*, fiecare tip de caiet reprezintă un eveniment, iar numărul de caiete cumpărate (cantitatea) influențează procentul de reducere a prețului:

- dacă avem (*cantitate* ≤ 4) atunci, *procent* = 0;
- dacă avem (*cantitate* ≥ 5) și (*cantitate* ≤ 7) atunci, *procent* = 20;
- dacă avem (*cantitate* ≥ 8) atunci, *procent* = 50.

Restricțiile sunt condiții care se pun anumitor date din problemă referitoare la proprietăți sau domenii de valori ale acestora. Se consideră corecte (*valide*) doar datele care îndeplinesc restricțiile din problemă. Încercarea de rezolvare a problemei cu date de intrare invalide poate avea consecințe logice greu de controlat în contextul problemei.

În aplicația *Robot 1*, laturile dreptunghiului trebuie să aparțină domeniului de valori $[0.2, 10]$; decupajul dintr-o bucată de material prea mică poate pune mașina în imposibilitatea de a croi.



TEME

- 1** Realizează clasificarea și gruparea datelor din aplicația *Casa de marcat*.
- 2** Realizează clasificarea și gruparea datelor din aplicația *Robot 1*.
- 3** Formulează o problemă ale cărei date de intrare să fie de tip întreg (spre exemplu, notele unui elev), iar datele de ieșire să fie de tip real (spre exemplu, media).
- 4** Formulează o problemă ale cărei date de intrare să fie de tip real (spre exemplu, media unui elev), iar datele de ieșire să fie de tip logic (spre exemplu, promovat).
- 5** Se cunosc numele și data nașterii unei persoane; data este specificată prin zi, lună, an.
Se dorește afișarea pe ecranul calculatorului a numelui și a unui mesaj.
Exemplu:

Nume	zi	luna	an	mesaj
Ion	28	02	1995	este elev
Ana	29	02	1995	luna eronată
Maria	30	08	1946	nu este elevă

Cerințe:

1. Formulează și alte exemple care să pună în evidență cât mai mult dintre mesajele posibile.
2. Formulează un exemplu de situație reală în care este necesară rezolvarea acestei probleme.
3. Precizează care sunt evenimentele specifice acestei probleme.
4. Precizează care sunt restricțiile din această problemă.
5. Oferă exemple de date invalide specifice acestei probleme.
6. Descrie raționamentul de rezolvare prin secvența pas cu pas a operațiilor elementare de gândire.

2.4. Noțiunea de algoritm. Proprietăți

Algoritm = succesiune precisă și finită de operații elementare ale gândirii, prin care datele de intrare ale problemei sunt transformate în date de ieșire.

Deși foarte modern, fiind specific astăzi informaticii, termenul de algoritm provine dintr-o formă mai veche, **algorism**, care însemna procesul efectuării operațiilor aritmetice, folosind cifrele arabe. Cuvintele „algoritm” și „algebră” sunt legate de numele matematicianului arab Abu Ja Far Mohammed Ibn Mûsâ al Khowarismî și de lucrarea sa *Kitab al-jabr wa'l-muqâbala*, scrisă în anul 825 d.Hr. în care sunt tratate operațiile asupra numerelor.

În vorbirea curentă, algoritmul reprezintă un proces sau procedeu prin care se rezolvă în amănunt o problemă oarecare, nu neapărat de matematică.

Proprietățile algoritmilor

a) Claritatea — datele și operațiile algoritmului să fie descrise foarte exact, fără ambiguități, să nu necesite completări.

Spre exemplu, pentru rezolvarea problemei:

„Cunoscând laturile unui triunghi, să se determine aria acestuia”, se propune următoarea secvență:

Pas 1. Se află laturile triunghiului.

Pas 2. Se aplică formula pentru determinarea ariei.

Pas 3. Se afișează aria triunghiului.

Secvența propusă nu reprezintă un algoritm, întrucât „se află laturile” și „se aplică formula” nu sunt acțiuni suficient de clar formulate: care este formula?

Transformarea secvenței în algoritm necesită clarificări și completări.

Pas 1. Se introduc laturile **a**, **b**, **c** (date de intrare).

Pas 2. Se calculează semiperimetrul $p = (a+b+c)/2$.

Pas 3. Se calculează aria cu formula $A = \sqrt{p(p-a)(p-b)(p-c)}$.

Pas 4. Se afișează aria (date de ieșire).

b) Generalitatea — un algoritm rezolvă o mulțime de probleme de același fel (clasă de probleme).

Exemplu:

Rezolvarea unei probleme de genul „Să se determine perimetrul unui triunghi cu laturile de 3 cm, 4 cm și 5 cm” nu conduce la un algoritm, întrucât formularea este pentru un caz particular, cu valori concrete. Pentru generalizare, abstractizăm problema: „Să se determine perimetrul oricărui triunghi cu laturile egale cu valori oarecare de a cm, b cm, c cm”. În acest caz, rezolvarea problemei conduce la un algoritm, întrucât oricare ar fi triunghiul dat va exista o soluție proprie calculată după relația

$$\text{perimetru} = a + b + c.$$

c) Completitudinea — tratarea cazurilor speciale. De exemplu, valorile a , b , c care nu formează triunghi.

d) Finitudinea — transformarea datelor de intrare în date de ieșire să se facă într-un timp măsurabil finit.

Calculatorul execută instrucțiunile transmise prin program și, la un moment dat, va trebui să se oprească, să termine ceea ce a început și să prezinte rezultatele.

O problemă de genul „Să se determine *toate* puterile unui număr întreg x ” nu poate fi rezolvată de nici un calculator. Secvența valorilor cerute este de forma x , x^2 , x^3 , x^4 , x^5 ... și reprezintă o secvență fără sfârșit, **infinită**, care nu poate fi algoritmicizată. Deși valorile se obțin simplu prin înmulțirea repetată cu x , nu există o condiție de oprire. Până când va repeta calculatorul înmulțirea? Sau cât timp? În acest caz, problema trebuie reformulată prin introducerea unei condiții.

De exemplu: „Să se determine toate puterile, *mai mici decât 1.000*, ale unui număr întreg oarecare x ”.

Pentru această cerință se poate formula un algoritm: „Se repetă înmulțirea lui x cu el însuși, până când se obține o valoare mai mare decât 1.000”.

În acest caz, secvența operațiilor efectuate de calculator este numărabilă, adică finită.

e) Eficiența — algoritmul stă la baza programului executat de calculator. În timpul execuției sunt folosite resursele calculatorului — memorie, procesor. Se alege algoritmul care folosește memoria strict necesară (minimă) și se termină într-un timp cât mai scurt (minim).

Particularități

Pentru o clasă de probleme pot exista mai mulți algoritmi — gândirea algoritmică este creativă. Orice rezolvitor poate construi propriul său algoritm care să respecte toate proprietățile.

Există probleme pentru care se poate elabora cel puțin un algoritm (*probleme decidabile*).

Există probleme pentru care nu se pot construi algoritmi specifici — probleme în care acțiunile depind de evenimente întâmplătoare sau imprevizibile (*probleme nedecidabile*).



TEME

- 1** Analizează rezolvarea problemei *Casa de marcat* și precizează dacă secvența propusă reprezintă un algoritm. Justifică răspunsul.
- 2** Analizează rezolvarea problemei *Robot 1* și precizează dacă secvența propusă reprezintă un algoritm. Justifică răspunsul.
- 3** Se prezintă următoarea secvență de operații:
 - Pas 1.** Se introduc laturile **a, b, c** (date de intrare).
 - Pas 2.** Se calculează semiperimetrul $p = (a+b+c)/2$.
 - Pas 3.** Se calculează aria cu formula $A = \sqrt{p(p-a)(p-b)(p-c)}$.
 - Pas 4.** Se afișează aria (date de ieșire).
 - Pas 5.** Reia de la **Pasul 1**.
 Cerințe:
 - 3.1.** Precizează dacă secvența **Pas 1 – Pas 4** reprezintă un algoritm. Justifică răspunsul.
 - 3.2.** Precizează dacă secvența **Pas 1 – Pas 5** reprezintă un algoritm. Justifică răspunsul.
 - 3.3.** Modifică secvența propusă astfel încât ea să respecte toate proprietățile algoritmilor.

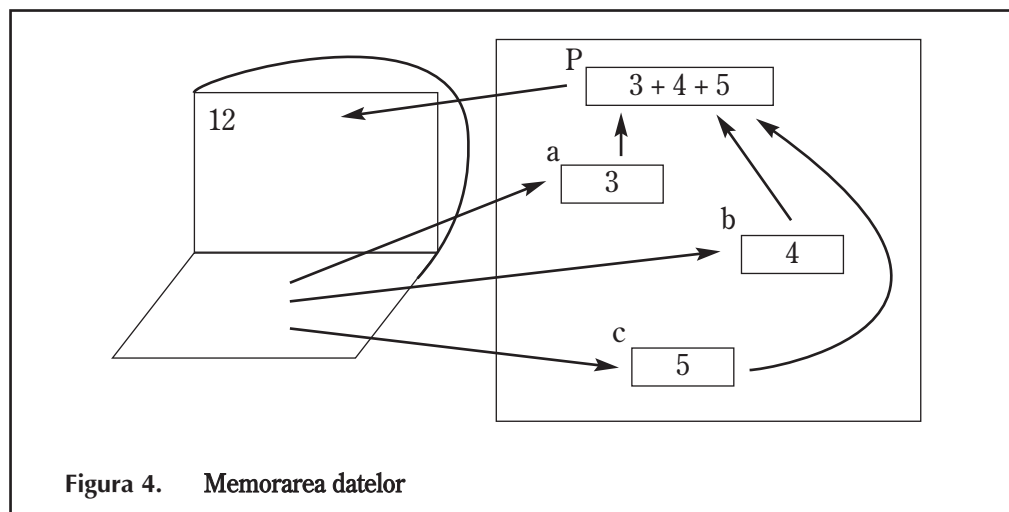
2.5. Datele cu care lucrează algoritmi

Algoritmul de rezolvare a unei probleme lucrează cu datele rezultate din analiza problemei. Algoritmul transformă *datele de intrare* ale problemei în *date de ieșire*. Transformarea efectivă are loc în timpul execuției programului.

Datele – valori concrete specifice problemei – vor fi reținute de calculator în zone de memorie solicitate prin instrucțiuni specifice limbajului de programare. Dimensiunea zonelor de memorie depinde de tipul datelor. Zonele de memorie rămân la dispoziția programului cât timp durează execuția acestuia. Memoria calculatorului este o resursă limitată care trebuie folosită cu măsură. Programatorul are sarcina de a se gândi atât la algoritmul de rezolvare a problemei, cât și la folosirea eficientă a memoriei.

Intuitiv, memoria calculatorului poate fi asemănată cu o suprafață formată din puncte (locații) învecinate (adiacente). Fiecare locație are o adresă prin care poate fi identificată și accesată. Mai multe locații formează o zonă de memorie cu o dimensiune sau capacitate bine definite.

Spre exemplu, pentru determinarea perimetrului **P** al unui triunghi cu laturile **a, b, c**, sunt necesare patru zone de memorie (Figura 4).



Pentru calculator, valorile sau datele unei probleme sunt informații codificate prin doar două valori (0 și 1) specifice sistemului **binar** (baza de numerație 2).

Dimensiunea sau capacitatea zonelor de memorie se exprimă în unități de măsură specifice informației. Cea mai mică unitate de măsură pentru informație este **bitul**.

Multiplii bitului sunt:

- 1 Byte (B) = 8 biți = 1 octet
- 1 KiloByte (KB) = 2^{10} B = 1024 B
- 1 MegaByte (MB) = 2^{10} KB
- 1 GigaByte (GB) = 2^{10} MB
- 1 TerraByte (TB) = 2^{10} GB

Reguli pentru utilizarea memoriei:

- Datele problemei sunt memorate în zone de memorie care pot fi adresate din program (algoritm) prin nume asociate lor.
 - O zonă de memorie păstrează la un moment dat o singură valoare.
 - Valoarea reținută într-o zonă de memorie poate fi modificată prin operații matematice.
 - Conținutul unei zone de memorie poate fi copiat (transmis) în altă zonă de memorie.
 - Conținutul unei zone de memorie poate fi comparat cu conținutul altei zone de memorie.
 - O zonă de memorie poate primi valori prin citire direct de la tastatură.
 - Conținutul unei zone de memorie poate fi afișat prin scriere pe ecranul calculatorului sau la imprimantă.

2.6. Operații asupra datelor

Transformarea datelor de intrare în date de ieșire, conform raționamentului de rezolvare a problemei, se face prin operații elementare corespunzătoare tipurilor de date din problemă.

Fiecare tip de dată poate fi prelucrat prin operatori specifici.

A. Operații și operatori pentru datele numerice (Tabelul 1)

Operator	Operație
adunare	+
scădere	-
înmulțire	*
câtul împărțirii	div
restul împărțirii	mod
împărțirea zecimală	/ (numai pentru valori reale)

Tabelul 1.

Datele și operatorii pot fi grupați, cu ajutorul parantezelor, în expresii care respectă regulile de prioritate cunoscute din matematică.

Exemplu: pentru determinarea valorii caietelor cumpărate de un client (aplicația *Casa de marcat*), se poate construi următoarea expresie:

$$\text{valoare} = \text{cantitate} * (\text{preț} - \text{preț} * \text{procent} / 100)$$

B. Operații relaționale

Pentru determinarea unor evenimente sau verificarea unor restricții, este necesară compararea datelor de același tip prin formularea unor condiții.

Pentru formularea condițiilor, se folosesc enunțuri simple numite **propoziții și operatori relaționali**:

- operatorul de egalitate (=);
- operatorul diferit (< >);
- operatorul mai mare (>);
- operatorul mai mare sau egal (>=);
- operatorul mai mic (<);
- operatorul mai mic sau egal (<=).

În aplicația *Casa de marcat*, stabilirea procentului de reducere a prețului s-a făcut prin operații relaționale: $\text{cantitate} \leq 4$, $\text{cantitate} \geq 5$, $\text{cantitate} \geq 8$.

C. Operații logice

În multe situații, mai multe condiții simple formează o condiție compusă. Spre exemplu: „Dacă ai înălțimea peste 1,70 și dacă ai greutatea sub 60 kg, te poți înscrie la concursul de modeling”. Cele două condiții simple „dacă ai înălțimea peste 1,70” și „dacă ai greutatea sub 60 kg” condiționează în același timp înscrierea la concurs. Compunerea lor s-a făcut cu operatorul logic ȘI (AND) numit **conjuncție**.

Un alt exemplu: „Dacă nu mă prezint la un antrenament sau dacă ratez o minge, sunt eliminat din echipă”. În acest caz, compunerea s-a făcut cu operatorul logic SAU (OR) numit **disjuncție**. S-a folosit și **negația**, NU (NOT), prin care valoarea de adevăr a unei propoziții este transformată în opusul ei.

Operatorii logici NOT, AND, OR se aplică în această ordine de priorități valorilor de adevăr — **adevărat** (A) și **fals** (F) — ale propozițiilor ce compun condiția.

Stabilirea corectă a valorii de adevăr pentru o condiție compusă se face conform regulilor de compunere a operatorilor logici:

NOT	A	F
	F	A

AND	A	F
A	A	F
F	F	F

OR	A	F
A	A	A
F	A	F

Aplicația 3 SEMAFOR

Ca exercițiu, să considerăm un semafor cu două faze: verde și roșu.

- Dacă semaforul este verde, atunci traversez.

Pentru stabilirea valorii de adevăr în toate cazurile posibile, scriem propoziția din condiție (P).

P: „semaforul este verde” cu semnificația **adevărat**. Negația acestei propoziții este de forma:

P1: „semaforul nu este verde” adică „semaforul este roșu”.

P1 = NOT (P).

Dacă negăm și propoziția **P1** obținem:

P2 = NOT (P1): „semaforul nu este roșu” adică „semaforul este verde”, care este chiar propoziția **P**.

Putem scrie **P2 = NOT (P1) = NOT (NOT (P)) = P**.

- Dacă semaforul este verde și dacă toate mașinile au oprit, atunci traversez.

În acest caz, avem de evaluat o condiție compusă.

Scriem propozițiile simple din condiție:

P1: „semaforul este verde” cu semnificația **adevărat**.

P2: „toate mașinile au oprit” cu semnificația **adevărat**.

Rezultă:

NOT (P1): „semaforul este roșu”.

NOT (P2): „mai trece cel puțin o mașină”.

Realizăm tabelul de adevăr pentru operatorul **AND**.

P1 AND P2	A	F
A	A	F
F	F	F

Conform enunțului, traversarea se face când condiția compusă este adevărată. Din tabel rezultă că valoarea **adevărat** se obține când **P1** este adevărată („semaforul este verde”) și **P2** este adevărată („toate mașinile au oprit”).

Folosind operatori relaționali și operatori logici, se pot construi expresii logice — expresii cu valoarea **adevărat** sau **fals**.

Exemplu: expresia $43 + x \geq 100$

este o expresie logică cu valoarea **adevărat** pentru orice valoare $x \geq 57$.

De reținut!

Rezolvarea problemelor cu calculatorul necesită următoarele etape:

1. Analiza problemei.
2. Formularea raționamentului de rezolvare.
3. Compunerea raționamentului pas cu pas. Algoritmul problemei.
4. Implementarea algoritmului.

Se codifică algoritmul într-un limbaj accesibil calculatorului — limbajul de programare. Un limbaj de programare se aseamănă cu o limbă străină: are un alfabet, vocabular sau dicționar de cuvinte, sintaxă și reguli gramaticale. Exemple de limbaje de programare: *Basic*, *Pascal*, *C*, *Java*. Programatorul alege limbajul de programare în funcție de specificul problemei și de propria sa experiență.

În general, un algoritm poate fi codificat în orice limbaj de programare. În acest manual, algoritmi vor fi implementați în limbajul *Pascal*. Elementele de vocabular și sintaxă specifice acestui limbaj de programare sunt prezentate succint în *Anexă*.



TEME

1 Evaluează următoarele expresii pentru x și y întregi, cu valorile $x = 3$ și $y = 2$.

- a) $x * y + x * (4 * y - 2 * x) =$
- b) $(x + y) > 7 =$
- c) $x \text{ div } y =$
- d) $3 * x \text{ div } y =$

- e) $x \bmod y =$
- f) $x/y =$
- g) $3 * (x \text{ div } y) =$
- h) $1 + x * y - x/y =$
- i) $x + 2 * y * (5 - (2 * x)/y) =$

2 În ce condiții pot fi calculate următoarele expresii?

- a) $E = 1/x + 1 / y$
- b) $E = 1 / (x + y)$

3 Evaluează următoarele expresii logice:

- a) fals și adevărat =
- b) fals și adevărat sau adevărat =
- c) fals sau adevărat și adevărat =
- d) adevărat și fals sau nu adevărat =

4 Fie a = adevărat; b = adevărat; c = fals.
Evaluează următoarele expresii logice:

- a) not (a or b) and (b or c) =
- b) not (a and b) or (b and c) =
- c) not (a or b) =
- d) not (a and b) =
- e) a or b and (c or b) =

5 Evaluează următoarele expresii:

- a) $a + b < a - 3$ pentru $a = 7$ și $b = -3$
- b) $a + 2.5 > 5$ pentru $a = 1.5$
- c) $(a < 2)$ AND $(a > 10)$ pentru $a \in \mathbb{R}$
- d) $(a < 2)$ OR $(a > 5)$ pentru $a \in \mathbb{R}$

6 Formulează expresii logice care să corespundă următoarelor cerințe:

- a) $x \in [0, 100]$
- b) intervalele $[a, b]$ și $[c, d]$ sunt disjuncte
- c) intervalul $[a, b]$ este inclus în intervalul $[c, d]$

CUPRINS

CAPITOLUL 1. ÎN LUMEA INFORMATICII	3
1.1. Informatica – știință interdisciplinară	3
1.2. Impactul informaticii asupra societății	6
1.3. Studiu de caz. Biblioteca școlii	7
1.4. Informatica în România – repere istorice	9
CAPITOLUL 2. REZOLVAREA ALGORITMICĂ A PROBLEMELOR	10
2.1. Descompunerea unui raționament în operații elementare	10
2.2. Rezolvarea problemelor cu calculatorul. Etape	15
2.3. Etapa de analiză	16
2.4. Noțiunea de algoritm. Proprietăți	19
2.5. Datele cu care lucrează algoritmi	21
2.6. Operații asupra datelor	23
CAPITOLUL 3. ALGORITMI ELEMENTARI	27
3.1. Reprezentarea algoritmilor în pseudocod	27
3.2. Principiile programării structurate	28
3.3. Algoritmi liniari (secvențiali)	30
3.4. Algoritmi cu structuri alternative	33
3.5. Algoritmi cu structuri repetitive	39
3.6. Algoritmi fundamentali. Aplicații	47
3.7. Probleme propuse	65
CAPITOLUL 4. APLICAȚII INTERDISCIPLINARE	67
4.1. Geometrie plană	67
4.2. Ecuații	68
4.3. Determinarea masei moleculare a unui compus chimic	69
4.4. Determinarea molarității unei soluții de acid	70
4.5. Determinarea normalității unei soluții de acid	71
4.6. Parașutiștii	72
4.7. La țintă	74
4.8. Boli contagioase	74
4.9. Croitorul	75
4.10. Probleme propuse	76
Anexă	78