

Mihaela Giurgiulescu  
Valeriu Benedicth Giurgiulescu



# Informatică și TIC

## Clasa a VII-a



Acest manual este proprietatea Ministerului Educației.

Acest manual școlar este realizat în conformitate cu Programa școlară aprobată prin Ordinul ministrului educației naționale nr. 3393/28.02.2017.

119 – număr unic de telefon la nivel național pentru cazurile de abuz împotriva copiilor

116.111 – numărul de telefon de asistență pentru copii

Mihaela Giurgiulescu  
Valeriu Benedicth Giurgiulescu



# Informatică și TIC

## Clasa a VII-a



Manualul școlar a fost aprobat de Ministerul Educației prin ordinul de ministru nr. 5420/04.07.2024.  
Manualul este distribuit elevilor în mod gratuit, atât în format tipărit, cât și în format digital, și este transmisibil timp de patru ani școlari, începând din anul școlar 2024-2025.

Inspectoratul Școlar .....

Școala/Colegiul/Liceul .....

ACEST MANUAL A FOST FOLOSIT DE:

Anul	Numele elevului	Clasa	Anul școlar	Aspectul manualului*			
				format tipărit		format digital	
				la primire	la predare	la primire	la predare
1							
2							
3							
4							

\* Pentru precizarea aspectului manualului se va folosi unul dintre următorii termeni: **nou, bun, îngrijit, neîngrijit, deteriorat**.

- Cadrele didactice vor verifica dacă informațiile înscrise în tabelul de mai sus sunt corecte.
- Elevii nu vor face niciun fel de însemnări pe manual.

**Referenți științifici:**

Lector univ. dr. Alexandra Băicoianu, Facultatea de Matematică și Informatică, Universitatea „Transilvania” din Brașov  
Prof. grd. I Simona Mihaela Popa, Colegiul Național „Gheorghe Lazăr”, București

Editor-coordonator: Ionuț Popa

Coordonator redacție: Cătălin Georgescu

Redactare: Diana Șerban

Tehnoredactare: Florin Paraschiv

Design copertă și layout: Faber Studio

DTP copertă: Florin Paraschiv

Credite foto: Dreamstime

Înregistrări și procesare sunet, activități digitale interactive și platformă e-learning: Infimedia Pro;

website: <https://infomediapro.ro>

ISBN 978-606-076-824-1

Pentru comenzi puteți contacta Departamentul Difuzare

C.P. 12, O.P. 63, cod 014880, sector 1, București

Tel.: 021 796 73 83, 021 796 73 80

Fax: 021 369 31 99

[www.art-educational.ro](http://www.art-educational.ro)

Toate drepturile asupra acestei lucrări sunt rezervate Editurii Art Klett.

Nicio parte a acestei lucrări nu poate fi reproducă, stocată ori transmisă, sub nicio formă (electronic, mecanic, fotocopiere, înregistrare sau altfel), fără acordul prealabil scris al editurii.

# Cuvânt-înainte

Motto: „Toate visurile tale se pot împlini dacă ai curajul să le urmezi.”  
Walt Disney

Acum sunteți adolescenți, mâine adulți... Învățați la fiecare pas cum să vă construiți un drum pentru a trece cât mai ușor și mai sigur prin viață, independenti și în ritm propriu, mereu încurajați de tehnologie, utilizând și creând în același timp instrumentele informatiche necesare activităților voastre. Sunteți construcțorii propriei voastre vieți, sunteți căpitanii propriei corăbii.

Începem prin a vă spune că această carte este realizată cu bucurie, cu bucuria de a împărtăși, cu bucuria de a participa și noi activ la dezvoltarea voastră, în fond, la dezvoltarea noastră a tuturor. Această carte este doar o cărămidă din fundația pe care urmează de aici încolo să o construți voi îնșivă. Citiți-o și învățați cu drag. Îi mulțumim din suflet băiatului nostru, care ne-a fost sursă de inspirație și model, ne-a oferit toate argumentele necesare pentru susținerea demersului nostru educațional.

Cartea este structurată în două părți, prima parte conținând trei unități legate de utilizarea tehnologiei informației și comunicațiilor în activitățile voastre zilnice – tehnoredactarea diferitelor lucrări, prelucrarea audio/video și lucru colo-  
borativ cu alte persoane, a doua parte, unitatea patru, fiind dedicată realizării de algoritmi și programe informatiche, având două teme: utilizarea unui nou limbaj de programare, foarte puternic, C++, și programarea roboților virtuali.

Pe parcursul lucrării, la finalul fiecărei unități, sunt prevăzute activități de auto-evaluare pentru ca fiecare dintre voi să poată să își aprecieze gradul de înțelegere și învățare a noțiunilor. La finalul fiecărei părți există activități de evaluare a cunoștințelor acumulate până la acel moment.

Vă dorim să aveți o experiență de învățare plăcută și atractivă, mereu cu zâmbetul pe buze!

*Autorii*

# Prezentarea manualului

Varianta digitală a manualului este similară cu cea tipărită, având în plus peste 90 de AMII, activități multimedia interactive de învățare, cu rolul de a spori valoarea cognitivă. Activitățile multimedia interactive de învățare sunt de trei feluri și sunt simbolizate pe parcursul manualului astfel:

-  **AMII static**, de ascultare activă și de observare dirijată a unei imagini semnificative
-  **Activitate animată**, filmulet sau scurtă animație
-  **Activitate interactivă**, de tip exercițiu sau joc, în urma căreia elevul are feedback imediat

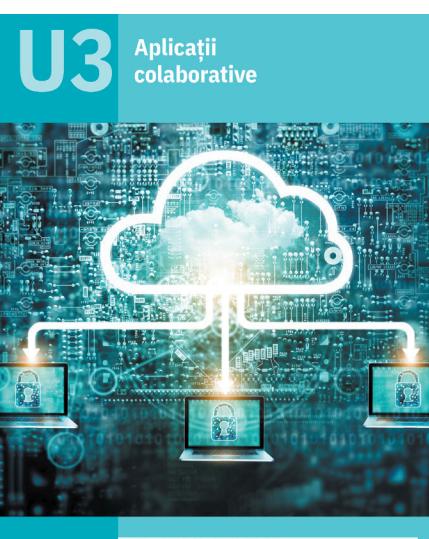
## Alte butoane folosite în varianta digitală:

-  Cuprins manual
-  Fereastră de afișare a rezultatelor
-  Ajutor general manual
-  Micșorarea/mărirea imaginii
-  Salt la începutul/finalul manualului
-  Pagină înainte/pagină înapoi
-  Mărire pe tot ecranul
-  Schimbarea modului de afișare
-  Includerea unei notițe
-  Introducerea de marcaje colorate
-  Informații despre utilizator

## Ce propune acest manual

Manualul se prezintă asemenea unui ghid al elevului, fiind conceput ca un tutorial interactiv și prietenos în lumea practică a tehnologiei informației și comunicațiilor. Acesta este structurat în 4 mari unități de învățare și răspunde viziunii și cerințelor actualului curriculum. Prima unitate acoperă integral, în cele 5 lecții, domeniul de conținut *Editorul de texte* din programa școlară. Urmează a doua unitate, structurată în 6 lecții, ce prezintă conținutul despre *Aplicații de prelucrare audio-video*. A treia unitate, în 3 lecții, va introduce elevul în lumea fascinantă a *Aplicațiilor colaborative*, iar unitatea 4, cu 10 lecții, este dedicată *Limbajului de programare*. Nu lipsesc recapitulările secvențiale marcate distinct pentru fiecare unitate, precum și o recapitulare finală, urmată de răspunsuri la multe dintre problemele și exercițiile date. Sunt propuse, de asemenea, și metode complementare de evaluare: *observarea sistematică a activității și comportamentului elevilor* (vezi *Fișa de autoevaluare a activității și comportamentului*), *investigația*, *proiectul*, *portofoliul*, *autoevaluarea*. Specificul disciplinei implică utilizarea unor metode didactice activ-participative, care sunt bogat sugerate de activitățile prezentate în acest manual, astfel încât să fie atinse toate cele trei competențe generale vizate.

## Manualul este structurat în 4 unități de învățare

U1 Editorul de texte		U2 Aplicații de prelucrare audio-video	
			
<b>Lecție 1</b> 10-11 Editarea/rezordonarea de documente utilizând aplicații specializate	<b>Lecție 2</b> 12-15 Microsoft Word. Elemente de interfață	<b>Lecție 1</b> 28-29 Aplicații de prelucrare audio-video. Scop și avantaje	<b>Lecție 2</b> 30-32 Aplicație WavePad. Elemente de interfață
<b>Lecție 3</b> 16-17 Operații de gestionare a fișierelor documente	<b>Lecție 3</b> 33-35 Prelucrarea sunetelor	<b>Lecție 3</b> 36-39 Programare de editare video. Elemente de interfață. VideoPad	<b>Lecție 4</b> 40-42 Prelucrarea video
<b>Lecție 4</b> 18-20 Editeră un document	<b>Lecție 4</b> 43-45 Prelucrări suprapuse în proiectul video	<b>Lecție 5</b> 46	<b>Lecție 6</b> 46
<b>Lecție 5</b> 21-24 Formatarea unui document. Aspectul paginii			
<b>Concurs</b> 25 Cel mai inspir... - REBUS			
<b>Esercizi recompensabile</b> 26			
<b>Autoevaluare</b> 26			
U3 Aplicații colaborative		U4 Limbajul de programare	
			
<b>Lecție 1</b> 48-50 Aplicații colaborative. Noțiuni introductive	<b>Lecție 1</b> 64-67 Modulul de dezvoltare Code:Blocks	<b>Lecție 2</b> 51-56 Google Apps (Docs, Slides). Interfață și utilizare	<b>Lecție 2</b> 68-73 Limbajul C++ – elemente de bază
<b>Lecție 3</b> 57-59 Noțiuni despre GIS și aplicabilitate	<b>Lecție 3</b> 74-76 Structura înțelită	<b>Lecție 3</b> 77-80 Structura intermediară	<b>Lecție 3</b> 81-85 Structura repetitivă condiționată anterior
<b>Lecție 4</b> 60	<b>Lecție 4</b> 86-89 Structura repetitivă condiționată posterior	<b>Lecție 4</b> 90-95 Structura repetitivă cu control (în modă cunoscut de pas)	<b>Lecție 4</b> 96-100 Aplicații C++ de firski
<b>Lecție 5</b> 60	<b>Lecție 5</b> 101-102 Aplicații C++ pentru Geogafie	<b>Lecție 5</b> 102 Hărți dezvoltării	<b>Lecție 5</b> 103-105 Programarea robotelor virtuale. Utilizarea senzorilor
<b>Lecție 6</b> 60	<b>Lecție 6</b> 106	<b>Lecție 6</b> 106	
<b>Lecție 7</b> 60			
<b>Lecție 8</b> 60			
<b>Lecție 9</b> 60			
<b>Concurs</b> 60			
<b>Esercizi recompensabile</b> 60			
<b>Autoevaluare</b> 60			

# Structura unităților de învățare

## Lecții de predare-învățare

Modul de abordare modern facilitează învățarea și asigură, totodată, stimularea creativității elevilor care își vor folosi cunoștințele dobândite în realizarea unor miniproiecte, pe teme de interes caracteristice vârstei lor (respectând creditarea informației și drepturile de autor). Principalele rubrici ale lecțiilor sunt marcate diferit și consecvent pe parcursul manualului.

**Descoperiți** prezintă, într-o manieră plăcută și graduală, conținutul nou de învățat. Sunt oferite numeroase exemple, iar conținuturile noi sunt prezentate având suportul unor imagini reprezentative ce sunt explicate în detaliu, acolo unde este cazul.

**Aplicați și investigați** asigură aplicarea noțiunilor noi, elevii fiind ghidați pas cu pas în realizarea unor sarcini specifice. Sunt implicați, astfel, în mod activ în procesul de învățare, cunoștințele căpătând un caracter operațional accentuat.

**Ewersați** propune itemi ce vor consolida învățarea, din care nu lipsesc temele de portofoliu, lucrul în echipe sau individual, activitățile de documentare/investigare și realizarea unor sarcini interesante.

**Amintiți-vă și Retineți** sunt alte rubrici care apar pe parcursul lecțiilor, atunci când conținutul prezentat necesită reamintirea unor noțiuni sau când este evidentiat conținutul important, de bază pentru achizițiile viitoare.

**Știați că ...?** prezintă câteva informații sau curiozități despre aplicațiile descrise în manual, oferind astfel un plus de informație.

## Portofoliu

Pe parcursul lecțiilor sunt recomandate mai multe teme de dezbatere și pentru portofoliul elevilor. Metode moderne de învățare-evaluare, acestea sunt prezentate în cadrul rubricii **Ewersați**. Proiectele au atât caracter informativ, cât și formativ. Se propun teme pentru documentare și investigații, manualul oferind o serie generoasă de exemple despre cum putem folosi diverse aplicații online sau instalate pe propriile calculatoare.

## Recapitulare

La fiecare final de capitol, înainte de testul de autoevaluare/interevaluare aferent, se află câte un test de recapitulare. Probele concepute conțin selecții de itemi obiectivi, semiobiectivi și subiectivi, cu nivel de la simplu la complex. Recapitularea face referire atât la noțiunile învățate, cât și la capacitatea copilului de asociere, interpretare și aplicare a cunoștințelor noi. Manualul propune, de asemenea, o recapitulare sumativă pentru partea manualului legată de tehnologia informației și comunicațiilor (TIC), precum și o recapitulare finală.

## Autoevaluare

Autoevaluările sunt însoțite de punctajele de notare și de răspunsuri care se află la finalul manualului, elevii putând astfel să se verifice singuri. Exercițiile și problemele date sunt astfel concepute încât să asigure o bună verificare a cunoștințelor acumulate, precum și operaționalizarea acestora în situații noi, variate. Si aici, elevii au la dispoziție, la sfârșitul manualului, răspunsurile la exercițiile propuse, pentru a se putea autoevalua cu ușurință.

**Amintiți-vă**

Într-o secțiune anterioră am învățat că era un mic program C++.

**Descoperiți**

În continuare începeți să explorezi și să descoperiți în continuare în ceea ce urmărește.

**Retineți**

Numim cod sursă al unui program C++ și îl scriem într-o fișă de lucru.

Un mic program C++ este un program scris în limba C++, care poate fi executat pe un calculator.

**PrimerProgram**

Este un prim proiect în limbajul C++. Este un proiect în limba C++ care include fișierele principale și auxiliare.

**includeiostream**

Este o instrucție de inclusiune a unei fișiere auxiliare.

**intmain**

Este o funcție principală a programului.

**#include<iostream>**

Este o instrucție de inclusiune a unei fișiere auxiliare.

**usingnamespacestd;**

Este o instrucție de inclusiune a unei fișiere auxiliare.

**intmain()**

Este o funcție principală a programului.

**cout << "Hello world!" << endl;**

Este o instrucție de afișare a unei mesaje.

**return0;**

Este o instrucție care returnează valoarea 0.

```

1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     cout << "Hello world!" << endl;
6     return 0;
}

```

● **maișă** este funcția principală a programului. Accesează unul dintre programe, cu această funcție începe întreprinderea programului. Observăm că se disting două părți componente, și anume anumite funcții: **int main()** și **funcție numită main**.

Totodată, în cadrul codului sursă se include biblioteca standardă pentru compilarea și execuția programului, prezentându-se o referință la codul sursă al unui program înainte de compilație.

● **cout << "Hello world!" << endl;** este o instrucție de afișare a unei mesaje.

● **return 0;** este o instrucție care returnează valoarea 0.

● **using namespace std;** este o instrucție de inclusiune a unei fișiere auxiliare.

● **#include<iostream>** este o instrucție de inclusiune a unei fișiere auxiliare.

● **int main()** este o funcție principală a programului. Accesează unul dintre programe, cu această funcție începe întreprinderea programului. Observăm că se disting două părți componente, și anume anumite funcții: **int main()** și **funcție numită main**.

Aceasta conține codul sursă care realizează prelucrările done. Instrucțiunile include și această funcție sunt:

- **cout << "Hello world!" << endl;** este o instrucție de afișare a unei mesaje.
- **return 0;** este o instrucție care returnează valoarea 0.

**Amintiți-vă**

În anul 1979, în cadrul laboratorului Bell Telephone, se realiză primul program în limba de programare Scratch.

**Descoperiți**

În cadrul laboratorului Bell Telephone se realizează primul program în limba de programare Scratch.

**Stiați că ...?**

Thomas Edison a fost primul care a inventat filmul.

**1. Mediaul de dezvoltare Code::Blocks**

**Amintiți-vă**

În anul anterior la învățare și realizări programe în limbajul visual de programare Scratch, următorul programător încearcă să realizeze un program în limba de programare C++.

**Descoperiți**

În cadrul laboratorului Bell Telephone se realizează primul program în limba de programare C++.

**Stiați că ...?**

Thomas Edison a fost primul care a inventat filmul.

**1. Mediaul de dezvoltare Code::Blocks**

**Amintiți-vă**

În anul anterior la învățare și realizări programe în limbajul visual de programare Scratch, următorul programător încearcă să realizeze un program în limba de programare C++.

**Descoperiți**

În cadrul laboratorului Bell Telephone se realizează primul program în limba de programare C++.

**Stiați că ...?**

Thomas Edison a fost primul care a inventat filmul.

**2. Mediaul de dezvoltare Code::Blocks**

**Amintiți-vă**

În cadrul laboratorului Bell Telephone se realizează primul program în limba de programare C++.

**Descoperiți**

În cadrul laboratorului Bell Telephone se realizează primul program în limba de programare C++.

**Stiați că ...?**

Thomas Edison a fost primul care a inventat filmul.

**3. Când să utilizezi un debugger**

**Amintiți-vă**

În cadrul laboratorului Bell Telephone se realizează primul program în limba de programare C++.

**Descoperiți**

În cadrul laboratorului Bell Telephone se realizează primul program în limba de programare C++.

**Stiați că ...?**

Thomas Edison a fost primul care a inventat filmul.

**3. Când să utilizezi un debugger**

**Amintiți-vă**

În cadrul laboratorului Bell Telephone se realizează primul program în limba de programare C++.

**Descoperiți**

În cadrul laboratorului Bell Telephone se realizează primul program în limba de programare C++.

**Stiați că ...?**

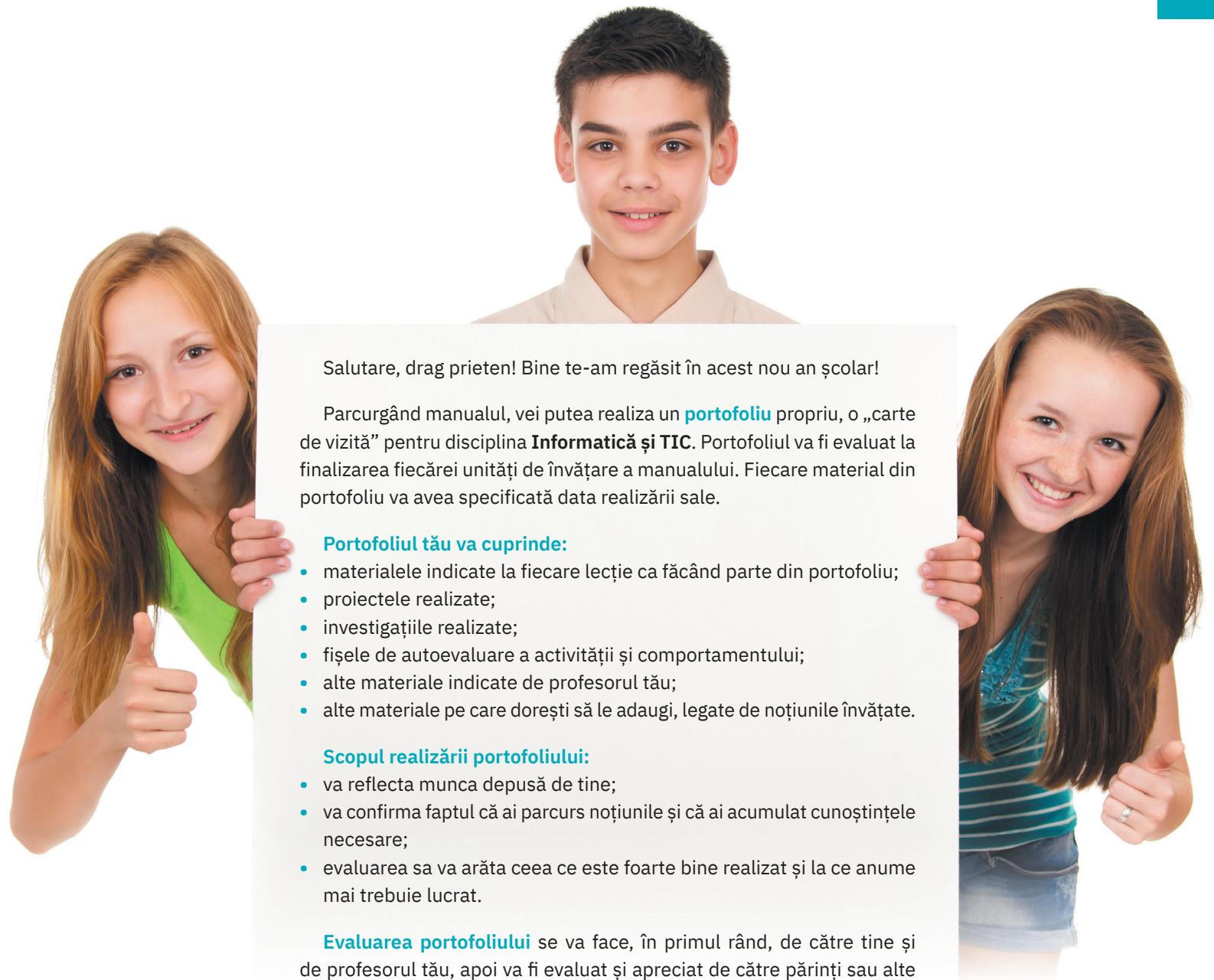
Thomas Edison a fost primul care a inventat filmul.

# Cuprins și competențe generale și specifice

	Nr. pag.	Lecții	Competențe specifice
	8	Să ne amintim! Scurtă recapitulare a cunoștințelor din anii anteriori	
<b>UNITATEA 1</b> Editorul de texte	10 12 16 18 21 25 26 26	L1: Editarea/tehnoredactarea de documente utilizând aplicații specializate L2: Microsoft Word. Elemente de interfață L3: Operații de gestionare a fișierelor documente L4: Editarea unui document L5: Formatarea unui document. Aspectul paginii Concurs: Cel mai inspirat ...REBUS Exerciții recapitulative Autoevaluare	1.1, 3.1
<b>UNITATEA 2</b> Aplicații de prelucrare audio-video	28 30 33 36 40 43 46 46	L1: Aplicații de prelucrare audio-video. Scop și avantaje L2: Aplicația WavePad. Elemente de interfață L3: Prelucrarea sunetelor L4: Programe de editare video. Elemente de interfață. VideoPad L5: Prelucrarea video L6: Prelucrări superioare în proiectul video Exerciții recapitulative Autoevaluare	1.2, 3.2
<b>UNITATEA 3</b> Aplicații colaborative	48 51 57 60 60 61 62	L1: Aplicații colaborative. Noțiuni introductive L2: Google Apps (Docs, Slides). Interfață și utilizare L3: Noțiuni despre GIS și aplicabilitate Exerciții recapitulative Autoevaluare Recapitulare sumativă TIC Evaluare sumativă TIC	1.1, 1.3, 3.1
<b>UNITATEA 4</b> Limbajul de programare	64 68 74 77 81 86 90 96 100 102 103 106 106 107 108 109 110 112	L1: Mediul de dezvoltare Code::Blocks L2: Limbajul C++. Elemente de bază L3: Structura liniară L4: Structura alternativă (de decizie) L5: Structura repetitivă condiționată anterior L6: Structura repetitivă condiționată posterior L7: Structura repetitivă cu contor (cu număr cunoscut de pași) L8: Aplicații de fizică în C++ L9: Aplicații de geografie în C++ Concurs: Micii programatori L10: Programarea roboților virtuali. Utilizarea senzorilor Exerciții recapitulative Autoevaluare Recapitulare finală Evaluare finală Fișă de autoevaluare a activității și a comportamentului Răspunsuri (selectiv) Dicționar de termeni	1.1, 1.4, 2.1, 2.2, 3.1, 3.3  1.1, 1.4, 2.1, 2.2, 3.1, 3.3

Competențe generale	Competențe specifice
1. Utilizarea responsabilă și eficientă a tehnologiei informației și comunicațiilor	<b>1.1.</b> Editarea/tehnoredactarea de documente utilizând aplicații specializate <b>1.2.</b> Documentarea pe diferite teme prin utilizarea aplicațiilor audio respectiv audio-video <b>1.3.</b> Utilizarea aplicațiilor colaborative în scopul dezvoltării în echipă a unor materiale digitale <b>1.4.</b> Utilizarea unui mediu de programare pentru implementarea algoritmilor
2. Rezolvarea unor probleme elementare prin metode intuitive de prelucrare a informației	<b>2.1.</b> Analizarea enunțului unei probleme simple în vederea rezolvării ei printr-un algoritm <b>2.2.</b> Construirea unor algoritmi elementari care combină structurile secvențiale, alternative, repetitive în scopul rezolvării unor probleme
3. Elaborarea creativă de miniproiecte care vizează aspecte sociale, culturale și personale, respectând creditarea informației și drepturile de autor	<b>3.1.</b> Elaborarea unor documente utile în situații cotidiene folosind aplicațiile studiate <b>3.2.</b> Elaborarea unor materiale audio-video pentru a ilustra o temă dată, folosind aplicații dedicate <b>3.3.</b> Implementarea algoritmilor într-un mediu de programare în scopul rezolvării creative a unor probleme având caracter aplicativ

# Introducere



Salutare, drag prieten! Bine te-am regăsit în acest nou an școlar!

Parcurgând manualul, vei putea realiza un **portofoliu** propriu, o „carte de vizită” pentru disciplina **Informatică și TIC**. Portofoliul va fi evaluat la finalizarea fiecărei unități de învățare a manualului. Fiecare material din portofoliu va avea specificată data realizării sale.

#### Portofoliul tău va cuprinde:

- materialele indicate la fiecare lecție ca făcând parte din portofoliu;
- proiectele realizate;
- investigațiile realizate;
- fișele de autoevaluare a activității și comportamentului;
- alte materiale indicate de profesorul tău;
- alte materiale pe care dorești să le adaugi, legate de noțiunile învățate.

#### Scopul realizării portofoliului:

- va reflecta munca depusă de tine;
- va confirma faptul că ai parcurs noțiunile și că ai acumulat cunoștințele necesare;
- evaluarea sa va arăta ceea ce este foarte bine realizat și la ce anume mai trebuie lucrat.

**Evaluarea portofoliului** se va face, în primul rând, de către tine și de profesorul tău, apoi va fi evaluat și apreciat de către părinți sau alte persoane.

Fișă de autoevaluare a portofoliului	Da	Nu
Am adăugat la portofoliu toate materialele indicate.		
Am rezolvat toate sarcinile în întregime.		
Am întâmpinat dificultăți în realizarea anumitor exerciții.		
Am cerut ajutorul colegilor sau prietenilor pentru a rezolva sarcinile de lucru.		
Am cerut ajutorul cadrului didactic în rezolvarea sarcinilor.		

**Fișele de autoevaluare a activității și comportamentului** se vor completa la finalul fiecărei unități de învățare și se vor ataşa portofoliului tău.

# Să ne amintim! Scurtă recapitulare a cunoștințelor din anii anterioiri

## Amintiți-vă

Principalele operații în lucrul cu fișierele și folderele sunt: creare, deschidere, editare, salvare, închidere, selectare, copiere, mutare, ștergere.

Pentru lucrul cu elementele conținute în fișiere, principalele operații sunt: selectare, copiere, mutare, ștergere.

Comenzile corespunzătoare operațiilor de mai sus se găsesc, în general, la majoritatea programelor, sub formă de opțiuni de meniu, în meniurile **File (Fișier)** și **Edit (Editare)** sau **Home (Acasă – cu sensul de „fila de bază”)**.

Pentru lucru mai eficient, puteți folosi următoarele comenzi rapide (combinații de taste):

- **Creare (New) – Ctrl+N**
- **Deschidere (Open) – Ctrl+O**
- **Salvare (Save) – Ctrl+S**
- **Închidere (Close sau Exit) – Alt+F4**
- **Selectare integrală (Select All) – Ctrl+A**
- **Copiere (Copy) – Ctrl+C**
- **Mutare** – aceasta este o combinație de două acțiuni: **Decupare** din locația inițială (**Cut**) – **Ctrl+X**, și **Lipire** în noua locație (**Paste**) – **Ctrl+V**
- **Ștergere (Delete)** – tasta **DEL** sau **Delete** (în funcție de producător, pe tasta respectivă marcajul poate să difere)

Dacă ati greșit ceva, puteți reveni imediat la starea dinaintea ultimei acțiuni efectuate, folosind comanda **Undo (Anulare) – Ctrl+Z**.

## Exersați

Sisteme de operare	Dați trei exemple de sisteme de operare foarte utilizate în prezent.
Realizarea unei prezentări electronice	Realizați o scurtă prezentare electronică, pe tema „Cinci lucruri importante pe care le-am învățat la Informatică și TIC în clasa a VI-a”. Folosiți în prezentare texte, imagini și sunete.
Siguranța pe Internet	Care sunt caracteristicile principale ale unei aplicații de tip malware?
Poșta electronică	Care sunt cele două categorii de programe prin intermediul cărora puteți utiliza poșta electronică?
Reguli de comunicare online	Neticheta impune norme de comportament general în mediul online. Enumerați cinci reguli specifice comunicării în acest mediu.
Operații cu e-mailuri	La ce folosiți Address Book (agenda cu adrese)?
Grafică 3D	Numiți două aplicații dedicate realizării graficii 3D.
Animații 3D	Numiți o aplicație de realizare a animațiilor 3D.
Realitatea virtuală	Prezentați un avantaj al utilizării realității virtuale.
Algoritmi	Ce este un algoritm? Care sunt caracteristicile unui algoritm?
Structuri repetitive	Care sunt cele trei tipuri de structuri repetitive învățate?
Structura repetitivă condiționată posterior	Dați exemplu de un algoritm în care se folosește structura repetitivă condiționată posterior.
Aplicabilitatea practică a algoritmilor	Realizați un personaj simplu în Paint 3D. Folosiți acest personaj într-un program pe care îl faceți în Scratch, program care să permită mișcarea personajului cu ajutorul tastelor <b>Săgeată Sus/Jos, Săgeată Stânga/Dreapta</b> , astfel încât personajul să nu iasă din ecran. La atingerea unei margini a ecranului, personajul să scoată un sunet scurt.

# U1

## Editorul de texte



Lecția 1	10-11	Editarea/tehnoredactarea de documente utilizând aplicații specializate
Lecția 2	12-15	Microsoft Word. Elemente de interfață
Lecția 3	16-17	Operații de gestionare a fișierelor documente
Lecția 4	18-20	Editarea unui document
Lecția 5	21-24	Formatarea unui document. Aspectul paginii
Concurs	25	Cel mai inspirat... REBUS
Exerciții recapitulative	26	
Autoevaluare	26	

# 1 Editarea/tehnoredactarea de documente utilizând aplicații specializate

## Descoperiți

În zilele noastre, putem discuta despre o adevarată generalizare a fenomenului de editare și tehnoredactare a documentelor. La nivel global, oamenii au posibilitatea de a-și redacta singuri diferite documente, utilizând tehnica de calcul actuală (calculatoare, tablete, telefoane inteligente) și aplicații software specializate.

Redactarea unui document presupune, de fapt, toate operațiile necesare pentru a oferi o experiență de citire optimă:

**1. editarea** – crearea, revizuirea și îmbunătățirea conținutului și a structurii generale a documentului:

- inserarea/modificarea textelor;
- adaptarea stilului de scriere la audiență;
- eliminarea erorilor de sintaxă sau gramaticale;

**2. tehnoredactarea (Desktop Publishing – DTP)** – procesul de formatare și aranjare a conținutului documentului:

- stabilirea aspectului general al documentului (aliniere, font, stilurile paragrafelor, marginile etc.);
- inserarea de diferite elemente, cum ar fi: imagini, tabele, grafice, ecuații;
- creația de liste, etc.

## Rețineți

Noțiunea de **tehnoredactare** este prezentată în *Dicționarul explicativ al limbii române* ca fiind „pregătirea tehnică și grafică a unui manuscris înainte de a începe operația de tipărire”.

Rezultatul tehnoredactării este lucrarea pregătită de tipar, care întrunește o serie de calități importante pentru viitorul cititor: să fie lizibilă (adică să se citească ușor), aspectul să fie plăcut și ordonat, calitatea grafică să fie foarte bună (imagini, litere, culori etc.), forma de prezentare să fie adaptată la conținut.

## Procesoare de text

O aplicație software specializată pentru redactarea de documente trebuie să asigure toate funcționalitățile importante pentru editarea și tehnoredactarea acestora.

## Rețineți

Aplicațiile software care permit utilizatorilor crearea, editarea, formatarea, aranjarea în pagină și tipărirea materialelor poartă numele de **procesoare de text** (*word processors*).

Instrumentele de bază ale unei aplicații de realizare a documentelor sunt:

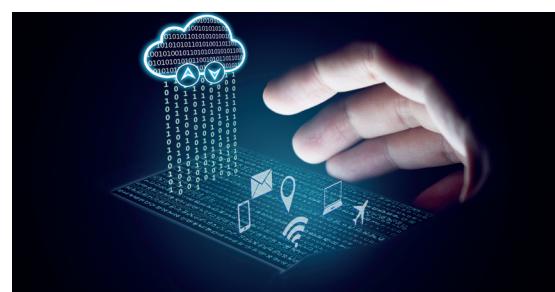
- instrumente pentru editarea textului;
- instrumente pentru formatarea textului;
- instrumente de verificare lingvistică (ortografic și grammatical);
- instrumente adiționale pentru aranjarea în pagină;
- instrumente adiționale pentru editarea unui document, cum ar fi: editorul de ecuații, editorul de grafică, editorul de tabele, editorul de diagrame etc.

Aplicațiile procesoare de text sunt unele dintre cele mai utilizate, mai ales în munca de birou. Astăzi, ele pot fi instalate pe calculator direct sau sunt puse la dispoziție de către producători pentru a fi accesate pe Internet. Procesoarele de text lucrează cu fișiere numite **documente**.

## Rețineți

► **Documentul** reprezintă, de fapt, ansamblul format din texte și alte elemente inserate (tabele, imagini, grafice etc.), care sunt prelucrate în mod unitar de către aplicația procesor de text. Un document poate fi: o carte, un referat, o scriere, o lucrare oarecare, un articol etc.

Procesoarele de text pot prelucra fișiere (documente) stocate local sau pe medii de stocare conectate la calculator (de exemplu, memory stick). Există și procesoare de text care pot accesa, prelucra și salva fișiere într-un spațiu virtual numit „cloud” (nor). Stocarea unui fișier în cloud înseamnă stocarea acestuia pe anumite calculatoare (servere) aflate oriunde în lume, fără ca utilizatorul să cunoască amplasarea sau configurația lor. Accesul la fișierele stocate în cloud se face online, pe Internet. Practic, puteți avea acces la datele voastre de oriunde din lume și le puteți utiliza și partaja ușor, cu oricine dorîți.



## Editarea/tehnoredactarea de documente utilizând aplicații specializate

Câteva exemple de servicii foarte cunoscute pentru păstrarea și accesarea datelor în cloud sunt:

- **OneDrive** (operat de către Microsoft) – este disponibil gratuit tuturor celor care dețin un cont Microsoft;
- **Google Drive** (operat de către Google) – este disponibil gratuit tuturor celor care dețin un cont Google;

Exemple de **aplicații procesoare de text** utilizate pentru tehnoredactarea de documente:

- **MS Word** (<https://products.office.com/ro-ro/word>) – aplicație dezvoltată de compania Microsoft; face parte din suita de aplicații MS Office. Se utilizează pe bază de licență; elevii și profesorii din instituțiile de învățământ pot utiliza gratuit suita Office 365 Education (include Word, Excel, PowerPoint etc.);
- **LibreOffice Writer** (<https://ro.libreoffice.org/>) – face parte din suita de aplicații pentru birou și poate fi utilizată gratuit. LibreOffice este un proiect al organizației nonprofit The Document Foundation;

### Avantajele editării/tehnoredactării computerizate

1. Ușurință în editarea documentului, în crearea și realizarea modificărilor. Odată scris, documentul se poate salva pe un mediu de stocare și poate fi oricând afișat pe ecran și completat/modificat, poate fi tipărit în nenumărate exemplare.
2. O bună parte a operațiilor de procesare a documentului sunt automat realizate de către calculator.
3. Calitatea grafică a documentului este mare din punct de vedere tehnic; se pot schimba caracteristicile literelor care compun textul, se pot insera diferite elemente (imagini, grafice, tabele etc.), se pot adăuga diferite efecte.
4. Se poate face căutare automată în fișier pentru a găsi diferite informații dorite.
5. Documentul poate fi transmis ușor către altcineva sau poate fi pus la dispoziția altor persoane în mediul online.



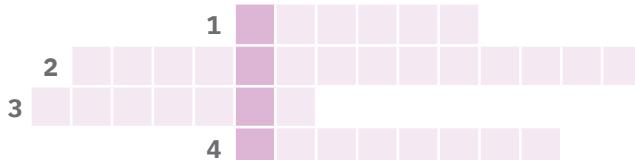
### Aplicații și investigații

- ▶ Căutați pe Internet informații despre aplicațiile pentru editarea/tehnoredactarea de documente. Accesați hiperlinkurile prezentate pentru procesoarele de text și descoperiți modul în care funcționează fiecare dintre ele.



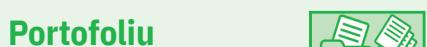
### Exesări

1. Dați exemple de trei procesoare de text.
2. Argumentați de ce este ușor de corectat un document realizat cu ajutorul unui procesor de texte.
3. Scrieți în caiete care dintre afirmațiile următoare sunt adevărate:
  - a. Un procesor de texte produce fișiere video.
  - b. Un document scris cu ajutorul unui procesor de texte poate fi tipărit de nenumărate ori, la intervale diferite de timp.
  - c. Într-un document scris cu ajutorul unui procesor de texte se poate realiza căutarea automată a unui text.
4. Care sunt avantajele tehnoredactării computerizate? Argumentați.
5. Copiați în caiete rebusul de mai jos. Completăți-l și descoperiți cuvântul care se formează pe verticala colorată:

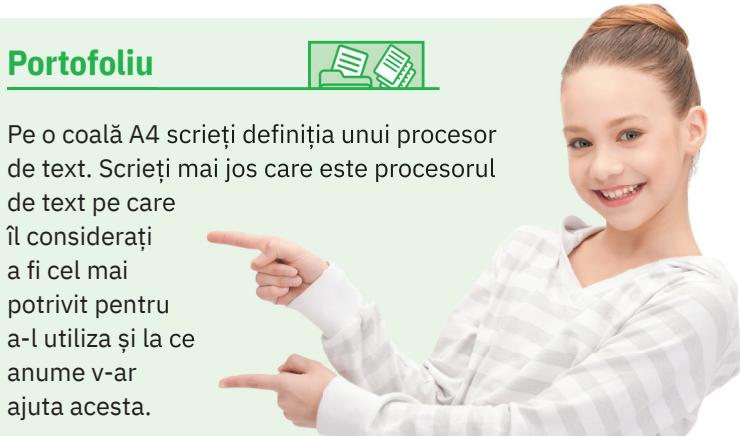


1. Procesor de texte care face parte din suita Apache OpenOffice.
2. Prelucrarea tehnică și grafică a unui document în vederea tipăririi sale.
3. Operația de creare, revizuire și îmbunătățire a conținutului unui document.
4. Cum se numește un fișier realizat cu un procesor de text?

### Portofoliu



Pe o coală A4 scrieți definiția unui procesor de text. Scrieți mai jos care este procesorul de text pe care îl considerați a fi cel mai potrivit pentru a-l utiliza și la ce anume v-ar ajuta acesta.



## 2 Microsoft Word. Elemente de interfață

### Amintiți-vă

**Microsoft Word (MS Word sau, simplu, Word)** este una dintre cele mai cunoscute aplicații procesor de text, componentă a pachetului de programe Microsoft Office.

Un fișier document lucrat în această aplicație poate avea una dintre următoarele extensii: **.doc, .docx, .docm**.

Există mai multe versiuni ale aplicației Microsoft Word. În continuare, pentru exemplificări, va fi utilizată versiunea Microsoft Word din suita de programe Microsoft 365º Personal, instalată în limba română pe un calculator cu un sistem de operare Windows 11.

Pentru conectarea la aplicația Microsoft 365 s-a utilizat un cont de utilizator numit **Elev Cls7**. A avea un cont presupune să aibă un nume și o parolă, unic pentru fiecare utilizator în parte, cu care să conectați la aplicație. Pentru elevi, contul poate fi atribuit gratuit de către școală.



### Ştiati că...?

Prima versiune a aplicației Microsoft Word a fost lansată în octombrie 1983. Aceasta a fost dezvoltată pentru platforma de operare DOS (Disk Operating System) și a făcut parte din prima suita de produse Microsoft Office, care includea Word, Excel și PowerPoint. Inițial, Microsoft Word a fost lansat sub numele de „Multi-Tool Word” pentru a evidenția caracteristicile sale de editare avansate.

### Descoperiți

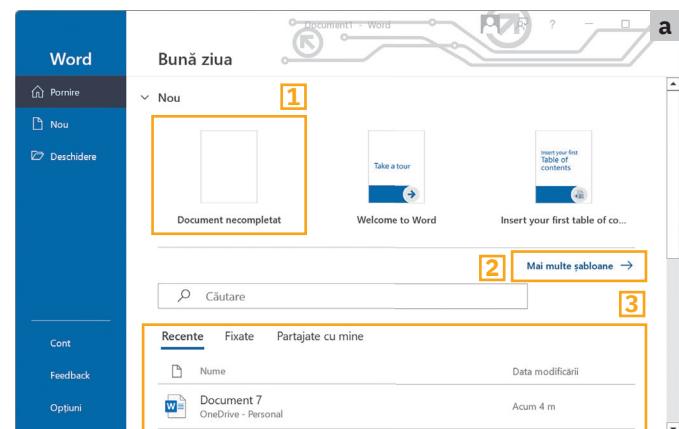
Mai departe, vom descoperi împreună cele mai importante elemente ale interfeței aplicației Microsoft Word.

#### Pornirea aplicației Microsoft Word

Lansarea în execuție a aplicației se face accesând **Start** -> **All Apps** -> **Word** sau se tastează cuvântul „Word” în căsuța de căutare (Search) de pe **Bara de activități (Toolbar)** a interfeței Windows-ului și se apasă Enter.

Aplicația Microsoft Word se deschide cu un ecran de start (*imaginea a*). De aici putem alege:

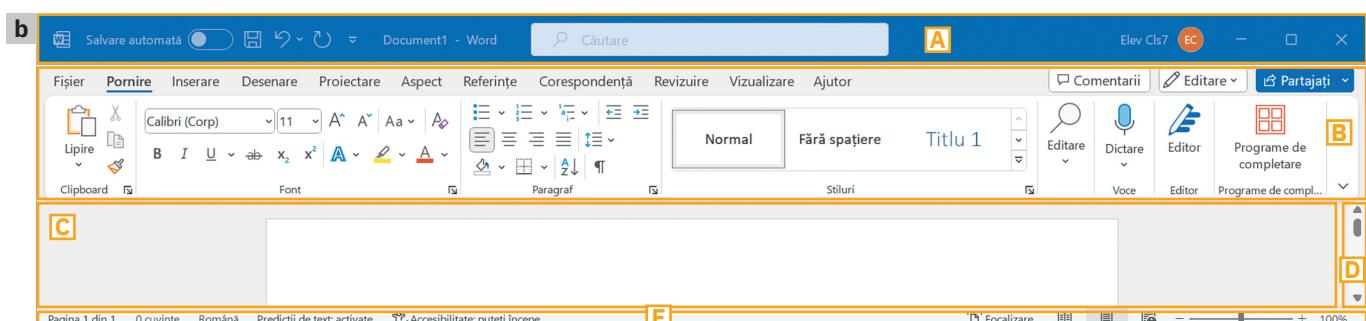
1. crearea unui document necompletat: se alege **Document necompletat (Blank document)**;
2. crearea unui document nou utilizând un şablon predefinit: se alege unul dintre şabloanele existente;
3. deschiderea unui document realizat anterior: se selectază un document din listele propuse.



#### Detalierea elementelor de interfață ale aplicației Microsoft Word

Vom alege să realizăm un document nou, necompletat. Alegerea făcută inițiază deschiderea în interfața Microsoft Word a unui document care este denumit automat **Document1**. El este acum în lucru, nesalvat încă.

Fereastra interfeței utilizator a Microsoft Word este alcătuită din următoarele elemente (*imaginea b*):



**A Bara de titlu (Title Bar);**

**D Bara de defilare (Scroll Bar);**

**B Panglica (Ribbon);**

**E Bara de stare (Status Bar).**

**C Zona documentului;**

#### A. Bara de titlu – detaliere



**Bara de titlu (Title Bar) conține (*imaginea c*):**

1. **Bara de acces rapid (Quick Access Toolbar)** – permite accesarea rapidă a operațiilor de bază cu fișiere tip document: **Salvare (Save, Ctrl+S)**, **Anulare (Undo, Ctrl+Z)**, **Refacere (Redo, Ctrl+R)**. Se mai pot adăuga și alte comenzi în această

## Microsoft Word. Elemente de interfață

zonă prin selectarea/deselectarea opțiunilor respective din meniul care apare la apăsarea săgeții din dreapta. Comanda **Salvare automată** permite salvarea periodică a documentului, în timpul lucrului (doar pentru fișiere stocate în cloud);

- 2. Numele fișierului** (documentului) aflat în lucru;
- 3. Căutare Microsoft** (Alt+G) – ajută la găsirea rapidă a textelor în fișier, a diferite acțiuni, a mai multor informații despre un cuvânt/o expresie sau chiar a fișierelor (în cazul în care sunteți logați în contul de utilizator Office);
- 4. Numele contului de utilizator;**
- 5. Panoul de management al ferestrei** – permite minimizarea, restaurarea și, respectiv, maximizarea ferestrei.

### B. Panglica – detalieri



► **Panglica (Ribbon)** – include toate comenziile aplicației Microsoft Word grupate pe file (tabs), pe tipuri de acțiuni (*imaginea d*). O filă poate avea în componență sa:

- 1. grupuri logice de comenzi** (gruparea se face în funcție de utilizarea lor) – dacă există în partea dreaptă jos o săgeată, apăsarea sa afișează mai multe opțiuni;
- 2. comenzi (butoane) individuale.**

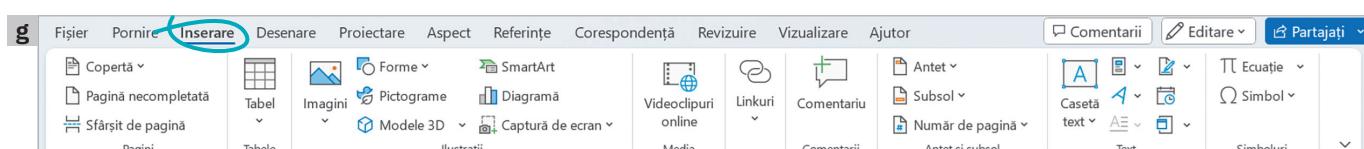
+ Modalitatea de afișare a panglicii (să se afișeze doar fiile sau să se afișeze panglica întreagă) se face apăsând pe butonul **3** din *imaginea d* și alegând opțiunea dorită.

Principalele file prezente pe panglica MS Word sunt:

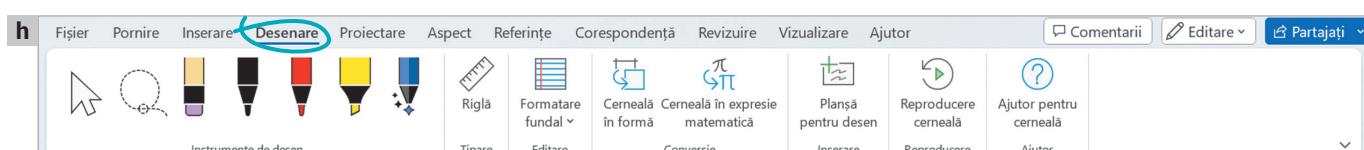
- 1. Fila Fișier (File)** – la selectarea acestei file, se deschide o vizualizare nouă, numită Backstage (*imaginea e*), care conține comenziile pentru acțiunile care pot fi realizate asupra fișierului. Principalele acțiuni sunt: crearea unui document nou, deschiderea unuia existent, salvarea, tipărirea, partajarea acestuia.
- 2. Fila Pornire (Home)** – conține comenzi și elementele necesare pentru editarea și formatarea textului unui document: caracteristici litere, aliniere, stil (*imaginea f*).



- 3. Fila Inserare (Insert)** – conține comenziile necesare pentru adăugarea de alte elemente de conținut într-un document, spre exemplu tabele, ilustrații (*imaginea g*). Tot de aici pot fi inserate în text comentarii, antet sau subsol, diferite simboluri etc.



- 4. Fila Desenare (Draw)** – conține diferite instrumente de desenare, inclusiv pentru dispozitivele cu tehnologie tactilă (*imaginea h*).



- 5. Fila Proiectare (Design)** – conține comenzi pentru formatarea documentului. De aici se pot alege diferite teme care includ o paletă de culori, fonturi și efecte deja armonizate, pentru a conferi documentului un aspect profesionist (*imaginea i*).



- 6. Fila Aspect (Page Layout)** – conține comenzi pentru configurarea aspectului paginilor documentului (*imaginea j*). Aici se configuraază aspectul fiecărei pagini, marginile, dimensiunea, tipul de orientare: **Portret** (*Portrait*) – vertical sau **Vedere** (*Landscape*) – orizontal. Tot aici găsim comenzi pentru spațierea paragrafelor, de aranjare (pozitionarea, încadrarea, alinierea) a diferențelor obiecte incluse în document (de exemplu, imagini) față de text.



- 7. Fila Referințe (References)** – conține comenzi necesare pentru gestionarea a diferite tipuri de referințe în cadrul unui document (*imaginea k*). Aceste referințe sunt, de fapt, hiperlinkuri, fie către diferite părți ale același document, permitând saltul direct acolo, fie către alte documente externe. De exemplu, de aici se poate crea un Cuprins bazat pe titlurile din document, se pot insera note de subsol, se poate crea și întreține Lista de referințe bibliografice dintr-o lucrare.



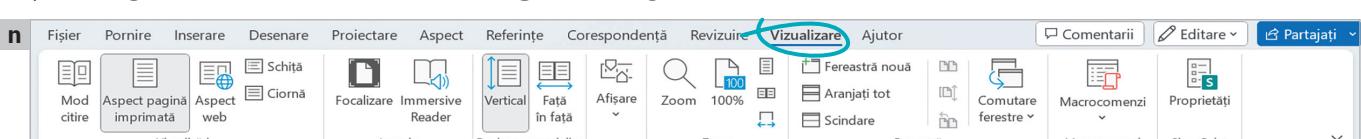
- 8. Fila Corespondență (Mailings)** – conține comenzi necesare pentru gestionarea operațiunilor de trimis a corespondenței, de exemplu crearea de mai multe adrese pentru plicuri odată, fiecare plic cu adresa lui proprie, sau realizarea de diferențe etichete cu diverse informații fiecare (*imaginea l*).



- 9. Fila Revizuire (Review)** – permite verificarea ortografică a textelor documentului, conform cu limba în care acesta este scris, adăugarea de comentarii, compararea documentelor. De asemenea, conține comenzi necesare gestionării modificărilor realizate într-un document (*Track Changes*), vizualizarea, acceptarea sau respingerea modificărilor (*imaginea m*).



- 10. Fila Vizualizare (View)** – permite alegerea modalității de vizualizare a documentului în fereastră (*imaginea n*). Există aici posibilitatea de afișare a unor instrumente care ajută la formatarea paginilor (**Rigla** – *Ruler* și **Linii de grilă** – *Gridlines*) și la navigarea în document (**Panou de navigare** – *Navigation Pane*).



- 11. Fila Ajutor (Help)** – conține comenzi care permit căutarea de diferențe informații de ajutor pentru utilizarea programului.

### C. Zona documentului – detaliere

Este zona de lucru, zona în care este efectiv editat documentul.

### D. Bara de defilare – detaliere

Este o bară verticală sau orizontală, localizată la dreapta sau în partea de jos a ecranului și care permite utilizatorului să mute aria de afișare a documentului sus/jos și, respectiv, stânga/dreapta.

### E. Bara de stare – detaliere



**Bara de stare** este situată în partea de jos a ecranului (*imaginea o*). Conține:

- Zona de afișare a paginii curente și a numărului total de pagini (contorul de pagini);
- Zona de afișare a numărului de cuvinte din document (contorul de cuvinte); la click pe ea se afișează o statistică legată de numărul de pagini, cuvinte, caractere, paragrafe și linii;

3. Butonul de afișare a listei de erori ortografice sau gramaticale descoperite în text; aceasta se afișează la click;
  4. Butonul de configurare a limbii pentru textul din document; la click se poate configura care este limba în care este scris un text selectat și dacă se dorește verificarea ortografică și gramaticală;
  5. Activarea/dezactivarea opțiunii de predicție de texte (anticiparea următoarelor cuvinte și sugerarea de cuvinte sau expresii pe măsură ce tastăți).
  6. Verificarea accesibilității documentului (ușurință în utilizare pentru o varietate de utilizatori, inclusiv cei cu diverse nevoi și dizabilități) și propunerea de îmbunătățiri, dacă este cazul.
  7. Zona de control a modului de vizualizare a documentului expune patru opțiuni: **Mod Focalizare (Focus)**, **Mod citire (Read Mode)**, **Aspect pagină imprimată (Print Layout)**, **Aspect Web (Web Layout)**.
  8. Instrumentul de panoramare (Zoom) – permite mărirea/micșorarea documentului. Pune la dispoziție două butoane („+” pentru mărire și „-“ pentru micșorare), un potențiometru care se poate muta cu ajutorul mouse-ului (la stânga pentru micșorare și la dreapta pentru mărire) și o zonă de afișare a nivelului de panoramare (zoom), în partea dreaptă a zonei (în procente).
- Bara de stare poate fi configurață să conțină zonele și informațiile dorite de către utilizator prin selectarea/deselectarea opțiunilor respective din meniu care apare la click dreapta pe o zonă liberă a acesteia.
- Dacă stați cu mouse-ul deasupra unui buton din interfața Word, se afișează un text de ajutor cu descrierea funcționalității butonului respectiv.

## Aplicații și investigații

Realizați următoarele acțiuni în ordinea în care sunt specificate. Observați, la fiecare pas, ce se întâmplă.

1. Deschideți Microsoft Word și creați un document nou necompletat; observați cum arată interfața și documentul vostru, apoi închideți aplicația Word.
2. Deschideți Microsoft Word și creați un document nou folosind un şablon (de exemplu, *CV clar și îngrijit proiectat de MOO*). Observați acum aspectul acestui document.
3. Parcurgeți, de la stânga la dreapta, toate filele și observați comenzile și grupurile de comenzi puse la dispoziție de fiecare filă în parte.
4. Schimbați modurile de vizualizare a documentului din **Mod citire** în **Aspect pagină imprimată**, **Aspect Web** sau **Focalizare**. Reveniți la **Aspect pagină imprimată**.
5. Micșorați și apoi măriți fereastra Microsoft Word. Observați transformările aspectului grupurilor de comenzi din fiecare filă în parte.
6. Schimbați modul de panoramare din bara de stare a ferestrei Word. Observați cum se modifică dimensiunile documentului din zona de editare. Faceți click pe numărul care indică nivelul de panoramare și descrepeți-i funcționalitatea.



## Exerciții

1. Scrieți în caiete răspunsul corect.
  - a. MS Word este:
    - un procesor de texte;
    - un procesor de sunete;
    - o aplicație de prelucrare video.
  - b. Panglica MS Word conține fila:
    - Corespondență;
    - Tranzitii;
    - Date.
2. Enumerați cel puțin două moduri de vizualizare a unui document.
3. Configurați nivelul de panoramare (zoom) la 55%. În câte moduri puteți modifica acest nivel?
4. Configurați panglica aplicației MS Word astfel încât să fie vizibile doar filele.
5. Adăugați în **Bara de acces rapid** butoanele **Deschidere (Open)** și **Imprimare rapidă (Quick Print)**.
6. Scrieți în caiete **A** (Adevărat) sau **F** (Fals), în funcție de valoarea de adevăr a propozițiilor:
  - a. Un document pregătit cu ajutorul aplicației MS Word poate avea extensia **.doct**.
  - b. Pe bara de stare a aplicației MS Word poate apărea numărul de cuvinte din document.
  - c. MS Word poate afișa în bara de stare câte imagini conține documentul.
  - d. În MS Word, fila **Revizuire (Review)** conține grupul de comenzi **Urmărire (Tracking)**.
7. **Dezbateră.** Organizați o dezbatere cu tema „Gestionarea modificărilor într-un document – o acțiune utilă sau împovărtătoare?“.

## Portofoliu



Scrieți pe o coală de hârtie 3 comenzi aflate pe panglica aplicației MS Word a căror denumire începe cu litera inițială a numelui sau a prenumelui vostru și la ce este utilizată fiecare dintre acestea.

A	F
A	F
A	F

# 3 Operații de gestionare a fișierelor documente



## Amintiți-vă

Cu ajutorul aplicației Microsoft Word se realizează editarea/tehnoredactarea computerizată a documentelor. Acestea pot fi salvate în format electronic pe un mediu de stocare (ca fișiere) și/sau pot fi tipărite la imprimantă.



## Descoperiți

### Operații de bază pentru gestionarea fișierelor

Operațiile de bază pentru gestionarea fișierelor documente sunt: crearea unui fișier nou, deschiderea unui existent, vizualizarea, salvarea în diverse formate, tipărirea și închiderea unui fișier.

#### A. Crearea unui fișier document nou

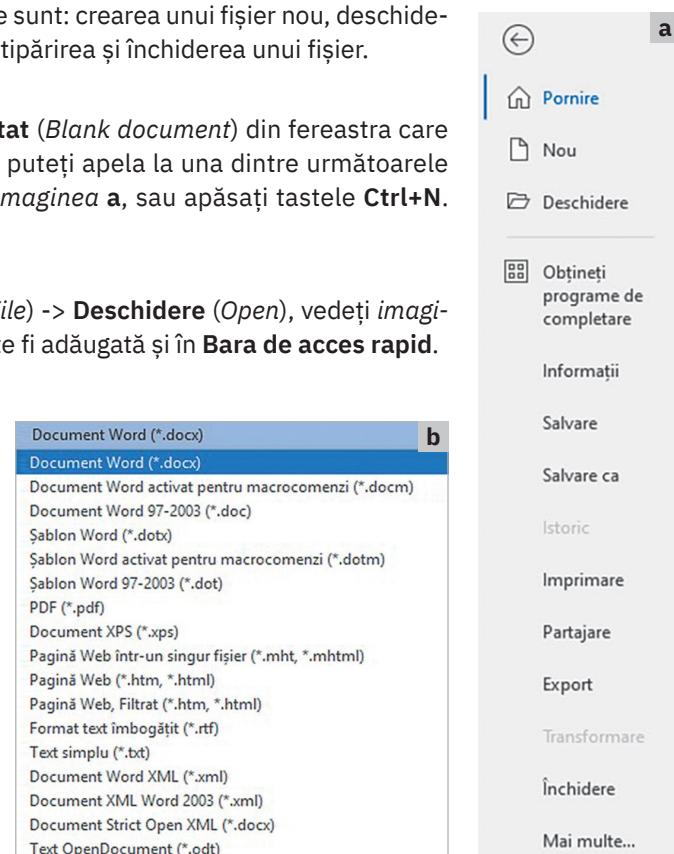
Pentru a crea un fișier nou, alegeți **Document necompletat** (*Blank document*) din fereastra care se deschide la pornirea aplicației Word. De asemenea, mai puteți apela la una dintre următoarele modalități: selectați fila **Fișier (File)** -> **Nou (New)**, vedeti imaginea **a**, sau apăsați tastele **Ctrl+N**. Comanda **Nou** poate fi adăugată și în **Bara de acces rapid**.

#### B. Deschiderea unui fișier existent

Pentru a deschide un fișier existent, selectați fila **Fișier (File)** -> **Deschidere (Open)**, vedeti imaginea **a**, sau apăsați tastele **Ctrl+O**. Comanda **Deschidere** poate fi adăugată și în **Bara de acces rapid**.

#### C. Salvarea unui fișier existent:

- fără a-i schimba numele, tipul (extensia) sau locația – accesați fila **Fișier (File)** -> **Salvare (Save)**, vedeti imaginea **a**, sau apăsați tastele **Ctrl+S**. Opțiunea **Salvare** poate apărea și în **Bara de acces rapid**.
- cu un alt nume, într-un alt format (\*.docx, \*.docm, \*.pdf, \*.htm etc.) sau la o altă locație – accesați fila **Fișier (File)** -> **Salvare ca (Save as)**, vedeti imaginea **a**, sau apăsați tasta **F12**. Opțiunea **Salvare ca** poate apărea și în **Bara de acces rapid**. Imaginea **b** prezintă tipuri de fișiere în care poate fi salvat un document Word. Salvarea se poate face fie local sau pe un mediu de stocare conectat la calculator, fie în cloud. Stocarea în cloud se poate face doar dacă sunteți conectați la aplicație cu un cont de utilizator Microsoft.



#### D. Imprimarea (tipărirea) unui document

Pentru a tipări un document, trebuie să accesați fila **Fișier (File)** -> **Imprimare (Print)**, vedeti imaginea **a**. Documentul se poate tipări la o imprimantă instalată în prealabil sau poate fi tipărit într-un fișier de tip **.pdf**, care se salvează pe unul dintre mediile de stocare aflate la dispoziție. Acest lucru se alege din caseta **Imprimantă (Printer)**.

#### E. Închiderea unui fișier

Pentru a închide un fișier document deschis, selectați fila **Fișier (File)** -> **Închidere (Close)**, vedeti imaginea **a**.

## Ştiați că...?



Începând cu Microsoft Office 2010 pentru Windows și Microsoft Office 2011 pentru Mac, suita Microsoft Office permite salvarea și deschiderea documentelor din cloud. Pe astfel de documente se permite lucrul simultan de către mai mulți utilizatori.



## Aplicați și investigați

Realizați următoarele acțiuni în ordinea în care sunt specificate și observați rezultatele.

1. Creați un folder de lucru cu numele **Lucru\_clsVII**.
2. Deschideți MS Word și creați un document nou.
3. Apăsați butonul **Salvare (Save)** din **Bara de acces rapid** și salvați fișierul cu numele **PrimulMeuDocument** în folderul **Lucru\_clsVII**.
4. Mai apăsați o dată butonul **Salvare (Save)** și observați comportamentul acum.
5. Închideți fișierul folosind comanda **Închidere**.
6. Deschideți MS Word, creați un document nou și alegeți pentru el un şablon.
7. Salvați-l cu numele **DocumentDupaSablon** în folderul **Lucru\_clsVII**.
8. Tipăriți documentul într-un fișier cu extensia **.pdf** în folderul **Lucru\_clsVII**.
9. Închideți fișierul folosind comanda **Închidere**, apoi deschideți-l din nou.

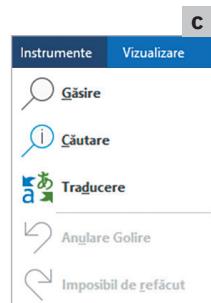


## Descoperiți

### F. Vizualizarea unui fișier

Există patru **modalități principale de vizualizare** a unui document deschis în Word:

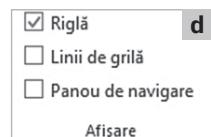
- 1. Mod Focalizare (Focus)** – este tipul de vizualizare potrivit pentru a elimina orice distragere, pentru a vă putea concentra asupra documentului (se elimină tot de pe ecran, cu excepția riglei și a textului pe care îl scrieți). Pentru a ieși din acest mod de vizualizare, apăsați tasta ESC.
- 2. Mod citire (Read Mode)** – este tipul de vizualizare potrivit pentru citit. Include instrumente potrivite citirii, accesibile din meniu **Instrumente (Tools)**, care apare imediat după selectarea acestui mod de vizualizare (*imagină c*). De exemplu: **Găsire** (Find) – găsirea unui text în document, **Căutare** – căutarea de informații pe Internet despre textul selectat sau **Traducere** (Translate) – traducerea în altă limbă a unui text selectat.
- 3. Aspect pagină imprimată (Print Layout)** – se poate vedea cum va arăta documentul tipărit.
- 4. Aspect Web (Web Layout)** – se poate vedea cum va arăta documentul ca pagină Web.



Aceste moduri de vizualizare se accesează fie din zona de control a modului de vizualizare din bara de stare, fie din grupul de comenzi **Vizualizări (Views)** din fila **Vizualizare (View)**. Puteti alege cât de mult din document să fie vizualizat pe ecran la un moment dat, utilizând comenzi de panoramare din grupul **Zoom**, prezent în fila **Vizualizare (View)**.

 Pentru a putea vedea mai ușor încadrarea în pagină a documentului și organizarea acestuia, accesați comenzi aflate în grupul **Afișare (Show)** din fila **Vizualizare (View)**, vedeti *imagină d*:

- 1. Riglă (Ruler)** – afișează cele două rigle (orizontală și verticală) lângă document. Cu ajutorul lor se pot face mult mai ușor alinieri și dimensiuni de obiecte;
- 2. Linii de grilă (Gridlines)** – afișează o grilă din linii pe fundalul documentului, pentru a ajuta la poziționarea perfectă a obiectelor;
- 3. Panou de navigare (Navigation Pane)** – afișează o fereastră care conține structura documentului, cu hiperlinkuri către pagini. Informațiile cuprinse aici sunt similare cu cele din Cuprinsul documentului.




## Aplicați și investigați

1. Deschideți fișierul creat anterior, **DocumentDupaSablon**.
2. Faceți click pe comanda **100%** din grupul **Zoom** al filei **Vizualizare (View)**.
3. Selectați, pe rând, comenziile **O pagină (One Page)**, **Mai multe pagini (Multiple pages)**, **Lățime pagină (Page Width)** din grupul **Zoom** al filei **Vizualizare (View)** și observați care este rezultatul.
4. Faceți click pe butonul **Zoom** din grupul **Zoom** al filei **Vizualizare (View)**, alegeti unul dintre procentele definite sau introduceți un alt procent după cum doriti.
5. Selectați, pe rând, cele patru moduri de vizualizare, observând care sunt modificările de aspect și de instrumente.
6. Închideți documentul.



## Exesăti

1. Scrieți în caiete asocierea corectă dintre cele două coloane:



- |  |  |
|--|--|
| A. Mod Focalizare<br>B. Panou de navigare<br>C. Zoom | I. afișarea mărită/micșorată a unui document<br>II. grilă din linii pe fundalul documentului<br>III. tip de afișare care favorizează concentrarea asupra documentului<br>IV. fereastră care afișează structura documentului cu hiperlinkuri către pagini |
|--|--|

2. Care este rezultatul accesării comenzi **Salvare ca (Save as)** din fila **Fișier (File)**?
3. Alegeti răspunsul corect la următoarele întrebări și scrieți-l în caiete:
  - I. Extensia unui fișier creat în MS Word poate fi:
    - a. .wav;
    - b. .docx;
    - c. .mp4.
  - II. Alegerea unui mod de vizualizare se realizează accesând:
    - a. grupul de comenzi **Vizualizări (Views)** din fila **Vizualizare (View)**;
    - b. grupul de comenzi **Afișare (Show)** din fila **Vizualizare (View)**;
    - c. grupul de comenzi **Fereastră (Window)** din fila **Vizualizare (View)**.




## Portofoliu

Alcătuiți un eseu de maximum o pagină, cu tema „Ce s-ar întâmpla dacă ați uită să salvați un document, iar termenul de predare este în două zile?”. Includeți în eseu propuneri de acțiuni atât pentru remedierea problemei, cât și proactive, de preîntâmpinare a unui astfel de eveniment.

# 4 Editarea unui document

## Descoperiți

### Inițializarea paginii

 Editarea unui document ar trebui începută cu inițializarea paginii, adică stabilirea exactă a caracteristicilor ei. Acestea pot fi configurate din fila **Aspect (Page Layout)** -> grupul **Inițializare Pagină (Page Setup)**, *imaginea a*.

De la stânga la dreapta avem următoarele comenzi:

1. **Margini (Margins)** – stabilirea marginilor paginilor documentului;
2. **Orientare (Orientation)** – stabilirea orientării paginii: **Porțret (Portrait)** – vertical sau **Vedere (Landscape)** – orizontal;
3. **Dimensiune (Size)** – stabilirea dimensiunii paginii;
4. **Coloane (Columns)** – stabilirea numărului de coloane în care se împarte scrisul pe pagină;



5. **Înteruperi (Breaks)** – introducerea de interruperi de pagină și secțiune, pentru ca textul să se continue din pagina, secțiunea sau coloana următoare;
6. **Numere de linie (Line numbers)** – opțiuni pentru afișarea numărului fiecărei linii (al fiecărui rând) din document;
7. **Despărțire în silabe (Hyphenation)** – opțiuni pentru despărțirea în silabe a cuvintelor la sfârșit de rând.

### Introducerea de diferite obiecte în document

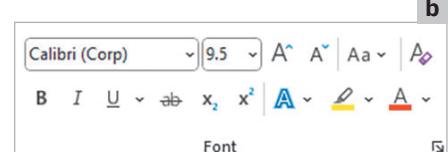
#### A. Introducerea de text

Când spunem „document”, ne gândim în primul rând la **text**.

Word pune la dispoziția utilizatorului instrumente pentru editarea textului, pentru configurarea proprietăților sale: tip, dimensiune, grosime, culoare, aspect etc.

 Acestea pot fi configurate utilizând comenzi care apar pe fila de lucru **Pornire (Home)** -> grupul **Font** (*imaginea b*). Urmărind ordinea de la stânga la dreapta și de sus în jos, comenzi sunt:

1. Alegerea tipului de font (caracter, literă) folosit;
2. Alegerea dimensiunii fontului;
3. Mărirea dimensiunii fontului;
4. Micșorarea dimensiunii fontului;
5. Modificarea textului selectat cu majuscule sau minuscule;
6. Ștergerea tuturor formatărilor;
7. Transformarea textului selectat în Aldin (**Bold**) (**Ctrl+B**) – text îngroșat (*imaginea c*);
8. Transformarea textului în Cursiv (**Italic**) (**Ctrl+I**) – text înclinat (*imaginea c*);
9. Sublinierea textului (**Underline**) (**Ctrl+U**) (*imaginea c*);
10. Tăierea textului cu o linie (*Strikethrough*) (*imaginea c*);
11. Literele devin mici, în dreapta jos a textului anterior, indice (*Subscript*) (**Ctrl+,=**) (*imaginea c*);
12. Literale devin mici, în dreapta, sus a textului anterior, exponent (*Superscript*) (**Ctrl+Shift+,+**) (*imaginea c*);
13. Adăugarea de efecte la un text;
14. Configurarea culorii de evidențiere a unui text (culoarea din spatele său);
15. Configurarea culorii fontului unui text.



#### Text Aldin (Bold)

#### Text italic

#### Text subliniat (Underline)

#### Text tăiat (Strikethrough)

#### Text<sub>subscript</sub>

#### Text<sup>superscript</sup>

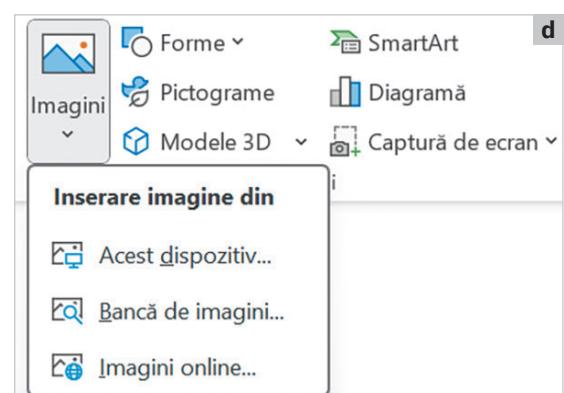
Toate aceste comenzi se aplică unui text selectat și reprezintă, de fapt, o formatare la nivel de caracter a textului introdus.

#### B. Introducerea de imagini

Găsiți instrumente pentru inserarea imaginilor accesând fila **Inserare (Insert)** -> grupul **Ilustrații (Illustrations)** *imaginea d*.

Puteți introduce într-un document:

1. imagini salvate local pe echipamentul vostru sau pe un echipament sau mediu de stocare extern, conectat la calculator (de exemplu, pe memory stick) – selectați comanda **Imagini (Pictures)** -> **Acest dispozitiv...** (*This Device...*), apoi navigați la locul unde este salvată imaginea dorită și alegeti-o, va fi inserată în document;
2. imagini de calitate foarte bună dintr-o bancă (bibliotecă) de imagini, pusă la dispoziție de aplicație – selectați **Imagini (Pictures)** -> **Bancă de imagini (Stock Images...)**;



## Editarea unui document

- 3.** imagini online, preluate de pe Internet – selectați comanda **Imagini (Pictures)** -> **Imagini Online (Online Pictures)** și apoi alegeți imaginea dorită (*imaginea e*). Aceasta va fi inserată în document.

Din același grup, **Ilustrații (Illustrations)**, din fila **Inserare (Insert)**, se mai pot adăuga și alte elemente grafice, cum ar fi:

- 1.** forme – apăsați butonul **Forme (Shapes)**;
- 2.** pictograme – sunteți foarte obișnuiți să inserați emoticoane în conborbirele voastre online; puteți acum și în Word ( ) – apăsați butonul **Pictograme**;
- 3.** modele 3D:

- dintr-un fișier salvat și stocat la voi – selectați **Modele 3D (3D Models)** -> **Acest dispozitiv... (This Device...)**;
- din biblioteca de modele 3D – accesați **Modele 3D (3D Models)** -> **Modele 3D din stoc... (Stock 3D Models...)**. *Imaginea f* conține un model 3D importat din această bibliotecă.

### C. Introducerea de tabele

Un **tabel** este o structură organizată

în rânduri (linii) orizontale și coloane verticale. O **celulă** a unui tabel este căsuță care se formează la intersecția unui rând cu o coloană a tabelului.

Puteți insera un tabel selectând fila **Inserare (Insert)** -> grupul **Tabele (Tables)** -> butonul **Tabel (Table)**. Alegeți apoi numărul de rânduri și coloane dorit, trecând cu mouse-ul peste tabela din meniu derulant care se deschide (*imaginea g*). La finalizarea acestei operații de marcare a numărului de rânduri/coloane dorite, faceți un click și tabelul se va insera în document.

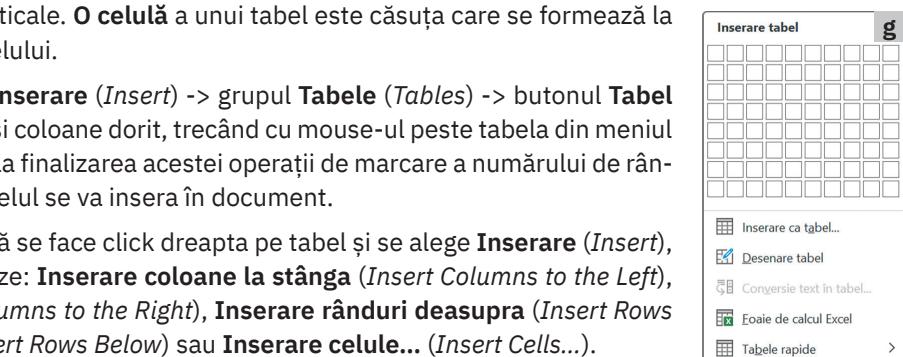
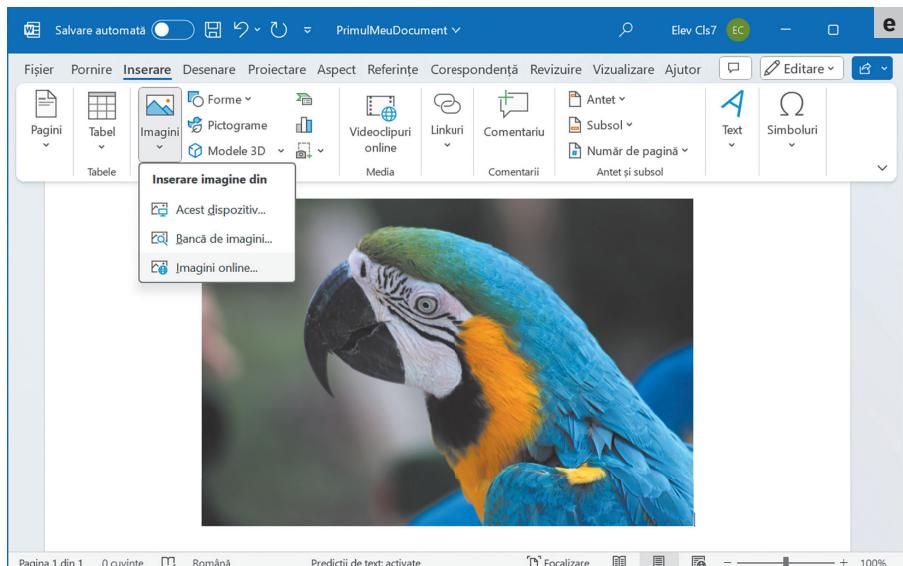
Pentru a adăuga un rând sau o coloană se face click dreapta pe tabel și se alege **Inserare (Insert)**, apoi se alege ce se dorește să se insereze: **Inserare coloane la stânga (Insert Columns to the Left)**, **Inserare coloane la dreapta (Insert Columns to the Right)**, **Inserare rânduri deasupra (Insert Rows Above)**, **Inserare rânduri dedesubt (Insert Rows Below)** sau **Inserare celule... (Insert Cells...)**.

### Amintiți-vă

Dacă doriți să includeți în document informații care nu vă aparțin (texte, imagini, tabele, modele 3D etc.), trebuie să vă asigurați că persoana care are dreptul de autor asupra lor permite acest lucru. Persoana care deține drepturile de autor poate fi creatorul operelor sau cineva care a primit aceste drepturi de la autor.

**Respectarea drepturilor de autor** este o obligație a fiecărui dintre noi, atât din punct de vedere legal, cât și din prisma respectului față de munca și creația persoanelor respective.

În momentul folosirii anumitor materiale în scop documentar sau dacă folosiți citate sau inserturi limitate din acestea, trebuie, întotdeauna, să menționați sursa de proveniență într-o secțiune Bibliografie a documentului.





## Descoperiți

### Operații de editare într-un document

Pentru a putea realiza operațiile de **copiere, mutare, ștergere**, este obligatoriu să aveți **selectate în prealabil** textele sau obiectele asupra cărora dorîți să faceți acțiunile.

#### A. Copiere la o altă locație (duplicare)

1. **Copiere** text/obiecte selectate: apăsați combinația de taste **Ctrl+C** sau selectați fila **Pornire (Home) -> grupul Clipboard -> Copiere (Copy)**.
2. **Lipire** la locul dorit: poziționați cursorul la locul destinație și apăsați combinația de taste **Ctrl+V** sau selectați fila **Pornire (Home) -> grupul Clipboard -> Lipire (Paste)**.

#### C. Ștergere

Pentru a șterge un text sau alte obiecte selectate, apăsați tasta **Delete (Ștergere)** de la tastatură.

#### B. Mutare la o altă locație

1. **Decupare** text/obiecte selectate: apăsați combinația de taste **Ctrl+X** sau selectați fila **Pornire (Home) -> grupul Clipboard -> Decupare (Cut)**.
2. **Lipire** la locul dorit: poziționați cursorul la locul destinație și apăsați combinația de taste **Ctrl+V** sau selectați fila **Pornire (Home) -> grupul Clipboard -> Lipire (Paste)**.



## Aplicați și investigați

Realizați următoarele acțiuni în ordinea în care sunt specificate și observați rezultatele:

1. Deschideți documentul creat anterior, **PrimulMeuDocument**, din folderul **Lucru\_clsVII**.
2. Scrieți un titlu în document: „Învăț ceva atunci când mă joc?”. Configurați ca fontul pentru acest text să fie Arial, cu dimensiunea 18, Aldin (Bold).
3. Scrieți un text de minimum 5 și maximum 20 de rânduri, cu fontul Times New Roman, de dimensiune 14. În acest text, prezentați-vă opinia despre ce considerați că învățați atunci când vă jucați jocul preferat.
4. Descărcați de pe site-ul <https://pixabay.com> două imagini gratuite despre jocul preferat pe care îl jucați. Poate fi un joc pe calculator/tabletă sau un alt tip de joc, la alegere. Inserați în document cele două imagini descărcate de pe Internet.
5. Introduceți în textul vostru un tabel în care să precizați cât timp vă propuneți să vă jucați în fiecare zi a săptămânii.

	Luni	Marți	Miercuri	Joi	Vineri	Sâmbătă	Duminică
Timp propus (în minute)							

6. Salvați documentul în folderul **Lucru\_clsVII**.



## Exesăti

1. Scrieți, în caiete, răspunsurile corecte.

- I. Dacă dorim să dublăm un element într-un document, facem operațiile:
  - a. selectare element, **Copiere (Copy)**, **Lipire (Paste)**;
  - b. **Copiere (Copy)**, **Lipire (Paste)**;
  - c. selectare element, **Decupare (Cut)**, **Lipire (Paste)**.
- II. Pentru adăugarea unui tabel într-un document Word accesăm:
  - a. fila **Referințe (References) -> grupul Tabele (Tables) -> Tabel (Table)**;
  - b. fila **Inserare (Insert) -> grupul Pagini (Pages) -> Tabel (Table)**;
  - c. fila **Inserare (Insert) -> grupul Tabele (Tables) -> Tabel (Table)**.
- 2. Dați trei exemple de elemente care pot fi inserate într-un document.
- 3. Completați, în caiete, cuvintele lipsă din următoarele enunțuri:
  - a. Combinarea de taste **Ctrl+U** se folosește pentru ..... unui text, combinarea de taste **Ctrl+B** se folosește pentru transformarea textului în ....., iar combinarea de taste **Ctrl+I** se folosește pentru transformarea textului în .....
  - b. O **celulă** a unui tabel este căsuța care se formează la intersecția unui ..... cu o ..... a tabelului.



## Portofoliu

Realizați o scrisoare de maximum o pagină către bunici, în care îi rugați să vă ajute să faceți o campanie de colectare de haine și jucării pentru copiii dintr-un centru de plasament. Documentul trebuie conțină minimum o imagine și un tabel, iar textul să fie scris cu font Tahoma de dimensiune 12. Salvați scrisoarea în folderul **Lucru\_clsVII** cu numele **Scrisoare**.



# 5 Formatarea unui document. Aspectul paginii



## Descoperiți

### Operații de aranjare în pagină și formatare a obiectelor dintr-un document

#### A. Text

Vom explica, în cele ce urmează, formatarea textului la nivel de paragraf.



#### Rețineți

Numim paragraf un capitol sau o subdiviziune dintr-un capitol al unui text, un fragment unitar care prezintă o anumită idee.

Puteți configura caracteristicile paragrafelor din text (aliniere, distanță dintre rânduri, orientare etc.) selectând butoanele din fila **Pornire (Home)** -> grupul **Paragraf (Paragraph)**, *imaginea a*.

Urmărind ordinea de la stânga la dreapta și de sus în jos, comenzi sunt:

- 1. Marcatori (Bullets)** – crearea unei liste cu marcatori;
- 2. Numerotare (Numbering)** – crearea unei liste numerotate;
- 3. Crearea unei liste cu mai multe niveluri (Multilevel List);**
- 4. Mutarea paragrafului mai aproape de marginea din stânga (Decrease Indent);**
- 5. Mutarea paragrafului mai departe de marginea din stânga (Increase Indent);**
- 6. Ordonarea alfabetica/numerica a unei selecții;**
- 7. Afisarea integrală a textului, inclusiv a tuturor marcajelor de paragraf sau a altor simboluri de formatare ascunse;**
- 8. Alinierea la stânga a textului (Align Left, Ctrl+L);**
- 9. Centrarea textului (Center, Ctrl+E);**
- 10. Alinierea la dreapta a textului (Align Right, Ctrl+R);**
- 11. Alinierea stânga-dreapta a textului (Justify, Ctrl+J) – distribuirea uniformă a textului pe linie;**



- 12. Configurarea spațiului dintre liniile unui paragraf sau dintre paragrafe;**
- 13. Umbrire (Shading) – schimbarea culorii care apare în spatele unui text;**
- 14. Borduri (Borders) – adăugarea de borduri (margini) pentru o selecție.**

Amintiți-vă că la un program informatic numeam **indentare** aranjarea liniilor astfel încât lizibilitatea să fie crescută (să se poată citi ușor). Se poate aplica acest termen și la un text simplu, reprezentând aranjarea liniilor față de marginile stânga/dreapta ale documentului.

Aspectul unui paragraf mai poate fi controlat și prin comenzi aflate în grupul **Paragraf (Paragraph)** din fila **Aspect** (*imaginea b*). Urmărind ordinea de sus în jos și de la stânga la dreapta, comenzi sunt următoarele:

#### Operații de indentare

- 1. Mutarea paragrafului mai departe de marginea din stânga (Indent Left)**
- 2. Mutarea paragrafului mai departe de marginea din dreapta (Indent Right)**



#### Operații de spațiere

- 3. Modificarea spațiului care apare deasupra paragrafului (Spacing Before)**
- 4. Modificarea spațiului care apare după paragraf (Spacing After)**

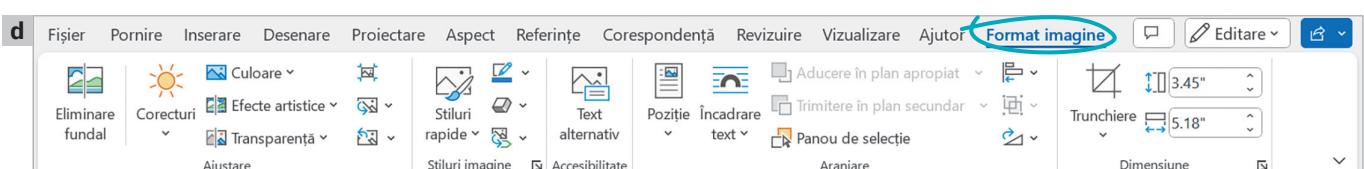
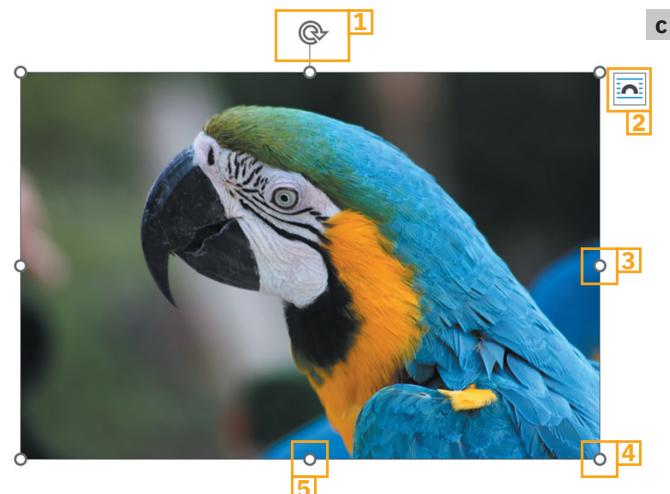
#### B. Imagini

În cadrul unui document, imaginile fac și ele parte din paragrafe, astfel că li se pot aplica și comenzi din grupul **Paragraf (Paragraph)** al filei **Pornire (Home)**. Așadar, la fel ca în cazul textului, li se pot aplica și imaginilor aceleași operații de aranjare în pagină. Unei imagini selectate i se pot aplica, de asemenea, comenzi de aliniere în cadrul unui paragraf (comenzi 8, 9, 10 și 11 descrise mai sus pentru texte).

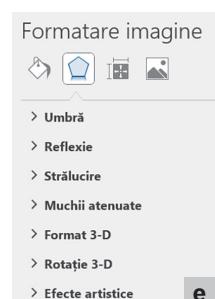
► Și aspectul unei imagini poate fi modificat, fiind puse la dispoziție câteva comenzi (*imaginea c*):

- 1. Rotirea imaginii în jurul centrului dreptunghiului care o include;**
- 2. Alegerea modului în care imaginea se încadrează față de textul din jur (este așezată în linie cu textul sau se întrepătrunde cu textul etc.);**
- 3. Mărirea/micșorarea pe orizontală a imaginii;**
- 4. Mărirea/micșorarea proporțională (orizontal și vertical) a imaginii;**
- 5. Mărirea/micșorarea pe verticală a imaginii.**

Un click pe imagine face să apară o filă nouă, **Format imagine**, care conține comenzi extinse pentru formatarea imaginilor (*imaginea d*). Această filă se poate deschide direct cu dublu click pe imaginea dată.



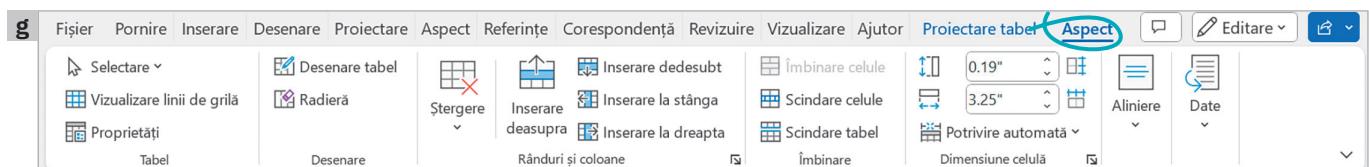
O altă variantă de a afișa instrumente pentru formatarea imaginilor este să faceți click dreapta pe imagine, apoi selectați **Formatare imagine** (*Format Picture*). Se va deschide un nou panel în care se pot alege caracteristicile dorite (*imaginea e*).



## C. Formatare tabele

În cadrul unui document, tabelele fac și ele parte din paragrafe, astfel că li se pot aplica și comenzi din grupul **Paragraf** (*Paragraph*) al filei **Pornire** (*Home*). Se pot aplica astfel și tabelelor aceleași operații de aranjare în pagină ca în cazul textului. Unui tabel selectat î se pot aplica, de asemenea, comenzi de aliniere în cadrul unui paragraf (comenzile 8, 9, 10 și 11 descrise pentru texte).

Un click pe tabel face să apară două file noi, **Proiectare tabel** (*imaginea f*) și **Aspect** (*imaginea g*), care conțin instrumente diverse pentru tabele.



## D. Copierea unei anumite formatări

În cazul în care aveți anumite elemente al căror aspect (formatare) vă place și dorîți să îl aplicați și altor elemente din document, puteți face acest lucru utilizând comanda **Descriptor de formate** (*Format Painter*) aflată în grupul **Clipboard** al filei **Pornire** (*Home*), *imaginea h*. Se selectează elementul „model”, al cărui aspect dorîți să îl copiați, apoi selectați comanda **Descriptor de formate** (*Format Painter*) și, la pasul al treilea, selectați zona din document căreia dorîți să îl aplicați formatul (zona „tîntă”).

Comanda **Descriptor de formate** (*Format Painter*) poate fi selectată și din meniul care se deschide la click dreapta pe zona model selectată (*imaginea i*).

Dacă dorîți să aplicați formatul elementului model mai multor zone din document, în loc să selectați comanda **Descriptor de formate** (*Format Painter*), faceți dublu click pe ea. Selectați apoi toate zonele dorite și se va aplica și acestora formatarea aleasă. Finalizați această copiere de format cu un nou click pe comandă.



## Reguli generale de tehnoredactare

- +**
1. Documentele scrise în limba română trebuie să conțină diacritice.
  2. La finalul unui rând nu se apasă tasta **Enter** decât dacă se dorește începerea unui paragraf nou. Altfel, se continuă scrierea, iar trecerea pe rândul următor se va face automat.
  3. Semnele de punctuație (punctul, virgula, punctul-și-virgula, două-puncte, semnul întrebării, semnul exclamării) se pun lipite de ultimul caracter de dinainte. Nu se lasă niciun spațiu înaintea lor, în schimb, se lasă un spațiu imediat după ele. Excepția este la sfârșitul unui paragraf, când nu se mai pune spațiu la final, sau în cazul prescurtărilor de tipul S.R.L. în care nu se mai pune spațiu după anumite semne.
  4. Nu se lasă niciun spațiu înainte de parantezele de închidere și de ghilimele de închidere, ci se lasă un spațiu după acestea. Excepție: la sfârșit de paragraf nu se mai pune spațiu după semn.
  5. Înainte de ghilimelele de deschidere și de parantezele de deschidere se lasă un spațiu, însă nu se lasă niciun spațiu după ele.

6. Liniuța de unire „-” și bara oblică „/” se pun lipite de ambele cuvinte din stânga și din dreapta.
7. Între două cuvinte se lasă întotdeauna un singur spațiu.
8. Se lasă un singur spațiu între valori și unitățile de măsură. Niciodată nu se desparte la capătul rândului valoarea de unitatea de măsură.
9. Pentru o mai bună lizibilitate a unui text se lasă un spațiu mai mare între paragrafe.



## Reguli de estetică a paginii tipărite

1. Un document trebuie să aibă un aspect unitar, armonios.
2. Într-un document nu trebuie să se întâlnească prea multe fonturi. Documentul trebuie să fie cât mai unitar. De asemenea, este importantă dimensiunea fonturilor pentru a se putea citi ușor.
3. Textele marcate cu Aldin (**Bold**), Cursiv (*Italic*) și cele subliniate nu se folosesc excesiv. Ele au rolul de a scoate în evidență anumite aspecte sau idei și trebuie să fie cât mai vizibile.
4. Trebuie să existe întotdeauna un contrast între fundal și scrisul de pe acesta, pentru a se putea citi ușor.
5. Elementele care se introduc într-un document nu trebuie să depășească zona acestuia; pentru a vedea încadrarea, se poate folosi instrumentul **Linii de grilă (Gridlines)** din grupul **Afișare (Show)** al filei **Vizualizare (View)**.
6. Tabelele introduse trebuie să fie de mici dimensiuni; dacă se întind pe mai multe pagini, devin greu de citit.

## Mici trucuri care pot fi de ajutor

- Pentru aranjarea în pagină a elementelor se recomandă plasarea acestora în tabele fictive, care nu au liniaturi vizibile. Pentru aceasta, se inserează un tabel cu câte rânduri și coloane avem nevoie, în celulele sale se plasează elementele care dorim să fie aliniate, apoi facem invizibilă liniatura tabelului selectând **Pornire (Home) -> Paragraf (Paragraph) -> Borduri (Borders) -> Fără bordură (No border)**.
- Formatul de pagină A4 vă scutește de multe probleme care pot să apară la o imprimare normală pe coli de hârtie A4, cea mai uzuală cu puțință. Deci, configurați acest format dacă nu aveți alte constrângeri. Pentru aceasta, selectați A4 după ce accesați fila **Aspect (Page Layout) -> Inițializare pagină (Page Setup) -> Dimensiune (Size)**.
- Dacă pregătiți un document care urmează să fie tipărit și îndosariat, atunci trebuie să configurați ca marginea din stânga să fie suficient de mare, astfel încât la îndosariere să se poată citi textul tipărit. Pentru aceasta, selectați fila **Aspect (Page Layout) -> Inițializare pagină (Page Setup) -> Margini (Margins)**.



## Aplicați și investigați

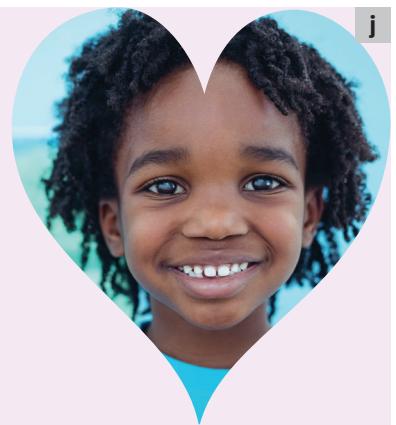
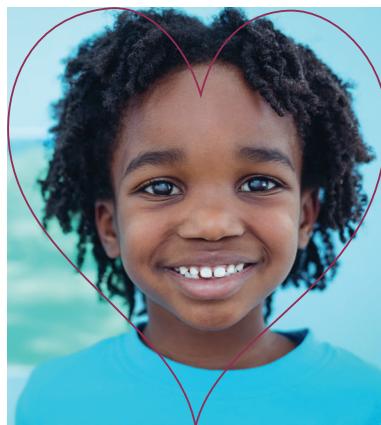
Realizați următoarele acțiuni în ordinea în care sunt specificate și observați rezultatele:

1. Creați un document nou, necompletat, și salvați-l cu numele **Suntem veseli** în folderul **Lucru\_clsVII**.
2. Scrieți un text de minimum trei și maximum cinci rânduri, cu fontul Arial, de dimensiune 12, aliniat stânga-dreapta (*Justify*). Acest text trebuie să conțină câteva idei legate de beneficiile râsului asupra sănătății și a stării de bine.
3. Descărcați de pe site-ul <https://pixabay.com/> (sau de pe un alt site care pune la dispoziție imagini gratuite) două imagini pe care să le utilizați în acest document.

4. Scrieți ca titlu în document: „De ce râsul este sănătos”. Configurați ca fontul pentru acest text să fie Calibri, cu dimensiunea 20, Aldin (Bold), iar textul să fie centrat.

5. Inserați cele două imagini descărcate de pe Internet. Rotiți imaginile și schimbați-le dimensiunea după cum doriți.

6. Având selectată prima imagine, deschideți fila **Format imagine** care conține instrumente pentru imagine și parcurgeți comenzi din grupurile **Ajustare** și **Stiluri imagine**. Alegeti aspectul dorit pentru aceasta.



j

7. Având a doua imagine selectată, alegeti fila **Format imagine** -> grupul **Dimensiune (Size)** -> **Trunchiere (Crop)** -> **Trunchiere la formă**, iar apoi alegeti forma de „inimă” (exemplu de trunchiere în *imaginea j*).

8. Introduceți în textul vostru un tabel în care să inserați răspunsul vostru la un minichestionar. Tabelul va avea forma de mai jos și va conține trei afirmații pe care trebuie să le analizați și să spuneți în ce măsură sunt adevărate pentru voi.

Pentru a da răspunsul, considerați o scală de la 1 la 5, unde 1 înseamnă foarte puțin, iar 5 înseamnă foarte mult. Pentru fiecare afirmație, scrieți un X în tabel, în dreptul numărului care reprezintă răspunsul vostru.

	1	2	3	4	5
Râsul te face să fii mai optimist(ă)?					
Râsul îmi creează o stare de bine.					
În fiecare zi, am momente în care râd cu poftă.					

9. Având tabelul selectat, din fila **Proiectare tabel** alegeti un stil de tabel, accesând comenzi din grupul **Stiluri (Table Styles)**, și o bordură, utilizând grupul de comenzi **Borduri (Borders)**.

10. Salvați documentul în folderul **Lucru\_ClsVII**.

## Exercăti



1. Enumerați patru reguli generale de tehnoredactare a unui document electronic.

2. Selectați A (Adevărat) sau F (Fals), în funcție de valoarea de adevăr a propozițiilor:

- a. Butoanele pentru alinierarea textului dintr-un paragraf se află în fila **Pornire (Home) -> Paragraf (Paragraph)**. A F
- b. Un tabel inserat în Word poate fi rotit la 90°. A F
- c. O imagine poate fi mărită/micșorată pe orizontală și verticală. A F

3. Explicați cum se creează o listă cu marcatori (*bullets*). Dați exemple de alte două elemente care pot fi incluse în această listă în afară de text.

4. Alegeti varianta corectă pentru enunțul de mai jos.

Decuparea unei imagini după o anumită formă se face utilizând comanda:

- a. fila **Format imagine -> Trunchiere -> Trunchiere la formă**;
- b. fila **Pornire -> Decupare -> Decupare la formă**.

5. Asociați corect, pentru aplicația Word, acțiunile din prima coloană cu tastele din cea de-a doua coloană:

- |  |                              |
|--|------------------------------|
| A. Centrarea unui text selectat                | I. Se apasă tastele Ctrl+R   |
| B. Alinierarea la dreapta a unui text selectat | II. Se apasă tastele Ctrl+L  |
| C. Alinierarea la stânga a unui text selectat  | III. Se apasă tastele Ctrl+J |
|  | IV. Se apasă tastele Ctrl+E  |

6. **Dezbateră.** Inițiați o dezbatere în clasă pe tema „Utilizarea de imagini și tabele într-un document Word aduce sau nu plus valoare?”.



## Portofoliu



Realizați în Microsoft Word o diplomă pentru colegii voștri mai mici din clasa I, în care să fie trecute și calificativele obținute la materiile de studiu (într-un tabel). Salvați-o cu numele **Diploma** în folderul **Lucru\_clsVII**.

# Concurs: Cel mai inspirat... REBUS

U1

## Organizare

Formați echipe de câte trei-patru elevi. Fiecare echipă i se va aloca un număr începând cu 1.

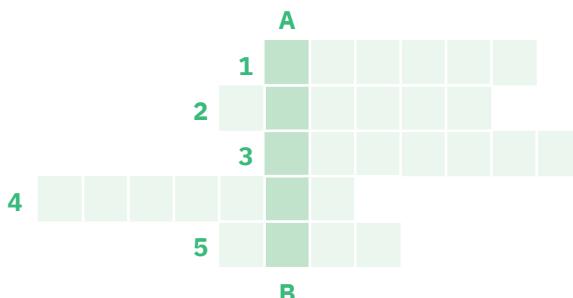
## Tema concursului

Fiecare echipă are ca sarcină să realizeze un proiect: să creeze un rebus (careu și definiții) pe o temă la alegere și să îl redacteze în MS Word. Rebusul trebuie să conțină între patru și șase cuvinte, iar acestea trebuie să fie aranjate astfel încât pe o verticală să se formeze un cuvânt. La finalul activității, realizați un concurs de rezolvat cuvinte încrucișate.

Un exemplu de astfel de rebus este următorul (soluția se ține secretă de către fiecare echipă până la finalizarea concursului):

### Tema: Toamna

Pe verticala A-B se va forma numele unui fruct care se coace toamna (la plural).



- Precipitație atmosferică sub formă unor picături de apă.
- Se colorează în galben toamna (plural).
- O purtăm atunci când plouă.
- Se coc toamna, cresc în pomi și au un înveliș cu țepi. Se mănâncă fierte sau coapte.
- Fructe cu formă rotundă, turtită, care pot avea mai multe culori (roșu, galben, verde).

### Răspuns:

		A							
1	P	L	O	A	I	E			
2	F	R	U	N	Z	E			
3	U	M	B	R	E	L	A		
4	C	A	S	T	A	N	E		
5	M	E	R	E					
		B							

## Timpul de lucru

Alocat de cadrul didactic.

## Cum veți lucra

Stabiliti tema rebusului și cuvântul de pe verticala A-B. Stabiliti sarcinile fiecărui membru al echipei. Documentați-vă și creați rebusul (careul și definițiile) și realizați documentul pe calculator, în MS Word. Salvați-l cu numele **Rebus**, în folderul **Lucru\_ClsVII**.

## Cum se face evaluarea

Se fac biletele pe care sunt scrise numere de la 1 la câte echipe sunt în total. Fiecare echipă va trage la întâmplare unul dintre aceste biletele, până când trage un număr diferit de numărul său de echipă și va rezolva rebusul pregătit de echipa cu acel număr tras la sorti. Va fi câștigătoare echipa care va rezolva cel mai repede rebusul selectat.

Rebusurile pot fi tipărite pentru a fi rezolvate sau echipele se pot muta astfel încât fiecare dintre acestea să ajungă la calculatorul pe care s-a pregătit rebusul tras la sorti și să îl rezolve acolo.



## Lista de verificare a proiectului Rebus

- Am respectat tema proiectului:
  - În întregime
  - Parțial
- În timpul de lucru alocat am terminat proiectul:
  - În întregime
  - Parțial
- Nivelul de documentare în vederea elaborării proiectului este:
  - Foarte bun
  - Bun
  - Slab
- Comunicarea cu colegii a fost:
  - Foarte bună
  - Bună
  - Slabă
- Nivelul de creativitate la realizarea proiectului a fost:
  - Foarte bun
  - Bun
  - Slab
- Rebusul proiectat le-a plăcut colegilor:
  - În mare măsură
  - În măsură normală
  - În mică măsură

# Exerciții recapitative

Rezolvați pe caiete sarcinile de lucru de mai jos.

1. Ce este un procesor de text?
2. Dintre următoarele elemente, care pot fi introduse într-un document Word?
 

<b>a.</b> tabele;	<b>c.</b> pictograme;	<b>e.</b> sunete;
<b>b.</b> desene;	<b>d.</b> text;	<b>f.</b> modele 3D.
3. Asociați corect, pentru aplicația Word, acțiunile din prima coloană cu tastele din cea de-a doua coloană:

A. copiere text selectat	I. Se apasă tastele Ctrl+C
B. lipire text copiat	II. Se apasă tastele Ctrl+X
C. decupare text selectat	III. Se apasă tastele Ctrl+V
	IV. Se apasă tastele Ctrl+S

4. Dintre aplicațiile următoare, scrieți-le în caiete pe cele care sunt procesoare de text:
 

<b>a.</b> MS Word;	<b>c.</b> LibreOffice;	<b>e.</b> Apache OpenOffice Writer;
<b>b.</b> MS Outlook;	<b>d.</b> Notepad;	<b>f.</b> Google Docs.
5. Completați, în caiete, enunțurile următoare, folosind cuvintele potrivite:
  - a. Noțiunea de ..... este prezentată de *Dicționarul explicativ al limbii române* (2009) ca fiind „pregătirea tehnică și grafică a unui manuscris înainte de a începe operația de tipărire”.
  - b. Aplicațiile software care permit utilizatorilor crearea, editarea, formatarea, aranjarea în pagină și tipărirea materialelor poartă numele de .....
  - c. Procesoarele de text permit ..... textului, stocarea acestuia în format electronic, ..... pe ecran, modificarea și ..... lui la imprimantă.
6. Dați trei exemple de operații de bază, din MS Word, pentru gestionarea fișierelor.
7. Ce sunt și la ce se folosesc liniile de grilă din MS Word?

## Autoevaluare

Timp de lucru: **15 minute**

Rezolvați în caiet exercițiile date. Evaluați-vă singuri rezultatele.

1. Realizați, în caiete, corespondența dintre definiție și noțiune, din tabelul de mai jos:
 

A. Include toate comenzile aplicației Microsoft Word grupate pe file (tabs), pe tipuri de acțiuni	1. Bara de titlu
B. Bară verticală sau orizontală care permite utilizatorului să mute aria de afișare a documentului sus/jos și, respectiv, stânga/dreapta	2. Panglica (Ribbon)
C. Contine bara de acces rapid, numele fișierului, opțiunile de afișare a panglicii și panoul de management al ferestrei	3. Bara de defilare
	4. Bara de stare
2. Numiți două avantaje practice ale stocării unui fișier document în cloud.
3. Scrieți **A** sau **F** pentru fiecare dintre afirmațiile de mai jos, legate de un document Word:
 

<ol style="list-style-type: none"> <li>a. Un paragraf are întotdeauna același spațiu deasupra sa.</li> <li>b. Textul editat poate să aibă diferite tipuri de font și culori.</li> <li>c. O imagine de pe Internet inserată în document poate fi redimensionată.</li> </ol>	A   F A   F A   F
--	-------------------------

### CUM VĂ APRECIATI?

- 1 Desenați ● în dreptul sarcinilor pe care le-ați rezolvat corect.

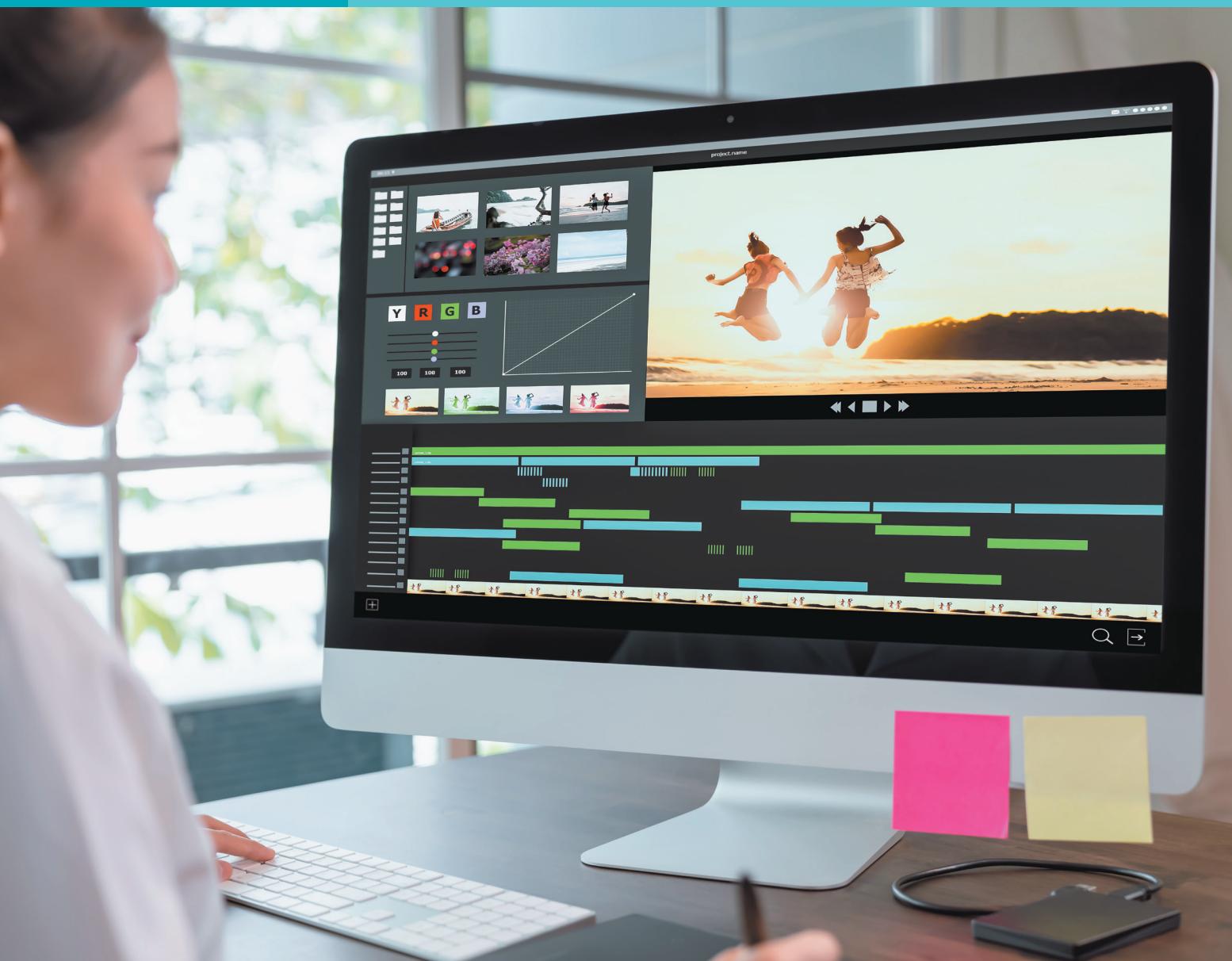


- 2 Scrieți cum vă simțiți după ce ați rezolvat acest test, alegând unul dintre cuvintele de mai jos:
  - nemulțumit(ă);
  - mulțumit(ă);
  - încântat(ă).



# U2

## Aplicații de prelucrare audio-video



Lecția 1	28-29	Aplicații de prelucrare audio-video. Scop și avantaje
Lecția 2	30-32	Aplicația WavePad. Elemente de interfață
Lecția 3	33-35	Prelucrarea sunetelor
Lecția 4	36-39	Programe de editare video. Elemente de interfață. VideoPad
Lecția 5	40-42	Prelucrarea video
Lecția 6	43-45	Prelucrări superioare în proiectul video
Exerciții recapitulative	46	
Autoevaluare	46	

# 1 Aplicații de prelucrare audio-video. Scop și avantaje

## Descoperiți

În jurul nostru este o abundență de informații, de conținut multimedia. Internetul, rețelele de socializare, radiourile și televiziunile, toate acestea ne furnizează informații text, audio, video și chiar interactive. Și, dintre toate, sunt preferate tot mai mult cele audio-video.

Dar oare cum sunt produse aceste materiale audio-video care ajung la noi? În principiu, totul este simplu:

- sunetele și/sau imaginile video sunt captate (înregistrate) cu ajutorul unor aparate specializate (reportofon, cameră video sau echipamente de studio de sunet – microfon, mixer, placă de sunet, calculator);
- apoi sunt prelucrate (prelucrările sunt de ordin calitativ, dar și cantitativ – se reține doar ce se consideră necesar din materialul înregistrat, material care poate fi mai lung sau care poate avea mai multe versiuni, provenite din mai multe încercări care au fost înregistrate);
- se adaugă diferite efecte, dacă este cazul;
- se mai pun două secvențe „șablon” (câteva secunde de intro și câteva secunde de încheiere);
- apoi se exportă (sau se salvează direct, de la caz la caz) în formatul de fișier dorit.

Pentru ce putem folosi aplicațiile audio-video și ce avantaje avem dacă le folosim? Aceste aplicații pot face din înregistrările voastre audio sau video superproduții ca de studio profesional.



De la an la an apar noi versiuni ale programelor, cu tot mai multe și mai spectaculoase opțiuni disponibile, oferindu-ne tuturor posibilitatea de a realiza creații multimedia apropriate de calitatea profesională. Este suficient să avem un calculator sau chiar un telefon intelligent. Telefoanele inteligente revoluționează industria telefoanelor, iar camerele foto din componentă lor nu mai sunt „accesoriu” banale, ci le transformă în adevarate studiouri de sunet și imagine.

► Un simplu telefon intelligent, dotat cu aplicații audio-video, vă poate transforma videoclipurile în capodopere. Cu aplicațiile potrivite, puteți decupa secvențe din înregistrările voastre, puteți adăuga efecte speciale și filtre de care experții de la Hollywood ar fi mândri.

## Rețineți

**Prelucrarea audio-video** este acea succesiune de operații tehnice și artistice prin care un material înregistrat, audio sau video, suferă anumite transformări, astfel încât rezultatul final să fie un material de mai bună calitate, care să îndeplinească anumite specificații (să aibă o anumită durată, anumite caracteristici tehnice) și să atingă scopul pentru care a fost creat.

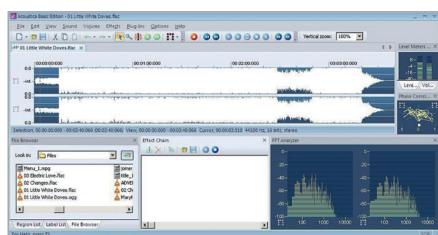
Există o multitudine de aplicații dedicate înregistrării și editării fișierelor audio. Unele se pot folosi online, în browserul web, altele trebuie instalate local, pe calculator sau telefonul intelligent, și sunt disponibile pentru diferite sisteme de operare (Windows, macOS, Linux, Android, iOS, Windows Mobile). Diferențele dintre ele sunt date de numărul de prelucrări pe care le pot face, de efectele speciale disponibile (efect de ecou, efect de sală de concerte, efect de voce robotizată și multe altele), de numărul de canale (track-uri audio) pe care le pot înregistra simultan (pentru a putea înregistra, de exemplu, toate instrumentele dintr-o formație, fiecare pe câte un canal, astfel încât ulterior să putem face reglaje fine pe fiecare instrument în parte), dar toate permit realizarea operațiunilor de bază:

- înregistrarea;
- reglarea volumului;
- egalizarea tonurilor;
- decuparea unor fragmente din înregistrare;
- lipirea mai multor fragmente;
- mixarea a două înregistrări;
- salvarea fișierelor și exportul lor în diferite formate audio (.wav, .mp3, .wma, .ogg, .au, .m4a, .vox etc.).

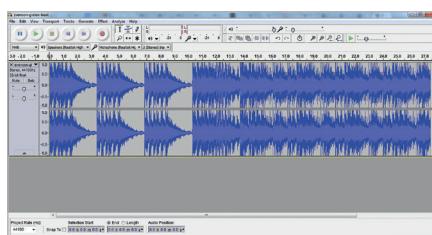
Câteva dintre cele mai populare **aplicații de prelucrare audio** ale momentului (și gratuite sau, cel puțin, gratuite pentru utilizare în scop necomercial) sunt: Audio Recorder for Free, Audacity (poate lucra multitrack), WavePad, Ocenaudio, Hya-Wave, TwistedWave, Wavosaur, Soundation, Acoustica Basic, Audio Cutter Pro, Nero Wave Editor, WaveShop, Qtractor, Audio MP3 Cutter Mix Converter.

La fel ca pentru aplicațiile audio, există o multitudine de aplicații dedicate înregistrării și editării fișierelor video. Operațiile de bază sunt disponibile în toate aplicațiile, iar decizia de a folosi o aplicație sau alta poate fi dată de preferințe de ordin estetic, de nevoile concrete de prelucrare, de ușurința utilizării, de cantitatea de memorie ocupată, de sistemul de operare pe care vrem să lucrăm etc.

## Aplicații de prelucrare audio-video. Scop și avantaje



Acoustica Audio Editor



Audacity



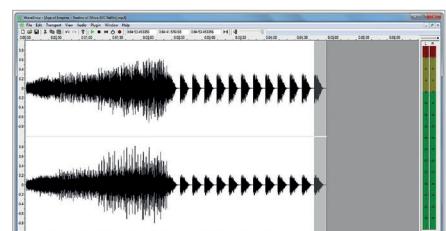
Nero Wave Editor



Audio MP3 Cutter Mix Converter

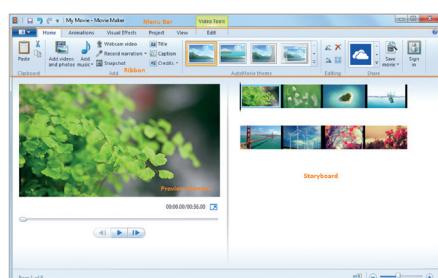


Ocenaudio

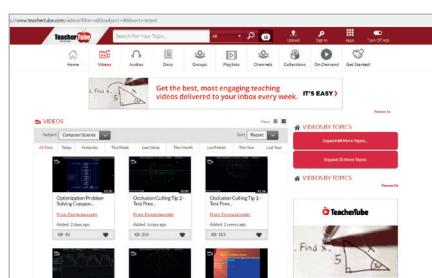


WaveShop

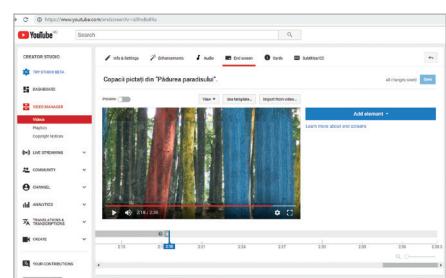
 Iată câteva exemple de **aplicații utilizate pentru redarea și procesarea fișierelor multimedia**: VirtualDub, VLC media player, BS Player, MovieMaker, TeacherTube, YouTube, VideoFX Live, VideoShow etc.



MovieMaker



TeacherTube



YouTube

Aplicațiile de editare video fac mai mult decât să vă ajute să decupați porțiuni din înregistrarea video. Unele permit aplicarea de efecte speciale, de efecte de trecere între cadre diferite etc. Pe lângă efectele care sunt ușor de aplicat, aplicațiile de editare video au multe alte caracteristici. Acestea vă permit să adăugați muzică de fundal, titluri (text), să vă înregistrați propriile narațiuni, să aplicați efecte sonore (așa cum se întâmplă și cu imaginile).

Majoritatea aplicațiilor de prelucrare video permit și prelucrarea sunetelor. Se pot aplica anumite operații pe partea de sunet, direct în aplicația de procesare a fișierelor video.

### Aplicați și investigați

► Căutați pe Internet informații despre trei aplicații utilizate pentru prelucrarea audio/video. Realizați un document care să conțină câteva informații despre fiecare aplicație studiată, inclusiv funcții principale, scopul utilizării, avantaje și dezavantaje, precum și exemple concrete de proiecte sau situații în care aceste aplicații sunt eficiente. Este important să menționați sursele de unde ati obținut informațiile.

### Exersați

1. Scrieți, în caiete, A (Adevărat) sau F (Fals), în funcție de valoarea de adevăr a propozițiilor:
  - a. Cu o aplicație audio putem edita și secvențe video. A F
  - b. Cu o aplicație video putem edita și prelucra și sunete. A F
  - c. Cu o aplicație audio putem edita și prelucra documente. A F
2. Care sunt operațiunile de bază pentru a obține un material audio de bună calitate (de exemplu, un interviu de două minute luat unui coleg)?
3. Explicați, în cuvintele voastre, ce este prelucrarea audio-video.
4. **Dezbateră.** Organizați o dezbatere cu tema „Editarea fișierelor multimedia poate duce la manipularea ascultătorului/privitorului. Etica în realizarea productiilor media. Ce este și ce nu este etic să facem într-un montaj multimedia”.

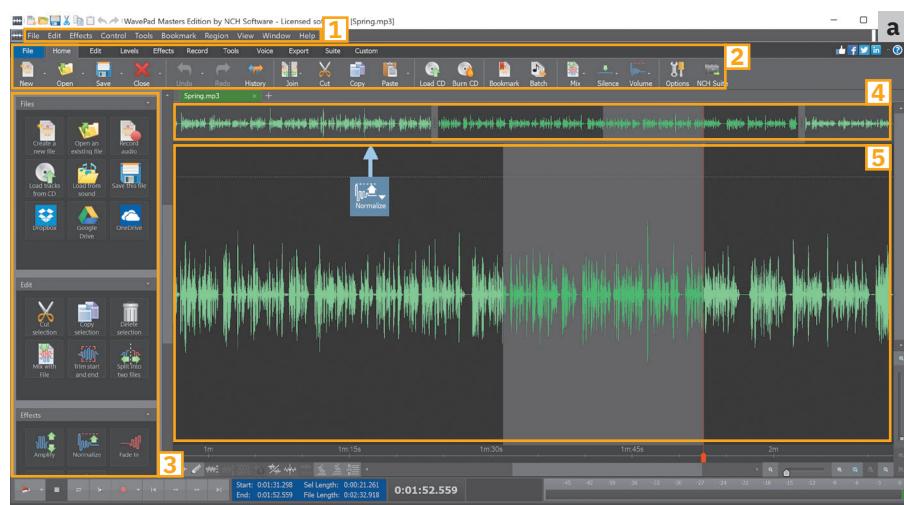
## 2 Aplicația WavePad. Elemente de interfață

### Descoperiți

#### Elemente de interfață ale aplicației WavePad

În cele ce urmează vom face cunoștință cu aplicația numită **WavePad**, una dintre cele mai populare aplicații pentru înregistrare și prelucrare de sunete. Pentru utilizare necomercială (deci și pentru utilizare în scopuri educative) este gratuită. Interfața programului WavePad arată astfel (*imaginea a*):

- 1. Bara de meniu** – conține acțiunile disponibile în WavePad;
- 2. Filele (Tabs)** – acestea organizează instrumentele și funcționalitățile programului WavePad, pentru a vă eficientiza fluxul de lucru. Dați click pe o filă pentru a vedea funcțiile și instrumentele disponibile;
- 3. Bara de comenzi (în laterală din stânga)** – conține legături către cele mai utilizate acțiuni, comenzi din WavePad. Pentru a controla afișarea barei de comenzi sau poziția în care este afișată, faceți click pe meniul **View (Vizualizare)** și apoi pe **Command Bar (Bara de comenzi)**;
- 4. Fereastra de sus a formei de undă** (a fișierului audio) – felul în care este reprezentat grafic sunetul seamănă cu un val, de unde vine și numele acestor forme de afișare (wave în engleză). Unui sunet mai tare, mai puternic, îi corespunde o linie mai mare în „valul” formei grafice care reprezintă sunetul întreg din fișierul de sunet;
- 5. Fereastra de jos a formei de undă** (a fișierului audio).



După cum vedem mai sus, pe ecranul principal sunt două ferestre pentru forma de undă a sunetului. Fereastra de sus, mai mică (4), afișează înălțimea unei forme de undă a fișierului audio în întregime, în toată lungimea sa – este utilă pentru a săi unde sunete posizionați în fișier. Puteți face click oriunde pe forma de undă din această fereastră pentru a vă reposiționa rapid într-o altă zonă a fișierului.

Fereastra de jos, mai mare (5), afișează forma de undă a fișierului audio în funcție de factorul de mărire setat. Se poate mări astfel încât să se vadă doar un mic fragment din fișierul audio – această facilitate este utilă atunci când avem nevoie să facem o prelucrare care necesită o foarte mare precizie. Dacă ați ales să vedeați mărit, în fereastra de jos nu se va mai vedea întregul fișier de sunet, dar îl puteți derula și vedea de la un capăt la altul, folosind cursorul barei de derulare orizontală. Făcând click și trăgând într-o parte sau alta a formei de undă, veți crea o selecție (veți selecta o zonă din fișier).

Alte elemente importante ale interfeței principale WavePad:

- 1. Instrument de panoramare – mărire/micșorare (Zoom)** se aplică la ceea ce este afișat în fereastra de jos a formei de undă (*imaginea b*). De aici puteți mări sau micșora pentru a vedea reprezentarea grafică a sunetului, mai amănuințit sau în ansamblu.

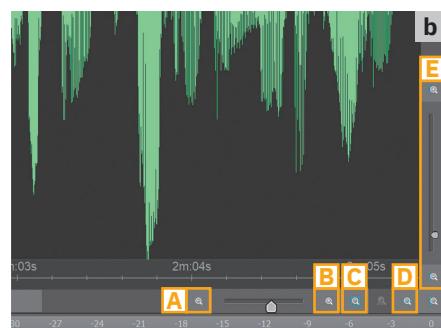
A – buton de micșorare (Zoom Out);

B – buton de mărire (Zoom In);

C – afișează întregul fișier de sunet;

D – mărește, pe întreaga fereastră de jos, doar porțiunea de sunet selectată;

E – **Instrument de panoramare – mărire/micșorare pe verticală (Vertical Zoom)**, în general, rar folosit în practică.



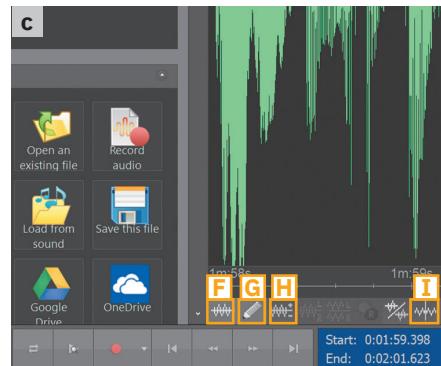
- 2. Butoane pentru diferite moduri de afișare (*imaginea c*):**

F – restaurarea afișării inițiale;

G – modul de editare eşantion – permite modificarea unui eşantion individual din fișierul de sunet folosind mouse-ul;

H – afișarea formei de undă + sus/- jos (această variantă de afișare combină și afișează canalul stâng și canalul drept – cele două canale corespunzătoare unei înregistrări stereo – într-o singură formă de undă);

I – afișează marcajele de editare (linii verticale pe părțile editate).



### 3. Display-ul de selecție (J) (imaginea d)

- **Start** – afișează poziția de început a zonei selectate (ora, minutul, secunda, miimea de secundă);
- **End (Sfârșit)** – afișează poziția de sfârșit a zonei selectate (ora, minutul, secunda, miimea de secundă);
- **Sel Length (Lungimea selecției)** – afișează lungimea (durata) zonei selectate;
- **File Length (Lungimea fișierului)** – afișează lungimea (durata) întregului fișier de sunet.

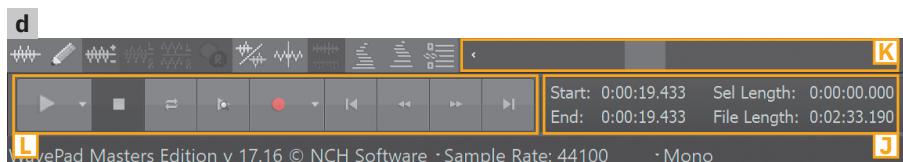
Pentru a selecta o porțiune din fișier, puteți fie să utilizați mouse-ul, să faceți click și să trageți până unde dorîți să selectați, fie să țineți apăsată tasta **Shift** și să apăsați apoi tastele cu săgeți – stânga sau dreapta.

### 4. Bara de derulare orizontală (K)

(imaginea d)

### 5. Zona butoanelor de bază (L). De la stânga la dreapta, acestea sunt (imaginea d):

- **Play (Redare)** – porniți redarea sunetului din fișier;
- **Stop** – opriți redarea, respectiv înregistrarea;
- **Loop (Redare în buclă)** – porniți sau opriți redarea în buclă;
- **Scrub (Frecare)** – pentru a găsi cu acuratețe poziția de editare (folosind auzul), apăsați **F6** sau apăsați acest buton, pentru a activa funcția de „frecare”. Aceasta permite o mișcare foarte fină a cursorului (cu tastele săgeți) în timp ce ascultați, până identificați exact punctul în care dorîți să începeți sau să terminați editarea;



- **Record (Înregistrare)** – porniți înregistrarea;
- **Go to Start (Home) (Salt la început, acasă)** – poziționați cursorul direct la începutul înregistrării/fișierului de sunet;
- **Rewind (Repede înapoi)** – derulați înapoi cu viteză mărită (trebuie ținut apăsat și sunetul se derulează înapoi cu viteză mărită);
- **Forward (Repede înainte)** – derulați înainte cu viteză mărită;
- **Go to End (End) (Salt la sfârșit)** – poziționați cursorul direct la sfârșitul înregistrării/fișierului de sunet.

## Operații de gestionare a fișierelor de sunet

Principalele operații pe care le putem face în lucrul cu fișierele de sunet în aplicația audio WavePad sunt (imaginea e):

### 1. Crearea unui nou fișier de sunet – înregistrarea de sunete. Pentru a crea un nou fișier, apăsați **Ctrl+N** sau alegeți opțiunea **New File...** (Fișier nou...) din meniu **File** (Fișier).

<b>e</b>	New File...	Ctrl+N
	New File in New Window...	Ctrl+Shift+N
	Open File...	Ctrl+O
	Open File in New Window...	
	Save File	Ctrl+S
	Save File As...	Ctrl+Shift+S
	Save All Files	
	Preview Waveform from Previous Save	
	Close File	Ctrl+W
	Close All	
	Open Project...	
	Close Project	
	Save Project	
	Save Project As...	
	Download From Sound Library...	
	Download File from Cloud Service...	
	Upload File to Cloud Service...	
	Upload File to YouTube	
	Send File...	
	Load Audio CD Track(s)...	
	Copy All Open File(s) to CD...	>
	Register Upgrade to Master's Edition...	
	Launch	>
	Exit	Alt+F4

### Retineti

Prima și cea mai importantă operație pe care o putem face cu WavePad este **înregistrarea sunetelor**. Este important să avem grijă cum facem înregistrarea, deoarece o înregistrare făcută îngrijit ne asigură o bună calitate a fișierului nostru de sunet și ne scutește de multă muncă și timp pierdut cu diferite operații de curățare zgomote, corectare/reglare niveluri sonore diferite etc.

### 2. Deschiderea unui fișier audio existent. Pentru a deschide un fișier existent, apăsați **Ctrl+O**, navigați prin structura de foldere până găsiți fișierul dorit, selectați-l și apăsați **Open (Deschidere)**.

Puteți, de asemenea, să încărcați și piste audio de pe CD-uri, selectând **Load Audio CD Track(s)** (Încărcare piste audio de pe CD) din fila **File** (Fișier). Selectați pista audio dorită făcând click pe ea.

### 3. Vizualizarea (redarea) fișierelor. Apăsați tasta **Spațiu**. La prima apăsare, redarea pornește automat de la începutul fișierului. La următoarea apăsare, redarea se oprește. La o nouă apăsare, pornește de unde a rămas.

Apăsați butonul **Play (Redare)** din zona butoanelor de bază (zona L). Acolo aveți disponibile toate comenziile uzuale pentru redarea înregistrărilor audio: redare (play), pauză, salt la început, salt la sfârșit, repede înapoi, repede înainte.

### 4. Salvarea unui fișier audio. Pentru a salva fișierul audio curent (după efectuarea unor modificări) cu numele actual, apăsați **Ctrl+S** (sau butonul **Save (Salvare)**, care are ca simbol o dischetă, din fila **Home (Pornire)**).

Pentru a salva fișierul curent cu un nume diferit sau într-un alt format audio, folosiți meniu **Save File As (Salvare ca)** din fila **File (Fișier)** sau apăsați **Ctrl+Shift+S**. Cele mai folosite formate de fișiere audio sunt: Wave (\*.wav), MPEG Layer-3 (\*.mp3), Windows Media Audio (\*.wma), Ogg Vorbis (\*.ogg), Apple AIFF (\*.aif, \*.aiff, \*.aifc), iPhone Ringtone (\*.m4r), Sun AU (\*.au), FLAC (\*.flac), AAC (\*.aac), MPEG-4 Audio (\*.m4a), GSM (\*.gsm), Dialogic Vox (\*.vox), Raw Audio (\*.raw), RSS Podcast Audio (\*.rss).

## Rețineți

Când lucrați cu fișiere audio, nu salvați într-un format comprimat decât după ce terminați toate operațiunile de editare, pentru că de fiecare dată când redeschideți sau salvați un fișier într-un format comprimat, pierdeți din calitate. Salvați în format WAVE pentru a păstra calitatea.

5. **Prelucrarea fișierelor audio** se referă la: decupare, lipire, inserare, eliminare de zgomote, aplicări de diferite efecte, mărire/micșorare de volum sau doar egalizare de volum – aducerea întregii înregistrări la același nivel de volum sonor, astfel încât să nu se audă „în salturi”, unele porțiuni mai tare și altele mai încrețit.
6. **Închiderea unui fișier audio.** Din meniul **File (Fișier)**, alegeti opțiunea **Exit (Ieșire)**. O altă posibilitate este să închideți fereastra programului de editare video de la butonul **X (Close)** din colțul din dreapta sus, aşa cum se poate închide orice fereastră Windows.
7. **Încărcarea fișierelor audio pe Internet.** Selectând opțiunea **Upload File to Cloud Service...** puteți încărca fișierele audio într-un spațiu de stocare din cloud. Aveți și opțiunea de a încărca un fișier audio direct în contul vostru de YouTube (**Upload File to YouTube**). Desigur, aceasta necesită să aveți deja creat un cont de YouTube.
8. **Trimiterea unui fișier audio.** WavePad vă permite să trimiteți un fișier de sunet prin e-mail sau printr-o conexiune directă la Internet. Puteți selecta cum vreți să trimiteți fișierul folosind meniul **Send file... (Trimite fișier...)** din fila **File (Fișier)**.

## Rețineți

Prelucrarea sunetului este foarte importantă pentru multe activități și ramuri ale industriei, dar cu siguranță se folosește cel mai mult în showbiz pentru filme, seriale, spoturi video, precum și în mass-media, care se bazează pe sunetul prelucrat calitativ, precum stații de radio, concerte etc. Etapele prelucrării de bază a sunetului sunt:

- Înregistrarea sunetului (fie înregistrare de voce, de muzică, fie de diverse alte sunete, efecte speciale, zgomote etc.);
- introducerea materialului audio în calculator (se poate înregistra direct pe calculator, folosind chiar WavePad, dar se poate înregistra și cu un alt aparat, un telefon, un reportofon etc.);
- prelucrarea de bază a materialului audio (eliminarea zgomotelor nedoreite, îmbunătățirea caracteristicilor tehnice ale fișierului de sunet);
- montajul (decuparea fragmentelor înregistrate în plus, lipirea laolaltă a tuturor sunetelor necesare proiectului etc.);
- aplicarea efectelor audio (dacă este cazul);
- producția părții audio finale și salvarea în fișier.

## Aplicați și investigați

- Realizați următoarele acțiuni în ordinea dată:
1. Creați un fișier de sunet nou, salvați-l în folderul **Lucru\_ClsVII** cu numele **SunetulMeu.wav** și închideți-l.
  2. Deschideți fișierul creat anterior și salvați-l sub un alt nume: **SunetulMeu1.wav**.
  3. Descoperiți elementele de interfață ale aplicației WavePad și vedeți care sunt acțiunile care nu se pot face asupra unui fișier de sunet gol.

## Exersați

1. Scrieți, în caiet, doar extensiile de nume de fișiere care definesc formate de fișiere de sunet:
 

a. .mp3;	b. .mp4;	c. .au;	d. .xls;
e. .wma;	f. .m4r;	g. .avi;	h. .aif;
i. .ogg;	j. .mkv;	k. .vox;	l. .mov.
2. Care sunt principalele cinci elemente ale interfeței WavePad?
3. Când este recomandat să salvați fișierul audio într-un format comprimat? De ce?
4. **Proiect:** Căutați pe Internet informații despre două formate de fișiere de sunet: **.mp3 și .wav**. Găsiți două avantaje și două dezavantaje ale fiecărui format. Care sunt situațiile în care este de preferat să se folosească fiecare dintre ele?



# 3 Prelucrarea sunetelor

## Descoperiți

Să presupunem că avem nevoie să realizăm o înregistrare audio și dorim să facem câteva prelucrări asupra acestei înregistrări, astfel încât, la final, să obținem un fișier audio de care să fim mândri. Ce avem de făcut?

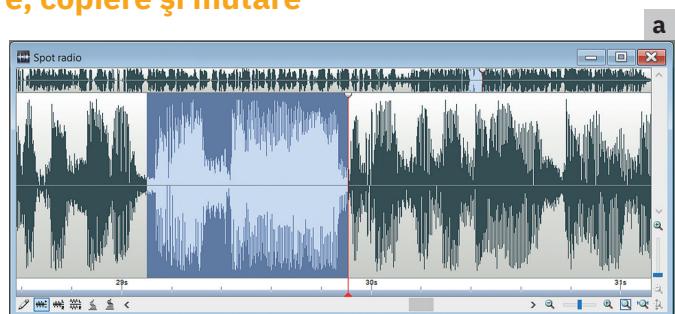
### 1. Realizarea înregistrării brute

Mai întâi, porniți aplicația WavePad. Apoi, presupunând că aveți conectat un microfon la calculator (dacă e laptop, are microfon încorporat), apăsați butonul **Record** (*Înregistrare*) sau **F5** din colțul din stânga jos al interfeței (butonul roșu). WavePad va crea automat un fișier nou și va porni să înregistreze sunet în acest fișier. O altă posibilitate este să creați mai întâi voi un nou fișier (**New File... (Ctrl+N)** din meniul **File**), apoi să începeți înregistrarea. În timpul înregistrării vorbiți clar, cursiv și aveți grijă să fie liniște deplină în jur. Mai mult, aveți grijă să alegeti o cameră în care vocea să nu se audă cu ecou. Când ati terminat de înregistrat, apăsați butonul **Stop**.

### 2. Selecția unor secvențe audio pentru ștergere, copiere și mutare

 Acum, că aveți înregistrarea făcută, să presupunem că aveți o porțiune în care v-ați bâlbâbit și ati reluat de la început acea frază. Evident, porțiunea respectivă va trebui eliminată (ștersă).

Pentru aceasta, ascultați și identificați cu precizie (ajutându-vă de reprezentarea grafică din fereastra de jos a interfeței WavePad) de unde începe și unde se termină porțiunea de șters. Dacă este nevoie, măriți zona respectivă (folosind butonul de mărire – **B**) pentru a putea poziționa cursorul cu mai mare precizie. Selectați întregul fragment (faceți click la începutul fragmentului, țineți apăsat butonul mouse-ului și trageți până la sfârșitul zonei care trebuie eliminată). În *imaginea a* se observă fragmentul selectat (cu albastru).



Odată ce ati selectat fragmentul de eliminat, nu vă mai rămâne decât să îl ștergeți apăsând tasta **Delete** (sau click cu butonul drept al mouse-ului pe fragmentul selectat și, din meniul contextual care se deschide, alegeti opțiunea **Delete – Ștergere**).

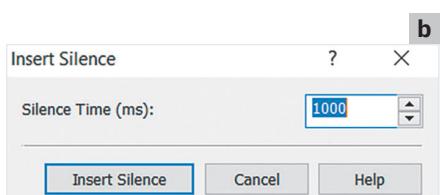
Să presupunem că vreți să schimbați două fraze între ele. Procedați ca mai sus pentru a selecta prima frază; din meniul **Edit** alegeti **Cut (Decupare)** sau folosiți combinația de taste **Ctrl+X**, apoi poziționați cursorul cu un click de mouse la sfârșitul celei de-a doua fraze, acolo unde doriți să înceapă fraza pe care ati tăiat-o, și tot din meniul **Edit (Editare)** alegeti **Paste (Lipire)**. În felul acesta ati mutat o frază (o porțiune din fișierul de sunet) dintr-un loc în altul.

Dacă în înregistrarea voastră aveți 2 s de aplauze ale unui coleg, dar vă doriți să aveți 6 s de aplauze, ce puteți face?

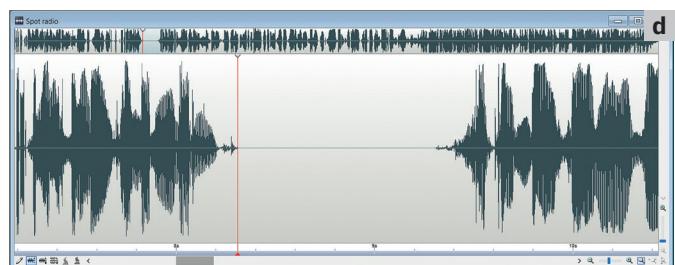
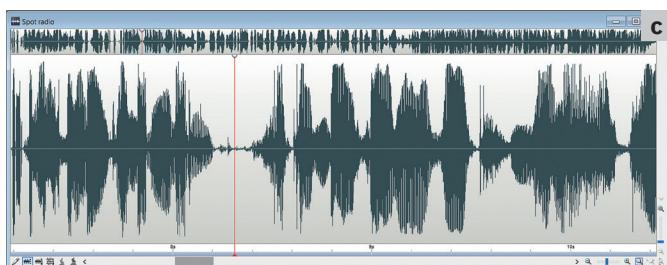
Soluția este următoarea: selectați porțiunea cu cele 2 s de aplauze și alegeti opțiunea **Copy (Copiere)** (**Ctrl+C**) (tot din meniul **Edit**). Apoi poziționați cursorul acolo unde se termină cele 2 s de aplauze și dați comanda **Paste (Lipire)** sau **Ctrl+V**. În acest moment aveți 4 s de aplauze. Dacă veți mai da încă o dată aceleași comenzi, veți obține cele 6 s dorite de aplauze. Simplu, nu-i aşa?

### 3. Inserarea unei secunde de liniște

Ne poziționăm cursorul în fișierul de sunet acolo unde vrem să inserăm secunda de liniște (aşa cum se vede în *imaginea c*), apoi din fila **Edit (Editare)**, apăsăm butonul **Silence (Linisțe)** și alegeti **Insert silence at cursor... (Inserare liniște la locul cursorului...)**. În fereastra de dialog care se deschide (*imaginea b*) introducem (în milisecunde) durata spațiului de liniște de inserat (implicit este 1000 ms, adică 1 s).



Vom obține, de la poziția curentă a cursorului înainte, 1 s de liniște (după cum vedeti, avem un spațiu în care nu este prezentă nicio formă de undă, cât de mică). Desigur, durata totală a fișierului nostru de sunet a crescut cu 1 s (*imaginea d*).



## 4. Reglarea nivelurilor sunetelor

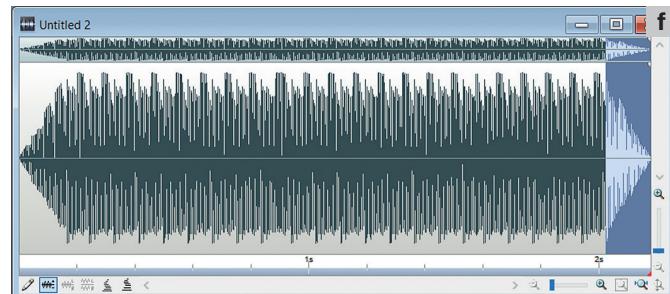
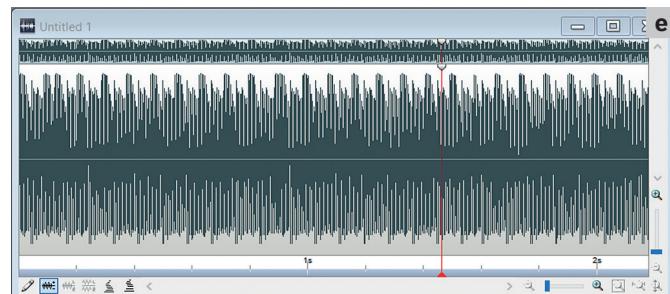
Pentru a avea o înregistrare audio de bună calitate, trebuie ca sunetul să fie suficient de puternic: nici prea slab, dar nici prea puternic încât să ajungă în zona în care este distorsionat. De asemenea, trebuie să fie uniform pe parcursul întregului fișier (să nu aibă zone în care se aude mai încet și zone în care se aude mai tare).

Aceste acțiuni (mici corecții) se pot realiza cu următoarele comenzi care se găsesc în fila **Levels (Niveluri)**:

- **Amplify (Amplificare)**. A amplifica înseamnă a crește volumul sonor al zonei selectate. Pentru a face o parte din înregistrare mai slabă sau mai tare, selectați-o și apoi utilizați meniu **Levels (Niveluri) -> Amplificare**. Volumul este introdus în procente (dacă valoarea este 100%, atunci nu se schimbă nimic, volumul sonor rămâne așa cum era; valoarea de 50% înseamnă -6 dB, deci puțin mai încet; valoarea de 200% se traduce în +6 dB mai tare).
- **Normalize (Normalizare, uniformizare)**. A normaliza înseamnă a ajusta volumul astfel încât cel mai puternic vârf să fie egal cu sau să fie un procent din semnalul maxim care poate fi utilizat în fișierul audio digital. Metoda de normalizare „vârf” (Peak) găsește mostra de cel mai mare nivel din fișier. Normalizarea se face apoi cu această valoare ca vârf maxim. Este bine să normalizați fișierele până la 100% ca ultimă etapă a procesării audio, pentru a face fișierul rezultat cât mai tare posibil, fără distorsiuni. Un alt motiv pentru a aplica normalizarea este să aveți întregul fișier de sunet la fel de tare, altfel spus, să aveți o intensitate medie egală.
- **Fade (Estompare de început/Estompare de sfârșit)**:
  - **Fade in (Estompare de început)** – se aplică pentru a obține o intrare lină a sunetului, crescător, de la zero la volumul maxim al sunetului.
  - **Fade out (Estompare de sfârșit)** – se aplică pentru a obține o terminare lină a sunetului, descrescător, de la volumul maxim al sunetului până la zero.

În imaginile alăturate puteți vedea reprezentarea grafică a unui fișier audio. În imaginea e este sunetul brut. Se observă că începe brusc, la un nivel ridicat și se menține așa până la final, când se termină tot brusc. În imaginea f s-a aplicat un efect de estompare la început, astfel încât să înceapă crescător, lin, iar la final i s-a mai aplicat un efect de estompare, pentru a se termina descrescendo, lin. Estomparea s-a aplicat pe porțiunea selectată de la final. Dacă doriți o estompare mai lină, care să se întindă pe o durată mai mare de timp, selectați o zonă mai mare și apoi aplicați estomparea (**Fade**).

- **Fade Out and Trim (Estompare și Trunchiere)**. Opțiunea de estompare și trunchiere este o funcție combinată care estompează sfârșitul zonei selectate, apoi marchează sfârșitul selecției ca sfârșit al fișierului. Această prelucrare este aplicată frecvent la finalul pieselor muzicale.



## 5. Efecte speciale aplicate sunetelor

▶ Puteți aplica înregistrării voastre, sau numai unei părți a ei (selectate), diferite efecte speciale disponibile în WavePad. Dintre acestea, enumerăm câteva mai spectaculoase:

- **Echo (Ecou)** – aplică un efect de ecou peste sunetul selectat;
- **Reverb (Reverberație)** – aplică un efect de reverberație asupra sunetului selectat;
- **Telephone (Telefon)** – aplică secvenței selectate un efect audio ca și cum s-ar auzi la telefon;
- **Speed Change (Schimbare viteză)** – modifică viteza cu care este redat sunetul. Poate fi redat mai rapid (pe repede înainte) sau mai rar (cu încetinitorul);
- **Voice Change (Schimbare voce)** – în funcție de parametrii pe care îi modificați după dorință, obțineți intonații diferite de voce. La ambele extremități, rezultatele obținute sunt de-a dreptul hazlii.

Efectele speciale se găsesc în meniu **Effects (Efecte)**, grupurile **Effect Chain, Speed (Viteză)** și **Voice (Voce)**.

## 6. Mixarea semnalului audio din mai multe surse

Ne dorim ca pe înregistrarea de voce să suprapunem un fond muzical (ușor, în fundal), adică să mixăm o înregistrare de voce cu o melodie. Pentru aceasta vom folosi funcția **Paste Mix (Lipire și amestecare)** din fila **Edit**, care ia porțiunea de sunet pe care anterior ați selectat-o și ați copiat-o (în clipboard) și o mixează cu porțiunea curent selectată. Cum faceți? După ce ați selectat și copiat porțiunea de sunet pe care doriți să o mixați, selectați noua zonă, cea cu care doriți să faceți mixarea, și folosiți **Paste Mix** din meniu (sau combinația de taste **Ctrl+Alt+V**). Puteți specifica și ce volum să aibă sunetul mixat.

Aproape întotdeauna lungimile sunetelor selectate vor fi diferite. Dacă sunetul din clipboard este mai scurt decât selecția curentă de sunet, atunci sunetul mixat se va sfârși când și selecția curentă de sunet se sfârșește. Dacă sunetul din clipboard este mai lung decât selecția curentă, atunci poate fi aleasă una dintre opțiunile:

- **Increase selection** (*Mărește zona selectată*) – însemnând că mixajul se poate extinde până când se sfârșește și sunetul din clipboard;
- **Insert silence** (*Inserează liniste*) – însemnând că după ce se termină zona curent selectată, mixajul se continuă cu liniste până se sfârșește și sunetul din clipboard. Practic, se prelungește cu liniste zona selectată și se mixează cu ceea ce era în clipboard, adică se va auzi în continuare doar ceea ce era în clipboard, dar la nivelul de volum sonor selectat pentru mixare;
- **Mix With File** (*Mixează cu un fișier*) – funcționează la fel ca **Paste Mix** (*Lipește și amestecă*), cu deosebirea că trebuie să specificați un fișier pe care să îl mixeze, în loc să copiați mai întâi în clipboard portiunea de sunet de mixat.

## Rețineti

- **Undo (Ctrl+Z)** – este operația de restaurare a fișierului la stadiul de dinaintea ultimei modificări (a ultimei acțiuni de editare). Aceasta este utilă dacă vreți doar să încercați o editare sau dacă ati făcut o greșală. Pentru a anula ultima acțiune făcută, doar apăsați **Ctrl+Z**. Notă: cu această facilitate puteți să anulați maximum 32 de acțiuni (ultimele 32 de acțiuni realizate pe fișierul de sunet).
- **Redo (Ctrl+Y)** – dacă, după anularea unei operații și revenirea la starea anterioară (**Undo**), decideți că modificarea făcută chiar era necesară, atunci puteți să o refaceți folosind această operație (**Redo**).

## 7. Finalizarea proiectului – salvarea proiectului și salvarea fișierului audio

Dacă ati terminat toate prelucrările de făcut, toate efectele de aplicat și proiectul vostru audio este gata, este timpul să îl salvați pe calculator. Puteți, pe de o parte, să salvați proiectul cu toate componentele și setările lui, astfel încât să aveți ulterior posibilitatea de a mai lucra pe acest proiect, pe de altă parte, puteți să salvați direct fișierul de sunet prelucrat.

Pentru salvarea proiectului, selectați meniul **File -> Save Project As...** (*Salvează proiectul ca...*). Nu mai trebuie decât să îi dați un nume sugestiv și **Save** (*Salvează*).

Pentru salvarea fișierului, alegeti meniul **File -> Save Audio File As...** (*Salvează fișierul audio ca...*). În afară de a-i da un nume sugestiv, mai puteți alege în ce tip de fișier vreți să salvați. Sunt disponibile mai multe tipuri de formate audio în care puteți salva (unele comprimate, altele necomprimate), printre care: MPEG Layer-3 (.mp3), Wave (.wav), Windows Media Audio (.wma), Ogg Vobis (.ogg), Apple AIFF (.aif), Dialogic Vox (.vox) etc. Nu uitați, *formatul Wave este necomprimat și asta vă oferă posibilitatea de a salva fișierul audio fără pierdere de calitate*. Desigur, asta înseamnă și că va ocupa mai mult spațiu în memoria unde se va salva.

## Aplicați și investigați

- Faceți o înregistrare de cel puțin 30 de secunde și aplicați pe această înregistrare, pe rând, toate efectele speciale disponibile în cele trei grupuri (**Effect Chain, Speed, Voice**) din fila **Effects** (*Efecte*). Scrieți într-un fișier .doc efectele care v-au plăcut cel mai mult, câteva observații despre ele și idei de proiecte în care ati putea utiliza aceste efecte audio.



## Exersați

1. Dați trei exemple de efecte speciale care se pot aplica unui sunet, în aplicația WavePad.
2. Descrieți cele două modalități de selectare a unui fragment de sunet (cu ajutorul mouse-ului și cu ajutorul tastaturii).
3. Selectați, în caiete, **A** (Adevărat) sau **F** (Fals), în funcție de valoarea de adevăr a propozițiilor:
 

<b>a.</b> Formatul Wave este un format comprimat de fișier. <b>b.</b> Avantajul unui format de fișier comprimat este că ocupă mai puțin spațiu de stocare în memorie. <b>c.</b> Formatul comprimat de fișier audio oferă o calitate mai bună decât formatul necomprimat de fișier.	<b>A</b> <b>F</b> <b>A</b> <b>F</b> <b>A</b> <b>F</b>
--	---

## Portofoliu

Folosind WavePad, realizați un fișier audio cu răspunsurile părintilor sau bunicilor la întrebarea „Care este pasiunea ta de weekend?”. Inserați o muzică de fond al cărei volum să înceapă crescător și să se termine descrescător. Salvați fișierul în folderul **Lucru\_clsVII**.

# 4 Programe de editare video. Elemente de interfață. VideoPad

## Descoperiți

 Pentru redarea și procesarea fișierelor video puteți utiliza o multitudine de programe dedicate, unele gratuite, altele contra cost. Câteva dintre cele mai populare sunt: VirtualDub, VLC media player, Sony Vegas, Pinnacle Studio, BS Player, MovieMaker, TeacherTube, YouTube, VideoPad Video Editor etc.

Să vedem cum arată interfețele a trei dintre ele:

### YouTube Studio

 Principalele elemente de interfață sunt (*imagină a*):

1. zona de meniu cu diverse opțiuni și setări;
2. zona de previzualizare;
3. instrumentul de mărire/micșorare;
4. zona pistelor video și audio.

Editorul video din YouTube Studio nu este ca o suită completă de unelte de editare, ca un program profesional de editare video, dar are un set de bază foarte util de funcționalități pentru creația de proiecte video interesante.

### VideoPad Video Editor

Principalele elemente de interfață din acest program sunt (*imagină b*):

1. butonul de meniu;
2. și 3. – barele de acces rapid și de unelte;
4. zona cutiilor de elemente;
5. zona de previzualizare;
6. zona secvențelor și pistelor.

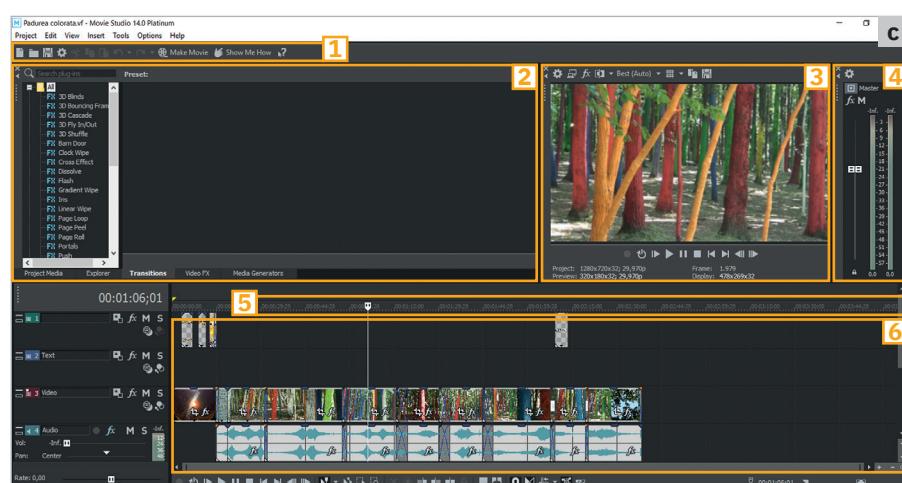
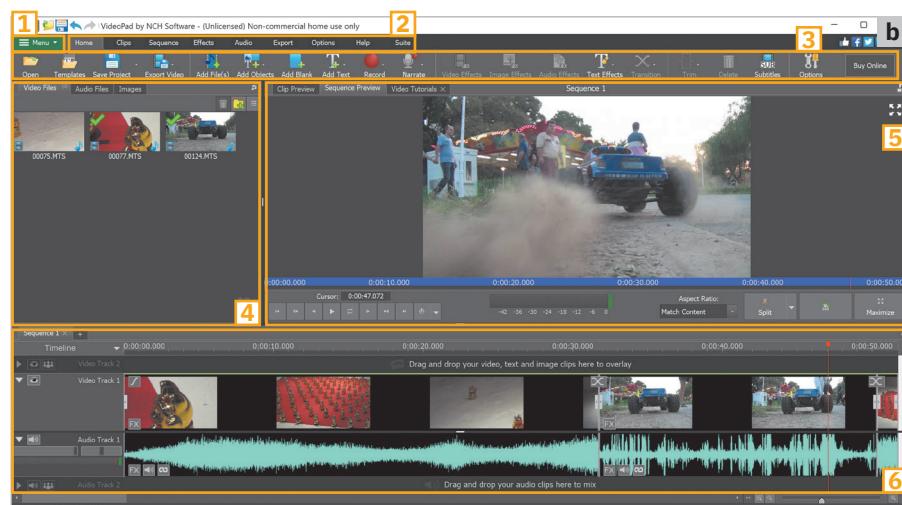
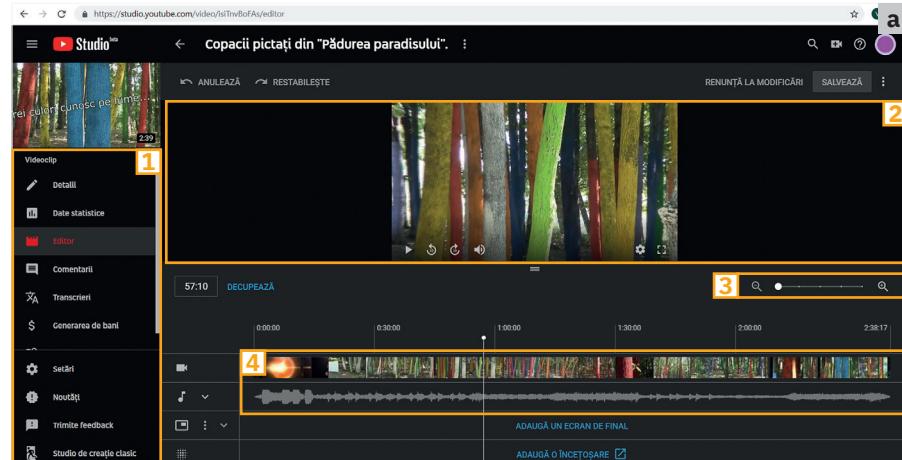
VideoPad este un program care oferă un set bogat de unelte de editare pentru fișierele video, intuitiv în utilizare odată ce v-ați obișnuit cu concepțile editării video, destul de apropiat de programele profesionale de editare video.

### Vegas Movie Studio Platinum

Principalele elemente de interfață ale programului Movie Studio sunt (*imagină c*):

1. bara principală de unelte;
2. zona elementelor media, a efectelor, a tranzițiilor etc.;
3. fereastra de previzualizare;
4. zona de vizualizare grafică a volumului audio;
5. linia de timp;
6. zona pistelor video, audio, a textelor, a efectelor speciale etc.

Movie Studio este probabil cel mai puternic program de editare și procesare video pentru amatori. Este „fratele mai mic” al lui Vegas Pro, care deja poate fi considerat o unealtă de editare din gama profesională. Îl puteți instala și încerca, programul beneficiind de o versiune de test, dar pentru a vă putea bucura fără îngădări de tot ceea ce poate face acest program, trebuie să achiziționați o licență Vegas Movie Studio. Aceasta se poate face online, direct de pe site-ul producătorului.



## Programe de editare video. Elemente de interfață. VideoPad

### Retineți

Un clip video este interesant în primul rând prin calitatea conținutului său și prin ideile prezentate în el. Nu capabilitățile tehnice ale programului de procesare video fac un clip interesant.

Vom discuta mai pe larg despre **VideoPad Video Editor**. Este ușor de folosit și vă dă posibilitatea de a edita clipurile video, de a personaliza pista audio și oferă multe alte prelucrări, precum decuparea unei secvențe video sau lipirea de mai multe fișiere video. Dacă vreți, puteți să adăugați efecte video și audio, precum și să inserați tranzitii în clipurile video pe care le editați. Are chiar și o opțiune de a include subtitrări. În plus, în afară de faptul că este foarte popular în prezent, are și meritul că este gratuit pentru utilizarea necomercială, acasă (pentru uz personal) sau la școală (pentru uz didactic).

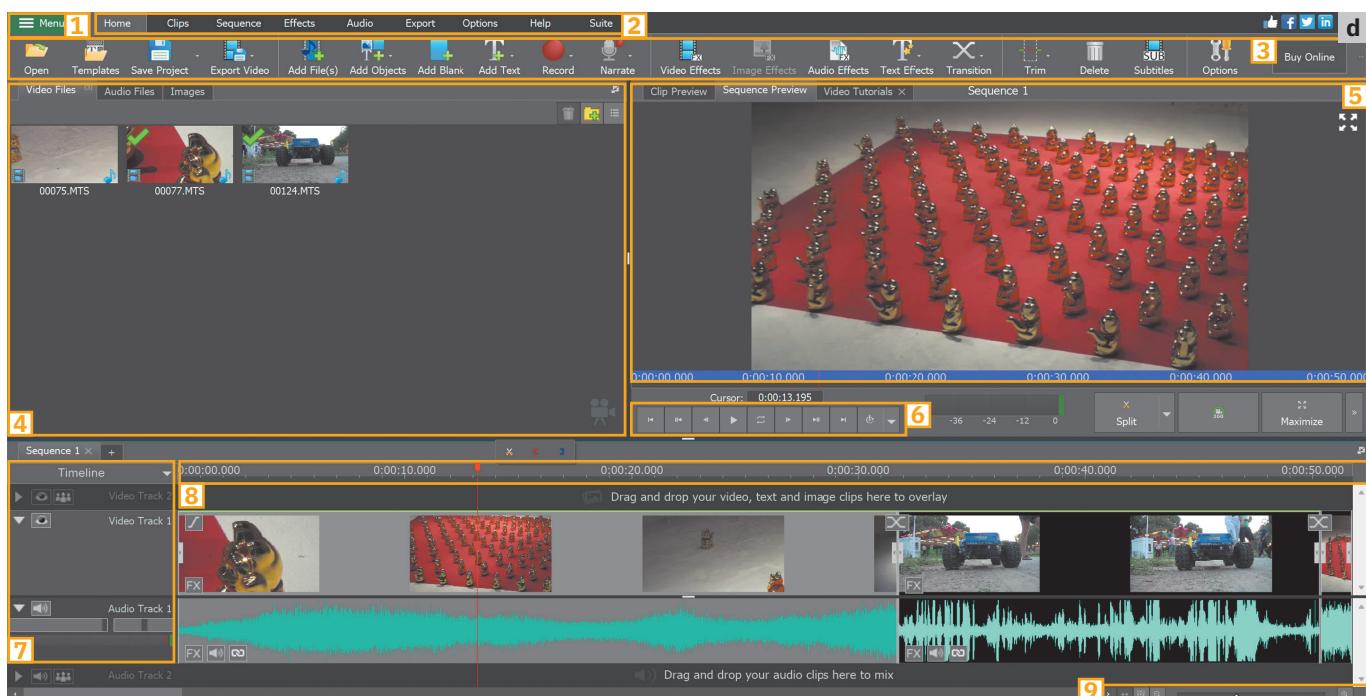
Interfața **VideoPad Video Editor** este fereastra principală în care veți edita proiectul vostru. Ecranul este împărțit în câteva zone importante.

Să vedem care sunt aceste zone, cum arată interfața mai în amănunt și care sunt elementele principale ale programului VideoPad Video Editor, numit, pe scurt, **VideoPad** (*imaginea d*):

- 1. Buton de meniu** – se regăsesc grupate, într-un meniu vertical, opțiunile clasice de meniu ale unui program de calculator, cum sunt **File, Edit, View, Help**, precum și opțiuni de meniu specifice unui program de editare video, **Control, Clip, Track, Sequence, Video/Image Effects, Transitions, Export Tools**;
- 2. Bară de unelte** – aici regăsiți grupate în mai multe file, pe categorii, majoritatea opțiunilor disponibile în VideoPad;
- 3. Bară de acces rapid** – vă permite accesul ușor și rapid la cele mai uzuale acțiuni din VideoPad. Între acestea,

regăsim **Open (Deschide), Save Project (Salvează proiectul), Export Video (Exportă video), Add Text (Adaugă text)**;

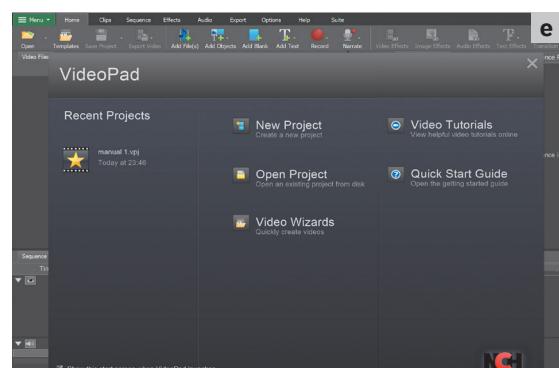
- 4. Zona cutiilor de elemente** – aici, fiecare fișă este vizibilă ca o cutie în care sunt păstrate clipuri și/sau foldere. Conținutul cutiilor poate fi reprezentat prin miniaturi sau ca listă de nume ale fișierelor corespunzătoare;
- 5. Zona de previzualizare** – după cum îi spune și numele, aici puteți vedea cum să arate rezultatul final al proiectului vostru video;



- 6. Butoanele pentru controlul vizualizării;**
- 7. Zona secvențelor și pistelor;**
- 8. Rigla de timp;**
- 9. Instrumentul de panoramare – mărire/micșorare (zoom).**

În general, principalele operații pentru gestionarea unei aplicații audio-video sunt: creare, salvare, deschidere, vizualizare, închidere.

Încă de la pornirea programului VideoPad sunteți întrebați dacă dorîți să creați un nou proiect video (opțiunea **New Project**) sau să deschideți un proiect existent, de pe discul calculatorului (un proiect salvat local, în memoria permanentă a calculatorului, de orice fel ar fi ea, **HDD (Hard Disk Drive)** sau **SSD (Solid State Drive)**) – ca în *imaginea e*.



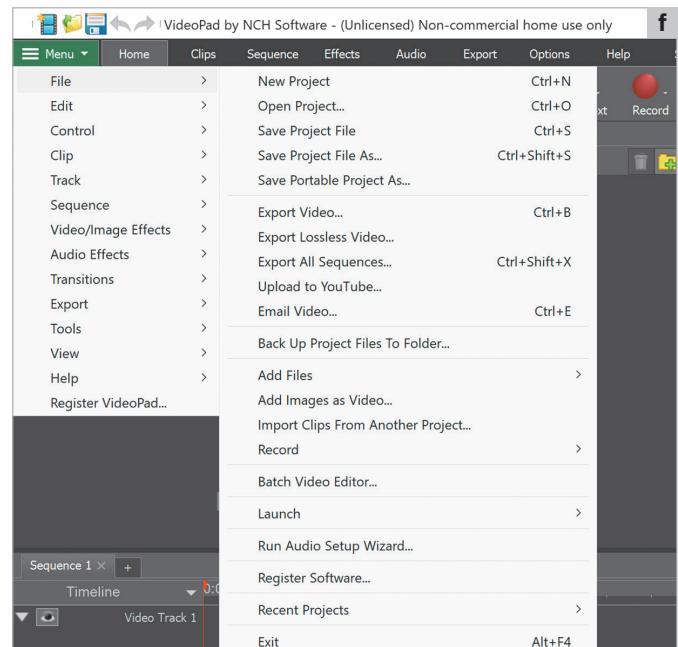
Altă modalitate de a crea un nou proiect video sau de a deschide unul existent este accesarea meniului **File** (*Fisier*), de unde puteți alege (imaginea f):

- **New Project** (*Project nou*) – același rezultat puteți obține și prin utilizarea combinației de taste (numită și „comandă rapidă”) **Ctrl+N**;
- **Open Project...** (*Deschide proiectul...*) – sau comanda rapidă **Ctrl+O**. Apoi răsfoiți folderele voastre de lucru până ajungeți la fișierul corespunzător proiectului dorit, faceți click pe el, apoi **Open** și proiectul va fi afișat în interfața VideoPad.

De menționat că ultimele proiecte deschise pot fi accesate mai rapid, fără să mai răsfoiți folderele de lucru, din meniul **Recent Projects** (*Proiecte recente*).

Odată creat un nou proiect, puteți începe să adăugați elementele pe care le veți folosi (înregistrări video, una sau mai multe, înregistrări audio suplimentare, fișiere cu muzică de fundal, dacă este cazul, fotografii, desene etc.). Le puteți trage direct cu ajutorul mouse-ului în zona **4**, zona cutiilor de elemente. În acest fel veți avea la îndemâna, asociate proiectului vostru, toate resursele necesare, toate elementele la care v-ați gândit, chiar dacă nu le veți utiliza efectiv pe toate în realizarea proiectului final.

Dacă aveți un proiect video simplu, în care nu vreți decât să utilizați o înregistrare video pe care să aplicați mici prelucrări și câteva efecte, este suficient să o trageți direct pe zona **7**, zona secvențelor și a pistelor audio și video.



## Aplicați și investigați

- ▶ Faceți acești pași cu o secvență video: creați un nou proiect, apoi trageți o secvență video pe zona **7**, pe pista video. Proiectul vostru nou creat trebuie acum salvat în calculator (și pentru a-l păstra, dar și pentru a continua ulterior lucrul la el).

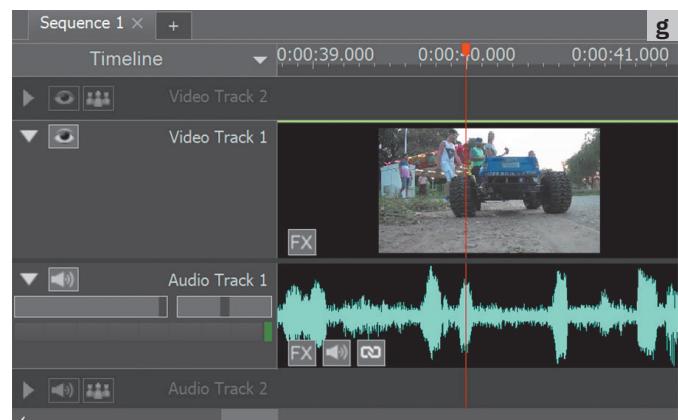
## Descoperiți

**Salvarea proiectului** se face tot din meniul **File** (*Fisier*), de unde alegeți opțiunea **Save as...** (*Salvează ca...*), sau cu ajutorul combinației de taste **Ctrl+S**. În continuare, trebuie să alegeți folderul în care doriți să salvați proiectul și să-i dați un nume. Nu uitați, numele fișierului cu proiectul vostru trebuie să fie sugestiv pentru a-l putea identifica ușor, mai ales dacă veți avea multe proiecte de acest fel.

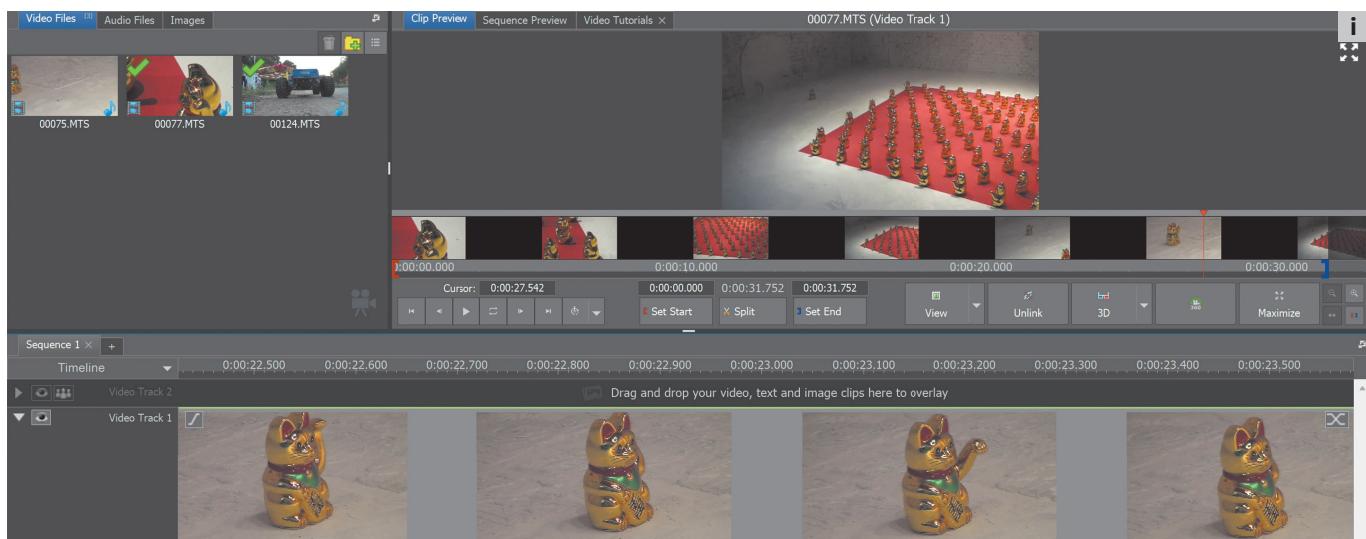
Țineți minte că în majoritatea programelor, operațiile de creare, salvare, deschidere și închidere de fișiere se găsesc în meniul **File** (*Fisier*).

Cum vizualizați proiectul vostru, odată ce l-ați deschis (dacă era făcut și salvat undeva în calculator) sau dacă l-ați creat și ați introdus cel puțin o înregistrare video în el?

- Apăsați tasta **Spațiu**. La prima apăsare, redarea pornește automat de la începutul proiectului. La următoarea apăsare, redarea se oprește. La o nouă apăsare, pornește de unde a rămas.
- Apăsați butonul **Play** din zona de previzualizare (zona **5**). Aici aveți disponibile toate comenziile uzuale pentru redarea înregistrărilor audio și video: redare (play), pauză, salt la început, salt la sfârșit, repede înapoi, repede înainte etc.
- Pentru a vă poziționa într-un anume punct al înregistrării (de exemplu, la secunda 40 a înregistrării sau în orice alt punct ales vizual), puteți trage cursorul pe **timeline** (*linia de timp*) până în punctul dorit sau puteți face click direct în acel punct (imaginea **g**).
- Pentru o poziționare mai precisă, puteți mări zona de secvențe și piste de la instrumentul de mărire/micșorare din colțul din dreapta jos al interfeței VideoPad (imaginea **h**).



## Programe de editare video. Elemente de interfață. VideoPad



O altă modalitate de afișare a ecranului principal este în mod **Sequence** (Secvențe) pentru lucrul cu secvențele care se montează și cadrele care le compun. Observați în *imaginea i* cât de mult sunt mărite cadrele de pe bara de timp, tocmai pentru a asigura o mai bună vizibilitate și precizie la operațiunile de prelucrare.

Există și un mod de vizualizare **Custom** (Personalizat), în care vă puteți aranja singuri, după necesități și preferințe, atât dimensiunile unor zone de lucru, cât și o multitudine de butoane pentru acces rapid la unele comenzi, butoane care pot fi afișate în **Bara de acces rapid**.

Cum închideți proiectul? Nimic mai simplu: doar asigurați-vă că sunt salvate și ultimele modificări (cel mai sigur este să mai apăsați o dată combinația de taste **Ctrl+S**) și, din același meniu **File** (Fișier), alegeti opțiunea **Exit** (Ieșire). O altă posibilitate este să închideți fereastra programului de editare video de la butonul **X** (Închide) din colțul din dreapta sus, așa cum se poate închide orice fereastră Windows (*imaginea j*).



### Ştiati că...?



Un proiect VideoPad se închide automat și în următoarele două situații:

- dacă deschideți un alt proiect;
- dacă vreți să creați un nou proiect.



## Aplicați și investigați

1. Realizați, pe rând, următoarele acțiuni pentru proiectul creat anterior:
  - a. Apăsați succesiv tasta **Spațiu** pentru a începe/a stopa redarea proiectului.
  - b. Apăsați, pe rând, butoanele din zona de previzualizare și observați ce se întâmplă cu fiecare în parte.
  - c. Poziționați-vă într-un punct al înregistrării voastre și redați din acel punct.
2. Căutați pe Internet informații despre programele de editare video VirtualDub, VLC Media Player, Sony Vegas, Pinnacle Studio, BS Player, MovieMaker, TeacherTube, YouTube, VideoPad Video Editor.

## Exersați

1. Numiți patru elemente importante ale ferestrei principale a programului VideoPad și precizați rolul fiecărui.
2. Selectați, în caiete, **A** (Adevărat) sau **F** (Fals) pentru fiecare propoziție de mai jos:
  - a. Rigla de timp din interfața VideoPad ne ajută să vedem în ce dată și la ce oră a fost făcută înregistrarea. A F
  - b. Salvarea unui proiect video în VideoPad se face din meniu **File** (Fișier), de unde se alege opțiunea **Save as...** (Salvează ca...), sau cu ajutorul combinației de taste **Ctrl+S**. A F
  - c. În zona de previzualizare a interfeței VideoPad puteți vedea cum o să arate rezultatul final al proiectului vostru video. A F
3. Descrieți modalitatea prin care vă puteți poziționa la secunda 25 a unei înregistrări video, în programul VideoPad.



## Portofoliu

Realizați o înregistrare video (cu o cameră video, cu o cameră web conectată la calculator sau chiar cu telefonul mobil) cu un scurt interviu luat colegilor voștri pe tema „Sportul meu preferat”. Apoi creați un nou proiect VideoPad, încărcați secvențele video înregistrate, salvați în folderul **Lucru\_clsVII** și vizionați.

## 5 Prelucrarea video

### Amintiți-vă

În lecția anterioară am realizat un proiect video cu o înregistrare a unor interviuri luate colegilor pe tema „Sportul meu preferat”. Înregistrarea trebuia realizată cu o cameră video, cu o cameră web conectată la calculator sau cu telefonul mobil.

### Descoperiți

Să continuăm lucrul la proiectul video început în lecția anterioară. Vom învăța, în cele ce urmează, cum anume se pot face prelucrări pe acest material filmat.

Mai întâi, deschideți proiectul. Avem nevoie să facem următoarele operațiuni pe materialul filmat inclus în acesta:

- ștergerea unor mici secvențe pentru a păstra în proiect numai material video de bună calitate (din punct de vedere al imaginii) și reprezentativ pentru subiectul ales;
- tăierea unei secvențe video – poate doriți să inserați ceva în acel punct;
- copierea unei secvențe video;
- mutarea unei secvențe video (poate că înregistrările video au fost făcute în funcție de cum s-a ivit ocazia, dar ordinea lor nu este cea mai potrivită pentru scenariul imaginat de voi pentru filmul final);
- aplicarea unor efecte de tranziții între scenele care formează proiectul vostru.

Pentru toate operațiile enumerate mai sus, mai întâi trebuie să faceți selecția secvențelor video pe care doriți să aplicați acele prelucrări.

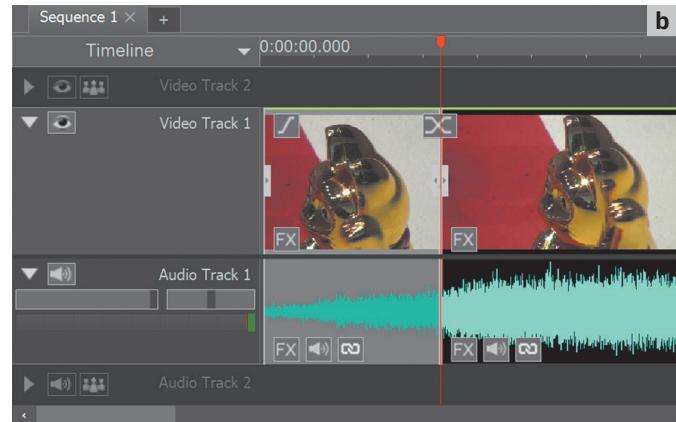
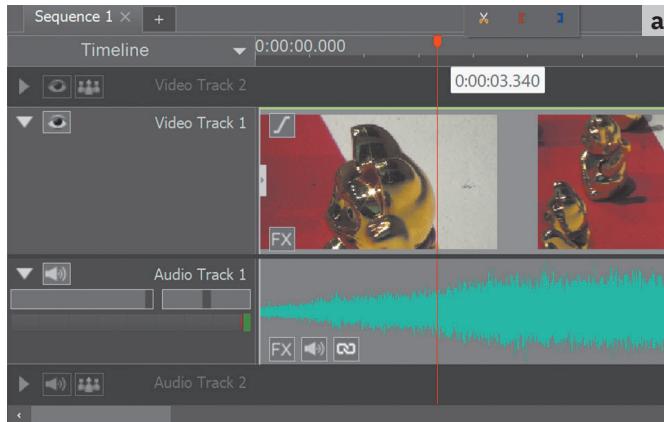
Să luăm, pe rând, fiecare operație și să vedem cum o puteți aplica pe proiectul vostru.

### Ștergerea unor mici secvențe din materialul filmat

Să presupunem că în înregistrarea voastră video există o porțiune inițială, de când s-a pornit înregistrarea și până la începutul propriu-zis al interviului, în care nu este tocmai liniște și oricum este un material care nu trebuie să apară în proiectul final. Vreți să ștergeți această porțiune.

Mai întâi, trebuie selectată porțiunea vizată, apoi poate fi ștersă.

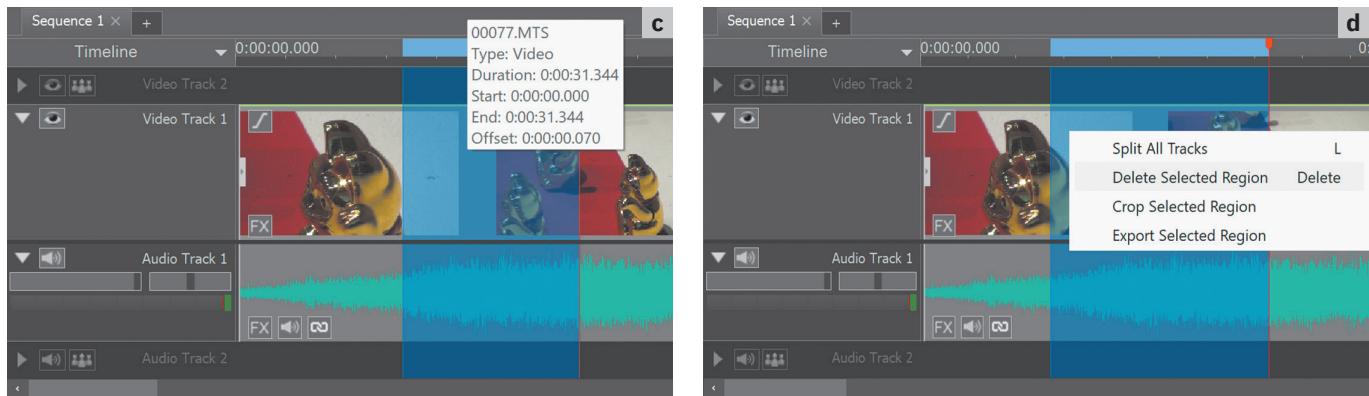
Pentru a o selecta, poziționați cursorul exact în punctul în care vreți să înceapă secvența video (*imaginea a*). Când faceți click și poziționați cursorul undeva, automat apare lângă cursor un buton cu o foarfecă. Faceți click pe acel buton și ați obținut o segmentare a filmului (porțiunea din stânga cursorului este acum ca o secvență de film separată). Faceți click pe această porțiune (cea din stânga, cea de început pe care vreți să o ștergeți – *imaginea b*) și apoi ștergeți-o fie apăsând tasta **DELETE**, fie alegând opțiunea **Delete** din meniul contextual care apare dacă faceți click dreapta pe fragmentul selectat.



O altă posibilitate de a selecta o secvență este să faceți click pe rigla de timp acolo unde vreți să înceapă zona de șters, să țineți apăsat butonul mouse-ului, apoi apăsați și mențineți apăsată tasta **SHIFT** și trageți cursorul până acolo unde doriți

## Prelucrarea video

să ștergeți. Observați că vi se spune cu precizie de unde începe și unde se termină porțiunea selectată (*imaginea c*). Odată terminată selecția, puteți apăsa tasta **DELETE** și ștergeți fragmentul selectat (*imaginea d*).



## Tăierea unei secvențe video

Această operație este utilă atunci când vreți să ștergeți o anumită porțiune din film, când vreți să inserați ceva într-un anumit punct, când vreți să mutați o porțiune de film, când vreți să selectați în mod repetat și cu ușurință o anumită porțiune sau în alte situații care se vor ivi în practică.

Cum se face? Cum am văzut mai sus, când am discutat de selectare: poziționați cursorul exact în punctul în care vreți să tăiați (*imaginea a*). Când faceți click și poziționați cursorul, automat apare lângă cursor un buton cu o forfecă. Faceți click pe acel buton și ati obținut o tăiere a secvenței video (porțiunile din stânga și din dreapta cursorului sunt acum două secvențe separate).

## Copierea unei secvențe video

Să zicem că doriți ca în montajul final să aveți aplauzele unor colegi după fiecare interviu, aplauze care, în materialul brut, au fost înregistrate doar după interviul primului coleg. Ce este de făcut?

Nimic mai simplu: faceți două tăieturi, la începutul și la sfârșitul secvenței cu aplauzele, selectați secvența astfel decupată și copiați-o în memorie (click cu butonul drept al mouse-ului și alegeti **Copy**). Apoi poziționați mouse-ul la sfârșitul fiecărui interviu și lipiți acolo (**Paste**) secvența cu aplauzele (copiată în memorie).

## Mutarea unor secvențe video

Poate că înregistrările video au fost făcute în funcție de cum s-a ivit ocazia, dar ordinea lor nu este cea mai potrivită pentru scenariul imaginat de voi pentru filmul final.

Ați făcut înregistrările interviurilor după cum i-ați găsit disponibili pe colegii voștri (dar presupunem că ați făcut toate înregistrările odată și că aveți o singură secvență video), însă montajul final vreți să îl faceți având colegii în ordine alfabetică.

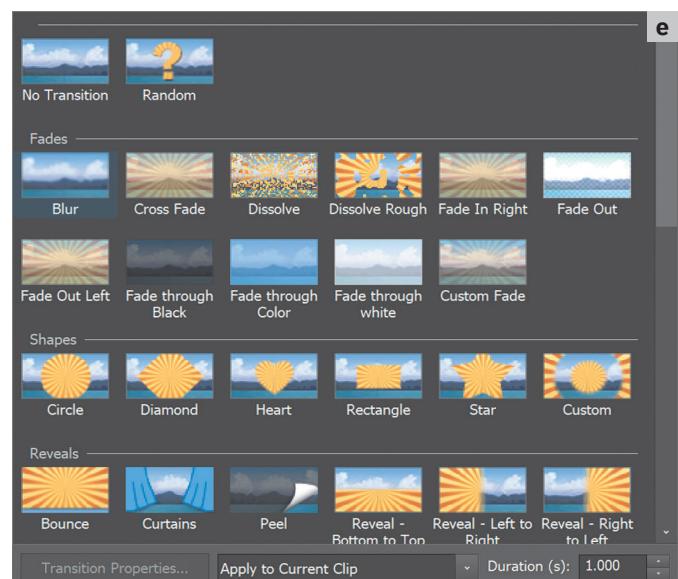
Pentru a face aceasta, segmentați fiecare interviu (vedeți în rândurile de mai sus tăierea unei secvențe video), apoi trageți pur și simplu cu mouse-ul fiecare secvență și mutați-o acolo unde doriți, până obțineți aranjarea dorită a tuturor secvențelor.

După ce ați pus secvențele cu interviurile colegilor în ordine alfabetică și ați copiat secvența cu aplauzele după fiecare interviu, urmează să puneti efecte de tranzitii între secvențe (între scene).

## Aplicarea unor efecte de tranzitii între scenele care formează proiectul

Între oricare două secvențe alăturate veți observa simbolul . Dacă veți face click pe el, se va deschide fereastra din *imaginea e*, alegeti de acolo una dintre opțiunile puse la dispozitie, și apoi veți putea insera un efect de tranzitie între cele două secvențe alăturate simbolului . Doar alegeti efectul dorit și faceți click pe el.

Aveți la dispozitie un set foarte bogat de efecte de tranzitii. și acestea nu sunt toate! Dacă veți merge cu mouse-ul pe una dintre imagini și îl veți ține puțin nemîscat, veți vedea animația corespunzătoare respectivului efect de tranzitie.



Opțiunile dintre care trebuie să alegeți înainte de a se aplica efectul sunt (*imaginea f*):

- **No, Freeze frames at the start/end of transition** (*Nu, înghețați cadrele la începutul/sfârșitul tranzitiei*) – previne schimbarea pozițiilor secvențelor;
- **Yes, Move the clip for an overlapping transition** (*Da, mutați clipul pentru o tranzitie suprapusă*) – schimbă poziția secvențelor, suprapunându-le puțin pentru a acoperi durata tranzitiei, asigurând o tranzitie mai fină.

O tranzitie mai aparte este tranzitie **Random** (*Aleatorie*). Este utilă dacă vă e greu să vă hotărăti ce efect de tranzitie să alegeti sau dacă vă plac surprizele și preferați să lăsați programul să vă facă o surpriză cu o tranzitie aleasă de el.

## Rețineti

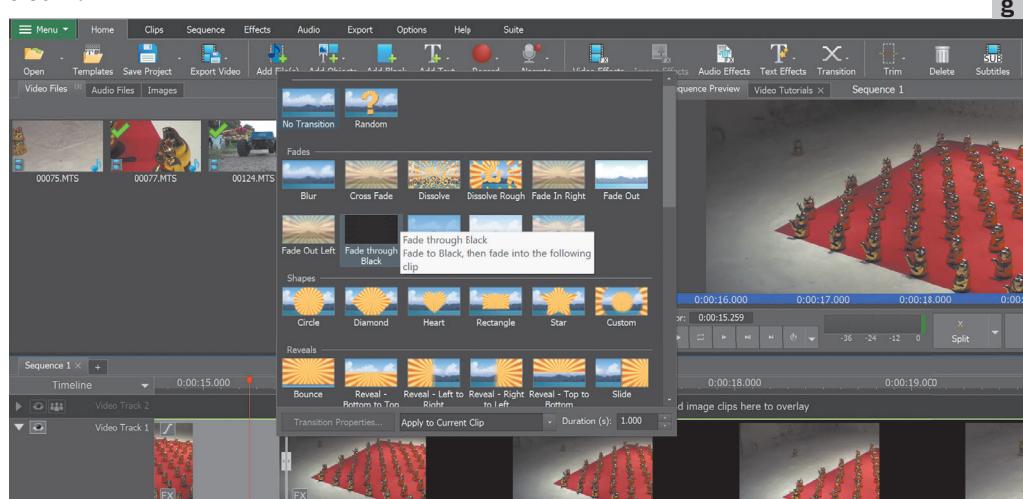
 **Tranzitie** este trecerea de la o secvență video la o altă secvență video. O tranzitie ascunde gradual secvența video curentă pe măsură ce aceasta se apropie de final și afișează treptat următoarea secvență video din montajul video. Această trecere se poate face prin estompare, întoarcere sau printr-o multitudine de alte efecte.

## Aplicați și investigați

- ▶ Încercați mai multe efecte și alegeti-le pe cele care se potrivesc cel mai bine cu secvențele între care se face tranzitie, ținând cont, desigur, și de preferințele voastre personale.

Două tranzitii mai speciale sunt:

- **Fade In** (*Estompare de început*) – tranzitie de la nimic la secvența video în derulare. Se folosește pentru începerea lină a unui video sau a unei secvențe dintr-un video;
- **Fade out** (*Estompare de final*) – tranzitie de la secvența curentă la... nimic (la negru). Aceasta este deseozi folosită la finalul unei secvențe video, sugerând astfel terminarea filmului.



Cum eliminați un efect de tranzitie dintre două secvențe, odată ce l-ați pus? Doar faceți click pe simbolul  (de la care ati și introdus efectul de tranzitie) și selectați **No transition** (*Fără tranzitie*).

Proiectul vostru deja a prins contur. Ați eliminat ceea ce era în plus, ați pus secvențele în ordinea dorită, ați copiat și ați inserat aplauze după fiecare secvență cu interviu și ați aplicat efecte de tranzitie între secvențe.

## Exersați

1. Parcurgeți enunțurile de mai jos și scrieți, în caiete, **A** (Adevărat) sau **F** (Fals) pentru fiecare dintre ele:
  - a. Pentru a șterge o porțiune dintr-o secvență video, mai întâi trebuie selectată porțiunea vizată. **A** **F**
  - b. Tăierea unei secvențe video este utilă atunci când vreți să inserați ceva într-un anumit punct al înregistrării sau când vreți să mutați o porțiune de film. **A** **F**
  - c. Nu se poate copia o secvență audio de mai multe ori pe o pistă audio a unui proiect video. **A** **F**
2. Descrieți pașii necesari pentru mutarea unei secvențe video în cadrul unui proiect VideoPad.
3. Descrieți trei exemple de efecte de tranzitii disponibile în aplicația VideoPad.
4. Ce efect are opțiunea **Yes, Move the clip for an overlapping transition** (*Da, mutați clipul pentru o tranzitie suprapusă*) la aplicarea unui efect de tranzitie între două secvențe video ale unui proiect?

# 6 Prelucrări superioare în proiectul video

## Descoperiți

În cele ce urmează vom discuta despre câteva prelucrări care fac proiectul vostru („Sportul meu preferat”) să fie unul desăvârșit:

- inserări de texte (suprapunerea textului peste scene pentru adăugare de titluri și/sau realizare de generice);
- suprapunerea unei coloane sonore peste scene;
- estomparea imaginii de final până la închiderea completă (**Fade Out**);
- exportarea filmului într-un format video potrivit scopului final.

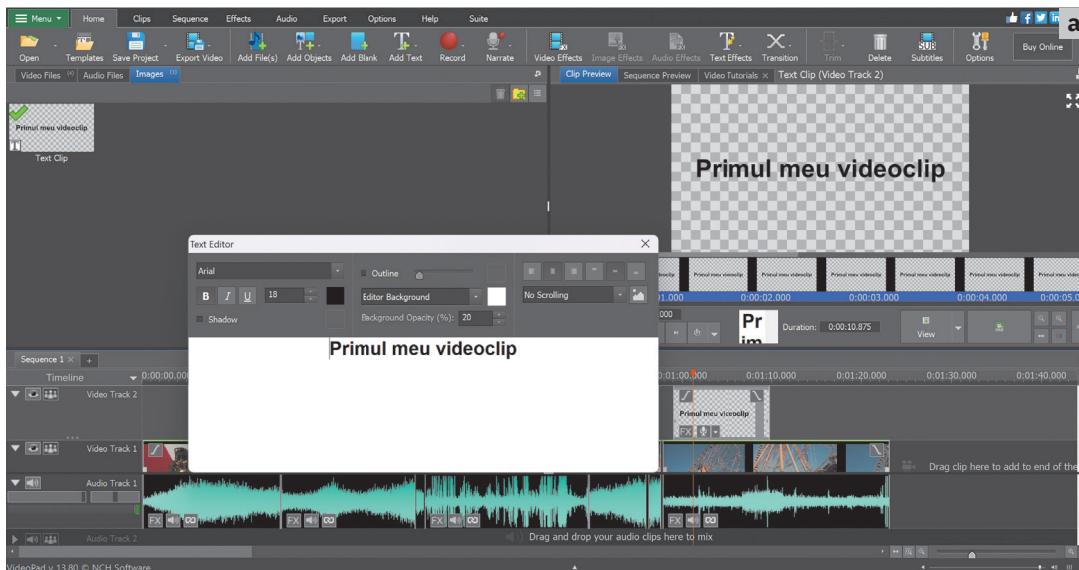
### Inserări de texte

Puteți avea nevoie să inserați texte în videoclipul vostru, în situații precum:

- punerea unui titlu videoclipului (text mare, pentru câteva secunde, pe cadrul de început);
- afișarea unor texte explicative la anumite secvențe;
- crearea unui generic de sfârșit, în care să fie trecute anumite informații, coautori, colaboratori etc.

Cum puteți inseră un text în videoclip? Mergeti în fila **Home** (Acasă) și faceți click pe butonul **Add Text** (Adaugă text). Se va deschide o mică fereastră, cum puteți vedea în imaginea de mai jos (*imaginea a*), în care introduceți textul dorit și îl setați cu diferite caracteristici:

- tip de font, dimensiune, culoare, îngroșat sau nu, aplecat sau nu, subliniat sau nu;
- grosimea conturului scrisului, dacă se afișează fundal sau nu (și se poate alege să se afișeze fundal pe întregul ecran sau numai un dreptunghi în jurul textului);
- alinierarea textului, atât pe orizontală (stânga, centrat, dreapta), cât și pe verticală (sus, centrat, jos);
- efecte de derulare a textului, atât pe orizontală, cât și pe verticală.



Observați că a apărut automat încă o pistă video (Video Track 2) pe care se vede secvența cu titlul. Implicit, aceasta are durata de 3 secunde și începe de unde era poziționat cursorul în momentul inserării (când ați apăsat butonul **Add text**). Secvența cu text poate fi trasă cu mouse-ul și repoziționată oriunde, astfel încât să înceapă de la orice moment al videoclipului vostru. De asemenea, trăgând de una dintre marginile ei, i se poate scurta sau lungi durata, în funcție de cum aveți nevoie.

## Rețineți

Este necesar să setați durata afișării textului și în funcție de lungimea lui, astfel încât cei care vor vedea videoclipul să aibă suficient timp pentru lecturarea textului. În special dacă textul este mai lung, încercați chiar să îl citiți și potriviti timpul de afișare astfel încât să vă ajungă să îl citiți integral.

 Pentru a vedea în timp real modificările efectuate asupra textului și cum se văd ele suprapuse pe înregistrarea video, faceți un click în zona mediană a ferestrei care este disponibilă cât timp sunteți în modul de editare a textului, și alegeți **Editor Background** (Fundalul Editorului), prin aceasta înțelegându-se secvența video existentă, peste care apare textul introdus de voi.

Ca să ieșiți din editarea textului și să vedeați rezultatul obținut, puteți face click oriunde în zona secvențelor și pistelor. Ca să reîntrați în modul de editare a textului, faceți dublu click pe secvența cu text de pe pistă video care îi corespunde.

În acest fel puteți inseră orice text, oriunde în videoclipul vostru (ca titlu la început, ca o explicație undeva pe parcurs sau ca un generic de sfârșit, la care puteți și să adăugați efectul de derulare pe verticală).

## Suprapunerea unei coloane sonore peste întregul videoclip

Este important să căutați un fundal muzical care să se potrivească subiectului ales, atât din punctul de vedere al conținutului, cât și al duratei. Sigur, în privința duratei, puteți aranja chiar voi să se potrivească: dacă este prea lung, îl tăiați și eliminați bucața care este în plus, iar dacă este prea scurt, îl puteți pune de două ori unul după altul (îl repetați) sau chiar de mai multe ori, până când acoperiți toată durata videoclipului.

Dar cum punem o coloană sonoră videoclipului? Doar trageți fișierul audio ales peste zona gri din zona secvențelor și pistelor, pe care, de altfel, vedeti scris **Drag and drop your audio clips here to mix** (*Trageți și puneți aici clipurile voastre audio*).

În acest moment aveți o pistă audio care conține fundalul vostru muzical (pista audio 1, dacă videoclipul vostru este format numai din fotografii, respectiv pista audio 2, dacă în videoclip ați inserat și secvențe filmate cu sunor, deci cu pistă audio proprie). Poziționați-vă la sfârșitul zonei video și (fiindcă este mai lungă) tăiați acolo fundalul audio și eliminați porțiunea care este în plus.

Acum nu vă mai rămâne decât să vă asigurați că secvența audio se sfărșește frumos, estompat, sugerând încheierea videoclipului (dacă am lăsa-o aşa cum a rezultat în urma tăierii, sigur ar fi un sfârșit brusc, lipsit de profesionalism). Pentru aceasta, selectați secvența audio (click pe ea), mergeți în fila **Audio** și de aici alegeți **Fade** (*Estompare*). Pentru estomparea de final alegeți **Fade Out** (*Estompare la ieșire*), iar în fereastra care se deschide setați durata pe care să se întindă acest efect de estompare. Implicit este setat la 3 s, dar puteți alege orice durată, de la 1 s în sus, pentru un efect suficient de bine conturat. În acest fel, ultimele secunde ale fundalului audio se vor estompa treptat de la volumul normal la zero (aşa cum puteți observa și vizual – linia albastră de pe reprezentarea grafică a fundalului sonor din *imaginie b*).

Încercați și vedeti rezultatul obținut. Dacă începutul melodiei alese pentru fundalul muzical este prea brusc, puteți aplica același efect, dar pentru începutul sunetului (**Fade In – Estompare pe intrare**).

Pentru cazul în care aveți și o secvență video cu sunor, deci aveți două piste audio, puteți seta volumul sunetului pe fiecare pistă în parte (de la cursoarele 1 și 2, după cum puteți vedea în *imaginie c*), astfel încât sunetul final al videoclipului să fie un mixaj potrivit între coloana sonoră care trebuie să se audă mai încet, să rămână în fundal, și sunetul care însoțește secvența video și care, cel mai probabil, conține cuvintele celui sau celor filmați și trebuie să se audă tare și clar.

## Estomparea imaginii de final

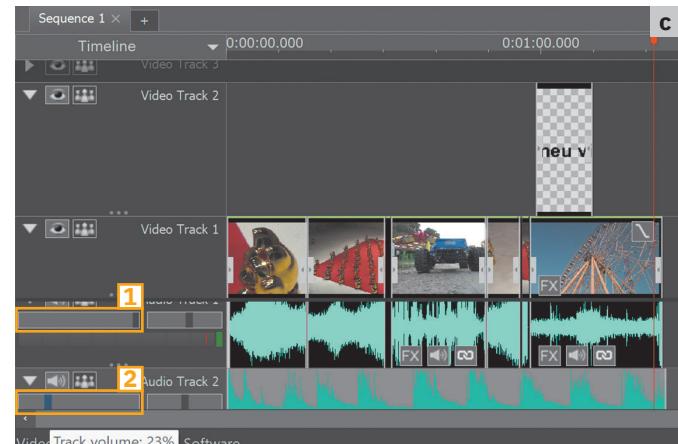
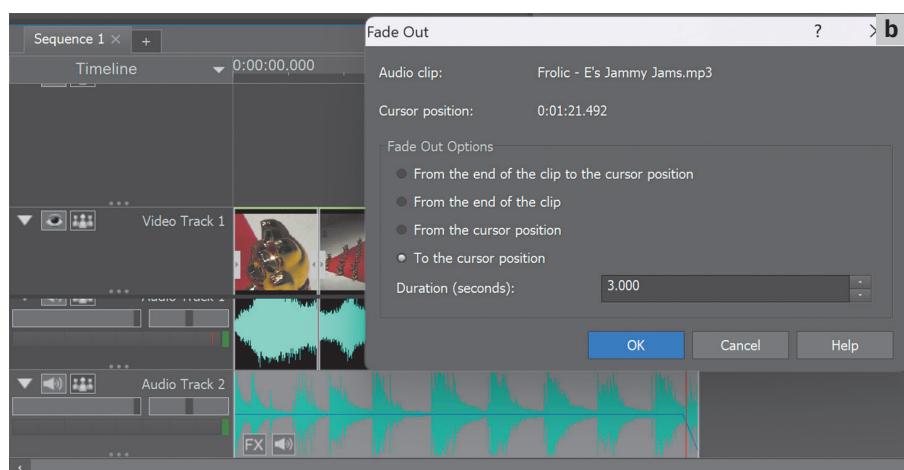
► Similar cu efectul de estompare a sunetului, puteți aplica un efect de estompare pe o secvență video. Atenție, este valabil numai pentru secvențele video, nu și pentru imagini, fotografii, chiar dacă sunt puse și ele pe pistă video. Selectați secvența finală a filmului și, apăsând butonul **Transitions** (*Tranzitii*), selectați durata și alegeți **Fade through Black** (*Estomparează prin negru*).

În acest moment lucrul la videoclipul vostru s-a terminat. Nu vă mai rămâne decât să îl exportați într-un fișier video. Dar nu uitați să salvați mai întâi proiectul în starea aceasta finală, cu toate operațiile efectuate.

Pentru a vedea cum arată rezultatul final, alegeți opțiunea **Preview Project** (*Previzualizare proiect*) din meniul **File** (*Fișier*).

## Exportarea filmului în diferite formate video

► Pentru a exporta videoclipul (filmul) vostru într-un fișier video, alegeți opțiunea **Export Video...** din meniul **File** (*Fișier*) sau din fila **Home** (*Acasă*). Aveți posibilitatea să încărcați videoclipul direct pe platforme ca YouTube, Flickr, Google Drive etc., să înregistrați videoclipul direct pe un disc (DVD sau Blu-ray) sau să îl exportați într-un fișier video. Acest fișier poate fi într-unul dintre formatele video: **.mp4, .3gp, .avi, .mov, .mpg, .ogv, .wmv** etc.



## Prelucrări superioare în proiectul video

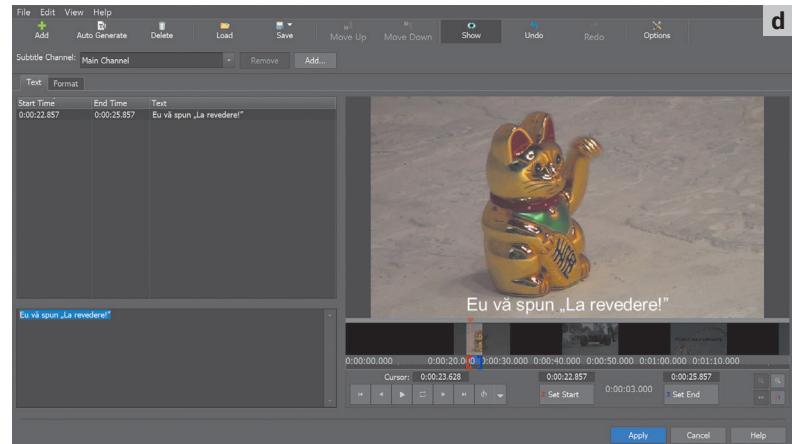
Alegeți formatul dorit, de exemplu **.avi**, și apăsați butonul **Create** (Creează). Nu vă speriați, această operație poate să dureze mai mult, în funcție de durata videoclipului vostru, de elementele pe care le-ați inserat în el și de puterea și viteza calculatorului pe care lucrați.

La final, veți obține un fișier video cu extensia **.avi** și cu același nume pe care i-l-ați dat și proiectului.

### Aplicați și investigați

► Urmați pașii de mai jos în ordinea în care sunt date:

1. Realizați o scurtă filmare cu tema „Hobby-ul meu”, care să conțină câteva dialoguri, sau găsiți pe Internet un film scurt care poate fi utilizat gratuit de către voi. Pe acesta urmează să îl prelucrați.
2. Deschideți filmul cu aplicația VideoPad.
3. Adăugați titlul „Hobby-ul meu”, utilizând un font de Arial de dimensiunea 22, bold, centrat la mijlocul ecranului și cu efect de defilare de jos în sus.
4. Adăugați o coloană sonoră.
5. Ajustați volumul acestia astfel încât să fie plăcut pentru ascultare și să nu deranjeze dialogurile din film.
6. Pe parcursul filmului adăugați texte de subtitrare la toate dialogurile existente, astfel încât filmul să poată fi urmărit și înțeles și de către persoane cu deficiențe de auz (vedeți exemplul din *imaginea d*). Acest lucru se realizează utilizând meniul **Subtitles** (Subtitrări) pentru fiecare secvență în parte. Sincronizarea subtitrării cu sunetul se realizează selectând cu atenție punctul de început (**Start**) și punctul de sfârșit (**End**) al afișării textului în cadrul secvenței sonore.
7. La sfârșitul filmului adăugați un generic în care să apară un mesaj de final, autorii și coautorii lui. Acest lucru se realizează similar cu adăugarea titlului filmului la început. Menționați sursele dacă este cazul.



### Exersați

1. În ce situații puteți avea nevoie să inserați texte într-un videoclip? Dați trei exemple.

2. Scrieți, în caiete, literele corespunzătoare enunțurilor corecte:

-  a. Pentru estomparea de final a unei secvențe video se alege opțiunea **Fade out**.
- b. Pentru estomparea de final a unei secvențe audio dintr-un proiect video se alege opțiunea **Fade In**.
- c. Exportarea filmului dintr-un proiect VideoPad se poate face într-unul dintre formatele video: **.mp4, .3gp, .avi, .mov, .mpg, .ogv, .wmv**.

3. Se poate încărca direct pe YouTube un clip video generat dintr-un proiect VideoPad?

4. **Proiect.** Realizați un videoclip cu durată între 30 de secunde și 3 minute, pe o temă socială, culturală, ecologică sau educațională (de exemplu, „Parcurile din jurul nostru îngrijite de copiii de la școlile din localitate”), pe următorul scenariu:

- să fie o succesiune de secvențe filmate și/sau fotografii pe tema aleasă (se pot folosi inclusiv fotografii ale panourilor expuse în școli, panouri realizate de copii pentru proiectele la care au participat);
- dacă se folosesc fotografii, acestea să fie în număr de cel puțin zece;
- dacă sunt folosite fotografii sau secvențe filmate luate de pe Internet, se vor folosi numai cu acordul scris al autorilor (dacă este specificat pe site-ul de pe care sunt luate că este permisă folosirea lor fără a se plăti drepturi de autor) sau cu menționarea numelor autorilor respectivelor materiale și a surselor pe genericul de final;

- să se pună mare, pe ecranul de început, titlul videoclipului;
- să se pună efecte de tranzitii între cadre;
- să se adauge un fundal muzical adecvat care să se întindă pe durata întregului videoclip.

Videoclipul se exportă la final

în format **.mp4**.



# Exerciții recapitative

Rezolvați pe caiete sarcinile de lucru de mai jos.

- Alegeți afirmația incorectă din cele de mai jos:
  - Fade Out** este o prelucrare care se poate aplica atât secvențelor audio, cât și secvențelor video.
  - În majoritatea programelor, operațiile creare, salvare, deschidere și închidere de fișiere sunt în meniu **File (Fișier)**.
  - Înregistrările de sunet se fac numai cu aparatură specializată, pe benzi speciale, și apoi sunt introduse în programele de prelucrare audio.
- Realizați corespondența între noțiunile din coloana stângă și explicațiile din tabelul de mai jos.

1. Paste Mix	a. prima și cea mai importantă acțiune care se poate face cu programul WavePad
2. Voice Change	b. aplică un efect de ecou peste sunetul selectat
3. Înregistrarea de sunet	c. mixează două secvențe audio
4. Echo	d. efect special prin care se obțin intonații diferite de voce

- Transcrieți în caiete, din lista următoare, doar aplicațiile de prelucrare video:
  - Audacity;
  - WavePad;
  - VirtualDub;
  - TwistedWave;
  - MovieMaker;
  - VideoFX Live;
  - Wavosaur;
  - Acoustica Basic;
  - WaveShop;
  - VideoShow.
- Alegeți afirmația corectă. WavePad este un program pentru:
  - editare video;
  - editare grafică;
  - editare audio.
- Completați textul următor în caiete, folosind cuvintele potrivite:
 

Prelucrarea ..... este acea succesiune de operații ..... și ..... prin care un material audio sau video suferă anumite ....., astfel încât rezultatul final să fie un material de mai bună calitate, care să îndeplinească anumite ..... (să aibă o anumită durată, să fie cu anumite caracteristici tehnice) și să atingă ..... pentru care a fost creat.
- Notați, în caiete, **A** (Adevărat) sau **F** (Fals) pentru fiecare propoziție de mai jos:
  - WavePad este o aplicație de prelucrare video.
  - Aplicațiile dedicate înregistrării și editării fișierelor audio se pot folosi online, dar și instalate local, pe calculator sau telefonul inteligent.
  - Efectele **Fade in/Fade out** se pot aplica atât fișierelor audio, cât și celor video.A F  
A F  
A F
- Ce înseamnă „mixarea semnalului audio”?

## Autoevaluare

Timp de lucru: **10 minute**

Rezolvați în caiet exercițiile date. Evaluați-vă singuri rezultatele.

- Completați, în caiete, fiecare dintre enunțurile următoare:
  - Pentru a insera un text în videoclip, mergem în fila **Home (Acasă)** și facem click pe butonul .....
  - O modalitate de a crea un nou proiect video este accesarea meniului **File (Fișier)** de unde puteți alege .....
  - Zona de previzualizare este acea zonă din interfața VideoPad în care puteți vedea cum o să arate rezultatul final al proiectului vostru .....
- Dați trei exemple de efecte speciale care se pot aplica unei înregistrări audio.
- Completând rebusul alăturat în caiete, veți obține pe coloana **AB** numele butonului de redare al unei secvențe audio sau video.
 

A  
 1      2  
 3      4  
 B

  - Editor video popular, ușor de folosit, care oferă foarte multe facilități de prelucrare video.
  - Acțiuni specifice care se realizează asupra unor secvențe video sau audio.
  - Ajustarea volumului astfel încât cel mai puternic vârf să fie *egal cu* sau să fie *un procent din* semnalul maxim care poate fi utilizat în fișierul audio. (în engleză)
  - A crește volumul sonor al zonei selectate. (în engleză)

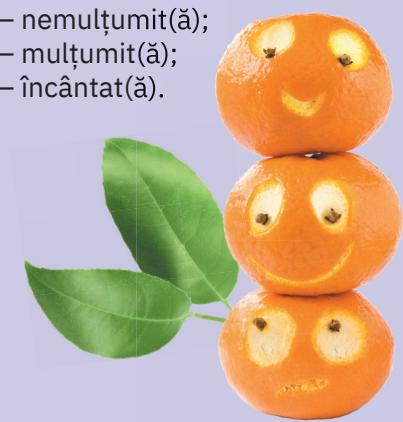
### CUM VĂ APRECIATI?

- 1 Desenați ● în dreptul sarcinilor pe care le-ați rezolvat corect.

●	●●	●●●
40 p.	70 p.	100 p.

- 2 Scrieți cum vă simțiți după ce ați rezolvat acest test, alegând unul dintre cuvintele de mai jos:

- nemulțumit(ă);
- mulțumit(ă);
- încântat(ă).



# U3

## Aplicații colaborative



Lecția 1

48-50 Aplicații colaborative. Noțiuni introductive

Lecția 2

51-56 Google Apps (Docs, Slides). Interfață și utilizare

Lecția 3

57-59 Noțiuni despre GIS și aplicabilitate

Exerciții recapitulative

60

Autoevaluare

60

# 1 Aplicații colaborative. Noțiuni introductive

## Amintiți-vă

Există aplicații software care permit salvarea în cloud a diferitelor fișiere lucrate, spre exemplu versiunile Microsoft Word apărute după Microsoft Office 2010 și Microsoft Office 2011 pentru Mac.

## Descoperiți

### Noțiunea de aplicație colaborativă

Apariția Internetului a facilitat comunicarea, interactivitatea dintre oameni, colaborarea pentru realizarea diferitelor tipuri de activități. Au apărut în mod natural aplicații care pun la dispoziție diferite funcționalități pentru lucrul în echipă, oferind astfel oportunitatea unei creșteri a productivității, realizarea unei comunicări foarte bune, acumularea rapidă de informații pe o temă dată, posibilitatea validării imediate a acestora de către utilizatori acreditați sau de către comunitate. Aceste aplicații poartă numele de **aplicații colaborative**.

## Rețineți

O **aplicație software colaborativă** este acea aplicație care ajută persoanele implicate în activități comune să își atingă scopul comun, lucrând împreună, în colaborare. *Colaborarea* este văzută aici ca muncă de echipă, membrii participanți completându-se în munca lor pentru a obține rezultate mai bune, într-un timp mai scurt, pentru realizarea acelorași obiective. *De exemplu:* scrierea unei cărți de către mai mulți autori; fiecare dintre aceștia scrie câte un capitol, iar munca lor pusă cap la cap formează cartea întreagă.

Aplicațiile colaborative se împart în două categorii majore: aplicații colaborative în timp real (sincrone) și aplicații colaborative care nu sunt în timp real (asincrone).

**Aplicațiile software colaborative sincrone** permit grupurilor de persoane să comunice, să colaboreze și să realizeze activități în același timp, în timp real, indiferent de locația lor. Câteva exemple de astfel de software colaborativ: aplicații de comunicații video (de exemplu, Skype) și chat, aplicații de editare colaborativă și partajare de documente.

Întreaga suită **Microsoft Office 365°** (Word, Excel, PowerPoint etc.) este un exemplu de suită de aplicații colaborative în timp real. Pentru a avea acces, utilizatorul trebuie să fie conectat la contul său Microsoft și își poate stoca fișierele lucrate în cloud, de exemplu, în OneDrive, permitând și altor utilizatori care dețin un cont Microsoft să aibă acces la acestea. Astfel, se poate lucra la acele fișiere în mod colaborativ, simultan. Fiecare utilizator care partajează accesul la un fișier din OneDrive vede în timp real ceea ce modifică un altul, fiecare persoană din echipă care editează fișierul poate renunța la modificările făcute sau să le salveze. Ultimul care salvează fișierul decide care sunt versiunile de modificări care rămân.

O altă suită de aplicații de același tip colaborativ este **Google Apps** (Docs, Sheets, Slides etc.) care permite, de asemenea, gestionarea și pregătirea facilă, în colaborare, a diferitelor tipuri de documente.



## Rețineți

Acste aplicații care permit utilizatorilor să lucreze simultan, în timp real, pe același fișier, având posibilitatea să renunțe la modificările făcute sau să le salveze, poartă numele de **aplicații colaborative pentru editare în timp real (Real Time Collaborative Editing – RTCE)**.

**Aplicațiile colaborative asincrone** nu se bazează pe comunicația în timp real. Utilizatorii lucrează în colaborare, dar nu în același timp. Câteva exemple de aplicații software de colaborare asincronă sunt: e-mailul, calendarele, sistemele de flux de lucru.

## Facilități și avantaje ale aplicațiilor de tip colaborativ

Printre funcționalitățile comune pe care le pot avea aplicațiile collaborative enumerăm: comunicarea prin mesaje, conferințele audio și video, managementul activităților, gestionarea documentelor, calendarele, urmărirea timpului, căutarea avansată.

Câteva dintre avantajele utilizării aplicațiilor collaborative sunt: creșterea productivității și atingerea mai rapidă a obiectivelor, comunicarea mai facilă, coordonarea persoanelor aflate oriunde în lume, facilitarea managementului de proiect, facilitarea lucrului în echipă și consolidarea relațiilor între membri, scăderea timpului de realizare a activităților, scăderea costurilor, creșterea nivelului de transparentă al procesului de realizare a activităților.

## Noțiuni de etică într-un mediu colaborativ; hacker și cracker

 Dezvoltarea utilizării aplicațiilor collaborative ridică noi probleme de ordin etic. Fiecare persoană care face parte dintr-o echipă de lucru trebuie să își respecte colaboratorii și munca acestora. Este important ca în totdeauna într-un mediu colaborativ să fie asigurate: distribuirea echitabilă și democratică a beneficiilor cunoașterii căpătate, păstrarea intimității membrilor echipei, securitatea lucrului, diseminarea corectă și în mod echitabil a informațiilor legate de realizarea de inovații în urma lucrului în colaborare, menținerea și îmbunătățirea obiectivelor.

Există persoane care însă doresc să profite de pe urma muncii celor din jurul lor sau care doresc să facă rău. S-a incetătenit pentru acestea denumirea de **hacker**, dar denumirea corectă este de **cracker**.

Un **hacker** este o persoană expertă în lucru cu calculatorul, care, folosindu-se de cunoștințele și experiența sa, poate obține informații și cunoștințe avansate legate de sistemele informatiche care nu sunt destinate publicului larg. Informațiile obținute pot fi puse în slujba dezvoltării acestor sisteme (odată descoperite anumite vulnerabilități, breșe de securitate, ele pot fi aduse la cunoștința proprietarilor sistemelor și pot fi eliminate). Există însă persoane care sunt experte în informatică și care acționează în mod dăunător și ilegal, cu scopul de a sparge sistemele informatiche pentru a produce pagube și a trage folioase proprii. Aceste persoane poartă denumirea de crakeri. Termenul **cracker** provine de la cuvântul în limba engleză *to crack* care înseamnă „a sparge”, relevant pentru spargerea parolelor, a sistemelor în general.



Un cracker poate fi foarte periculos, poate intercepta conversații sau transmisii de date, poate face modificări și intervenții pe datele utilizatorilor, poate sparge parole și poate fură bani din conturi, poate vírusa calculatoarele, poate să hărțuiască persoanele, și toate acestea în scopul unui câștig ilegal de bani.

## Protecția împotriva fraudei în mediul colaborativ online

O bună protecție împotriva fraudei într-un mediu colaborativ online o reprezintă întărirea securității în sistemele informatiche utilizate, înțelegerea și abordarea fenomenului infracțional cu mare responsabilitate. Protejarea conturilor și a parolelor de acces este un lucru principal. Este foarte important, de asemenea, ca în mediul colaborativ să se partajeze informațiile doar cu persoane de încredere și să existe în totdeauna un control.

La fel ca în viața reală, în mediul virtual nu se intră în discuții cu necunoscuți și nu se dau niciodată informații legate de viața personală, adrese, conturi etc.

## Accesare/conectare în aplicația colaborativă

Aplicațiile collaborative presupun stocarea fișierelor în cloud pentru a putea fi accesate de mai mulți utilizatori, din orice colț al lumii. Unul dintre cele mai populare sisteme de stocare în cloud este Google Drive, care este și gratuit. Astfel, prin suita de aplicații Google Apps se pot edita în colaborare documente (Google Docs), prezentări (Google Slides), foi de calcul (Google Sheets) etc. care sunt apoi stocate în Google Drive.

Un utilizator poate accesa aplicațiile Google Apps doar dacă are un cont activ Google. Toate aceste aplicații se accesează cu același cont (*în imaginea a*).

**ATENȚIE!** Google impune restricții de vîrstă pentru crearea unui cont personal. Verificați și citiți cu atenție specificațiile înainte de crearea unui cont Google (precum vîrsta minimă necesară pentru a-l crea). Pentru siguranță, solicitați ajutorul unui adult.

a

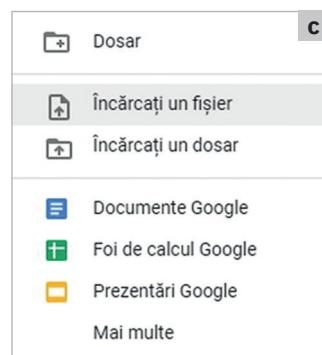
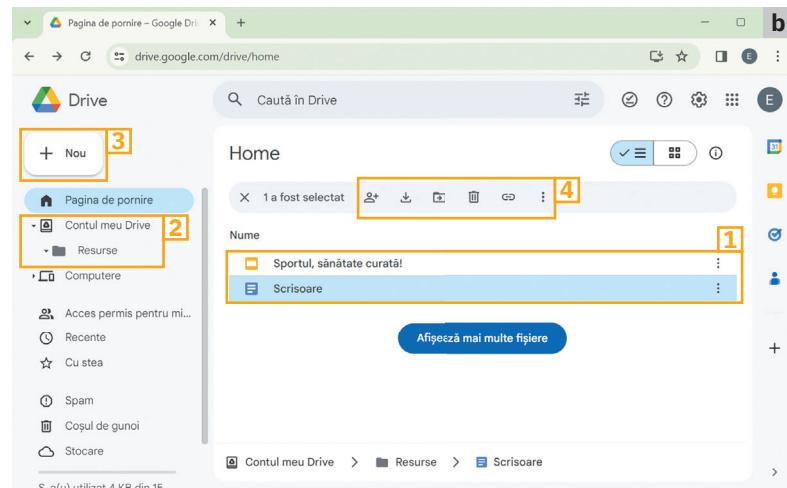
Dacă nu aveți un astfel de cont și nu vă puteți crea personal unul din cauza impunerii de vârstă minimă, vă prezentăm două posibilități pentru a-l obține:

- școala se poate înscrie în programul **G Suite for Education** și poate face câte un cont fiecărui elev care a împlinit vârsta de 13 ani;
- unul dintre părinți poate face un cont **Google For Families** – există această posibilitate special concepută de către Google – și poate să vă facă și vouă un cont de utilizator.

În cele ce urmează vom detalia cele mai importante elemente ale interfeței Google Drive.

► Google Drive se poate accesa introducând în orice browser: [drive.google.com](http://drive.google.com). Se afișează imediat fereastra de **Bun venit** în care se cere autentificarea în contul Google pentru conectare la acesta, deci trebuie introduse adresa de e-mail și parola. Interfața Google Drive este următoarea (*imaginea b*):

- Zona de afișare a fișierelor și folderelor – fișierele salvate în cloud în Google Drive pot fi organizate pe Foldere (dosare).
- Conținutul contului Google Drive – aici se afișează toate folderele stocate în contul Google Drive.
- Butonul **Nou** (New) – apăsarea acestui buton afișează un meniu derulant de unde se poate alege ce anume se dorește să se adauge la contul Google Drive: un nou folder, un document, o prezentare, o foaie de calcul etc. Meniul poate fi văzut în imaginea din dreapta (*imaginea c*).
- Butoanele din această zonă apar doar dacă este selectat un fișier sau folder. În ordine, de la stânga la dreapta, ele sunt folosite pentru adăugarea altor persoane cărora li se permite accesul la fișierul sau folderul selectat, pentru partajare în vederea accesării/lucrului colaborativ, respectiv descărcarea, mutarea, ștergerea sau copierea link-ului de acces la fișierul/folderul selectat. Ștergerea unui fișier nu este definitivă de la început. Fișierul este mutat inițial într-un folder, **Coșul de gunoi** (Trash), de unde se poate restaura (Restore) sau se poate șterge definitiv (Delete forever).



În lecția următoare vom detalia interfețele pentru lucrul colaborativ cu documente și prezentări în Google Apps.

## Aplicații și investigații

- Căutați pe Internet alte exemple de aplicații colaborative și identificați scopurile pentru care sunt utilizate.

## Exersații

- Scrieți, în caiete, A (Adevărat) sau F (Fals) pentru fiecare din propozițiile de mai jos:
  - Pentru a accesa Google Drive și Google Docs este nevoie de două conturi Google. A    F
  - Pe Google Drive se poate stoca orice tip de fișier. A    F
  - Pe Google Drive informațiile pot fi grupate în foldere făcute de utilizator. A    F
- Completați, în caiete, cuvintele lipsă din următorul enunț:  
*O aplicație software colaborativă este acea aplicație care ajută persoanele implicate în activități... să își atingă scopul comun, lucrând împreună, în...*
- Explicați care sunt diferențele dintre un hacker și un cracker. Dar asemănările?
- Ce este o aplicație software colaborativă pentru editarea în timp real?
- Faceți o paralelă între aplicațiile colaborative sincrone și asincrone.
- Dezbateră.** Inițiați o dezbatere în clasă având tema „Utilizarea aplicațiilor colaborative în procesul de învățare – avantaje și potențiale probleme”.



## Portofoliu

Realizați o prezentare PowerPoint în care să fie expuse avantajele utilizării aplicațiilor colaborative. Salvați-o în folderul **Lucru\_clsVII** cu numele **ApColaborative.pptx**.

## 2 Google Apps (Docs, Slides). Interfață și utilizare

### Descoperiți

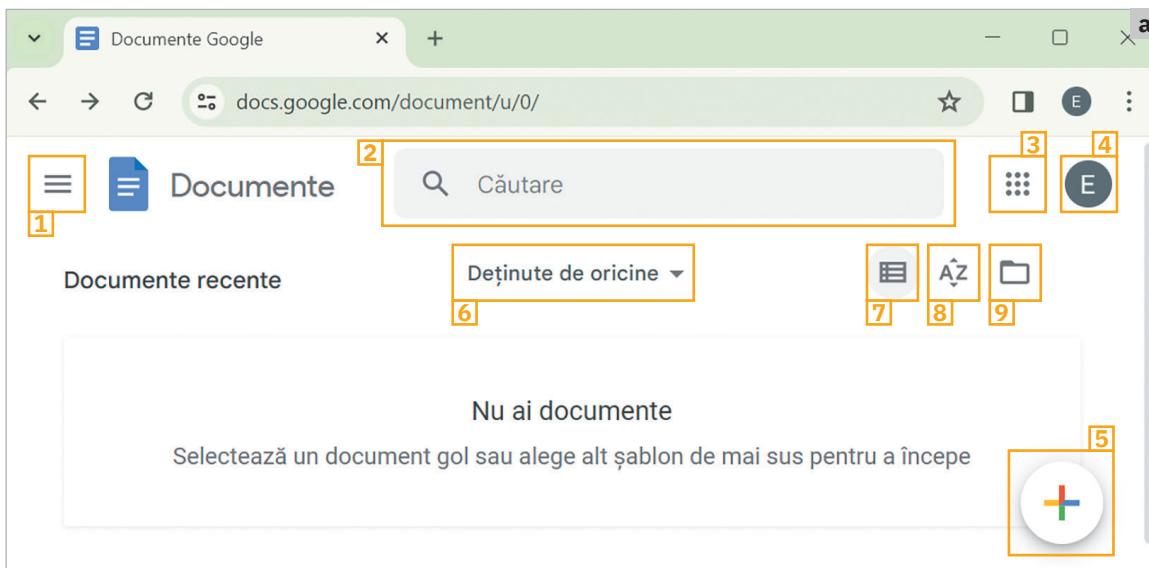
#### Aplicația colaborativă Google Docs

##### Pornirea aplicației

- Lansarea în execuție a aplicației se poate face în mai multe feluri dintre care prezentăm două, uzuale:
- accesați în orice browser: [www.docs.google.com](http://www.docs.google.com) sau [www.docs.google.com/document](http://www.docs.google.com/document). Se afișează imediat fereastra de **Bun venit** în care se cere autentificarea în contul Google pentru conectare la acesta (se solicită introducerea adresei de e-mail și a parolei). Dacă sunteți deja autentificat (conectat la contul Google), nu mai apare această fereastră;
  - accesați în orice browser: [www.google.com](http://www.google.com), apoi accesați butonul **Aplicații Google** (Google Apps). Se afișează imediat o fereastră din care selectați aplicația **Docs**. În cazul în care nu sunteți deja autentificați în contul Google, la selectarea aplicației apare fereastra de **Bun venit** în care se introduc datele contului Google.

#### Interfața aplicației Google Docs

Aplicația Google Docs se deschide cu un ecran de start (*imaginea a*).



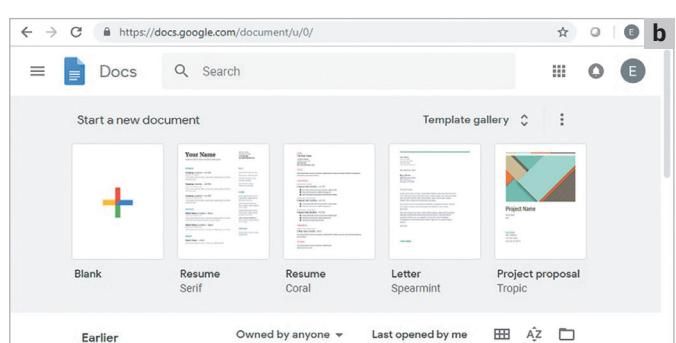
Bara Google conține:

- Meniu principal (Main menu)** – de aici se deschide o fereastră din care se poate comuta între aplicațiile Google sau se pot realiza diferite setări ale aplicației;
- Căutare (Search)** – se poate realiza căutarea unui fișier stocat pe Google Drive la care aveți acces. Se vor afișa toate fișierele care conțin în nume textul introdus aici;
- Aplicații Google (Google Apps)** – se deschide o fereastră din care se pot alege pentru deschidere alte aplicații Google pentru același utilizator;
- Gestionarea contului Google** – de aici se poate face deconectarea de la contul activ, comutarea între conturi (conectarea la un alt cont), gestionarea setărilor contului;
- Buton pentru crearea unui document nou.**

Zona de afișare a documentelor salvate în Google Drive pe care utilizatorul conectat le poate accesa sau modifica:

- Filtrul pentru afișarea documentelor** care au un anume deținător (utilizator care le-a creat); se poate alege dintre: deținute de oricine, deținute de mine, nedechinute de mine;
- Modalitatea de afișare** a documentelor: ca listă sau grilă;
- Opțiuni de sortare** a documentelor;
- Alegerea selectorului de documente** pentru a deschide un anume document dorit.

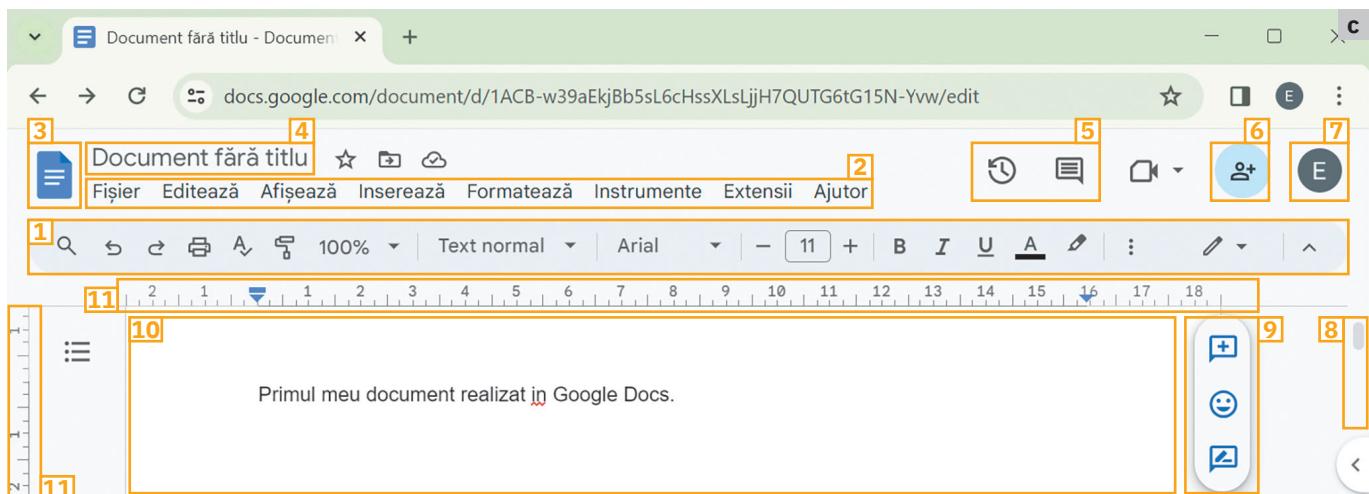
În versiunea în limba engleză sunt disponibile și diferite şabloni de documente care pot fi alese spre utilizare (*imaginea b*).



## Gestionarea unui document

## Interfață și funcționalități

Un document nou creat se deschide prin apăsarea butonului „+” din dreapta jos a ecranului (butonul nr. 5). La creare, implicit are numele **Document fără titlu** (*Untitled document, imaginea c*).



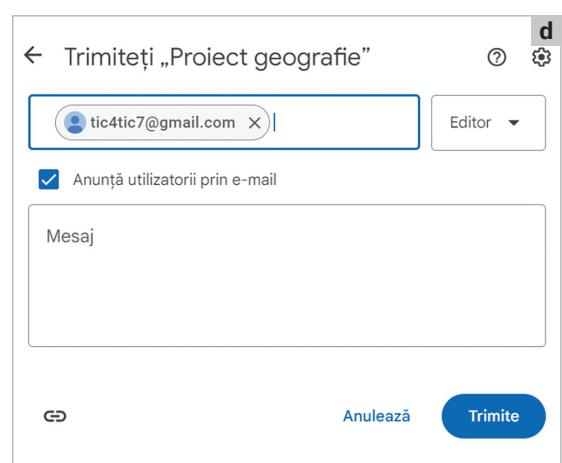
- 1. Bara de unelte (Tool Bar)** – conține comenzi rapide pentru diferite operații;
  - 2. Bara de meniuri (Menu Bar)** – conține meniuri sub care sunt grupate comenzi și opțiuni pentru lucrul cu documentul;
  - 3. Butonul de navigare la pagina de început** a Google Docs;
  - 4. Numele fișierului document** – Google Docs afișează aici numele documentului. La click pe el, numele se poate modifica;
  - 5. Notificări legate de editarea și salvarea documentului, precum și istoricul comentariilor** – aplicația Google Docs salvează documentul în mod automat, la fiecare câteva secunde în cloud, în Google Drive. Aici se poate vedea și urmări faptul că documentul a fost salvat. Pentru a regăsi documentele lucrate, deschideți aplicația Google Docs și verificăți lista de Documente recente;
  - 6. Distribuire (Share)** – se permite partajarea documentului cu alte persoane; se pot adăuga persoane sau grupuri, iar accesul poate fi de trei tipuri: cititor, comentator (poate să citească și să adauge comentarii) sau editor (are și drept de editare a documentului);
  - 7. Gestionarea contului Google (Google Account)** – permite conectarea, deconectarea de la contul activ sau comutarea între conturi;
  - 8. Bara de derulare pe verticală (Scroll Bar);**
  - 9. Butoane de adăugare comentariu, reacție cu emoji sau de sugerare modificări;**
  - 10. Zona document** – aici se realizează editarea efectivă a documentului;
  - 11. Rigla (Ruler)** – folositoare pentru formatarea documentului, aranjarea marginilor, indentare etc.

Comenzile și opțiunile pe care le pune la dispoziție Google Docs pentru tehnoredactare (editare, formatare, aranjare în pagină) sunt asemănătoare cu cele prevăzute de către Microsoft Word, inclusiv imaginile butoanelor de comandă. Dacă se stă cu mouse-ul deasupra unui buton, se afisează un text ajutător referitor la utilizarea butonului respectiv.

## **Lucrul colaborativ**

► Dacă se dorește ca un document să fie partajat cu o altă persoană pentru editare în colaborare, persoana respectivă trebuie să aibă cont Google. Dacă nu are cont Google, persoana respectivă primește un link înspre document și are doar posibilitate de vizualizare. Pentru partajare, se apasă butonul **6**. Se afișează o fereastră în care se completează adresele de e-mail ale persoanelor cu care se dorește partajarea și se apasă butonul **Trimite** (*Send, imaginea d*).

Cazul considerat spre exemplificare în cele ce urmează este următorul: avem doi utilizatori, cu două conturi Google, *tic4tic7@gmail.com* și *elev.c7@gmail.com*. Utilizatorul *elev.c7@gmail.com* a creat un document **Primul meu document...** utilizând aplicația Google Docs și partajează acest document cu utilizatorul *tic4tic7@gmail.com*. În imagine avem fereastra în care se specifică utilizatorii cu care se partajează documentul (*imaginea d*).



## Google Apps (Docs, Slides). Interfață și utilizare

The screenshot shows two parts of the Google Docs interface. On the left, a sharing invite from 'ElevCls7' is displayed, showing the document title 'Proiect geografie' and a link to 'Deschide'. On the right, the document itself is shown with the title 'Problemele Africii în lumea contemporană' and a subtitle '- studiu de caz -'. It includes a note 'Să facem puțină istorie a Africii!' and a signature 'Nume elev: Ion Popescu Cls. a VII-a D'. The toolbar at the top shows various editing tools.

Cel căruia î se acordă dreptul de a partaja documentul cu cel care l-a creat va primi un e-mail cu o invitație la editare, în care este anunțat că i s-a dat acces la document. Din acest moment, el poate să lucreze asupra documentului. În e-mail există un link direct către documentul partajat (**Deschide**).

În *imaginea e* este invitația la editarea documentului **Proiect Geografie** pe care a primit-o utilizatorul `tic4tic7@gmail.com` de la utilizatorul `elev.c7@gmail.com`.

Fiecărui participant la tehnoredactarea documentului î se permite modificarea conținutului acestuia. De asemenea, se poate vedea, în orice moment, care dintre utilizatorii cărora le-a fost partajat sunt activi. Totodată, se vede în timp real unde este cursorul acestora și ce modificări fac. Documentul este salvat automat în cloud, pe Google Drive.

În *imaginea f* este vizualizarea din contul `elev.c7@gmail.com`, cel care a și creat documentul. Se observă că utilizatorul **TIC CLS7** (`tic4tic7@gmail.com`) este activ, are și el deschis documentul pentru editare și se observă, de asemenea, unde este poziționat cursorul său în cadrul documentului.

Documentul partajat devine vizibil în Google Docs pentru toți cei care îl partajează. Pentru fiecare document este specificat cine l-a creat și data ultimei salvări. În *imaginea g* este vizualizarea din contul utilizatorului **TIC CLS7** (`tic4tic7@gmail.com`). În *imaginea h* se observă modificările realizate de către cei doi utilizatori (`tic4tic7@gmail.com` și `elev.c7@gmail.com`) asupra documentului.

Se poate vedea oricând istoricul versiunilor, accesând **Fișier (File) -> Istoricul versiunilor (Version History) -> Vezi istoricul versiunilor (See Version History)**. În *imaginea h* se observă modificările realizate de către cei doi utilizatori (`tic4tic7@gmail.com` și `elev.c7@gmail.com`) asupra documentului.

O versiune a unui document poate primi un nume pentru a fi regăsită mai ușor, dacă se apasă butonul marcat cu trei puncte din partea dreaptă a sa, alegându-se apoi **Redenumește (Rename)**. De asemenea, ea poate fi copiată ca document separat pe Google Drive, accesând comanda **Fă o copie (Make a copy)** din același meniu derulant care se deschide la apăsarea butonului marcat cu trei puncte.

The screenshot shows the Google Docs interface with the document 'Proiect geografie'. To the right, a detailed view of the version history is shown. It lists changes made by 'AZI' (Today) and 'IERI' (Yesterday). The 'AZI' section shows a change made on '6 martie, 01:01' with the note 'Versiunea actuală' and users 'Test Generic' and 'TIC CLS7'. The 'IERI' section shows a change made on '5 martie, 23:26' with the note 'Test Generic'. A checkbox at the bottom right allows users to 'Afipați modificările' (Show changes).

Documentul poate fi descărcat pe calculator din cloud, accesând **Fișier (File)** -> **Descarcă (Download)**. Se poate alege tipul de fișier pentru descărcare dintr-o listă pusă la dispoziție de către Google Docs: Microsoft Word (.docx), Format OpenDocument (.odt), Format text îmbunătățit (.rtf), Document PDF (.pdf), Text simplu (.txt), Pagină Web (.html, arhivă ZIP), Publicație EPUB (.epub).

-  De asemenea, documentul poate fi publicat pe Web, adică făcut vizibil pe Internet pentru oricine, accesând **Fișier (File)** -> **Trimite (Send)** -> **Publică pe Web...** (*Publish to the Web...*). Dacă se dorește ca documentul să fie transmis pe e-mail către o tertă persoană, se accesează **Fișier (File)** -> **E-mail (Email)** -> **Trimite fișierul prin e-mail (Email this file)**. În acest caz, fișierul se atașează ca document PDF (implicit) sau i se poate include conținutul în e-mail.

Dacă se dorește să se transmită colaboratorilor (celor cu care utilizatorul partajează documentul respectiv) un mesaj, se selectează **Fișier (File)** -> **E-mail (Email)** -> **Trimiteți un e-mail colaboratorilor (Email collaborators)**.

## Aplicați și investigați

► **Lucrați în perechi și parcurgeți pașii următori în ordinea în care sunt precizați:**

1. Deschideți aplicația Google Docs și creați un document nou, denumit **Scrisoare pentru bunici**.
2. Inserați un text de 5 rânduri, scris cu fontul Times New Roman de dimensiune 14, aliniat la stânga.
3. Partajați documentul cu colegul de bancă.
4. Colegul va insera un text de 4 rânduri cu font roșu, va folosi butonul **Culoarea textului** (*Text color*) din bara de unelte.
5. În același timp cu colegul, inserați un tabel de 2 linii și 3 coloane, accesând **Inserează (Insert)** -> **Tabel (Table)**.
6. Colegul va insera o imagine pe care o are pe calculator, accesând **Inserează (Insert)** -> **Desen (Image)** -> **Nou (New)**.
7. Urmăriți istoricul versiunilor pentru a vedea modificările făcute de fiecare în parte.
8. Descărcați pe calculator documentul.
9. Întoarceți-vă în Google Docs și regăsiți documentul vostru.

## Descoperiți

### Aplicația colaborativă Google Slides

#### Pornirea aplicației

Pornirea aplicației Google Slides se face asemănător cu pornirea aplicației Google Docs. Singura diferență este că se accesează [www.slides.google.com](http://www.slides.google.com) sau [www.docs.google.com/presentation](http://www.docs.google.com/presentation) din orice browser, sau se selectează aplicația **Slides (Prezentări)** din lista de aplicații Google Apps pusă la dispoziție de Google Chrome.

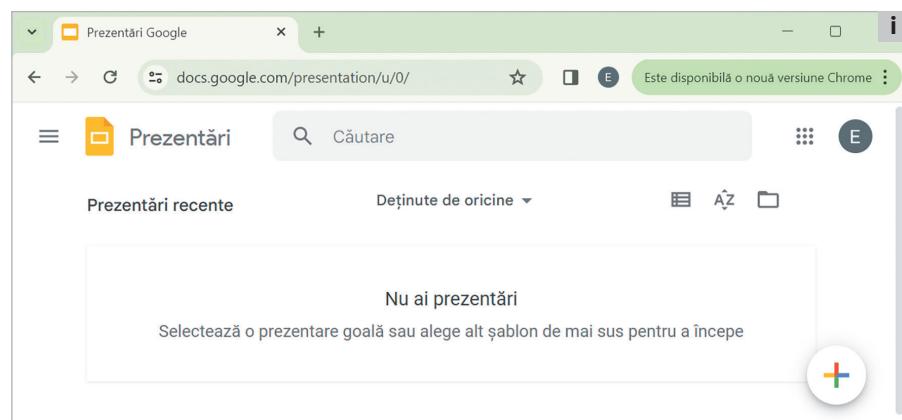
#### Interfața aplicației Google Slides

Fereastra de deschidere a aplicației Google Slides este similară cu cea a aplicației Google Docs (*imaginea i*).

#### Gestionarea unei prezentări

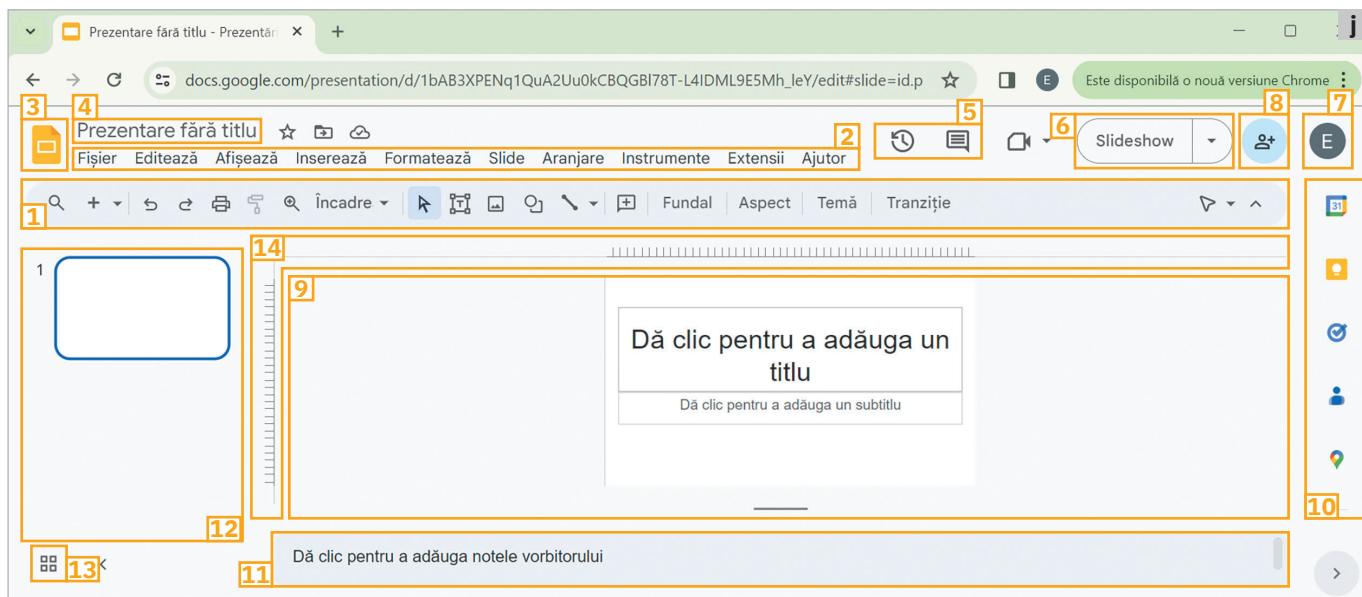
#### Interfață și funcționalități (*imaginea j*)

O prezentare nou creată se deschide prin apăsarea butonului „+” din dreapta, jos a ecranului de start al aplicației Google Slides. La creare, implicit are numele **Prezentare fără titlu** (*Untitled presentation*).



1. **Bara de unelte (Tool Bar)** – conține comenzi rapide pentru diferite operații; în funcție de elementul selectat pe ecran, bara de unelte afișează butoane speciale pentru lucrul cu acel element;
2. **Bara de meniuri (Menu Bar)** – conține diferite meniu sub care sunt grupate comenzi și opțiuni pentru lucrul cu documentul;
3. Buton de **navigare la pagina de început** a Google Slides;
4. **Numele fișierului** prezentare – Google Slides afișează aici numele dat de către utilizator prezentării. Numele se poate modifica dacă se dă click pe el;
5. **Notificări legate de editarea și salvarea documentului, precum și istoricul comentariilor** – aplicația Google Slides salvează prezentarea în mod automat, la fiecare

## Google Apps (Docs, Slides). Interfață și utilizare



câteva secunde în cloud, în Google Drive; aici se pot urmări toate modificările făcute pe prezentarea curentă, precum și toate comentariile făcute de cei care au acces la respectiva prezentare;

6. Butonul **Slideshow (Slideshow)** – pornește *slideshow*-ul și conține diferite opțiuni pentru pornirea prezentării;
7. **Gestionarea contului Google (Google Account)** – permite conectarea/deconectarea de la contul activ sau comutarea între conturi;
8. **Opțiuni pentru lucrul colaborativ** – se permite partajarea prezentării cu alte persoane; cei cărora li se permite accesul pot avea dreptul de a edita prezentarea, de a insera comentarii sau doar de a vedea prezentarea;
9. **Panoul diapozitiv** – este zona de editare a diapozitivului. Aici se poate lucra direct cu fiecare diapozitiv în parte;

Interfața Google Slides pune la dispoziția utilizatorilor comenzi asemănătoare cu cele prevăzute de către Microsoft PowerPoint, inclusiv imaginile butoanelor de comandă. Staționarea mouse-ului deasupra unui buton al barei de unele duce la afișarea unui text ajutător referitor la scopul acestuia.

Se pot include în prezentările realizate diferite tipuri de obiecte: texte, imagini, video, forme, tabele, diagrame din meniu **Inserează (Insert)**, prezentat în *imaginea k*.

Obiectele dintr-un diapozitiv pot avea diferite efecte, diferite animații, care se inseră din meniu **Inserează (Insert) -> Animatie (Animation)**.

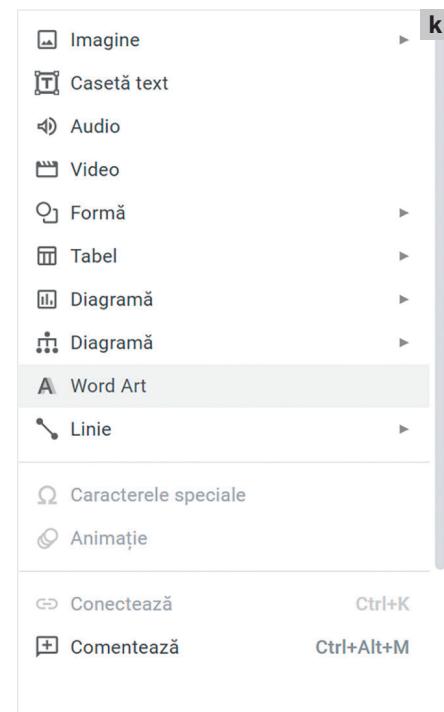
### Lucrul colaborativ

Lucrul colaborativ în Google Slides este identic cu cel din Google Docs. Întreaga suite Google Apps permite aceleași operații și funcționalități legate de partajarea de fișiere și de lucrul în echipă, simultan la ele.

### Ştiați că...?



Kahoot este o aplicație colaborativă ([kahoot.com/what-is-kahoot/](http://kahoot.com/what-is-kahoot/)) care poate fi utilizată cu succes în educație. Conținutul creat de cadrele didactice (teste cu răspuns grilă) poate fi accesat simultan de elevi de pe telefonul mobil sau de pe un calculator.





## Aplicații și investigații

► **Lucreți în grupuri** de câte doi elevi și parcurgeți pașii următori în ordinea în care sunt precizați:

1. Conectați-vă la unul dintre cele două conturi Google. Deschideți aplicația Google Slides și creați o prezentare nouă, **Sportul, sănătate curată!**, care să aibă ca subiect abordarea relației sport-sănătate.
2. Încărcați în Google Drive filmulețul realizat și prelucrat într-o lecție anterioară având tema „Sportul meu preferat”, care conține un interviu cu unul dintre colegi, ca mărturie video pentru placerea de a practica sportul. Pentru aceasta, accesați Google Drive, apăsați **Nou (New)** și selectați **Încarcă un fișier (File Upload)**.
3. Partajați prezentarea cu colegul/colega.
4. Adăugați la prezentarea voastră simultan 4 diapoziitive, fiecare câte unul, utilizând comanda **Inserează (Insert) -> Slide nou (New slide)** sau **Ctrl+M**.
5. Primul elev inserează pe primul diapoziтив o casetă de text, selectând **Inserează (Insert) -> Casetă text (Insert Textbox)**. Introduceți în acea casetă două propoziții.
6. Al doilea elev inserează pe al doilea diapoziтив o imagine pe care o are pe calculator, accesând **Inserează (Insert)**

-> **Imagine (Image)** -> **Încarcă din computer (Upload from computer)**.

7. Primul elev inserează pe al treilea diapoziтив texte și un tabel, accesând **Inserează (Insert) -> Tabel (Table)**. Tabelul trebuie să aibă 2 linii și 4 coloane și să conțină câteva informații despre popularitatea a 3 sporturi printre colegii voștri, astfel:

	Tenis	Fotbal	Înot
Nr. elevi din clasa ta cărora le place acest sport			

8. Al doilea elev inserează pe al patrulea diapoziтив filmulețul de la punctul 2. Pentru aceasta, se accesază **Inserează (Insert) -> Video (Video)**, apoi se selectează **Google Drive (Google Drive) -> Discul meu (My Drive)**.
9. Porneți prezentarea și vedeți ce a rezultat.
10. Afipați istoria versiunilor prezentării accesând **Fișier (File) -> Istoricul versiunilor (Version History) -> Vezi istoricul versiunilor (See Version History)**.
11. Întoarceți-vă în Google Slides și regăsiți prezentarea voastră.



## Exerciții

1. Faceți o paralelă între aplicațiile Google Docs și Microsoft Word. Enumerați trei asemănări și trei deosebiri.
2. De unde se poate vedea istoricul modificărilor într-un document Google Docs sau într-o prezentare Google Slides?
3. Se pot insera imagini într-un document creat cu Google Docs? Dar fișiere video lucrate de voi într-o prezentare Google Slides? Explicați cum se fac aceste acțiuni.
4. Notați, în caiete, **A** (Adevărat) sau **F** (Fals) pentru fiecare propoziție de mai jos:
  - a. O persoană care nu are cont Google poate doar să vizualizeze un fișier Google Docs partajat.
  - b. Un fișier Google Slides este salvat periodic.
  - c. O prezentare Google Slides poate fi descărcată pe calculator ca document **.docx**.
  - d. Într-un document Google Docs se pot insera sunete.

**5. Lucru în echipă.** Realizați, la nivelul clasei, o carte electronică cu titlul **Povești, povestiri, amintiri**. Parcugând toate acțiunile următoare, veți obține un minut proiect, realizat împreună:

- Mai întâi, alegeti un „manager de proiect”. El va iniția fișierul Word în care cu totii veți introduce câte ceva pentru realizarea cărții.
- Salvați fișierul pe Google Drive, cu titlul **Povești, povestiri, amintiri.doc**.
- Faceți ceea ce este necesar pentru ca toți elevii clasei să aibă acces la fișierul care va conține cartea.
- Asigurați-vă că aveți câte un cont Google pentru fiecare; dacă nu, creați conturile necesare.
- Managerul de proiect va da tuturor drept de acces la acel fișier (fiecare dintre voi va primi câte un e-mail de la managerul de proiect cu permisiunea de acces și cu link către fișierul care va conține cartea clasei).
- Scrieți, fiecare dintre voi, în documentul creat, o poveste, o povestire, o amintire care vi se pare interesantă sau hazlie și merită împărtășită. Puteți adăuga imagini și, de asemenea, puteți formața textul așa cum dorîți.



## Portofoliu

Scrieți un eseu de maximum o pagină cu tema „Aplicațiile collaborative – avantaje”. Producția voastră trebuie să vizeze, printre altele: descrierea avantajelor și a dezavantajelor, cu prezentarea unor exemple concrete.

### 3 Notiuni despre GIS și aplicabilitate

#### Amintiți-vă

Am discutat anterior despre aplicațiile colaborative ca fiind aplicații care le permit persoanelor implicate în activități comune să își atingă scopul comun, lucrând împreună, în colaborare. Munca în grup a unor oameni care comunică bine și rapid, oriunde s-ar afla ei, duce la obținerea de rezultate mai bune, într-un timp mai scurt, cu costuri reduse.

#### Descoperiți

Haideți să ne gândim ce înseamnă pentru o organizație, oricăr de mare ar fi, să realizeze o hartă a lumii, cu detalii și imagini pentru fiecare zonă în parte, începând cu toate detaliile de natură geografică și ajungând cu detalierea până la nivelul fiecărei clădiri în parte. Gândiți-vă la cum evoluează lumea și lucrurile, în general, și cât de repede se întâmplă modificări care ar trebui să se regăsească pe harta noastră.

Soluția este realizarea unor sisteme informatiche care pot lucra cu mari cantități de informații necesare construirii, accesării și întreținerii acestor hărți. Întreținerea hărților poate fi realizată de firmele care dețin aceste sisteme, dar există și sisteme care permit lucrul colaborativ, la alimentarea hărților cu detalii putând contribui orice persoană care poate accesa aplicația respectivă.

#### Ce este un Sistem Informațional Geografic (GIS)?

Un sistem care permite utilizarea informațiilor de natură geografică, a datelor și a caracteristicilor asociate acestor informații poartă numele de **GIS (Geographic Information System/Sistem Informațional Geografic)**. Un sistem GIS este format din resurse de calcul (calculatoare, servere etc.), programe (software) și operatori care permit înregistrarea, stoarea și analiza informațiilor de natură geografică. Un sistem GIS creează, actualizează și utilizează hărți geografice și topografice cu foarte multe informații asociate.

Imaginați-vă harta lumii (globul pământesc – *în imaginea a*) în format electronic, cu toate informațiile geografice pentru fiecare punct (latitudine, longitudine, altitudine) și cu o multitudine de alte date atașate pentru: oceane, mări, lacuri, ape curgătoare, ghețari, păduri, terenuri arabile, plaje, deșerturi, orașe, clădiri, străzi, căi ferate, localizarea și densitatea populației, resursele naturale din sol și subsol și multe altele. Există programe dedicate care pot afișa doar ceea ce avem nevoie la un moment dat din toate informațiile și date, pot face simulări, studii statistice sau permit chiar introducerea de noi date, date proprii obținute în urma unor cercetări care îmbogățesc sistemul și oferă tuturor posibilitatea de a beneficia de noile descoperiri.



Sistemele GIS sunt utilizate în diverse domenii de activitate: în cercetare, în scopuri științifice, în managementul afacerilor (în luarea unor decizii), în gestiunea bunurilor și a stocurilor, în cartografie, în managementul resurselor naturale, în realizarea planurilor de evacuare în cazul unor dezastre, în turism, în agricultură, în planificarea și conservarea resurselor naturale (de exemplu, în silvicultură), în analiza poluărilor etc. Chiar și biologii folosesc sisteme GIS pentru înregistrarea și analiza rutelor de migrație ale diferitelor păsări și animale.

Sunt, de asemenea, utilizate în aviație pentru diferite scopuri, ca dirijarea traficului aerian sau simularea rutelor de zbor integrate cu datele de înălțime. Sunt folosite (știți bine asta) la automobile pentru a ne ajuta să ajungem dintr-un loc în altul, pentru monitorizarea vitezei de deplasare și chiar pentru imobilizarea mașinii la trecerea frontierei în cazul în care nu sunt îndeplinite anumite condiții (bariere virtuale).

#### Retineti

**GIS** este un sistem care se bazează pe echipamente electronice de calcul (hardware) și pe programe (software) pentru a stoca, prelucra și a pune la dispoziția oamenilor date geografice și informații adiacente.

#### Istoria apariției Sistemelor Informaționale Geografice (GIS)

Primele sisteme GIS au fost aplicații de cartografie asistată de calculator și au fost realizate în perioada 1960-1970.

Primul sistem GIS cu adevărat operațional a fost creat în Canada, în anul 1967, de către Departamentul Pădurilor și al Dezvoltării Rurale pentru stocarea și analizarea informațiilor din programul Canada Land Inventory, dedicat stabilirii posibilităților de dezvoltare a zonei rurale canadiene prin cartografierea vegetației, solurilor, surselor de apă, folosinței terenurilor, ariilor protejate etc. Geograful Roger Tomlinson, dezvoltatorul acestui sistem informatic, este considerat creatorul GIS.

Primul pachet de programe GIS comerciale a fost lansat în 1982 (ESRI ARC/INFO 1.0).

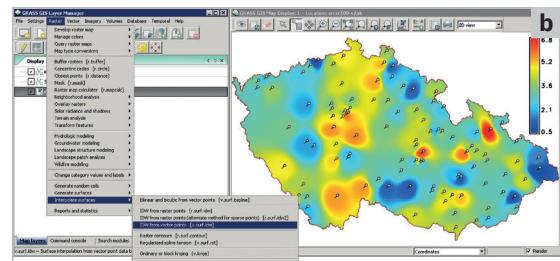
Începând cu anul 2004, au fost lansate mai multe programe din categoria GIS: versiunea GRASS 6.0, cu o nouă interfață utilizator și cu suport pentru baze de date; Google Maps (bazat pe tehnologii Web-GIS) și Google Earth.

### Exemple de Sisteme Informaționale Geografice (GIS)

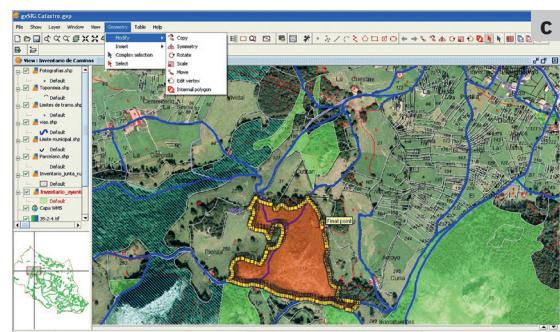
 S-au dezvoltat și se dezvoltă în continuare o multitudine de sisteme GIS. Să enumerez câteva dintre ele:

- **GRASS GIS** (în imaginea b) – sistem dezvoltat inițial de U.S. Army Corps of Engineers (Corpul de Ingineri al Armatei Statelor Unite);
- **gvSIG** (în imaginea c) – sistem GIS multiplatformă care funcționează pe diferite sisteme de operare (Linux, Unix, Mac OS X, Windows);
- **ILWIS** (Integrated Land and Water Information System – Sistem Integrat de Informații pentru Pământ și Ape) – sistem care integrează imagini, date și informații tematică;
- **JUMP GIS** – sistem care încorporează mai multe componente (aplicația OpenJUMP, SkyJUMP, deeJUMP și Kosmo), toate inițiate de compania JUMP;
- **MapWindow GIS** – aplicație gratuită pentru calculatoare personale, aplicație care este însotită de câteva componente de programare;
- **QGIS** (anterior cunoscută sub numele de **Quantum GIS**) – un alt sistem GIS care funcționează pe mai multe sisteme de operare: Linux, Unix, Mac OS X și Windows;
- **SAGA GIS** (System for Automated Geoscientific Analysis) – un software GIS hibrid care se remarcă printr-un număr mare de metode geoștiințifice pe care le conține.

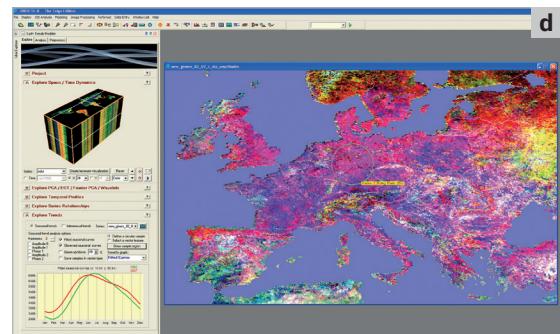
Să vedem câteva capturi de ecrane din diferite sisteme GIS. Observați cum sunt reprezentate hărțile geografice.



GRASS GIS 6.4



gvSIG 1.0 (imagine: Emilio Gómez Fernández)



IDRISI Taiga 16.05 (imagine: Shankar Dayal)

### Google Maps – sistem de Web mapping (Web GIS). Interfață

Operațiunea de cartare reprezintă, conform *Dicționarului explicativ al limbii române*, operațiunea de urmărire pe teren și transpunere, prin semne și prin culori convenționale, pe hărți topografice, a diferitelor elemente din natură (roci, formațiuni geologice, ape, soluri, vegetație, faună etc.). Această operațiune stă la baza realizării hărților.

O aplicație de Web mapping permite realizarea acestei operațiuni de cartare în mod colaborativ, pentru realizarea și întreținerea hărților puse la dispoziție de către sistemele GIS pe Internet. Întreținerea hărților, adăugările sau actualizările se pot realiza fie de către deținătorul acestei aplicații, fie de către utilizatorii acestia. Termenul *Web GIS* devine similar cu cel de *Web mapping*.

Google Maps este, de fapt, un sistem pentru vizualizarea informațiilor geografice, dar care nu permite utilizatorilor să facă multe dintre lucrurile pe care un Sistem Informațional Geografic profesional le poate face. Este restrâns la ceea ce folosește în mod ușor, dintr-un sistem GIS, majoritatea oamenilor care nu sunt specialiști.

 La ce folosim cu toții, cel mai frecvent, Google Maps? Mulți dintre noi utilizăm zilnic aplicații de tipul Google Maps, iGo, HERE WeGO, Waze sau altele pentru a ajunge la o destinație nouă, necunoscută, sau pentru a localiza un obiectiv turistic, de exemplu. Așadar, dacă oferim o explicație simplă, Google Maps pune pe hartă adrese și obiective turistice, economice, geografice etc. Un exemplu concret și ușor de imaginat referitor la aplicabilitatea Google Maps și a programelor similare este planificarea unei noi rute de călătorie. Doar tastăm cele câteva destinații pe care vrem să le atingem și programul (sistemul) ne va propune un traseu optim în funcție de mai mulți parametri setați de noi.

 Dar Google Maps mai face ceva extraordinar: vă permite să faceți tururi virtuale oriunde pe glob. De căte ori aveți o anumită destinație în minte, puteți să o căutați pe harta pusă la dispoziție de Google Maps și să o vedeați până la nivel de amânat, împreună cu împrejurimile sale, ca și cum ați trăi în zona respectivă.

Google Maps este probabil cea mai răspândită dintre platformele GIS pentru publicul larg. Deși nu este neapărat cel mai bun instrument pentru vizualizarea complexă a datelor, este o platformă extrem de robustă și ușor de utilizat, inclusiv pe dispozitivele mobile, și este foarte potrivită pentru simularea rutelor și a timpilor de călătorie.



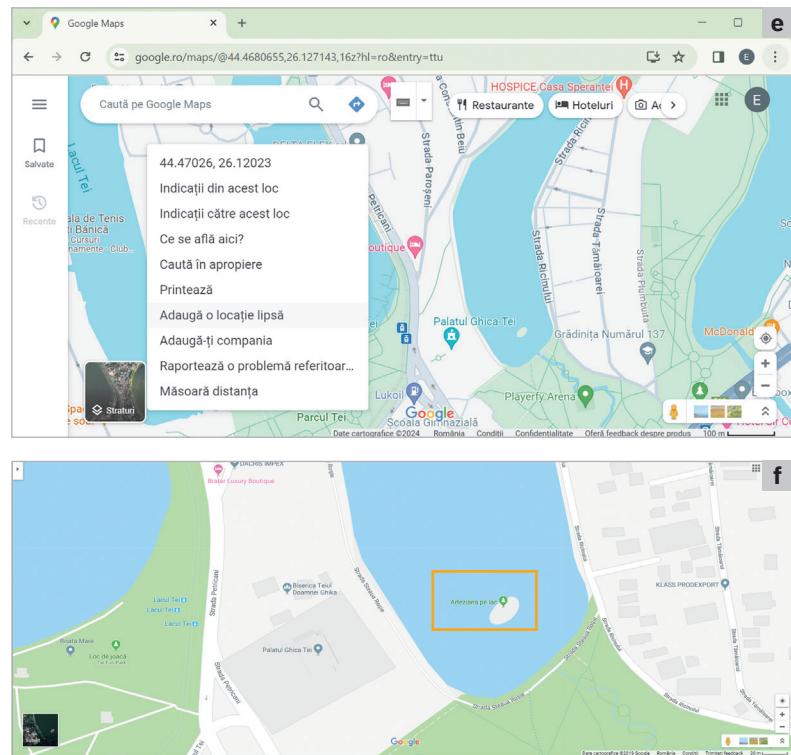
## Aplicații și investigații

- Să vedem cum puteți introduce un nou element, o nouă locație pe Google Maps (un element de identitate locală – școală voastră, gara din localitatea în care domiciliați, primăria, un teatru sau o casă de cultură, un monument, o piață publică, un parc sau orice altceva credeți că ar putea fi de interes pentru voi și pentru colegii și prietenii voștri, sau ar fi de interes public și/sau turistic).

Mai întâi, deschideți un browser de Internet și intrați pe Google Maps ([www.google.ro/maps](http://www.google.ro/maps)). Apoi, conectați-vă la contul vostru Google.

Localizați pe harta Google cât mai precis locația pe care vreți să o adăugați (pentru aceasta, măriți cât este necesar pentru o poziționare precisă pe hartă). Faceți click cu butonul drept pe locația aleasă (pe ecranele tactile apăsați și țineți puțin apăsat), iar din meniu care se deschide alegeți opțiunea **Adăugați o locație lipsă** (*imaginea e*).

În fereastra care se deschide trebuie să introduceți numele locației pe care doriți să o introduceți și categoria în care se încadrează (magazin, spital, benzinărie, bancă, bancomat, muzeu, cinematograf, parc public etc.). În câmpul următor, în care puteți vedea o mică parte de hartă, puteți face o poziționare mai precisă, în cazul în care nu ați reușit acest lucru de la început. Puteți adăuga chiar și o fotografie pentru această locație. Când ați terminat, apăsați butonul **Trimite**. O să dureze o vreme (în principiu, până la 24 de ore) până ce Google aproba și activează locația adăugată de voi, dar în scurt timp toată lumea va putea vedea respectiva locație. În imaginea de mai sus se observă locația noastră aprobată și afișată de Google după 12 ore (*imaginea f*).



## Știați că...?

<https://thetruesize.com> este un site interesant, prin intermediul căruia puteți localiza orice țară pe o hartă Google Maps și îi puteți vedea dimensiunile reale. Căutați o țară, folosiți mouse-ul pentru a o deplasa într-o altă zonă a globului și veți vedea că dimensiunea sa este distorsionată odată cu diferența de latitudine. Aceasta din cauză că o proiecție pe un plan al globului pământesc, care este o sferă, forțează zonele să distorsioneze, pe măsură ce latitudinea se apropie de poli. Vedeți cât de interesant este faptul că prin tragerea Canadei la ecuator, ea devine aproape de mărimea Braziliei.



## Exesăti

- Dintre elementele enumerate mai jos, scrieți-le în caiete pe cele care intră în componența unui sistem GIS:
  - a. resurse de calcul;
  - b. programe (software);
  - c. operatori;
  - d. date și informații.
- Când a fost creat primul sistem GIS cu adevărat operațional? De către cine?
- Ce este și la ce se folosește un sistem GIS? Dați un exemplu de utilizare practică în viața de zi cu zi.
- Proiect.** Plimbați-vă prin zona din jurul casei (pe o zonă cu raza de 1 km), pe jos sau cu bicicleta, și adăugați pe Google Maps toate cuiburile de barză/magazinele pe care le găsiți. Așteptați să fie avizate și să devină disponibile toate elementele adăugate, apoi formați grupuri de lucru și analizați și comparați ce elemente a adăugat fiecare, precizia localizării și multitudinea amănuntelor din descriere, precum și eleganța descrierii.



## Portofoliu

Căutați pe Internet informații și imagini legate de evoluția aparatelor GPS pentru publicul larg din ultimii 20 ani. Realizați o prezentare cu ceea ce ați găsit interesant.

# Exerciții recapitative

Rezolvați pe caiete sarcinile de lucru de mai jos.

1. Ce este un hacker? Care este diferența dintre un hacker și un cracker?
2. Abrevierea GIS vine de la:
  - a. Geometrical Information System;
  - b. Geographic Information System;
  - c. Gimnazial Infrastructure System.
3. Realizați, în caiete, corespondența corectă a termenilor din tabelul de mai jos:

Google Apps	Microsoft Office
1. Docs	a. Excel
2. Slides	b. Word
3. Sheets	c. Power Point

4. Transcrieți, în caiete, afirmațiile adevărate referitoare la Google Slides din lista de mai jos.

- a. Folosind Google Slides se pot crea și edita prezentări multimedia.
- b. Pentru a salva modificările făcute într-o prezentare, trebuie apăsat butonul **Save**.
- c. Mai multe persoane pot lucra simultan pe același fișier, folosind Google Slides.
- d. Fișierele Power Point pot fi convertite în prezentări Google și invers.
- e. Un fișier Google Slides poate fi accesat direct de pe laptop, tabletă sau telefon intelligent.

5. Prezentați două avantaje practice ale utilizării unei aplicații colaborative.

6. Scrieți, în caiete, DA sau NU pentru fiecare dintre afirmațiile de mai jos.

- a. Un document lucrat în Google Drive poate fi salvat local pe calculatorul nostru.
- b. Pe Google Drive se pot crea foldere și subfoldere.
- c. Se poate colora capul de tabel al unui tabel dintr-un document făcut cu Google Docs.

DA / NU  
DA / NU  
DA / NU

7. Transcrieți textul următor în caiete, completând spațiile libere:

Aplicațiile care le permit utilizatorilor să lucreze ..... , în timp real, pe un același ..... , având posibilitatea să renunțe la ..... făcute sau să le salveze, poartă numele de aplicații ..... pentru editare în .....

8. Se poate vedea istoricul modificărilor unui document redactat în Google Docs? Dacă da, cum?

## Autoevaluare

Timp de lucru: **10 minute**

Rezolvați, în caiet, exercițiile date. Evaluați-vă singuri rezultatele.

1. Care este meniul care nu apare în bara de meniu a aplicației Google Docs?
  - a. Fișier;
  - b. Editează;
  - c. Afisează;
  - d. Șterge;
  - e. Inserează;
  - f. Formatează.
2. Dați 3 exemple de domenii de activitate în care sunt folosite sisteme GIS sau Web GIS.
3. Realizați o prezentare în Google Slides cu titlul „Localitatea mea natală – prezentare”. Prezentarea va conține cel puțin cinci diapozițive. Apoi dați drept de editare unui coleg de clasă și rugați-l să insereze și el cel puțin două diazozi în prezentare.



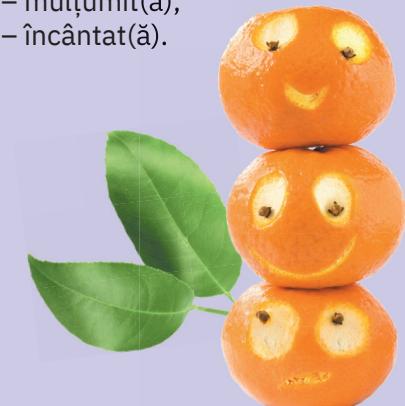
### CUM VĂ APRECIAȚI?

- 1 Desenați ● în dreptul sarcinilor pe care le-ați rezolvat corect.



- 2 Scrieți cum vă simțiți după ce ați rezolvat acest test, alegând unul dintre cuvintele de mai jos:

- nemulțumit(ă);
- mulțumit(ă);
- încântat(ă).



# Recapitulare sumativă TIC

U3

Toate cerințele se rezolvă în caiete.

1. Căutați cele opt cuvinte care au legătură cu operațiile de procesare computerizată de texte, ascunse între literele din careul de mai jos. Atenție! Cuvintele sunt scrise atât orizontal, cât și vertical.

A	O	N	E	D	R	I	V	E	N	S	I	C	B	T
E	B	U	S	U	D	O	C	U	M	E	N	T	R	I
X	C	A	D	L	O	E	S	I	A	D	X	U	I	P
O	E	K	O	W	Y	P	I	K	J	I	A	S	J	A
T	E	H	N	O	R	E	D	A	C	T	A	R	E	R
Q	S	L	I	R	J	O	W	A	F	A	F	E	W	I
C	L	O	U	D	O	Z	A	N	Y	R	Q	H	T	R
I	O	U	Q	W	U	J	T	O	T	E	X	T	I	E

2. Creați un document Word în care să descrieți cele mai utilizate zece opțiuni din meniurile aplicației VideoPad, pe care le-ați folosit voi în procesul de editare a unei secvențe video. Fișierul să fie redactat cu font Arial, cu dimensiunea de 14, scris la un rând și jumătate și să aibă inserate imagini, câte una pentru fiecare opțiune de meniu.
3. Microsoft Word este un procesor de text. Pe calculatorul pe care lucrăți voi, mai este disponibil un alt procesor de text? Dacă da, care este numele lui?

4. Deschideți MS Word și redactați textul următor:

*Îmi place cum râzi și cum zâmbești mereu,  
Când ochii tăi privesc lumina cerului meu.  
În sufletul meu, să știi, ești mereu prezent,  
Iar înima mea te iubește sincer și vehement.*

## POEZIE DE DRAGOSTE

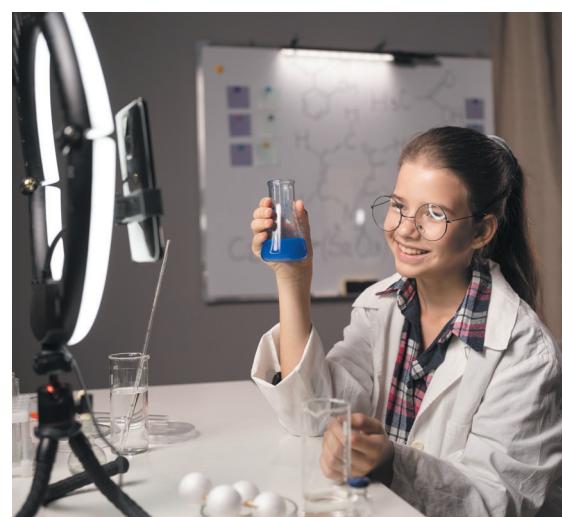
- Răspundeți la întrebarea: cum ați făcut să scrieți ultimul rând numai cu litere mari?
- Descrieți două modalități de a scrie cu majuscule.
- Mutăți ultimul rând al textului redactat mai sus, pe primul rând, ca titlu.
- Centrați și îngroșați textul titlului.
- În partea dreaptă a textului inserați o imagine potrivită cu tema textului.



5. Luați un fișier de sunet, editați-l în WavePad și măriți-i volumul sonor cu 5 decibeli. Apoi inserați o secundă de linie la secunda 45 a înregistrării.

6. În caiete, trasați câte o linie pentru fiecare element din coloana din stânga pentru a-l uni cu elementul care îi corespunde din coloana din dreapta. Exercițiul trebuie rezolvat într-un document Word, iar liniile le trageți cu ajutorul instrumentelor din fila „Desenare”.

Extensia fișierului	Tipul de fișier
.doc	fișier de sunet
.wav	fișier video
.jpg	fișier de sunet
.mp3	fișier video
.mp4	fișier de text
.txt	fișier imagine
.avi	fișier de text
	fișier imagine



7. Realizați un film de cel puțin trei minute cu un experiment de fizică sau de chimie. Inserați cel puțin cinci texte pe parcursul filmului, care să explice pașii experimentului. „Condimentați-l” și cu câteva efecte sonore!

# Evaluare sumativă TIC

Toate cerințele se rezolvă în caiete.

1. Adevărat sau fals? Verificați fiecare dintre afirmațiile de mai jos.
  - a. Documentele scrise în limba română trebuie să conțină diacritice. A / F
  - b. La finalul unui rând nu se apasă tasta **Enter** decât dacă se dorește începerea unui paragraf nou. A / F
  - c. Semnele de punctuație (punct, virgulă, punct-și-virgulă, două-puncte, semnul întrebării, semnul exclamării) se pun lipite de ultimul caracter de dinainte. A / F
  - d. Nu se lasă niciun spațiu înainte de parantezele de închidere și de ghilimele de închidere, ci se lasă un spațiu după acestea. A / F
  - e. Înainte de ghilimele de deschidere și de parantezele de deschidere se lasă un spațiu, însă nu se lasă niciun spațiu după ele. A / F
  - f. Liniuța de unire „-” și bara oblică „/” se pun lipite de ambele cuvinte din stânga și din dreapta. A / F
  - g. Între două cuvinte se lasă întotdeauna două spații. A / F
  - h. Între două paragrafe succesive nu este bine să lăsăm niciun spațiu. A / F

2. Într-un document tehnoredactat computerizat se poate folosi funcția de căutare pentru a regăsi rapid anumite informații?

- a. da;
- b. nu;
- c. depinde de cuvintele căutate.

3. Care sunt modalitățile principale de vizualizare a unui document deschis în Word?

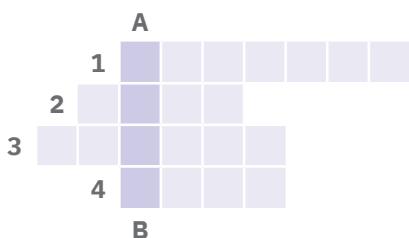
4. Ce utilitate are tasta **Spațiu** la redarea fișierelor audio în programul WavePad?

5. Enumerați cel puțin patru dintre operațiile de bază ale unei aplicații dedicate înregistrării și editării fișierelor audio.



6. Există mai multe sisteme GIS dezvoltate de-a lungul anilor? Dacă da, dați trei exemple de astfel de sisteme.

7. Completând rebusul următor, veți obține pe coloana **AB** numele unui tip de marcat de text (în limba engleză).



1. Folosite la liste, pentru a evidenția fiecare element (în limba engleză).
2. Una dintre cele mai populare aplicații procesor de text.
3. Căsuță care se formează la intersecția unui rând cu o coloană, într-un tabel.
4. Format de fișier WORD.

8. Ce este o aplicație colaborativă? Dați două exemple de astfel de aplicații.

9. Explicați, pe scurt, cum puteți introduce o locație nouă pe harta Google.

10. Enumerați minimum trei avantaje ale utilizării aplicațiilor de tip colaborativ.

## Punctaje

1 – 5 puncte;	6 – 10 puncte;
2 – 5 puncte;	7 – 10 puncte;
3 – 10 puncte;	8 – 10 puncte;
4 – 10 puncte;	9 – 10 puncte;
5 – 10 puncte;	10 – 10 puncte.

Se acordă 10 puncte din oficiu.

## Punctaj maxim: 100 de puncte

- Insuficient: 0-49 de puncte
- Suficient: 50-69 de puncte
- Bine: 70-89 de puncte
- Foarte bine: 90-100 de puncte

Timp de lucru: stabilit de cadrul didactic

# U4

## Limbajul de programare



Lecția 1	64-67	Mediu de dezvoltare Code::Blocks
Lecția 2	68-73	Limbajul C++. Elemente de bază
Lecția 3	74-76	Structura liniară
Lecția 4	77-80	Structura alternativă (de decizie)
Lecția 5	81-85	Structura repetitivă condiționată anterior
Lecția 6	86-89	Structura repetitivă condiționată posterior
Lecția 7	90-95	Structura repetitivă cu contor (cu număr cunoscut de pași)
Lecția 8	96-99	Aplicații de fizică în C++
Lecția 9	100-102	Aplicații de geografie în C++
Concurs	102	Micii programatori
Lecția 10	103-105	Programarea roboților virtuali. Utilizarea senzorilor
Exerciții recapitulative	106	
Autoevaluare	106	

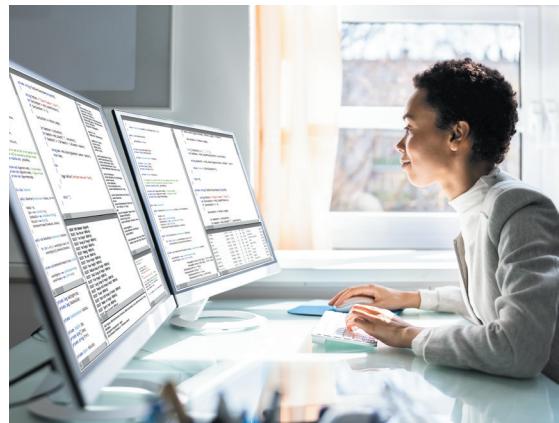
# 1 Mediul de dezvoltare Code::Blocks

## Amintiți-vă

În anii anteriori ati învățat să realizați programe în limbajul vizual de programare Scratch. Un program se realiza aici prin îmbinarea blocurilor de comenzi asemenea unui puzzle.

## Descoperiți

În cele ce urmează vom învăța să scriem programe într-un alt limbaj de programare, foarte puternic, numit C++, și să folosim pentru aceasta un mediu integrat de dezvoltare (IDE – Integrated Development Environment) numit **Code::Blocks**.



### Etape pentru dezvoltarea unei aplicații

Limbajul C este precursorul limbajului C++. El a fost dezvoltat, în anul 1972, de către Dennis Ritchie la laboratoarele Bell Telephone ca un limbaj de programare pentru scrierea de sisteme de operare. În anul 1979, a fost dezvoltat C++ de către Bjarne Stroustrup ca o extensie a limbajului C, acesta adăugând noi caracteristici/funcționalități (*features*), dintre care amintim programarea pe obiecte. Pentru a înțelege cum se execută un program C++ este necesar să înțelegem care sunt etapele pentru dezvoltarea sa (*îmaginea a*).

**1. Analiza problemei de rezolvat și proiectarea soluției** – la acest pas se identifică datele de intrare/ieșire, datele de manevră, se realizează fie algoritmul în pseudocod, fie schema logică a programului.

**2. Editarea codului sursă al programului** – codul sursă reprezintă, de fapt, algoritmul de mai sus, scris într-un limbaj de programare; pentru realizarea acestui pas se utilizează un **editor** în care se scrie efectiv codul. Codul sursă se salvează în fișiere cu extensia **.cpp** (această extensie indică faptul că este vorba despre un fișier C++). Programele mai complexe pot avea mai multe fișiere **.cpp**.

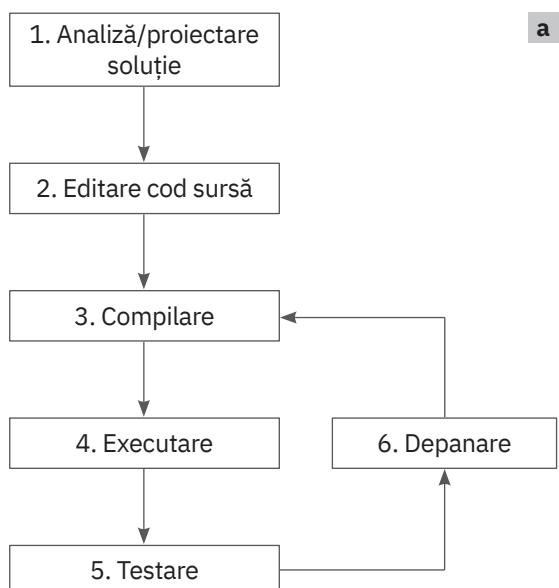
Ne amintim că, la programele Scratch, editarea codului presupunea de fapt tragerea blocurilor de comandă în zona de program și îmbinarea acestora, iar programele aveau extensia **.sb3**.

**3. Compilarea programului** – pentru realizarea acestui pas se utilizează un **compilator**. În primul rând, se verifică dacă sunt erori de sintaxă (dacă se respectă regulile de scriere ale limbajului C++); compilarea nu merge mai departe până când acestea nu sunt rezolvate. Apoi, se convertește codul sursă al programului într-un cod intermediar specific compilatorului, numit **cod obiect**. Dacă programul are mai multe fișiere sursă (**.cpp**), vor rezulta mai multe fișiere obiect. Fișierele obiect au același nume ca fișierele **.cpp** din care provin și au extensia **.o** sau **.obj**. Urmează o operație care se numește **link-editare** și care leagă codurile obiect cu bibliotecile de funcții, rezultând un fișier executabil cu extensia **.exe**.

**4. Executarea programului** – pornirea, lansarea în execuție sau rularea programului. La programele Scratch, acest pas presupunea apăsarea stegulețului verde.

**5. Testarea programului** – la acest pas se realizează parcurgerea efectivă a programului, cu scopul de a identifica posibilele probleme (abateri de la funcționarea dorită). Pentru o eroare care apare într-un program s-a întărit utilizarea termenului **bug** („gândac”, „insectă”, în limba română).

**6. Depanarea programului** – aici toate erorile găsite sunt rezolvate, iar programul este pus din nou în funcțiune, este compilat, executat, testat și apoi depanat până când sunt eliminate toate erorile. Termenul în limba engleză pentru realizarea depanării programului este **debug**.



## Ştiați că...?



Thomas Edison a fost cel care a utilizat pentru prima oară termenul **bug** în anii 1870. În anul 1947, niște ingineri care lucrau la calculatorul Mark II al Universității Harvard au găsit o molie reală blocată în calculator, aceasta provocând un scurtcircuit. Molia buclucașă și carnetul în care s-a raportat eroarea „Primul caz real de bug (eroare) găsit(ă)” (*First actual case of bug being found*) se găsesc acum la muzeul american Smithsonian.

## Mediul de dezvoltare Code::Blocks



### Retineti

Pentru realizarea pașilor 2, 3, 4 și 6 se pot utiliza programe software diferite. **Un mediu integrat de dezvoltare (IDE) conține toate facilitățile necesare pentru dezvoltarea unui program (editare, compilare, depanare)** și este mult mai ușor de utilizat.

## Mediul de dezvoltare integrat Code::Blocks

Code::Blocks este un mediu de dezvoltare integrat (IDE) care poate rula pe Windows, pe Linux și pe Mac OS.

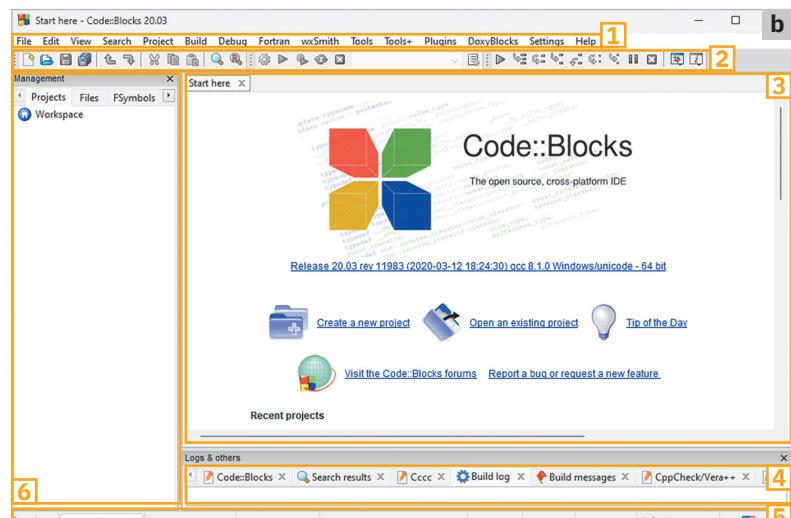
### A. Pornirea Code::Blocks

Lansarea în execuție a aplicației se face din meniul **Start**, accesând: **All Apps -> Code::Blocks**, sau se apasă **Start**, apoi se tastează **Code::Blocks** și se apasă **Enter** când să apăseze să fie afișat sigla și numele programului.

### B. Elemente de interfață Code::Blocks

Se deschide o interfață care conține mai multe zone (*imagină b*).

- 1. Zona de meniuri** – aici apar toate meniurile aplicației;
- 2. Bara de unelte** – conține butoane pentru diverse comenzi necesare pentru gestionarea fișierelor, pentru procesul de dezvoltare a programelor, pentru depanare (debug);
- 3. Zona de editare a codului sursă** – aici, doar la pornirea aplicației Code::Blocks apar butoanele de creare/deschidere de proiect, în rest apare codul sursă al programelor;
- 4. Zona de afișare a diferențelor observații** legate de diverse etape în dezvoltarea programului;
- 5. Bara de status** – aici se afișează diverse informații, dintre care enumerăm câteva mai importante: calea unde se află fișierul selectat, tipul aplicației, linia curentă din codul sursă;
- 6. Zona de management** – aici se află mai multe file, printre care și fila **Projects** (*Proiecte*) în care se pot vedea proiectele deschise în spațiul de lucru (*Workspace*).



### C. Facilități ale Code::Blocks pentru dezvoltarea unui program C++

Pentru a scrie un program într-un mediu de dezvoltare integrat (IDE) este necesar să fie creat un *project*. Proiectul este ca un container pentru toate fișierele care compun programul nostru. Fiecare proiect corespunde unui program.

#### a. Crearea/deschiderea unui proiect

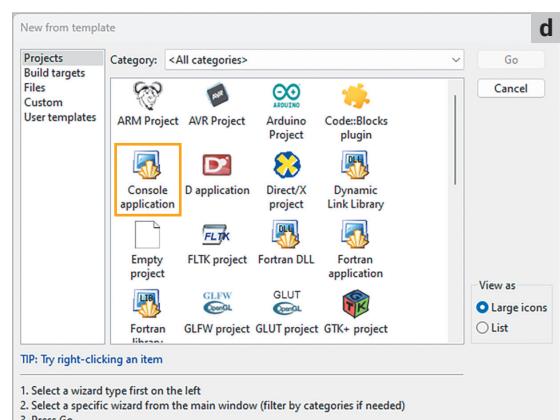
La pornirea aplicației Code::Blocks există următoarele posibilități:

- **crearea unui proiect nou** – se apasă **Create a new project** (*Crearea unui proiect nou*) care apare direct pe interfață (*imagină c*, stânga) sau se selectează comanda **File -> New -> Project... ;**
- **deshiderea unui proiect existent** – se apasă **Open an existing project** (*Deschiderea unui proiect existent*) care apare direct pe interfață (*imagină c*, dreapta) sau se selectează comanda **File -> Open... .**



Pentru crearea unui proiect nou, cu numele **PrimulProiect**, se parcurg următorii pași:

1. În fereastra care se afișează la inițierea creării unui proiect nou, se alege tipul de proiect **Console application** (*Aplicație consolă*) și se apasă **Go** (*Pornire*) (*imagină d*).



Implicit, aplicațiile consolă nu au interfață grafică pentru utilizator (*Graphical User Interface – GUI*), tipăresc textul direct pe monitor, citesc date de la tastatură, complexitatea este minimă și permit învățarea.

2. Se afișează o nouă fereastră în care se alege tipul aplicației C++, apoi se apasă butonul **Next** (*Înainte*).

**3.** Pasul următor este afişarea unei ferestre (*imaginea e*) în care se introduc:

- titlul proiectului (numele): în căsuța **Project title** (*Titlul proiectului*);
- folderul în care va fi salvat proiectul în căsuța **Folder to create project in** (*Folderul în care se creează proiectul*).

Automat, proiectul va fi plasat într-un folder care are numele similar cu titlul proiectului. Fișierul proiect va avea extensia **.cbp**. Se apasă apoi butonul **Next (Înainte)**.

**4.** Fereastra afişată la acest pas conține informații despre compilator și folderele în care vor fi create versiunile de program pentru depanare (**Debug**) și productiv (**Release**). Se apasă butonul **Finish (Finalizare)**.

În acest moment s-a creat proiectul nostru cu numele **PrimulProiect**. La crearea sa, un proiect este adăugat automat unui spațiu de lucru (**Workspace**). Acesta este un container care poate conține mai multe proiecte la un loc. El poate fi văzut în zona de management a proiectelor (*imaginea f*).

#### b. Editarea codului sursă

Fișierul care conține codul sursă este **main.cpp**, el poate fi accesat dacă în fereastra de Management se expandează folderul **Sources** al proiectului (click pe „+”-ul de lângă el). La selectarea fișierului **main.cpp** în panoul de management din stânga ecranului, codul sursă apare în zona de editare a codului și aici se poate scrie în continuare programul (*imaginea g*).

Editorul pus la dispoziție de Code::Blocks colorează diferit variate părți ale programului pentru a se citi mai ușor codul și adaugă numere la linile de cod pentru a putea identifica ușor o linie la care apare o eroare.

Comenziile clasice utilizate la editarea unui document se află în meniul **Edit (Editare)** (*imaginea h*) sau în bara de unelte: 1. **Undo** (Renunțare la modificări); 2. **Redo** (Refacere a modificărilor); 3. **Clear Change History** (Șterge istoria modificărilor); 4. **Cut** (Decupare); 5. **Copy** (Copiere); 6. **Paste** (Lipire).

#### c. Compilarea proiectului

Se realizează prin selectarea comenzii **Build** din meniul **Build**, sau se apasă **Ctrl+F9** ori butonul **Build** din bara de unelte. Inițierea acestei operații are ca urmare compilarea și link-editarea programului, adică transformarea codului sursă direct în fișier executabil **.exe**. Se poate realiza și o operație numită **Rebuild**, care șterge toate fișierelor create la operația anterioară și reface fișierul executabil. Pentru aceasta, se selectează **Rebuild** din meniul **Build** sau se apasă **Ctrl+F11** ori butonul **Rebuild** din bara de unelte.

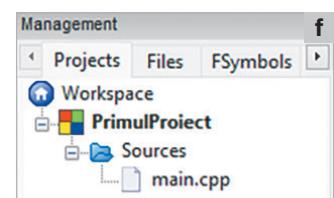
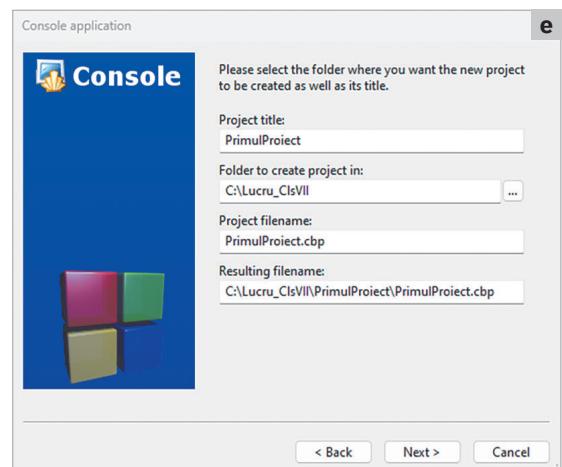
#### d. Rularea (executarea) proiectului

În cazul în care s-a realizat anterior etapa de compilare a programului, lansarea în execuție a programului se realizează fie prin selectarea comenzii **Run** din meniul **Build**, fie se apasă **Ctrl+F10** ori butonul **Run** din bara de unelte. În cazul în care nu s-a realizat etapa de compilare, putem selecta direct comanda **Build and run** din meniul **Build**: fie se apasă **F9**, fie se apasă butonul **Build and run** din bara de unelte.

#### e. Depanarea programului

Pentru a putea depana un program, el trebuie lansat în execuție într-un mod **Debug (Depanare)**, apoi se utilizează comenzi existente în meniul cu același nume sau cele corespondente de pe bara de unelte. Pentru a descoperi o eroare, un bug, este important să putem urmări pas cu pas ce se întâmplă în program, să vedem valorile variabilelor și pe ce fir de execuție merge programul la un moment dat. Sunt importante următoarele acțiuni:

- Adăugarea unui punct de oprire în program (breakpoint)** la o anumită linie de cod – până la acel punct programul rulează fără oprire, urmând ca în continuare, de la acel punct încolo, să preia controlul cel care face depanarea. Se face astfel: ne poziționăm la linia unde dorim să punem un punct de întrerupere și accesăm comanda **Toggle breakpoint** (*Pozitionare punct de întrerupere*) din meniul **Debug** sau apăsăm **F5**. Va apărea un punct mare, roșu în dreptul acelei linii.
- Ștergerea unui punct de oprire în program** – ne poziționăm pe linia de cod unde vrem să nu mai avem breakpoint și, fie accesăm comanda **Toggle breakpoint** (*Pozitionare punct de întrerupere*) din meniul **Debug**, fie apăsăm **F5**. De asemenea, un simplu click pe punctul roșu care apare în dreptul liniei îl va șterge.



## Mediul de dezvoltare Code::Blocks

3. Câteva dintre cele mai uzuale **comenzi pentru parcurgerea programului** în mod **Debug (Depanare)** sunt:

Comandă	Descriere	Accesare
<b>Start (Pornire)</b>	Realizează pornirea programului în mod <b>Depanare</b> atunci când nu este deja în acest mod.	Din meniu <b>Debug -&gt; Start/Continue</b> sau la apăsarea <b>F8</b> ori a butonului corespunzător din bara de unelte.
<b>Continue (Continuare)</b>	Odată fiind într-o sesiune <b>Depanare</b> , această comandă face să fie continuată execuția normală a programului de la un anume punct încolo.	Din meniu <b>Debug -&gt; Start/Continue</b> sau la apăsarea <b>F8</b> ori a butonului corespunzător din bara de unelte.
<b>Run to cursor (Mergi la cursor)</b>	Execută programul până când se ajunge la linia pe care este poziționat cursorul mouse-ului.	Din meniu <b>Debug -&gt; Run to cursor</b> sau la apăsarea <b>F4</b> ori a butonului corespunzător din bara de unelte.
<b>Next line (Linia următoare)</b>	Trece să execute următoarea linie a programului pe firul normal de execuție.	Din meniu <b>Debug -&gt; Next line</b> sau la apăsarea <b>F7</b> ori a butonului corespunzător din bara de unelte.
<b>Stop debugger (Oprire depanator)</b>	Oprește execuția programului în mod <b>Depanare</b> .	Din meniu <b>Debug -&gt; Stop debugger</b> sau la apăsarea <b>Shift+F8</b> ori a butonului corespunzător din bara de unelte.

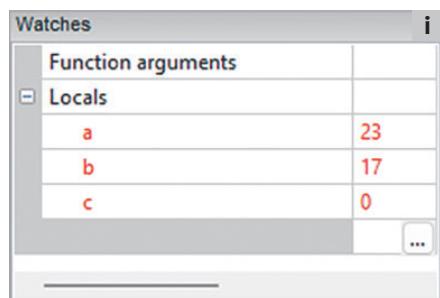
4. Urmărirea valorilor **variabilelor** se realizează apelând comanda **Debugging Windows (Ferestre pentru depanare) -> Watches (Urmărire)** de pe bara de unelte (*imaginea i*).

### f. Salvarea fișierelor unui proiect

Se pot salva fișierele care compun un proiect, fiecare în parte, cu același nume și tip, apelând comanda **File (Fișier) -> Save file (Salvare fișier)** sau apăsând butonul de pe bara de unelte. Dacă se dorește salvarea unui fișier cu alt nume sau tip, se accesează comanda **File (Fișier) -> Save file as... (Salvare fișier ca...)**.

### g. Salvarea proiectului

Un proiect se salvează, fără a-i se schimba numele, prin selectarea comenzii **Save project (Salvare proiect)** din meniu **File (Fișier)**. Comanda **Save project as... (Salvare proiect ca...)** se apelează pentru a salva proiectul cu un alt nume.



## Aplicați și investigați

- Creați un folder de lucru **Proiecte C++** în care veți salva proiectele voastre de aici încolo. Realizați în ordine pașii **a**, **b**, **c** și **d** pentru a crea și rula primul vostru proiect în C++. Salvați proiectul. Observați ce mesaj vă apare pe ecran după execuțarea pasului **d**. Este mesajul primului vostru program în C++, „Hello world!” (*Salutare lume!*). Salvați din nou proiectul.
- Adăugați un punct de întrerupere (*breakpoint*) în programul vostru, la linia 7. Porniți apoi execuția programului în mod **Debug (Depanare)** și treceți la linia următoare. Priviți la fiecare pas ce se întâmplă pe ecran.

## Exersați

- În termenul „IDE”, ce reprezintă litera „D”?
- Explicați ce presupune etapa de compilare a unui program scris în limbajul C++.
- Dați trei exemple de comenzi care se pot utiliza pentru editarea unui program scris în C++.
- Realizați, în caiete, asocierea corectă între butoanele de comandă și descrierea lor:

- |   |  |
|---|--|
| 1 | a Pornirea programului în mod <b>Depanare</b>          |
| 2 | b Salvarea fișierului curent                           |
| 3 | c Lipire   |
| 4 | d În modul <b>Depanare</b> se trece la linia următoare |
| 5 | e Lansarea în execuție a programului                   |

- Explicați pentru ce se utilizează modul **Depanare (Debug)** și cum poate fi urmărită valoarea unei variabile utilizate în program.



## Portofoliu

Realizați o prezentare PowerPoint în care să prezentați repere din istoria apariției limbajului de programare C++. Prezentarea trebuie să conțină imagini și sunete, eventual înregistrate de voi. Salvați prezentarea cu numele **IstoriaC++** în folderul **Lucru\_ClsVII**.



## 2 Limbajul C++. Elemente de bază

### Amintiți-vă

În lecția anterioară am învățat să realizăm un proiect în mediul de dezvoltare Code::Blocks, pe care l-am numit **PrimulProiect**. Era un mic program de bun venit care afișa pe ecran mesajul „Hello World!”.

### Descoperiți

Vom descoperi în continuare elementele de bază ale limbajului C++: structura unui program C++, vocabularul limbajului, variabile și constante, informații despre datele numerice (tipurile de date numerice, operatori).

### Rețineți

Numim **cod sursă** al unui program implementarea (editarea) unui algoritm într-un limbaj de programare, cu scopul de a atinge obiectivele dorite. Un program C++ poate conține mai multe fișiere cu cod sursă.

Numim **instrucțiune** o comandă care este dată calculatorului de către program atunci când acesta este lansat în execuție. Instrucțiunile în C++ se termină obligatoriu cu „;”.

Numim **funcție** un ansamblu de mai multe instrucțiuni care realizează anumite prelucrări și returnează un rezultat. Codul sursă al unui program poate conține una sau mai multe funcții.

Numim **bibliotecă** o colecție de resurse care pot fi utilizate de către un program software.

### Structura programelor C++

Vom descoperi în continuare structura unui program C++ plecând de la proiectul **PrimulProiect**.

Privim codul sursă al primului program (din fișierul **main.cpp**) și identificăm părțile componente ale acestuia (*îmaginea a*):

#### 1. #include <iostream>

„#include <iostream>” se numește **header**.

- <iostream> este o bibliotecă standard C++ care conține funcțiile utilizate pentru intrări și ieșiri de date (input/output). Orice program în care introducem date de la tastatură și/sau afișăm diferite informații pe ecran are nevoie să aibă inclusă această bibliotecă;

- sintagma `#include` adăugată înaintea numelui bibliotecii asigură faptul că pot fi efectuate operații cu funcțiile incluse în această bibliotecă. Liniile de program care încep cu `#` sunt directive pentru compilator (**directive de preprocessare**). În cazul de față, compilatorul ia la cunoștință că trebuie să includă biblioteca `iostream` la compilarea și execuția programului, preprocessarea fiind o prelucrare a codului sursă al unui program înainte de compilare.

#### 2. using namespace std;

Toate elementele din biblioteca standard C++ sunt declarate într-un **spațiu de nume (namespace)** cu titlul **std (standard)**. Exact cum două persoane pot avea același nume, și variabilele și funcțiile din C++ pot avea nume identice. Ca să evităm confuzia, le grupăm în aceste spații de nume. Dacă utilizăm această instrucțiune, toate elementele din acest spațiu de nume (funcții, variabile etc.) devin vizibile și pot fi utilizate direct, fără a mai fi nevoie să specificăm de unde provin. Dacă nu o utilizăm, trebuie să specificăm spațiul de nume de unde provine `cout`, adică utilizam `std::cout` în loc de `cout` în codul nostru.

#### 3. int main()

```
{  
cout << "Hello world!" << endl;  
return 0;  
}
```

- main()** este funcția principală a programului. Aceasta este unică într-un program, cu această funcție începe în totdeauna execuția programului. Observăm că se disting două părți componente, și anume antetul funcției: `int main()` și corpul funcției între `{}`. Acesta conține codul sursă care realizează prelucrările dorite. Instrucțiunile incluse în această funcție sunt:

- `cout << "Hello world!" << endl;` realizează scrierea pe ecran a mesajului „Hello world!”;
- `return 0;` înseamnă returnarea valorii **0** ca rezultat al funcției **main**. Returnarea acestui rezultat de către funcția **main()** este interpretată ca finalizare cu succes a programului.

```
1 #include <iostream> 1  
2  
3 using namespace std; 2  
4  
5 int main() 3  
6 {  
7     cout << "Hello world!" << endl;  
8     return 0;  
9 }
```

a

## Rețineți

Structura generală a unui program C++ este formată din: zona de directive de preprocesare, zona de declarații, zona de funcții – include obligatoriu funcția **main()**, vedeți *imaginea b*. Ca observație, constantele/variabilele pot fi declarate și oriunde este nevoie în program.

**b**

zona de directive de preprocesare  
(includerea bibliotecilor)

zona de declarații  
(constante, variabile)

zona de funcții

```
int main()
{
    zona de declarații de variabile
    instrucțiuni
    return 0
}
```

## Aplicați și investigați

1. Stergeți „;” de la linia 7 de cod, apoi compilați. Observați ce eroare de compilare se afișează.
2. Stergeți **using namespace std;** și apoi compilați. Observați erorile. Modificați apoi linia 7 în „**std::cout << "Hello world!" << std::endl;**” și compilați din nou.

## Descoperiți

### Vocabularul limbajului

Pentru a putea scrie programe în C++ trebuie să cunoaștem, în primul rând, vocabularul limbajului și sintaxa.

**Vocabularul** presupune: setul de caractere utilizate, identificatorii, cuvintele-cheie, separatorii, comentariile.

**Sintaxa** reprezintă totalitatea regulilor de scriere corectă a programelor.

#### a. Setul de caractere

Setul de caractere cu ajutorul căruia poate fi scris un program C++ este format din:

- literele majuscule și minuscule din alfabetul englez: a ... z, A ... Z;
- cifrele sistemului de numerație în baza 10: 0 ... 9;
- caractere speciale (+ - \* / = ^ < > ( ) [ ] . , : ; # \$ @ \_ <spațiu> ).

#### b. Identificatori și cuvinte-cheie

Un **identificator** este un șir de caractere format din litere, cifre și/sau caracterul „\_”. Acesta identifică în mod unic o variabilă, o constantă etc. (este numele cu care le utilizăm pe acestea în program). Obligatoriu, un identificator trebuie să înceapă cu o literă sau cu caracterul „\_”, să nu conțină spații și să nu coincidă cu niciun cuvânt-cheie din C++.

Un **cuvânt-cheie** este un cuvânt care face parte chiar din sintaxa limbajului C++, cum ar fi:

instrucțiuni (**if**, **for** etc.), tipuri de date (**int**, **long**) etc.

Exemple de identificatori corectă: pret, c1, contor\_cifre, nrCifre, \_masaCorpului

Exemple de identificatori incorectă: 3retete, \*nrMasina, pret de cost, produsul 1

Limbajul C++ este *case sensitive*, adică face diferențierea între literele majuscule și cele minuscule. Odată definit un identificator care conține anumite tipuri de litere (majuscule sau minuscule), acesta trebuie utilizat pe tot parcursul programului.

Exemple de identificatori diferenți: eValid, evalid, evaliD

#### c. Separatori

Separatorul este un caracter care delimită unitățile sintactice sau le separă între ele. Limbajul C++ utilizează următorii separatori: <spațiu>, caracterul CR, sfârșitul de linie sau caracterul punct și virgulă „;”.

#### d. Comentarii

Comentariile sunt niște texte care se introduc în codul sursă fără a avea vreun efect în execuția programului, ele sunt ignorate de către compilator. Este foarte important ca în orice program să existe astfel de comentarii pentru a se putea înțelege codul sursă mult mai ușor.

În C++ se pot introduce comentarii în două moduri, doar pe o linie (1) sau pe una sau mai multe linii (2):

- |                      |     |                       |
|----------------------|-----|-----------------------|
| 1. //text comentariu | sau | 2. /* text comentariu |
|                      |     | text comentariu       |
|                      |     | ....                  |
|                      |     | text comentariu */    |



## Tipuri de date numerice

Calculatoarele lucrează cu date. Memoria calculatorului este organizată în bytes (octeți). Un octet este compus din 8 biți. Un byte este cea mai mică unitate de memorie care poate fi gestionată de C++. Este astfel important pentru cei care scriu programe să știe exact cât ocupă în memoria calculatorului fiecare tip de date cu care lucrează.

Tipurile fundamentale în C++ sunt: **char, short, int, long, float, double și long double**. Primele patru tipuri de date se numesc **tipuri întregi** și sunt utilizate pentru a reprezenta numere întregi, ultimele trei tipuri de date se numesc **tipuri reale** (se mai numesc *în virgulă mobilă*) și memorează valori reale.

La denumirea unui tip de date se mai poate adăuga una dintre particulele **long, short și signed (cu semn), unsigned (fără semn)**. Acestea poartă numele de **calificatori** și modifică spațiul de memorie ocupat de către acea variabilă. Dacă nu se specifică pentru tipul unei variabile nici **signed**, nici **unsigned**, compilatorul consideră datele ca fiind **signed implicit**.

### Tipuri întregi

Denumire tip	Reprezentare (bytes)	Reprezentare (biți)		Interval de valori
char	1 byte cu semn	8 biți cu semn	$-2^7 \dots 2^7 - 1$	-128 ... 127
unsigned char	1 byte fără semn	8 biți fără semn	$0 \dots 2^8 - 1$	0 ... 255
short int	2 bytes cu semn	16 biți cu semn	$-2^{15} \dots 2^{15} - 1$	-32768 ... 32767
unsigned short int	2 bytes fără semn	16 biți fără semn	$0 \dots 2^{16} - 1$	0 ... 65535
int	4 bytes cu semn	32 biți cu semn	$-2^{31} \dots 2^{31} - 1$	-2147483648 ... 2147483647
unsigned int	4 bytes fără semn	32 biți fără semn	$0 \dots 2^{32} - 1$	0 ... 4294967295
long int	4 bytes cu semn	32 biți cu semn	$-2^{31} \dots 2^{31} - 1$	-2147483648 ... 2147483647
unsigned long int	4 bytes fără semn	32 biți fără semn	$0 \dots 2^{32} - 1$	0 ... 4294967295
long long int	8 bytes cu semn	64 biți cu semn	$-2^{63} \dots 2^{63} - 1$	
unsigned long long int	8 bytes fără semn	64 biți fără semn	$0 \dots 2^{64} - 1$	

### Tipuri reale

- float – reprezentat pe 4 octeți
- double – reprezentat pe 8 octeți
- long double – reprezentat pe 10 octeți

### Variabile și constante

La fel ca la programele Scratch, datele cu care lucrează un program C++ pot fi variabile (valoarea lor se poate modifica) sau constante (valoarea lor nu se modifică). O variabilă pe care o utilizați în program este obligatoriu să fie definită înainte, trebuie să se specifice care este tipul ei. Definirea variabilelor se face utilizând *instrucțiuni declarative*:

```
tip nume_variabila;
tip nume_variabila1, nume_variabila2, ... ;
```

Când se definește o variabilă, se poate prevedea chiar la acel moment o valoare inițială pentru ea. Această operațiune se numește inițializare. Limbajul C++ permite atribuirea unei valori inițiale variabilei chiar în cadrul declarației. Pentru inițializare se folosește operatorul **=**.

Numim valoare literală (sau literal) orice valoare constantă din program. Constantele se definesc astfel:

```
const tip nume_constanta = valoare;
const tip nume_constanta1 = valoare1, nume_constanta2 = valoare2, ...;
```

### Operatori în C++

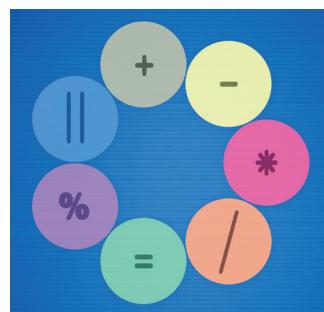
Când spunem operație, ne gândim automat la părțile sale componente: la operanzi și la operatori.

Operanzzii pot fi constante, variabile, rezultatele altor operații sau al unor funcții. Numim **expresie** o operație complexă care este construită din operatori și alte operații pe post de operanzi.

```
Exemplu: int varsta;
          long suma, nrLocuitoriEuropa;
          double pret;
```

```
Exemplu: int varsta=1;
          long suma=0, produs=1, nr1;
          double pret=3.2;
```

```
Exemplu: const int zile = 7;
          const double nrFrati = 2, pretVarza = 3.5;
```



### a. Operatori aritmetici



		Semnificație	Exemplu	
			Expresie	Valoare
+	adunare		56+23	79
-	scădere		56-23	33
*	înmulțire		56*23	1288
/	împărțire	Dacă operanții sunt de tip întreg ( <b>int</b> , <b>short</b> , <b>long</b> , ...), atunci rezultatul este tot un număr întreg, egal cu câtul întreg al împărțirii. Dacă cel puțin un operand este de tip real ( <b>float</b> , <b>double</b> , <b>long double</b> ), atunci rezultatul operației este rezultatul împărțirii zecimale a celor două numere, adică tot un număr real.	56/23	2
%	restul împărțirii a două numere întregi (modulo)		56%23	10

### b. Operatori relaționali

Rezultatul unei expresii în care avem un operator relațional poate fi Adevărat (1) sau Fals (0).

<b>==</b>	este egal cu	<b>&lt;</b>	este mai mic decât	<b>&gt;</b>	este mai mare decât
<b>!=</b>	este diferit de	<b>&lt;=</b>	este mai mic sau egal cu	<b>&gt;=</b>	este mai mare sau egal cu

*Exemplu:* `x>=2;` returnează Adevărat (1) pentru  $x=7$   
`x==4;` returnează Fals (0) pentru  $x=7$

### c. Operatori logici



Expresiile logice construite cu ajutorul operatorilor logici ca rezultat Adevărat (1) sau Fals (0).

Notăm A – Adevărat (1), F – Fals (0). Tabelele de adevăr pentru operatorii logici din C++ sunt cele știute:

<b>&amp;&amp; (ȘI logic)</b>	A	F	<b>   (SAU logic)</b>	A	F	<b>! (negație logică)</b>	A	F
A	A	F	A	A	A			
F	F	F	F	A	F		F	A

*Exemplu:* Dacă cerințele unei probleme cer ca un număr  $a$  fie natural nenul, mai mic sau egal cu 100, atunci scriem în C++ următoarea expresie: `(a>0) && (a<=100)`. Se va returna Adevărat (1) pentru  $a=77$  și Fals (0) pentru  $a=150$ .

### d. Operatori de atribuire

Atribuirea este o operație prin care o variabilă primește o anumită valoare. Prin adăugarea lui „;” la final, atribuirea devine *instructiune de atribuire*: **variabila = expresie**;

Se evaluatează expresia din dreapta, iar valoarea ei se atribuie variabilei din stânga egalului.

*Exemplu:* `nrCrt=7;` `produsul=(7+5)*4;` `a=a+56;`

**Important:** Faceți distincția între operatorul de atribuire **=** și cel relațional **==**.

Când dorim să modificăm valoarea unei variabile pornind de la valoarea sa anterioară și aplicând asupra acesteia o operație, putem să utilizăm niște operatori numiți **operatori compuși**.

Operator compus	Expresie	Expresie echivalentă
<b>+=</b>	variabila +=expresie;	variabila = variabila + expresie;
<b>-=</b>	variabila -=expresie;	variabila = variabila - expresie;
<b>*=</b>	variabila *=expresie;	variabila = variabila * expresie;
<b>/=</b>	variabila /=expresie;	variabila = variabila / expresie;
<b>%=</b>	variabila %=expresie;	variabila = variabila % expresie;

*Exemplu:* `a+=7;` este echivalentă cu `a=a+7;`  
`b*=a+3;` este echivalentă cu `b=b*(a+3);`  
`c%=5;` este echivalentă cu `c=c%5;`



### e. Operatori de incrementare/decrementare

**Incrementarea** unei variabile înseamnă mărirea valorii sale cu 1. **Decrementarea** unei variabile înseamnă micșorarea valorii sale cu 1. Operatorii de incrementare/decrementare pot apărea fie înaintea variabilei, prefixați (**++variabila**, **--variabila**), fie după variabilă, postfixați (**variabila++**, **variabila--**). Prin adăugarea lui „;” la final se transformă în *instrucțiune de incrementare/decrementare*.

*Exemplu:* `a++; b--;`

Pentru a explica mecanismul de funcționare al acestor operatori, utilizăm ca exemplu o instrucțiune de atribuire pentru două variabile întregi *a* și *b* (vedeți tabelul din dreapta).

Instrucțiune de atribuire	Echivalență
<code>a=++b;</code>	<code>b=b+1; a=b;</code>
<code>a=--b;</code>	<code>b=b-1; a=b;</code>
<code>a=b++;</code>	<code>a=b; b=b+1;</code>
<code>a=b--;</code>	<code>a=b; b=b-1;</code>

*Exemplu:*

```
int a=11, b=50, p;
a++;           //a ia valoarea 11
p=--a;         //a ia valoarea 10, p ia valoarea nouă a lui a, adică 10
p=b++;         //p ia valoarea lui b, adică 50, apoi b ia valoarea 51
```

### f. Operatorul de conversie explicită

De multe ori este necesar ca într-un program C++ să considerăm o expresie care are un anume tip ca fiind de un alt tip pentru a putea face anumite operații. Pentru aceasta folosim operatorul de conversie. Sintaxa: **(noulTip) expresie**

*Exemplu:* `cout<<(float)17/2+5; x=(float)y/2;`

### Operații de citire și afișare a datelor

În exemplul de program de care am discutat mai sus, am spus că linia de cod `cout << "Hello world!" << endl;` realizează scrierea pe ecran a mesajului „Hello world!”. În C++, intrarea și ieșirea datelor se realizează cu ajutorul fluxurilor de date.

Un **flux de date (stream)** este fluxul datelor de la o sursă (de exemplu, tastatură) la o destinație (de exemplu, ecranul monitorului). Pentru operațiile de intrare (citire) și ieșire (scriere) se folosesc instrucțiuni expresie create cu ajutorul operatorilor `>>` și, respectiv, `<<`.

Fluxurile de date standard sunt:

**1. `cin ("console in")`** – flux de date de intrare, permite trimitera datelor de la tastatură către program. Se utilizează împreună cu operatorul de intrare `>>`. Instrucțiunea are forma:

`cin >> variabila;`

`cin >> variabila1 >> variabila2...;`

<i>Exemplu:</i>	<code>cin&gt;&gt;s;           //se atribuie variabilei s valoarea introdusă de la tastatură</code>
	<code>cin&gt;&gt;a&gt;&gt;b;      //se citesc două valori de la tastatură și se atribuie celor</code>
	<code>//două variabile a și b, în ordinea introducerii.</code>

**2. `cout ("console out")`** – flux de date de ieșire, permite trimitera datelor către consolă și afișarea lor ca text pe ecran. Se utilizează împreună cu operatorul de ieșire `<<`. Instrucțiunea are forma:

`cout << variabila/constanta;`

`cout << variabila1/constanta1 << variabila2/constanta2...;`

Pentru a trece la rândul următor avem două posibilități:

`cout << endl;` // inserează o linie nouă și golește fluxul de date de ieșire

`cout << '\n';` // doar inserează o linie nouă

<i>Exemplu:</i>	<code>cout&lt;&lt;"Buna ziua tuturor!";           //afișează pe ecran "Buna ziua tuturor!"</code>
	<code>cout&lt;&lt;x;                           //afișează pe ecran valoarea variabilei x</code>
	<code>cout&lt;&lt; 675;                         //afișează pe ecran numărul 675</code>
	<code>cout&lt;&lt;"Suma numerelor este: " &lt;&lt;s; //afișează pe ecran mesajul "Suma numerelor</code>
	<code>//este: " urmat de valoarea variabilei s</code>
	<code>cout&lt;&lt;x+y;                         //afișează pe ecran rezultatul expresiei x+y</code>

Pentru a folosi `cin` și `cout`, se adaugă la începutul programului directiva `#include <iostream>`.





## Aplicații și investigații

Parcurgeți următorii pași în ordinea în care sunt specificați:

1. Creați un proiect nou în Code::Blocks și numiți-l **OperatoriAritmetici**.
2. Declarați două variabile *a* și *b* de tip întreg (**int a, b;**) și două variabile *c* și *d* de tip float (**float c, d;**).
3. Introduceți de la tastatură valori pentru toate cele patru variabile (**cin>>a; cin>>b; cin>>c; cin>>d;**).
4. Folosiți operatorii +, -, \*, /, % și afișați rezultatele pentru fiecare operație în parte pentru variabilele *a* și *b*, după exemplul: **cout<<"a+b="<<a<<"+"<<b<< "="<<a+b<<endl;**. Rulați și observați rezultatele.
5. Adăugați la program linia: **cout<<"c%d="<<c<<"%"<<d<< "="<<c%d<<endl;**; încercați să compilați și să rulați și observați eroarea de compilare: **error: invalid operands of types 'float' and 'float' to binary 'operator%'**.
6. Comentați linia anterior introdusă și adăugați linia de cod: **cout<<"c/d="<<c<<"/"<<d<< "="<<c/d<<endl;**. Rulați și introduceți de la tastatură valorile *a=3, b=4, c=3, d=4*. Observați diferența de calcul *a/b* și *c/d*. În primul caz, rezultatul este 0 (operanții sunt numere întregi), în al doilea caz este 0.75 (operanții sunt numere reale).
7. Folosiți operatorii de incrementare pentru a afișa succesorul valorii variabilei *a*. Încercați, pe rând, două variante; observați că doar a doua (cea cu prefixare) afișează corect rezultatul. Variantele sunt: **cout << "Afisam a++:"<<a++<< endl;** și **cout << "Afisam ++a:"<<++a<< endl;**. Investigați de ce se întâmplă acest lucru. Salvați programul.
8. Creați un nou proiect cu numele **OperatoriLogici**. Declarați două variabile *a* și *b* de tip întreg (**int a, b;**).
9. Introduceți de la tastatură valori pentru *a* și *b* (**cin>>a; cin>>b;**). Când rulați programul, introduceți doar valorile 0 sau 1 pentru aceste variabile.
10. Afișați, pe rând, operațiile cu operatorii logici (&&, ||, !), după exemplul:  
**cout<<"a&&b="<<a<<"&&"<<b<< "="<<(a&&b)<<endl;**  
Rulați programul și observați rezultatele când introduceți pentru *a* și *b* valorile 1 și 0, în toate combinațiile posibile.



## Eserții

1. Scrieți, în caiete, care dintre următoarele siruri de caractere pot reprezenta identificatori în C++:

- |                  |                      |                    |                      |                  |
|------------------|----------------------|--------------------|----------------------|------------------|
| <b>a.</b> nrCrt; | <b>c.</b> produs_56; | <b>e.</b> if;      | <b>g.</b> _partTime; | <b>i.</b> #form; |
| <b>b.</b> n&d;   | <b>d.</b> SnT;       | <b>f.</b> Suma 17; | <b>h.</b> max+1;     | <b>j.</b> sumă.  |

2. Alegeți și scrieți în caiete răspunsul corect pentru fiecare dintre cerințele următoare:

- A.** Care este rezultatul returnat în C++ pentru **15/19\*19?**
- a.** 1;      **b.** 0.0934;      **c.** 0.
- B.** Evaluați următoarea expresie: **!(1 || 0 && !1)**.
- a.** 0;      **b.** 1;      **c.** expresie invalidă.
- C.** Evaluați **47%3**.
- a.** 2;      **b.** 4;      **c.** 0.
- D.** Care tip de dată stochează numere zecimale?
- a.** char;      **b.** long;      **c.** float;
- E.** Ce tipărește linia de cod **cout<<(0==0)?**
- a.** 0;      **b.** 1;      **c.** eroare la compilare.
- F.** Dacă un program se execută corect, ce valoare ar trebui să întoarcă funcția **main()**?
- a.** 0;      **b.** 1;      **c.** -1.

3. Dați exemple de trei tipuri de date numerice întregi.

4. Scrieți, în caiete, valorile variabilelor întregi *a* și *b* după executarea liniei de cod **a+=b\*3;** dacă *a=1, b=5*;

## Operatori C++



### Portofoliu



Realizați un poster în care să prezentați toti operatorii învățați.

### 3 Structura liniară

#### Amintiți-vă

Ne amintim din anii anteriori că structura liniară sau secvențială este o succesiune de instrucțiuni care se execută în ordinea în care apar, una după alta, de la început până la sfârșit (*îmaginea a*).

#### Descoperiți

Vom descoperi, în cele ce urmează, cum anume putem realiza programe în C++ cu structură liniară, folosind diferitele instrucțiuni învățate.

Într-o structură liniară putem întâlni următoarele tipuri de instrucțiuni:

1. Instrucțiunea de declarare – declararea de variabile/constante, terminată prin „;”.

*Exemplu:* int nrCrt; long suma\_19; const int a=17;

2. Instrucțiunea expresie – o expresie formată din operanzi și operatori, terminată prin „;”. Poate fi:

a. *instrucțiune de atribuire*;

*Exemplu:* a=7; a=c\*234; a-=67; b=a++;

b. *instrucțiune de incrementare/decrementare*;

*Exemplu:* a++; b--;

c. *instrucțiune de intrare/iesire*.

*Exemplu:* cout<<"Este frumos sa faci programe in C++"; cout<<s; cin>>a;

3. Instrucțiunea compusă

O **instrucțiune compusă** (sau **bloc de instrucțiuni**) este o succesiune (grupare) de mai multe declarații și instrucțiuni, grupate între acolade {}. Instrucțiunile din corpul unui bloc se execută secvențial, în ordine. Sintaxa:

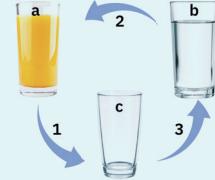
```
{
  Declarații
  Instrucțiuni
}
```

Corpul oricărei funcții (inclusiv al funcției **main()**) este un **bloc**. Un bloc poate conține și doar o singură instrucțiune. El nu trebuie terminat prin punct și virgulă.

*Exemplu:* Interschimbarea valorii a două variabile a și b:



```
int a,b,c;
c=a;
a=b;
b=c;
```



4. Instrucțiunea **return** – permite ieșirea dintr-o funcție. Forma instrucțiunii este:

**return expresie;**

*Exemplu:* Folosim **return 0;** la terminarea funcției **main()**, dacă aceasta s-a încheiat cu succes.

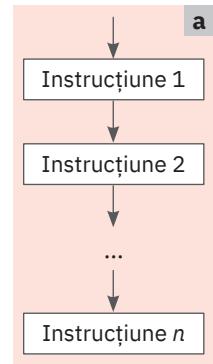
5. Instrucțiunea **vidă**

De multe ori, în C++ este nevoie de existența unei instrucțiuni pentru a corespunde sintaxei limbajului, dar aceasta nu trebuie să facă nimic. În aceste cazuri se utilizează doar „;” fără nimic altceva.

#### Rețineți

În C++, instrucțiunile se finalizează întotdeauna prin **punct și virgulă „;”**. Se pot regăsi mai multe instrucțiuni pe aceeași linie atâtă timp cât ele sunt despărțite prin punct și virgulă.

Mai departe, vom vedea câteva exemple de programe care utilizează structuri liniare. Vom lucra cu variabile de tip întreg și vom presupune că se introduc de la tastatură numere întregi, fără să fie nevoie să verificăm tipul datelor introduse.



## Calculul mediei școlare

► La școală, la o materie, elevii primesc trei note. Afipați media obținută de aceștia.

*Indicații:*

Haideți să construim algoritmul pentru acest program. Schema logică poate fi văzută în imaginea b.

**Date de intrare:**  $n_1, n_2, n_3$  = variabile de tip întreg, reprezentând notele la acea materie  
**Date de ieșire:**  $m$  = variabilă de tip real, reprezintă media la acea materie

Algoritmul construit este:

```
algoritm media
întreg n1, n2, n3;
real m;
Citește n1, n2, n3
m ← (n1+n2+n3)/3
Scrie 'Media notelor este:' m
Sfârșit
```

Programul scris în C++ este următorul:

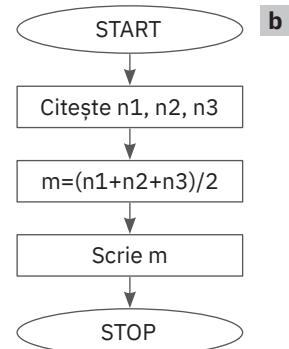
```
#include <iostream>
using namespace std;

int main()
{
    int n1, n2, n3;
    float m=0;

    cout<<"Introduceti prima nota: ";
    cin>>n1;
    cout<<"Introduceti a doua nota: ";
    cin>>n2;
    cout<<"Introduceti a treia nota: ";
    cin>>n3;

    m=(float)(n1+n2+n3)/3;
    cout<<"Media notelor este: "<<m<<endl;

    return 0;
}
```



Parcurgerea programului pe pași

n1=10  
n2=10  
n3=9  
m=9.66667  
Media notelor este: 9.66667

## Criptarea unui număr

Sunteți agenți secerți și trebuie să transmiteti un mesaj. În mesaj aveți un număr de 4 cifre care trebuie transmis criptat ca să nu poată fi interpretat de cineva care ar citi mesajul. Algoritmul de criptare este următorul: se adună la fiecare cifră a numărului câte 1 unitate. Dacă cifra este 9, atunci aceasta va fi transformată în 0. Scrieți un program care să realizeze criptarea numărului și să afișeze ca număr de control suma cifrelor inițiale necriptate.

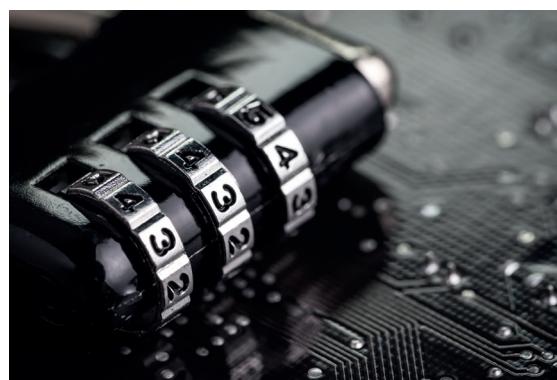
*Indicații:*

Pentru a rezolva această problemă, este necesar să extragem fiecare cifră în parte a numărului, să aplicăm algoritmul de conversie și apoi să afișăm numărul recompus din noile cifre. Transformarea cifrei 9 în 0 se traduce prin calcularea restului împărțirii (cifrei +1) la 10. De exemplu, dacă numărul este 3418, codificarea sa va fi 4529. Dacă numărul este 1956, numărul cifrat este 2067.

**Date de intrare:**  $n$  = variabilă de tip întreg, reprezintă numărul care se va cripta

**Date de manevră:**  $c1, c2, c3, c4$  = variabile de tip întreg, reprezintă cifrele numărului  $n$

**Date de ieșire:**  $s$  = suma cifrelor numărului  $n$ , necriptat



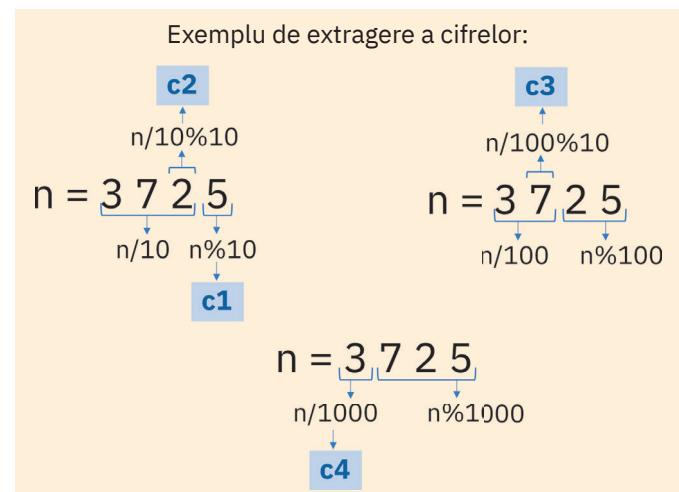
Algoritmul în pseudocod este:

 **algoritm** criptare  
**întreg** n, c1, c2, c3, c4, s;  
 s  $\leftarrow$  0  
**Citește** n  
 $c1 \leftarrow n \% 10;$   
 $c2 \leftarrow n/10 \% 10;$   
 $c3 \leftarrow n/100 \% 10;$   
 $c4 \leftarrow n/1000;$   
 $s \leftarrow c1 + c2 + c3 + c4$   
**Scrie** s  
**Scrie**  $(c4+1)\%10, (c3+1)\%10, (c2+1)\%10, (c1+1)\%10$   
**Sfârșit**

Programul C++ este următorul:

```
#include <iostream>
using namespace std;
int main()
{ int n, c1, c2, c3, c4, s=0;
    cout<<"Introduceti numarul: ";
    cin>>n;
    c1=n%10;           //cifra unitatilor
    c2=n/10%10;        //cifra zecilor
    c3=n/100%10;       //cifra sutelor
    c4=n/1000;         //cifra miilor

    s=c1+c2+c3+c4;   //suma cifrelor nr necriptat
    cout<<"Numarul de control: "<<s<<endl;
    cout<<"Nr codificat: "<<(c4+1)%10<<(c3+1)%10<<(c2+1)%10<<(c1+1)%10;
    return 0;
}
```



## Aplicați și investigați

- ▶ Rulați programul de mai sus pas cu pas, aşa cum ați învățat la lecția „Mediul de dezvoltare Code::Blocks, Depanarea programului”. Punei punctul de intrerupere la linia **c1=n%10;** (la aflarea cifrei unităților), apoi rulați pas cu pas utilizând comanda **Next line** (*Linia următoare*). Urmăriți în fereastra **Watches** (*Urmărire*) valoarea variabilelor.



## Exersați

1. Care este rezultatul returnat de programul prezentat alăturat în *imaginea c*, dacă se introduc de la tastatură valorile a=5 și b=7?
2. Scrieți un program C++ care să realizeze conversia din metri în centimetri și în kilometri. *Indicație:* Intrare: 10 m; Ieșire: 1000 cm, 0.01 km.
3. Bunicii vor să își împrejmuiască grădina și să o aranjeze cu gazon. Aceasta are formă dreptunghiulară. Realizați un program în care să se introducă valoarea lungimii și lățimii terenului în metri și să se calculeze câți metri de gard trebuie să cumpere dacă poarta are 1 metru lățime. Calculați, de asemenea, ce suprafață de gazon trebuie să acopere bunicii. Lungimea și lățimea terenului sunt numere naturale nenule, mai mici decât 1000.  
*Indicație:* Intrare: lungimea = 20 m, lățimea = 10 m; Ieșire: P = 59 m, A = 200 m<sup>2</sup>.
4. Care dintre următoarele instrucțiuni în C++ au ca rezultat eliminarea cifrei zecilor dintr-un număr de trei cifre?  
 a.  $a=a\%10+a\%100;$       b.  $a=a/100*10+a\%10;$       c.  $a=a/100+a\%10;$
5. Se dă două variabile numerice întregi *a* și *b*. Alegeti care dintre cele două variabile va avea valoarea de 3 ori mai mare decât valoarea inițială după executarea următoarelor instrucțiuni: **b=2\*a;** **a=a+b;** **b=b+b+b;**  
 a. variabila *a*;      b. variabila *b*;      c. ambele variabile.

```
#include <iostream>
using namespace std;
int main()
{ int a, b;
    cin>>a;
    cin>>b;
    cout<<(((a*b)%6)-1) /2;
    return 0;
}
```

# 4 Structura alternativă (de decizie)



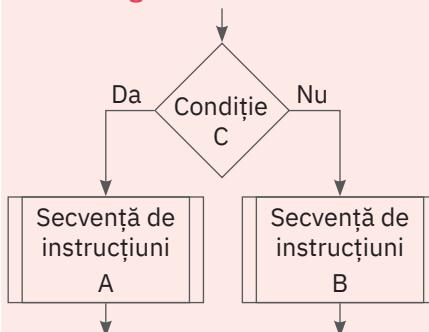
Forma structurii alternative (de decizie) este:



## Pseudocod

**Dacă** (<condiție C>) **atunci**  
<Secvență de instrucțiuni A>  
**Altfel** <Secvență de instrucțiuni B>

## Schema logică



## Algoritm de funcționare

**Pasul 1:** se evaluatează <condiție C>

**Pasul 2:**

**Dacă** valoarea <condiție C> este Adevărat  
**Atunci** se execută <Secvență de instrucțiuni A>  
**Altfel** se execută <Secvență de instrucțiuni B>

Pot fi și cazuri în care există doar o ramură a structurii (lipsește ramura **Altfel**).



## If... Then... Else

În C++, sintaxa instrucțiunii care permite implementarea în limbaj de programare a structurii alternative este:



```
if (condiție C)
{
    secvență de instrucțiuni A
}
else
{
    secvență de instrucțiuni B
}
```

```
if (condiție C)
{
    secvență de instrucțiuni A
}
```

Pentru anumite cazuri în care nu este necesar să se precizeze alternativa (secvența de instrucțiuni B).

## Switch

**+** În mai multe cazuri, este necesar să luăm decizii în funcție de mai mult de două valori pe care le-ar putea avea o variabilă. În C++ există o astfel de instrucțiune de decizie multiplă, **switch**, cu următoarea sintaxă:

```
switch (expresie_E)
{
    case expr_constantă_1: secvență_instrucțiuni_1;
    [break];
    case expr_constantă_2: secvență_instrucțiuni_2;
    [break];
    .....
    case expr_constantă_n-1: secvență_instrucțiuni_n-1;
    [break];
    [default: secvență_instrucțiuni_n];
}
```

## Algoritm de funcționare

Se verifică valoarea unei expresii **expresie\_E** pentru a se vedea dacă este egală cu vreuna dintre valorile tratate de către switch (**exprессie\_constantă\_1, ..., expresie\_constantă\_n-1**).

Dacă valoarea se regăsește în lista valorilor **exprессie\_constantă\_1, ..., expresie\_constantă\_n-1**, atunci se va executa secvența de instrucțiuni aferentă ramurii cu valoarea egală cu valoarea expresiei **expresie\_E**.

Altfel, dacă **expresie\_E** are o valoare care nu apare printre valorile **exprессie\_constantă\_1, ..., expresie\_constantă\_n-1**, atunci se execută secvența de instrucțiuni aferente cazului **default**, dacă acesta există (este optional).

Valorile expresiilor **exprессie\_constantă\_1, ..., expresie\_constantă\_n-1** etc. trebuie să fie întregi. Nu se admit duplicate (valori identice).

Exemple de astfel de expresii: corecte: 1+2+3, (6+2)/4 etc.; incorecte: a+b, a\*b+1 etc.

Instrucțiunea **break**; este optională, aceasta cauzează ieșirea imediată din **switch**. În momentul în care se întâlnește această instrucțiune, parcurgerea este întreruptă și se va continua cu execuția următoarei instrucțiuni aflate după **switch**.

Dacă instrucțiunea **break;** nu este întâlnită, atunci parcurgerea continuă la următorul caz al lui **switch**, până când se întâlnește următorul **break** sau se termină **switch**.

În cele ce urmează vom parcurge câteva programe care implementează structuri alternative. Variabilele cu care vom lucra sunt de tip întreg și vom presupune că se introduc de la tastatură numere întregi, fără să fie nevoie să verificăm tipul datelor introduse.

### Selectia acțiunilor la începutul unui joc

► Jocurile electronice afișează, de multe ori, la început, o fereastră în care poți alege dintre mai multe opțiuni. În funcție de ce alegi, programul face o anumită acțiune. Realizați un program care afișează o astfel de interfață simplă din care poți alege: dacă apeși tasta 1, începi jocul; dacă apeși tasta 2, poți alege nivelul jocului; dacă apeși tasta 3, poți vedea rezultatele. În loc de acțiuni, afișați doar ce s-a apăsat.

*Indicații: Date de intrare: a = variabilă de tip întreg, reprezintă numărul opțiunii alese de către utilizator.*

Programul C++ este următorul:

```
#include <iostream>
using namespace std;
int main()
{ int a;
cout<<"Selecteaza o optiune: "<<endl;
cout<<"1. Incepe joc"<<endl;
cout<<"2. Alege nivel"<<endl;
cout<<"3. Afiseaza rezultatele"<<endl;
cin>> a;

switch (a) {
case 1:           // Se foloseste ":", nu ";"
    cout<<"Ai ales 1";
    //aici se adauga codul pentru joc
    break;
case (6+2)/4:
    cout<<"Ai ales 2";
    //aici se adauga codul pentru alegerea nivelului
    break;
case 3:
    cout<<"Ai ales 3";
    //aici se adauga codul pentru afisarea rezultatelor
    break;
default:
    cout<<"Input gresit\n";
    break;
}
return 0;
}
```



### Aplicați și investigați

- Adăugați în program citirea de la tastatură a valorii variabilei *b*, apoi modificați expresia de la switch în *a+b*: **switch(a+b)**. Rulați aplicația și urmăriți rezultatele pentru diferite valori ale lui *a* și *b*.
- Modificați **case 3**: în **case 2**, apoi compilați și vedeți eroarea obținută.

### Descoperiți

#### Număr par/impar

Se introduce un număr natural mai mic decât 1000 de la tastatură. Afișați dacă este par sau impar.

*Indicații:* Pentru a decide dacă un număr este par sau impar, calculăm restul împărțirii sale la 2. Numerele pare dau restul 0, cele impare dau restul 1. Este clar vorba despre o decizie în funcție de valoarea de adevăr a unei condiții.

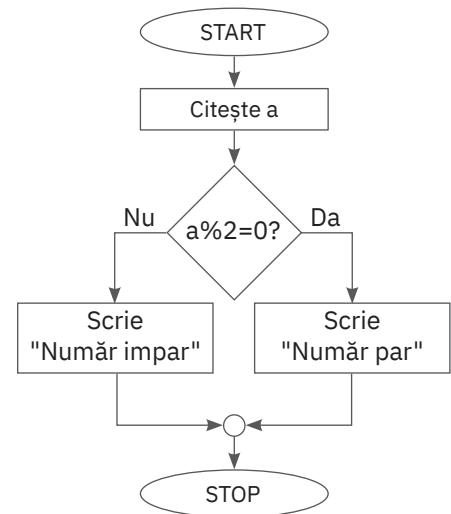
## Structura alternativă (de decizie)

Date de intrare:  $a$  = variabilă de tip întreg, reprezentând numărul introdus

```
#include <iostream>
using namespace std;
int main()
{
    int a;
    cout << "Introduceti un numar intreg:";
    cin >> a;

    if ( a % 2 == 0) //restul impartirii la 2 este 0
        cout << "Numar par";
    else //restul impartirii la 2 diferit de 0
        cout << "Numar impar";

    return 0;
}
```



## Necesar în bucătărie

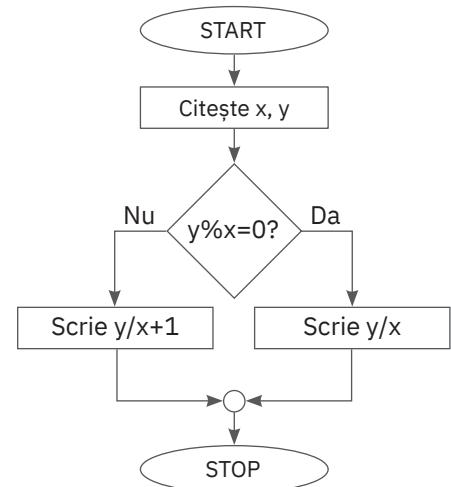
Calculați de câte sticle a câte  $x$  litri fiecare este nevoie pentru a umple un vas de  $y$  litri.

*Indicații:* Dacă  $x = 4$  l și  $y = 20$  l, atunci în vasul mare intră un număr întreg de sticle (restul împărțirii lui  $y$  la  $x$  este 0), deci este nevoie de 5 sticle.

Dacă  $x = 4$  l și  $y = 23$  l, atunci în vasul mare nu intră un număr întreg de sticle (restul împărțirii lui  $y$  la  $x$  este 3), deci avem nevoie de apa din 5 sticle întregi și o bună parte din apa dintr-o săsea stică, în total 6 sticle.

Scriem direct programul C++:

```
#include <iostream>
using namespace std;
int main()
{
    int x,y;
    cout<<"Introduceti cati l are sticla ";
    cin>>x;
    cout<<"Introduceti cati l are vasul ";
    cin>>y;
    if (y%x==0)
        cout<<"Avem nevoie de "<<y/x<<" sticle";
    else
        cout<<"Avem nevoie de "<<y/x+1<<" sticle";
    return 0;
}
```



## Calculul unei date anterioare

Se citesc de la tastatură trei numere naturale  $z$ ,  $l$  și  $a$ , care reprezintă o dată calendaristică:  $z$  este ziua,  $l$  este luna și  $a$  este anul. Afipați data zilei anterioare.

*Indicații:* Lunile anului pot fi împărțite în trei tipuri după numărul lor de zile: tip 1 – lunile cu 31 de zile sunt 1, 3, 5, 7, 8, 10, 12; tip 2 – lunile cu 30 de zile sunt 4, 6, 9, 11; tip 3 – luna cu 29/28 de zile (dacă anul este bisect sau nu) este 2.

Un an este **bisect** dacă se divide cu 4, excepție făcând anii divizibili cu 100, dar sunt ani bisecți și cei divizibili cu 400. Rezumând sub forma unei expresii logice: un an este bisect dacă se divide cu 4 și nu se divide cu 100 sau se divide cu 400. Exemple de ani bisecți: multiplii de 400 (1600, 2000, 2400 etc.), multiplii de 4 sau care nu se divid cu 100 sau 400 (2008, 2012, 2016, 2020, 2024 etc.). Să observăm ce cazuri ar fi pentru calculul zilei anterioare unei date.

Nr. crt.	Cazuri	Data introdusă	Data anterioară
1	Dacă ziua > 1	12.5.2024	11.5.2024
2	Dacă ziua = 1 și luna = 1	1.1.2024	31.12.2023
3	Dacă ziua = 1 și luna > 1 și luna anterioară = tip 1	1.2.2024	31.1.2007
4	Dacă ziua = 1 și luna > 1 și luna anterioară = tip 2	1.5.2024	30.4.2007
5	Dacă ziua = 1 și luna > 1 și luna anterioară = tip 3, an bisect	1.3.2024	29.2.2024
6	Dacă ziua = 1 și luna > 1 și luna anterioară = tip 3, an nebisect	1.3.2023	28.2.2023

Am marcat în culori diferite care sunt, pentru fiecare caz, elementele care se modifică pentru a calcula data anterioară (cu roșu schimbarea zilei, cu albastru schimbarea lunii, cu verde schimbarea anului). Transpunem într-un program C++ aceste cazuri.

```
#include <iostream>
using namespace std;
int main()
{ int z, l, a;
    cout<<"Introduceti ziua: ";
    cin>>z;
    cout<<"Introduceti luna: ";
    cin>>l;
    cout<<"Introduceti anul: ";
    cin>>a;

    cout<<endl<<"Data introdusa: "<<z<<". "<<l<<". "<<a<<endl;

    if (z>1) z=z-1;      //Caz 1, ziua>1
    else           //ziua=1
    { if (l==1)       //Caz 2, ziua>1 si luna=1
        { l=12; a=a-1; z=31; }
        else           //ziua>1 si luna>1
        { l=l-1;
            //Caz 3, luna anterioara = tip 1
            if (l==1 || l==3 || l==5 || l==7 || l==8 || l==10) z=31;
            //Caz 4, luna anterioara = tip 2
            else if (l==4 || l==6 || l==9 || l==11) z=30;
            //luna anterioara = tip 3
            else
            { //Caz 5, anul este bisect
                if((a%4==0 && a%100!=0) || (a%400==0)) z=29;
                //Caz 6, anul nu este bisect
                else z=28;
            }
        }
    }
    cout<<"Data anterioara: "<<z<<". "<<l<<". "<<a<<endl;
    return 0;
}
```



## Aplicați și investigați

- Rulați programul cu diferite valori și urmăriți pas cu pas valorile variabilelor, utilizând modul **Depanare (Debug)**.

## Exersați

- Se citesc de la tastatură două numere naturale  $a$  și  $b$  mai mici decât 10 000. Scrieți un program C++ care afișează dacă numărul  $a$  este divizibil cu  $b$ .

*Indicație:* Intrare:  $a=9$ ,  $b=0$ , Ieșire: „Operatie imposibila”; Intrare:  $a=4$ ,  $b=2$ , Ieșire: „4 este divizibil cu 2”; Intrare:  $a=5$ ,  $b=2$ , Ieșire: „5 nu este divizibil cu 2”.

- Scrieți un program C++ în care introduceți de la tastatură numărul unei luni și afișați câte zile are aceasta.

*Indicație:* Intrare: 9, Ieșire: „30 zile”; Intrare: 1, Ieșire: „31 zile”; Intrare: 2, Ieșire: „28/29 zile”.

- Utilizând instrucțiunea **switch**, faceți un program C++ care afișează numărul cel mai mic dintre două numere introduse de la tastatură.

*Indicație:* Intrare: 9, 3, Ieșire: 3.

- Având în vedere programul alăturat, pentru care dintre următoarele valori de intrare se vor afișa pe ecran două cifre?

**a.** 8;      **b.** 6;      **c.** 5;      **d.** 11.

```
int a; cin>>a;
if (a%2==0) cout<<"1";
if (a%3==0) cout<<"2";
else cout<<"3";
```

# 5 Structura repetitivă condiționată anterior

## Amintiți-vă

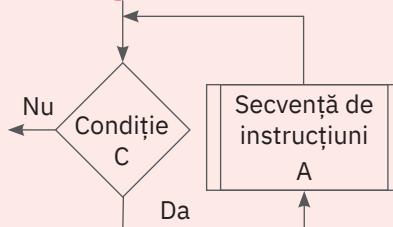
Forma structurii repetitive condiționate anterior este:



### Pseudocod

Cât timp (*<condiție C>*) execută  
*<Secvență de instrucțiuni A>*

### Schema logică



### Algoritm de funcționare

**Pasul 1:** se evaluatează *<condiție C>*

**Pasul 2:**

Dacă valoarea *<condiție C>* este

Adevărat

Atunci se execută *<Secvență de instrucțiuni A>* și se revine la Pasul 1

Altfel se ieșe din secvența repetitivă.

Utilizăm structura repetitivă condiționată anterior atunci când trebuie să executăm repetitiv niște instrucțiuni fără a ști exact numărul de repetări necesare. De asemenea, alegem această structură dacă în logica algoritmului nostru *nu este necesar să executăm obligatoriu măcar o dată instrucțiunile respective*.

De exemplu:

Avem un robot care trebuie să înainteze către un zid dintr-o poziție aleatorie. Care este algoritmul pe care îl construim pentru a înainta?

Cât timp (*Mai este loc până la zid pentru a face un pas*)

Robotul înaintează un pas

Cum poziția inițială a robotului este oarecare, este posibil ca robotul să fie mai departe de zid și să poată înainta, dar este și la fel de posibil ca robotul să fie poziționat chiar lângă zid în momentul în care pornește programul și să nu poată face niciun pas. În aceste condiții, este imperios necesar ca, înainte de a face fiecare pas (chiar și la primul), să verificăm dacă are loc să facă pasul sau nu. Pentru acest tip de problemă folosim *structura repetitivă condiționată anterior*.

## Descoperiți

În C++, sintaxa instrucțiunii repetitive condiționate anterior este următoarea:

```
while (condiție C)
{
    secvență de instrucțiuni A
}
```



Vom vedea, în cele ce urmează, cum anume folosim structura repetitivă condiționată anterior în limbajul C++. Vom lucra cu variabile de tip întreg și vom presupune că se introduc de la tastatură numere întregi, fără să fie nevoie să verificăm tipul datelor introduse.

## Iepurașul atlet

Un iepuraș participă la o întrecere sportivă. El sare la început 1 m, apoi fiecare salt este de două ori mai mare decât cel de dinaintea lui. Câte sărituri trebuie să facă iepurașul pentru a parcurge o distanță de  $n$  metri? Presupunem că pistă are minimum 1 m și maximum 1 000 m.

*Indicații:*

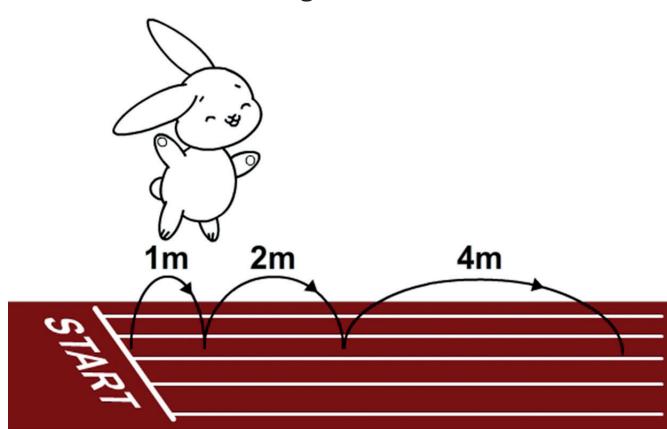
Haideți să construim algoritmul pentru acest program.

**Date de intrare:**  $n$  = variabilă de tip întreg, reprezintă câți metri are de parcurs în total iepurașul

**Date de manevră:**  $s$  = variabilă de tip întreg, reprezintă câți metri parcurge la un salt, un salt reprezintă un pas

$d$  = variabilă de tip întreg, distanța în metri parcursă până la un anume pas de către iepuraș

**Date de ieșire:**  $p$  = variabilă de tip întreg, numărul de pași parcursi de către iepuraș



Să luăm, spre exemplu,  $n=10$ , adică iepurașul ar avea de parcurs 10 m în total. În tabelul de mai jos avem o ilustrare a ceea ce se întâmplă la fiecare pas în parte până ajunge la destinație:

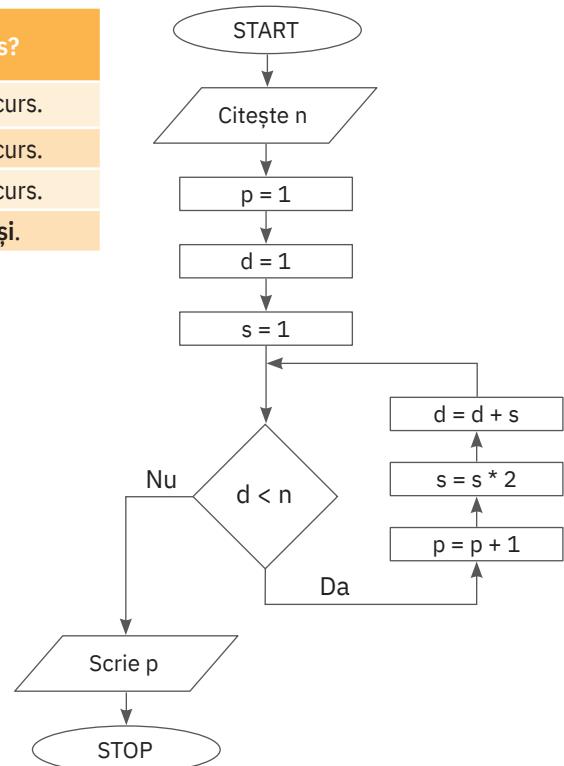
Nr. pas	Cât metri sare la acest pas	Cât metri a parcurs până acum	Mai are de parcurs?
1	1	1	$1 < 10$ , da, mai are de parcurs.
2	2	$1+2=3$	$3 < 10$ , da, mai are de parcurs.
3	4	$3+4=7$	$7 < 10$ , da, mai are de parcurs.
4	8	$7+8=15$	$15 > 10$ a terminat în 4 pași.

Algoritmul construit în pseudocod este:

```

algoritm iepuras
întreg n, s, d, p
Citește n
p  $\leftarrow$  1
d  $\leftarrow$  1
s  $\leftarrow$  1
Cât timp (d < n)
  p  $\leftarrow$  p + 1
  s  $\leftarrow$  s * 2
  d  $\leftarrow$  d + s
Scrie p
Sfârșit

```



Programul scris în C++ este următorul:

```

#include <iostream>
using namespace std;
int main()
{
    //n = cati metri are de parcurs in total iepurasul
    //s = cati metri parcurge la un salt
    //d = distanta in metri parcursa pana la un anume pas de catre iepuras
    //p = numarul de pasi parcursi de catre iepuras
    int n, s, d, p;

    //introducem numarul n de la tastatura
    cout<<"Introduceti distanta traseului de parcurs: ";
    cin>>n;

    //calculam pentru fiecare pas parcurs de iepuras cati metri a parcurs pana la acel moment
    //apoi comparăm cu distanța care trebuie parcursă
    p=1;
    d=1;
    s=1;
    while (d<n)
    {
        p=p+1; //mai fac un pas
        s=s*2; //calculez cat sare iepurasul la acest pas
        d=d+s; //calculez distanta parcursa pana la acest moment
        cout<<endl;
        cout<<"Pana la pasul "<<p<<" iepurasul a parcurs "<<d<<"m"<<endl;
    }

    cout<<endl;
    cout<<"Iepurasul a parcurs distanta in "<<p<<" pasi"<<endl;
    return 0;
}
  
```



## Aplicați și investigați

- Rulați programul pentru  $d=10$  și urmăriți rezultatele în paralel cu tabelul din indicații.
- Considerați că iepurașul nu sare de două ori mai mult la fiecare pas, ci sare cu 2 metri mai mult la fiecare pas. Modificați saltul iepurașului la fiecare pas, astfel:  $s=s+2$ . Observați numărul de pași necesari iepurașului în acest caz, spre deosebire de cazul anterior.



## Descoperiți

### Lucrul cu cifrele numerelor – cea mai mică cifră a unui număr

► Să se afișeze cea mai mică cifră a unui număr natural nenul  $n$ , mai mic decât 100 000, introdus de la tastatură.

*Indicații:*

**Date de intrare:**  $nr$  = variabilă de tip întreg, reprezintă numărul introdus de la tastatură

**Date de manevră:**  $c$  = variabilă de tip întreg, reprezintă ultima cifră a numărului

**Date de ieșire:**  $m$  = variabilă de tip întreg, reprezintă cifra minimă a numărului

Pentru a rezolva exercițiul, utilizăm o variabilă  $m$  care reține, la fiecare pas al programului, valoarea celei mai mici cifre întâlnite până atunci. Valoarea lui  $m$  se compară cu fiecare cifră a numărului în parte, începând cu cifra unităților. Dacă cifra este mai mică, înseamnă că cifra respectivă este minimul de până atunci și valoarea lui  $m$  devine valoarea acesteia. Inițializarea variabilei  $m$  se face cu 9, cea mai mare valoare posibilă pentru o cifră. În acest fel, imediat după realizarea primei comparații cu prima cifră, valoarea minimului se va actualiza corespunzător și rezultatul nu va fi alterat. Extragerea unei cifre, compararea sa cu  $m$  și apoi eliminarea din număr a cifrei pe care tocmai am comparat-o (trunchierea numărului), se realizează cât timp mai avem cifre în număr.

Să luăm, spre exemplu,  $nr=723$ . Inițializăm  $m=9$ .

Nr. pas	Extragem ultima cifră, c	Verificare: $c < m$ ?	Eliminăm ultima cifră din număr și continuăm dacă mai avem cifre de verificat
1	3	$3 < 9 \Rightarrow m=3$	$nr=72, nr>0 \Rightarrow$ continuăm
2	2	$2 < 3 \Rightarrow m=2$	$nr=7, nr>0 \Rightarrow$ continuăm
3	7	$7 > 2 \Rightarrow m$ nu se modifică	$nr=0 \Rightarrow$ nu mai continuăm

**algoritm** CifraMin

întreg  $nr, m, c$

Citește nr

$m \leftarrow 9$

Cât timp ( $nr > 0$ )

$c \leftarrow nr \% 10$  //restul împărțirii la 10, extragem ultima cifră

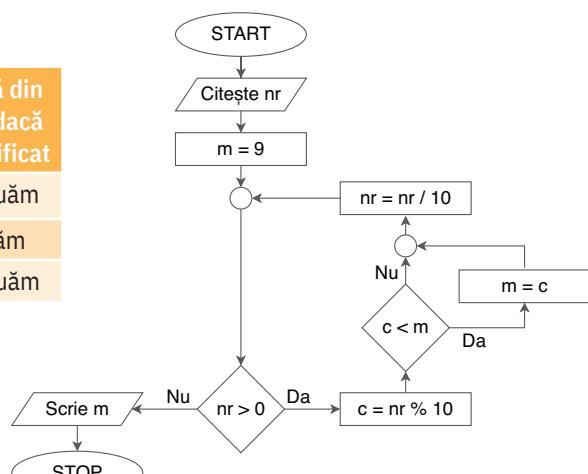
    Dacă ( $c < m$ ) atunci

$m \leftarrow c$

    nr  $\leftarrow nr / 10$  //câtul împărțirii la 10, trunchiem numărul

Scrie m

Sfârșit



## Rețineti

O secvență de cod în C++, generală pentru lucrul cu cifrele unui număr natural  $n$ , este:

```

int n, c;
cin >> n;
while(n > 0) // cat timp numarul mai are cifre
{
    c = n % 10;           //extragem ultima cifra a lui n
    ...prelucrare cifră c...
    n = n/10;            //eliminam ultima cifra a numarului
}
    
```

Modul de prelucrare a cifrei extrase c depinde de ceea ce se cere la un moment dat.

Programul C++ este următorul:

```
//nr = numarul caruia vrem sa ii determinam cifra minima
//c = ultima cifra a numarului
//m = cifra minima

#include <iostream>
using namespace std;

int main()
{
    int nr, c, m;
    cout<<"Introduceti numarul: ";
    cin>>nr;

    m=9;           //initializam minimul cu 9, cea mai mare cifra posibila

    while (nr>0)
    {
        c=nr%10; //extragem ultima cifra a numarului
        if (c<m) //verificam daca cifra este mai mica decat minimul de pana aici
            m=c;
        nr=nr/10; //eliminam ultima cifra pe care tocmai am prelucrat-o
    }
    cout<<"Cea mai mica cifra a numarului este "<<m;
    return 0;
}
```



## Aplicați și investigați

- ▶ Transformați problema anterioară astfel încât să se afișeze cea mai mare cifră a numărului. Pentru aceasta, inițializați variabila  $m$  (acum, aceasta fiind maximul) cu cea mai mică cifră posibilă, adică  $m=0$ . În continuare, verificați dacă fiecare cifră extrasă este mai mare decât maximul de până atunci (dacă  $c>m$ ). Dacă este mai mare, atunci noul maxim devine chiar acea cifră ( $m=c$ ). Investigați de ce trebuie să inițializăm maximul cu 0.



## Descoperiți

### Metoda înjumătățirii intervalului

Îl punem la încercare pe calculator. Trebuie să găsească un număr care este cuprins între 1 și 200, făcând cât mai puține încercări. Pentru aceasta, folosim o metodă de programare care poartă numele de *metoda înjumătățirii intervalului*. De fiecare dată se va propune spre verificare mijlocul intervalului în care știm că se încadreză numărul. Dacă acesta nu este egal cu numărul căutat, avem două variante:

- dacă numărul căutat este mai mic decât mijlocul, atunci mijlocul devine noul capăt la dreapta al intervalului;
- dacă numărul căutat este mai mare decât mijlocul, atunci mijlocul devine noul capăt la stânga al intervalului.

#### Indicații:

La fiecare pas se vor elimina jumătate din variante.

*Exemplu pentru numărul: 125*

Nr. pas	Limita inferioară	Limita superioară	Mijlocul (nr. pe care îl încercăm)	Verificare număr
1	1	200	100	$125 > 100$ , deci se consideră limitele 100 și 200
2	100	200	150	$125 < 150$ , deci se consideră limitele 100 și 150
3	100	150	125	$125 = 125$

**Date de intrare:**  $n$  = variabilă de tip întreg, reprezintă numărul introdus de la tastatură

$a, b$  = variabile de tip întreg, reprezintă capetele intervalului în care se caută numărul

**Date de manevră:**  $m$  = variabilă de tip întreg, reprezintă mijlocul intervalului

**Date de ieșire:**  $p$  = variabilă de tip întreg, reprezintă numărul de încercări până la găsirea numărului

## Structura repetitivă condiționată anterior

Programul C++ este următorul:

```
#include <iostream>
using namespace std;
int main()
{
    int n,a=1,b=200,p=1,m;

    cout<<"Introduceti numarul pe care doriti sa il caute calculatorul: ";
    cin>>n;

    m=(a+b)/2;
    cout<<"Se cauta intre "<<a<<" si "<<b<<", seincearca numarul "<<m<<endl;

    while(m!=n)
    {
        if(n<m) b=m;
        else a=m;
        p=p+1;
        m=(a+b)/2;
        cout<<"Se cauta intre "<<a<<" si "<<b<<", seincearca numarul "<<m<<endl;
    }
    cout<<endl<<"Numarul a fost gasit dupa "<<p<<" incercari"<<endl;

    return 0;
}
```



### Aplicați și investigați

- Înlocuiți cu două numere citite de la tastatură cele două limite (1 și 200), între care este necesar să se încadreze numărul care trebuie ghicit de către calculator.



### Exesăti

- 1.** Modificați programul *Iepurașul atlet*, astfel încât distanța  $n$  pe care o are de parcurs iepurașul să poată lua valoarea 0 m.

*Indicație:* Intrare: 0 m, Ieșire: 0 pași; Intrare: 3 m, Ieșire: 2 pași. Se va realiza inițializarea celor trei variabile  $p$ ,  $d$ ,  $s$  utilizate în program la 0 și se va trata separat, în corpul structurii repetitive, cazul în care iepurașul este la primul pas.

- 2.** Realizați un program C++ care să afișeze numărul de cifre al unui număr natural  $n$  introdus de la tastatură. Numărul  $n$  are minimum 3 și maximum 9 cifre.

*Indicație:* Intrare: 345, Ieșire: 3.

- 3.** Realizați un program C++ care să afișeze dublul celei mai mici cifre a unui număr întreg introdus de la tastatură.

*Indicație:* Intrare: 345, Ieșire: 6; Intrare: -57, Ieșire: 10.

- 4.** Scrieți un program C++ care să rezolve următoarea problemă:

Maria este în vacanță și citește o carte. În prima zi, citește două pagini. În cât timp citește Maria carteau, dacă aceasta are  $n$  pagini și citește în fiecare zi pară cu două pagini mai mult decât în ziua precedentă și în fiecare zi impară, cu trei pagini mai mult decât în ziua precedentă?

*Indicație:* Intrare: 14 pagini, Ieșire: 3 zile.



### Portofoliu



Realizați un program C++, **SumaCifre**, care calculează suma cifrelor unui număr întreg introdus de la tastatură. Dacă aceasta este pară, se afișează jumătatea ei. În caz contrar, se afișează pătratul ei. Salvați proiectul în **Proiecte C++**.

*Indicație:* Intrare: 826, Ieșire: 8; Intrare: 821, Ieșire: 121.

# 6 Structura repetitivă condiționată posterior



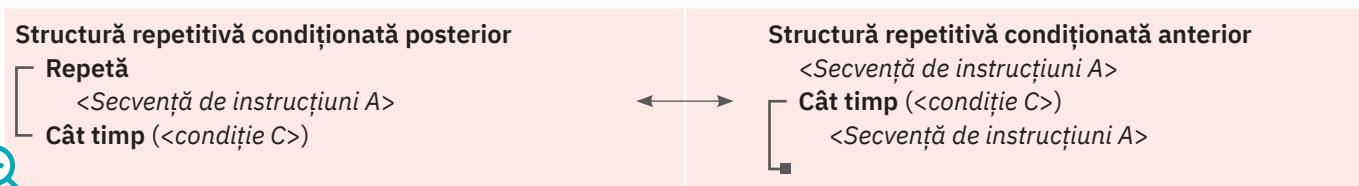
## Amintiți-vă

O structură repetitivă condiționată posterior se folosește atunci când avem un număr necunoscut de repetări necesare. Are două forme posibile: repetarea unei secvențe de instrucțiuni, *atât timp cât* este îndeplinită o condiție, sau, a doua formă, repetarea unei secvențe de instrucțiuni, *până când* o condiție devine adevărată.

Reamintim prima dintre formele structurii repetitive condiționate posterior:

Pseudocod	Schema logică	Algoritm de funcționare
<b>Repetă</b> <Secvență de instrucțiuni A> <b>Cât timp</b> (<condiție C>)	<b>Schema logică</b> 	<b>Pasul 1:</b> se execută <Secvență de instrucțiuni A> <b>Pasul 2:</b> se evaluează <condiție C> <b>Pasul 3:</b> Dacă valoarea <condiție C> este Adevarat Atunci se execută <Secvență de instrucțiuni A> și se revine la Pasul 1 Altfel se ieșe din secvență repetitivă.

Secvența de instrucțuni A se execută minimum o dată! Se poate transforma în secvență repetitivă condiționată anterior astfel:



## Descoperiți

În C++, sintaxa instrucțiunii repetitive condiționate posterior este următoarea:

```
Do
{
  secvență de instrucțiuni A
} while (condiție C);
```



Vom vedea, în cele ce urmează, cum anume folosim structura repetitivă condiționată posterior în limbajul C++. Vom lucra cu variabile de tip întreg și vom presupune că se introduc de la tastatură numere întregi, fără să fie nevoie să verificăm tipul datelor introduse.

## Validarea valorii unui număr introdus de la tastatură

Faceți un program pentru citirea unui număr de la tastatură până când numărul introdus este mai mic decât 10. La introducerea numărului apare mesajul „Introduceti un numar natural mai mic decat 10.”. Implementați mesajele de excepție conform tabelului de mai jos:

Intrare (a)	Ieșire	Explicații
14	Se solicită introducerea din nou a numărului. Apare mesajul „Introduceti un numar natural mai mic decat 10.”	Nu a fost respectată condiția ca numărul introdus să fie mai mic decât 10 și se solicită introducerea din nou a numărului.
10	Se solicită introducerea din nou a numărului. Apare mesajul „Introduceti un numar natural mai mic decat 10.”	Nu a fost respectată condiția ca numărul introdus să fie mai mic decât 10 și se solicită introducerea din nou a numărului.
3	Bravo! Ai introdus corect! Nr. introdus este: 3	A fost respectată condiția ca numărul introdus să fie mai mic decât 10.

## Structura repetitivă condiționată posterior

*Indicații:*

**Date de intrare:**  $a$  = variabilă de tip întreg, reprezintă numărul introdus de la tastatură

**Date de ieșire:**  $a$  = variabilă de tip întreg, reprezintă numărul corect introdus de la tastatură

Pentru a ne asigura că datele noastre de intrare sunt corecte din punctul de vedere al valorii, vom realiza citiri repetitive ale valorii cât timp aceasta este incorect introdusă. Algoritmul în pseudocod este:

**algoritm** validare

**întreg** a

**Repetă**

        Scrie 'Introduceti un numar natural mai mic decat 10: '

        Citește a

**Cât timp** ( $a > 10$ )

            Scrie 'Bravo! Ai introduc corect! Nr. introduc este: ', a

        Sfârșit

Programul C++ este:

```
+ #include <iostream>
using namespace std;
int main()
{    int a;
    do
    {    cout<<"Introduceti un numar natural mai mic decat 10: ";
        cin>>a;
    } while(a>10);
    cout<<"Bravo! Ai introduc corect! Nr. introduc este: " << a;
    return 0;
}
```



Dacă dorim să afișăm un mesaj pentru utilizator în cazul în care acesta introduce un număr care nu corespunde condiției, adică este mai mare decât 10, programul nostru se modifică astfel:

```
#include <iostream>
using namespace std;
int main()
{    int a;
    a=0;

    do
    {    if (a>10) cout<<"Numarul este prea mare!"<<endl;
        cout<<"Introduceti un numar natural mai mic decat 10: ";
        cin>>a;
    } while (a>10);

    cout<<"Bravo! Ai introduc corect! Nr. introduc este: " << a;
    return 0;
}
```

## Algoritmul lui Euclid

Aplicați algoritmul lui Euclid pentru a calcula cel mai mare divizor comun pentru două numere naturale mai mici decât 10000, introduse de la tastatură.

*Indicații:*

Presupunem că avem două numere  $a$  și  $b$ , naturale, nenule.

Notăm:

- **cmmdc(a, b)** sau **(a, b)** – cel mai mare divizor comun al numerelor  $a$  și  $b$
- **c** câtul împărțirii lui  $a$  la  $b$
- **r** restul împărțirii lui  $a$  la  $b$ .

deîmpărțit      împărțitor      cât  
 $a : b = c$  rest  $r$

Algoritmul lui Euclid este o metodă foarte eficientă pentru calculul **cmmdc(a,b)**. Vom detalia în cele ce urmează **algoritmul lui Euclid prin împărțiri repetitive**.

**Date de intrare:**  $a, b$  = variabile de tip întreg, reprezintă numerele introduse de la tastatură

**Date de manevră:**  $r$  = variabilă de tip întreg, reprezintă restul împărțirii numerelor

**Date de ieșire:**  $a$  = variabilă de tip întreg, reprezintă cmmdc( $a, b$ ).

#### Algoritmul în pseudocod este următorul:

Folosind structura repetitivă condiționată anterior

algoritm Euclid  
întreg  $a, b, c, r$

**Citește a, b**

**Cât timp** ( $b \neq 0$ )

```
r ← a % b
a ← b
b ← r
```

**Scrie** 'cmmdc='; a  
Sfârșit

Folosind structura repetitivă condiționată posterior

algoritm Euclid  
întreg  $a, b, c, r$

**Citește a, b**

**Dacă** ( $b \neq 0$ ) **atunci**

**Repetă**

```
r ← a % b
a ← b
b ← r
```

**Cât timp** ( $b \neq 0$ )

**Scrie** 'cmmdc='; a  
Sfârșit

Programul în C++ utilizând instrucțiunea repetitivă condiționată posterior este următorul:

```
#include <iostream>
using namespace std;

int main()
{
    int a, b, r;
    cout << "a="; cin >> a;
    cout << "b="; cin >> b;

    if (b != 0)
    {
        do {
            r = a % b;
            a = b;
            b = r;
        } while (r != 0);
    }

    cout << "cmmdc=" << a;
    return 0;
}
```

► Vom urmări un exemplu pentru funcționarea structurii repetitive condiționată posterior din program, pentru valorile de intrare  $a=24$  și  $b=18$ .

	Iterația	a	b	c	r	Valoarea de adevăr a expresiei ( $b \neq 0$ )	Acțiune
do {   r = a % b;	1	24	18	1	6		
a = b;		18	18	1	6		
b = r;		18	6	1	6		
} while (b != 0);						Adevărat	Reintrăm în instrucțiunea repetitivă do ...while
do {   r = a % b;	2	18	6	3	0		
a = b;		6	6	3	0		
b = r;		6	0	3	0		
} while (b != 0);						Fals	Ieșim din instrucțiunea repetitivă do ...while

#### Divizori

Se citește de la tastatură un număr natural nenul mai mic decât 10000. Afipați toti divizorii săi proprii (divizorii diferiți de 1 și de el însuși), apoi afipați numărul lor.

*Indicații:*

Cel mai mic posibil divizor propriu al unui număr natural nenul este 2, iar cel mai mare divizor posibil este  $n/2$ . Pentru a găsi toti divizorii unui număr trebuie să încercăm fiecare număr în parte de la 2 la  $n/2$ . Dacă acesta îl divide pe  $n$ , atunci este divizor, îl afișăm și îl numărăm.

**Date de intrare:**  $n$  = variabilă de tip întreg, reprezintă numărul căruia dorim să îl determinăm divizorii.

**Date de manevră:**  $d$  = variabilă de tip întreg, reprezintă un potențial divizor.

**Date de ieșire:**  $nrDiv$  = variabilă de tip întreg, reprezintă numărul de divizori ai numărului introdus.



## Structura repetitivă condiționată posterior

Programul C++ este următorul:

```
#include <iostream>
using namespace std;
int main()
{
    int n, d, nrDiv;
    cout<<"n="; cin>>n;

    nrDiv=0;      //initializez cu 0 numarul de divizori proprii
    d=2;          //cel mai mic potential divizor este 2

    if (d<=n/2) //cel mai mare posibil divizor este n/2
    {
        do {
            if (n%d==0)      //daca d este divizor
            {
                cout<<d<<" "; //afisez divizorul
                nrDiv++;       //numar acest divizor gasit
            }
            d++;             //trec la urmatorul posibil divizor
        } while (d<=n/2); //cel mai mare posibil divizor este n/2
    }

    if (nrDiv==0) cout<<endl<<"Nr "<<n<<" nu are divizori proprii.";
    else cout<<endl<<"Nr "<<n<<" are "<<nrDiv<<" divizori proprii.";

    return 0;
}
```



### Aplicați și investigați

- ▶ Parcurgeți cele trei programe anterioare pas cu pas în modul **Depanare (Debug)** și urmăriți valorile variabilelor.



### Erexsați

1. Modificați programul C++ *Divizori*, astfel încât să afișeze pe ecran dacă numărul introdus de la tastatură este prim sau nu.  
*Indicație:* Numerele prime sunt cele care nu au divizori proprii. Cel mai mic număr prim este 2. Intrare: 47, Ieșire: „Nr. 47 este prim”; Intrare: 46, Ieșire: „Nr. 46 nu este prim”.
2. Într-un program C++ se citesc de la tastatură două numere naturale. Afișați numărul care are cei mai mulți divizori.  
*Indicație:* Intrare: a=5, b=7, Ieșire: „Cele două numere sunt prime”; Intrare: a=6, b=3, Ieșire: „Nr. 6 are mai mulți divizori decât numărul 3”; Intrare: a=6, b=10, Ieșire: „Nr. 6 are același număr de divizori ca numărul 10”.
3. Maria are o sumă de bani și se duce la cumpărături. Faceți un program în care Maria să introducă ceea ce își dorește să cumpere (preț și cantitate), până când nu mai are bani. Programul o atenționează în cazul în care nu are suficienți bani ca să cumpere produsul introdus.  
*Indicație:* Intrare: Maria are 100 RON. Face cumpărături până își termină banii:

Cantitate achiziționată	Preț produse achiziționate	Suma de bani rămasă	Mesaj program
1	10 RON	100-1*10=90 RON	
2	15 RON	90-2*15=60 RON	
1	100 RON	60 RON<100 RON	Nu ai bani suficienți!
1	60 RON	60-60=0 RON	S-au terminat banii!



### Portofoliu



Realizați un program C++ pentru următoarea problemă:  
Afișați suma divizorilor unui număr natural de trei cifre.

*Indicație:* Intrare: 121, Ieșire: 1+11+121=133.

4. Victor are mai multe note la o materie și vrea să știe cât este media lui. Faceți un program C++ care citește aceste note până când este introdusă nota 0, apoi calculează media aritmetică și o tipărește.  
*Indicație:* Intrare: 10, 9, 0, Ieșire: 9.5.
5. Rezolvați problema în C++: Se citesc de la tastatură mai multe numere, până când se întâlnește primul număr multiplu de 3. Afișați pătratul celui mai mare număr citit până atunci.  
*Indicație:* Intrare: 2, 1, 11, 3, Ieșire: 121.

# 7 Structura repetitivă cu contor (cu număr cunoscut de pași)

## Amintiți-vă

Folosim această structură repetitivă ori de câte ori trebuie să **repetăm anumite operații de un număr cunoscut de ori**. Aceasta poate fi transformată, după nevoie, în structură repetitivă condiționată anterior sau posterior.

Forma structurii repetitive cu contor este:

### Pseudocod

pentru  $x \leftarrow$  <valoare inițială>, <valoare finală> [, <pas>]  
execută <Secvență de instrucțiuni A>



**Observație:** Dacă <pas> nu este specificat, atunci el va fi implicit 1.



### Caz 1:

Dacă: valoare inițială  $\leq$  valoare finală  
Atunci: pas este număr întreg pozitiv nenul

### Caz 1:

Dacă: valoare inițială  $\geq$  valoare finală  
Atunci: pas este număr întreg negativ nenul

Caz 1	Caz 2
<p><b>Algoritm de funcționare</b> Valoarea variabilei x crește de la &lt;valoare inițială&gt;, până la &lt;valoare finală&gt;, cu câte un &lt;pas&gt; (<i>la fiecare iterare, la valoarea lui X se adaugă un număr pozitiv</i>).   <b>Pasul 1:</b> <math>x =</math> &lt;valoare inițială&gt;  <b>Pasul 2:</b>            Cât timp <math>x \leq</math> &lt;valoare finală&gt; execută            &lt;Secvență de instrucțiuni A&gt;  <math>x = x + &lt;\text{pas}&gt;</math></p>	<p><b>Algoritm de funcționare</b> Valoarea variabilei X scade de la &lt;valoare inițială&gt;, până la &lt;valoare finală&gt;, cu câte un &lt;pas&gt; (<i>la fiecare iterare, la valoarea lui X se adaugă un număr negativ</i>).   <b>Pasul 1:</b> <math>x =</math> &lt;valoare inițială&gt;  <b>Pasul 2:</b>            Cât timp <math>x \geq</math> &lt;valoare finală&gt; execută            &lt;Secvență de instrucțiuni A&gt;  <math>x = x + &lt;\text{pas}&gt;</math></p>
<p><b>Schemă logică</b></p> <pre> graph TD     Start["x ← &lt;valoare inițială&gt;"] --&gt; Cond1{"x &lt;= &lt;valoare finală&gt;"}     Cond1 -- Da --&gt; BlockA["Secvență de instrucțiuni A"]     BlockA --&gt; Step["x ← x + &lt;pas&gt;"]     Step --&gt; Cond1     Cond1 -- Nu --&gt; End   </pre>	<p><b>Schemă logică</b></p> <pre> graph TD     Start["x ← &lt;valoare inițială&gt;"] --&gt; Cond2{"x ≥ &lt;valoare finală&gt;"}     Cond2 -- Da --&gt; BlockA["Secvență de instrucțiuni A"]     BlockA --&gt; Step["x ← x + &lt;pas&gt;"]     Step --&gt; Cond2     Cond2 -- Nu --&gt; End   </pre>



## Descoperiți

În cele ce urmează vom învăța care este sintaxa acestei instrucțiuni repetitive în C++ și cum se folosește, în cele două cazuri descrise mai sus:

Caz 1	Caz 2
<pre>for (X=valoare inițială; X&lt;=valoare finală; X=X+pas p) { secvență de instrucțiuni A }</pre> <p>Exemplu:  <code>for (x=1; x&lt;=10; x=x+1) cout&lt;&lt;"Bravo!"&lt;&lt;endl;</code></p>	<pre>for (X=valoare inițială; X&gt;=valoare finală; X=X+pas p) { secvență de instrucțiuni A }</pre> <p>Exemplu:  <code>for (x=10; x&gt;=1; x=x-1) cout&lt;&lt;"Bravo!"&lt;&lt;endl;</code></p>

## Structura repetitivă cu contor (cu număr cunoscut de pași)

O structură repetitivă devine repetitivă la infinit, dacă nu este specificată niciuna dintre expresiile pentru instrucțiunea **for**. Când este absentă condiția unui **for**, implicit se consideră că are valoarea adevărat. Repetarea la infinit se poate încheia apăsând **Ctrl+C**.

**for ( ; ; )**  
**{secvență de instrucțiuni}**      ↔      **while (1)**  
**{secvență de instrucțiuni}**

```
#include <iostream>
using namespace std;
int main () {
    for( ; ; ) {
        cout<<"Aceasta structura se repeta la infinit. \n";
    }
    return 0;
}
```



De asemenea, pot lipsi din cele trei expresii ale instrucțiunii **for**. Mai jos este un caz în care lipsesc două expresii.

**for ( ; expresie ; )**      ↔      **while (expresie)**  
**{secvență de instrucțiuni}**           **{secvență de instrucțiuni}**

Vom vedea câteva exemple de programe care utilizează structuri repetitive cu contor. Vom lucra cu variabile de tip întreg și vom presupune că se introduc de la tastatură numere întregi, fără să fie nevoie să verificăm tipul datelor introduse.

### Produsul a $n$ numere

Afișați valoarea lui  $n!$ , unde  $n$  este un număr natural mai mic sau egal cu 12, știind că  $n!=1 * 2 * ... * n$ , iar  $0!=1$ .

De exemplu: pentru  $n=3$  se afișează  $3!=6$ ; pentru  $n=10$  se afișează  $10!=3628800$

*Indicații:*

**Date de intrare:**  $n$  = variabilă de tip întreg, numărul pentru care trebuie să calculăm  $n!$

**Date de manevră:**  $i$  = variabilă de tip întreg, o utilizăm în cadrul unei structuri repetitive

**Date de ieșire:**  $p$  = variabilă de tip întreg, o utilizăm pentru stocarea produsului  $1*2*3*...*n$ , adică  $n!$

Pentru a calcula  $n!$ , trebuie să calculăm repetitiv produsul:  $1*2*3*...*n$ .

Întotdeauna, când avem de calculat produsul unor numere, folosim o variabilă ( $p$ ) care îl stochează și care se inițializează cu 1 (1 nu schimbă valoarea unui produs atunci când acesta se înmulțește cu el).

La fiecare pas, vom calcula  $p=i$ , unde  $i$  este o variabilă de manevră care ia pe rând valorile de la 1 la  $n$ , valoarea lui  $i$  crescând cu 1 de fiecare dată.

```
i=1 p = 1
i=2 p = p * 2 = 1 * 2 = 2
i=3 p = p * 3 = 1 * 2 * 3 = 6
...
i=n p = p * n = 1 * 2 * ... * (n-1) * n = n!
```



**+** Cel mai firesc, acest program se scrie utilizând structura repetitivă cu contor, însă poate fi ușor transformat astfel încât să utilizeze o structură repetitivă condiționată anterior sau posterior.

Utilizarea structurii repetitive cu contor	Utilizarea structurii repetitive condiționate anterior	Utilizarea structurii repetitive condiționate posterior
<b>algoritm factorial</b> întreg n, p, i <b>Citește</b> n $p \leftarrow 1$ <b>Pentru</b> $i \leftarrow 1, n, 1$ $\quad p \leftarrow p * i$ <b>Scrie</b> p <b>Sfârșit</b>	<b>algoritm factorial</b> întreg n, p, i <b>Citește</b> n $p \leftarrow 1$ $i \leftarrow 1$ <b>Cât timp</b> ( $i \leq n$ ) <ul style="list-style-type: none"> <li><math>\quad p \leftarrow p * i</math></li> <li><math>\quad i \leftarrow i + 1</math></li> </ul> <b>Scrie</b> p <b>Sfârșit</b>	<b>algoritm factorial</b> întreg n, p, i <b>Citește</b> n $p \leftarrow 1$ $i \leftarrow 1$ <b>Execută</b> <ul style="list-style-type: none"> <li><math>\quad p \leftarrow p * i</math></li> <li><math>\quad i \leftarrow i + 1</math></li> </ul> <b>Cât timp</b> ( $i \leq n$ ) <ul style="list-style-type: none"> <li><b>Scrie</b> p</li> <li><b>Sfârșit</b></li> </ul>

### 1. Utilizarea structurii repetitive cu contor:

```
#include <iostream>
using namespace std;
int main()
{   int i, n, p=1;

    cout<< "n=";
    cin>>n;

    for (i=1; i<=n; i++)
        p=p*i;      //inmultim fiecare termen in parte

    //afisam valoarea lui n!
    cout<< n<<"! = "<< p;
    return 0;
}
```

### 2. Utilizarea structurii repetitive condiționate anterior:

```
#include <iostream>
using namespace std;
int main()
{   int i=1, n, p=1;
    cout<< "n=";
    cin>>n;

    while (i<=n)
    {
        p=p*i; //inmultim fiecare termen in parte
        i++;   //crestem pe i cu o unitate
    }

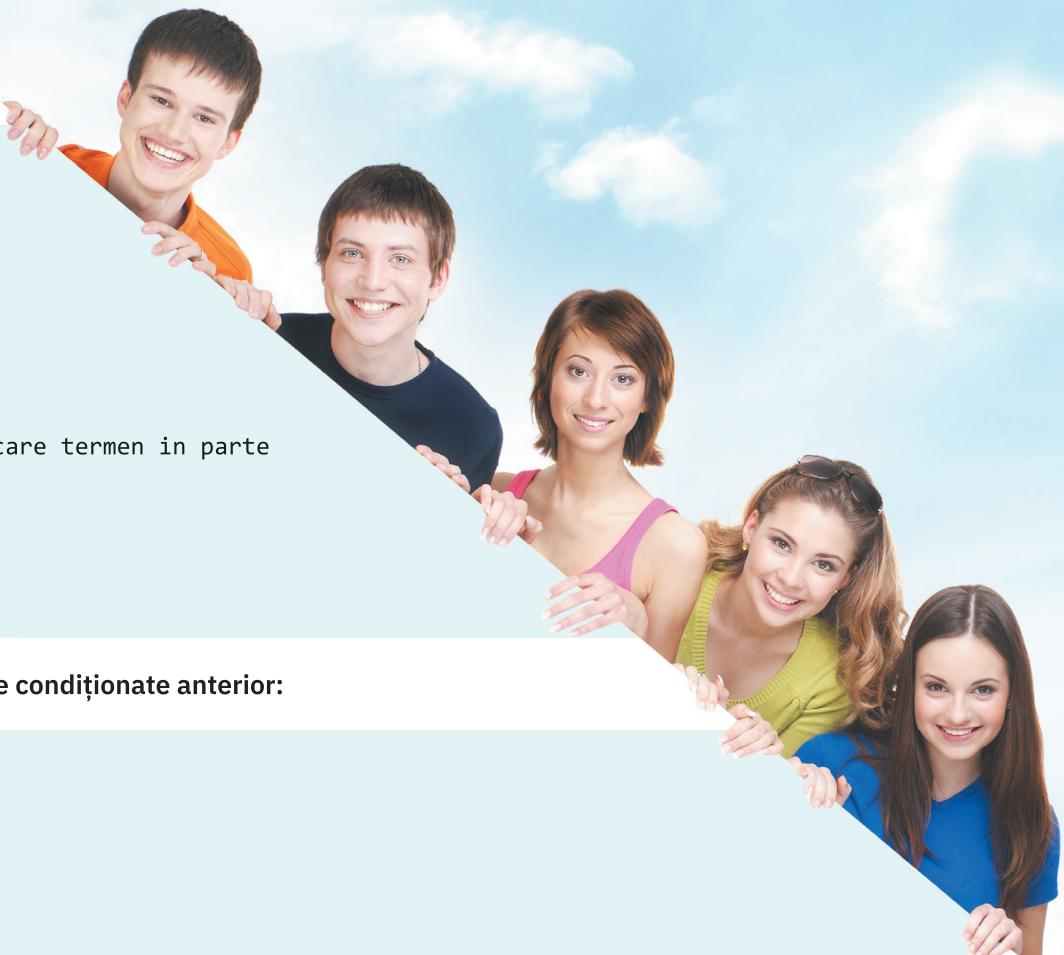
    //afisam valoarea lui n!
    cout<< n<<"! = "<< p;
    return 0;
}
```

### 3. Utilizarea structurii repetitive condiționate posterior:

```
#include <iostream>
using namespace std;
int main()
{   int i=1, n, p=1;
    cout<< "n= ";
    cin>>n;

    do
    {
        p=p*i; //inmultim fiecare termen in parte
        i++;   //crestem pe i cu o unitate
    } while (i<=n);

    //afisam valoarea lui n!
    cout<<n<<"! = "<< p;
    return 0;
}
```





## Aplicați și investigați

- Adăugați programului anterior validarea valorii numărului  $n$  introdus de la tastatură. Cât timp valoarea introdusă nu este corectă ( $n < 0 \text{ || } n > 12$ ), resolicitați introducerea. Executați programul, utilizând ca dată de intrare -4 și observați rezultatul.
- Ignorați limitarea numărului  $n$  la 12 și introduceți ca date de intrare pentru  $n$  valorile 13, 14 și 15. Observați ce se întâmplă cu valorile lui  $13!, 14!, 15!$ . Citiți informația din caseta **Ştiăt că...?** și explicați rezultatele obținute.
- Modificați programul de mai sus astfel încât să calculeze suma  $S = 1 + 2 + 3 + \dots + n$ , unde  $n$  este numărul natural introdus de la tastatură, utilizând o structură repetitivă cu contor. În cazul acestui program, vom inițializa suma cu 0, acesta fiind numărul neutru la adunare.
- Faceți un program care să calculeze aceeași sumă folosind formula lui Gauss:  $1 + 2 + \dots + n = n * (n + 1) / 2$ . Comparați rezultatele.



## Descoperiți

### Ultima cifră a unei valori calculate

Determinați ultima cifră a numărului  $n!$ , unde  $n$  este un număr natural introdus de la tastatură, mai mic sau egal cu 12.

#### Indicații:

Vom dezvolta programul de calcul al produsului numerelor naturale consecutive prezentat anterior, adăugând la final extragerea ultimei cifre a acestuia (restul împărțirii la 10).

**Date de intrare:**  $n$  = variabilă de tip întreg, numărul pentru care trebuie să calculăm  $n!$

**Date de manevră:**  $i$  = variabilă de tip întreg, o utilizăm la calculul lui  $n!$

$p$  = variabilă de tip întreg, stocarea valoare lui  $n!$

**Date de ieșire:**  $c$  = variabilă de tip întreg, o utilizăm pentru stocarea ultimei cifre a lui  $p$

**Restricții:**  $0 \leq n \leq 12$



Valoare la intrare ( $n$ )	$n!$	Ultima cifră a lui $n!$
3	$3!=1*2*3=6$	6
5	$5!=1*2*3*4*5=120$	0

Algoritmul scris în pseudocod este:

```

algoritm ultimaCifra întreg n, c, i
Citește n, c, i
p ← 1
Pentru i ← 1, n, 1
    p ← p * i
    c ← p%10
    Scrie c
Sfărșit

```

Programul C++ este cel alăturat.

### Ştiăt că...?



În cazul în care numărul  $n$  ar fi mai mare decât 12, atunci valoarea lui  $n!$  ar depăși valoarea maximă pe care o poate lua o variabilă de tip întreg (2147483647).

```

#include <iostream>
using namespace std;
int main()
{
    //n = numarul solicitat in problema, variabila de intrare
    //p = rezultatul lui n!, variabila de manevră
    //i = variabila de manevră
    //c = ultima cifră a lui n!, variabila de ieșire
    int n, i, p, c;

    cout << "n= ";
    cin >> n;

    //calculam p=n!
    p=1;
    for (i=1; i<=n; i++)
        p=p*i;

    //calculam ultima cifră a lui p=n!
    c=p%10;

    //afisam ultima cifră a lui p=n!
    cout << "Ultima cifră a lui n! este: " << c << endl;
}

return 0;
}

```

## Căciula lui Moș Crăciun

Desenați căciula lui Moș Crăciun din simboluri „&”. Pentru aceasta, citiți un număr natural  $n$ , cu cel mult două cifre. Afisați pe ecran o figură formată din caracterul „&”, ca în exemplul de mai jos, pentru  $n=3$ .

```
&
&&&
&&&&
```

*Indicații:*

Vom vedea în continuare cum trebuie să gândim acest algoritm. Vom construi pe rând fiecare linie din spații și caractere „&”. Vom nota cu  $i$  numărul liniei și cu  $S$  spațiul. Se observă, din tabelul de mai jos, că pe fiecare rând din cele  $n$  rânduri avem  $(n-i)$  spații și  $(2*i-1)$  caractere „&”.



Cazuri pt $n=3$	$i$	$S$	&
SS&	$i=1$	2	1
S&&&	$i=2$	1	3
&&&&	$i=3$	0	5
<b>Regula</b>		$n-i$	$2*i-1$

**Date de intrare:**  $n$  = variabilă de tip întreg, numărul de linii

**Date de manevră:**  $i$  = variabilă de tip întreg, o utilizăm pentru rândul curent pe care îl construim

**Restricții:**  $0 < n \leq 99$

Programul C++ este următorul:

```
#include <iostream>
using namespace std;

int main()
{
    int i, j, n;
    for (i=1; i<=n; i++)
    {
        for (j=1; j<=n-i; j++) cout<<" ";
        for (j=1; j<=2*i-1; j++) cout<<"&";
        cout<<endl;
    }
    return 0;
}
```

```
Introduceti numarul: 3
&
&&&
&&&&

Process returned 0 (0x0)  execution time : 4.150 s
Press any key to continue.
```



## Aplicați și investigați

► Transformați programul anterior, astfel încât căciula lui Moș Crăciun să fie acoperită cu steluțe, apoi cu cifre.

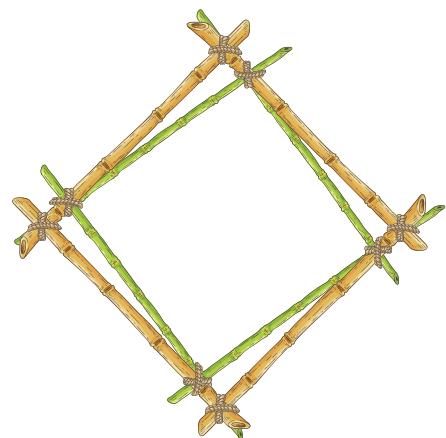
## Descoperiți

### Romb din numere

Se citește un număr natural  $n$  cu cel mult două cifre. Afisați pe ecran o figură sub formă de romb formată cu numerele naturale de la 1 la  $n$ , ca în exemplul alăturat pentru  $n=4$ .

*Indicații:* La fel ca la programul anterior, considerați fiecare linie în parte și gândiți-vă la regula sa de formare; câte spații și câte cifre avem la fiecare rând.

```
1
222
33333
4444444
33333
222
1
```



## Structura repetitivă cu contor (cu număr cunoscut de pași)

**Date de intrare:**  $n$  = variabilă de tip întreg, numărul conform căruia construim rombul

**Date de manevră:**  $i, j$  = variabile de tip întreg, le utilizăm la realizarea rombului

**Restricții:**  $0 \leq n \leq 99$

Programul C++ este următorul:

```
#include <iostream>
using namespace std;
int main()
{ int n, i, j;
cout<<"Introduceti numarul: ";
cin>>n;
for(i=1;i<=n;i++)
{ for(j=1;j<=n-i;j++)
    for(j=1;j<=2*i-1;j++)
        cout<<i;
    cout<<endl;
}
for(i=n-1;i>=1;i--)
{ for(j=1;j<=n-i;j++)
    for(j=1;j<=2*i-1;j++)
        cout<<i;
    cout<<endl;
}
return 0;
}
```

```
Introduceti numarul: 5
1
222
33333
4444444
555555555
4444444
33333
222
1

Process returned 0 (0x0)   execution time : 3.457 s
Press any key to continue.
```



### Aplicați și investigați

- ▶ Modificați programul anterior, astfel încât spațiile să fie înlocuite cu caracterul „\*”.



### Exersați

- Realizați un program C++ care să determine ultima cifră a numărului  $n!+6n$ , unde  $n$  este un număr natural introdus de la tastatură, mai mic decât 10.

*Indicații:* Intrare:  $n=2$ , Ieșire:  $n!+6n=2+36=38$ , se afișează 8.  
Restricții:  $0 \leq n \leq 10$ .

- Realizați un program C++ care să tipărească un triunghi cu vârful în jos, din steluțe. Acest triunghi trebuie să aibă atâtea linii câte indică valoarea unui număr natural introdus de la tastatură.

*Indicații:* Intrare:  $n=3$ , Ieșire: se va afișa triunghiul:  
\* \* \* \* \*  
\* \* \*  
\*

- Se dă secvența alăturată (A). Scrieți în caiet numărul care reprezintă repetările afișării textului „Uraaa!”.

**A.** 100;      **B.** 70;      **C.** 700;      **D.** 7000.

- Care este valoarea variabilei  $a$  de tip întreg, după rularea secvenței de program alăturate (B)?

- Structura repetitivă cu contor se poate transforma în structură repetitivă condiționată anterior? Cum?

- Ce afișează secvența alăturată de cod (C) pentru  $r=3$ ?

**A.** 1 2 3      **B.** 1 2 3      **C.** 1 1 1  
1 2            1 2            2 2  
1                1                3

### Portofoliu



Realizați un program C++ pentru a calcula suma primelor  $n$  numere naturale pare, folosind structura repetitivă cu contor ( $n \leq 1000$ ).

*Indicație:* Intrare:  $n=3$ , Ieșire: 6; Intrare:  $n=0$ , Ieșire: 0.

**A**  
int i, j, k;  
for(i=1;i<=10;i++)  
 for(j=1;j<=7;j++)  
 for(k=1;k<=10;k++)  
 cout<<"Uraaa!";

**B**  
int a, i, j;  
a=0;  
for(i=1;i<=2;i++)  
 for(j=1;j<=7;j++)  
 a++;

**C**  
for(int i = r; i >= 1; --i)  
{ for(int j = 1; j <= i; ++j)  
 cout << j << " ";  
 cout << endl;  
}

## 8 Aplicații de fizică în C++



## Amintiti-vă

După cum deja știți, pentru a realiza o animație, mișcarea elementului dorit se descompune pe mai multe cadre care sunt afisate succesiv, mai rapid sau mai lent, având câte o mică pauză între ele.



# Descoperiți

În C++, cadrele unei animații sunt de fapt reafișări ale aceluiași element, iar animația va fi o afișare succesivă a acestora, cu o pauză între ele.

Vom învăța în cele ce urmează două funcții care ne vor ajuta să realizăm programele noastre:

- 1. system()** – funcție utilizată pentru a apela o comandă a sistemului de operare dintr-un program C/C++. Este o funcție standard definită în biblioteca `<cstdlib>` în C++. Vom utiliza această funcție în două scopuri:

- a. **Ştergerea ecranului consolă.** Dacă o apelăm cu parametrul „`cls`”, atunci aceasta va avea ca rezultat ştergerea conţinutului ferestrei consolă (fereastra în care apare rezultatul programului).

*Exemplu:* system("cls");

- b. Modificarea culorilor în ecranul consolă.** Putem schimba culorile fundalului și textului ferestrei consolă, apelând funcția cu parametrul „color XY”, unde X este culoarea fundalului și Y este culoarea textului. Tabelul din dreapta prezintă posibilele valori pentru X și Y.

*Exemplu: schimbarea culorii de fundal în azuriu și a culorii textului în verde: system("color B2");*

- 2. sleep()** – funcție utilizată pentru a opri execuția programului pentru o perioadă de timp. Durata pauzei este dată în milisecunde. Pentru a o putea utiliza trebuie să includem <windows.h>.

*Exemplu:* introducerea unei pauze de 1 s (1000 ms) în program:  
`Sleep(1000);`

Vom vedea în cele ce urmează două aplicații dezvoltate în C++ pentru disciplina Fizică. În prima aplicație vom învăța cum să realizăm o animație C++ în fereastra consolă.



0	Negru	8	Gri
1	Albastru	9	Albastru-deschis
2	Verde	A	Verde-deschis
3	Azuriu (Aqua)	B	Azuriu-deschis
4	Roșu	C	Roșu-deschis
5	Violet	D	Violet-deschis
6	Galben	E	Galben-deschis
7	Alb	F	Alb-luminos

## Miscarea rectilinie si uniformă



Cum realizăm un program care afisează pe ecran miscarea rectilinie uniformă a unei bărcute?

Să ne amintim definiția: *Mișcarea rectilinie uniformă este mișcarea a cărei traectorie este o linie dreaptă și de-a lungul căreia vectorul viteză rămâne constant*.

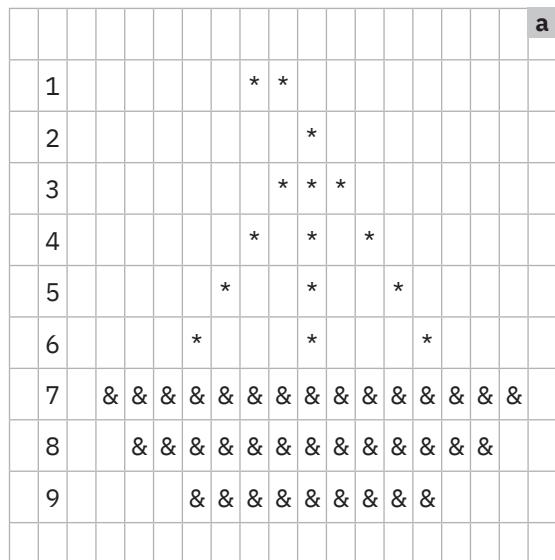
Vom deplasa bărcuța pe orizontală, înspre dreapta, în linie dreaptă, miscarea va fi rectilinie.



 Mai întâi, ne gândim cum desenăm barca. O variantă simplă de desen ar fi cea din *îmaginea a*. Pentru a o afișa în C++, trebuie să numărăm câte spații și câte caractere avem pe fiecare linie în parte, apoi să afișăm fiecare linie în parte.

Pentru animația bărcii (mișcarea la dreapta) vom repeta următoarele acțiuni (de 30 ori):

- 1. Afisare cadru:** afisăm barca pe ecran. La fiecare afisare vom mări numărul de spații de la începutul liniei, cu atâtea spații cât este numărul iterării. Acest lucru va duce la desenarea bărcuței din ce în ce mai la dreapta.
  - 2. Pauză:** Lăsăm barca vizibilă pe ecran un timp ( $t$  milisecunde) – pentru aceasta, vom insera în programul nostru o pauză de  $t$  milisecunde.
  - 3. Stergem ecranul.**



Dacă pauza are aceeași lungime  $t$  la fiecare iterare de afișare a bărcuței (adică a cadrului animației), atunci mișcarea va fi uniformă.



```
#include <iostream>
#include <windows.h>
#include <cstdlib>
using namespace std;

int main()
{ int t=100; //timpul de pauza dintre cadre, in milisecunde
    system("color B2"); //coloreaza fundalul in azuriu-deschis si textul in verde
    for (int i=1; i<=30; i++) //repetam de 30 ori desenarea bărcutei, din ce in ce mai la dreapta
    {
        //afisam bărcuta linie cu linie
        for (int j=1; j<=i; j++) //afisam „i” spatii la inceputul liniei
            cout<<" ";
        cout << " ** " << endl; //afisam linia 1 din desenul bărcii
        for (int j=1; j<=i; j++) //afisam „i” spatii la inceputul liniei
            cout<<" ";
        cout << " * " << endl; //afisam linia 2 din desenul bărcii
        for (int j=1; j<=i; j++) //afisam „i” spatii la inceputul liniei
            cout<<" ";
        cout << " *** " << endl; //afisam linia 3 din desenul bărcii
        for (int j=1; j<=i; j++) //afisam „i” spatii la inceputul liniei
            cout<<" ";
        cout << " * * * " << endl; //afisam linia 4 din desenul bărcii
        for (int j=1; j<=i; j++) //afisam „i” spatii la inceputul liniei
            cout<<" ";
        cout << " * * * " << endl; //afisam linia 5 din desenul bărcii
        for (int j=1; j<=i; j++) //afisam „i” spatii la inceputul liniei
            cout<<" ";
        cout << " &&&&&&&&&&&&& " << endl; //afisam linia 7 din desenul bărcii
        for (int j=1; j<=i; j++) //afisam „i” spatii la inceputul liniei
            cout<<" ";
        cout << " &&&&&&&&&&& " << endl; //afisam linia 8 din desenul bărcii
        for (int j=1; j<=i; j++) //afisam „i” spatii la inceputul liniei
            cout<<" ";
        cout << " &&&&&&&& " << endl; //afisam linia 9 din desenul bărcii
        Sleep(t); //facem o pauza de t ms
        system("cls"); //stergem fereastra consola
    }
    return 0;
}
```



## Aplicații și investigații

- Modificați valoare variabilei  $t$  din programul anterior și investigați cum crește sau scade viteza animației.
- Modificați culorile de afișare a textului și a fundalului și observați care este cea mai plăcută combinație.
- Investigați cum trebuie să faceți ca să deplasați bărcuța cu viteze diferite în animație. Modificați programul de mai sus în consecință.

*Indicație:* pentru a micșora viteza, puneti o pauză din ce în ce mai mare între cadrele afișate. Pentru a mări viteza, durata pauzei trebuie să scadă de la o iterare la alta.


**Descoperiți**

## Lucrul mecanic

Dar dacă avem o problemă de fizică legată de lucrul mecanic? Oare ne putem ajuta de C++ în rezolvarea ei? Să vedem împreună problema și codul programului pentru rezolvarea ei. Desigur, calculatorul ne ajută în rezolvarea problemelor, dar nu poate înlocui cunoștințele noastre de fizică!

### Problemă:

Este iarnă. Alex împinge o sanie cu masa de 20 kg pe o suprafață orizontală, pe o distanță de 100 m. Coeficientul de frecare dintre sanie și suprafață pe care se deplasează ( $\mu$ ) se introduce de la tastatură. Forța de tracțiune are direcția orizontală. Se presupune că viteza saniei este constantă. Să se calculeze:

- lucrul mecanic efectuat de Alex asupra saniei;
- lucrul mecanic al forței de frecare.

#### Indicații:

Forțele care acționează asupra unui corp împins pe o suprafață orizontală sunt prezentate în *îmaginea b.*

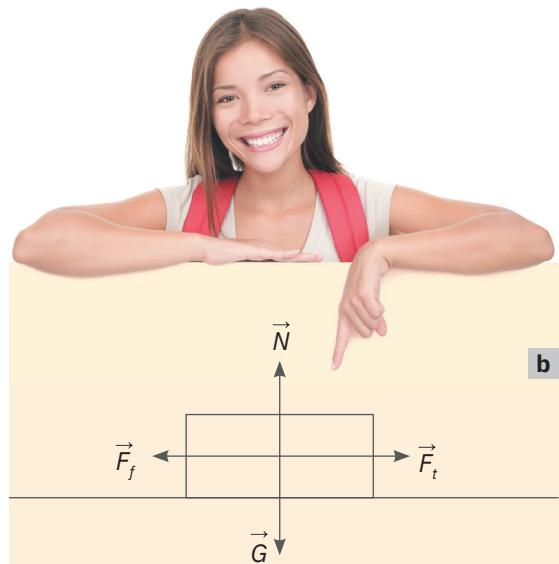
Notăm: masa saniei  $m = 20 \text{ kg}$ , distanța parcursă de Alex  $l = 100 \text{ m}$ .

Lucrul mecanic efectuat de Alex asupra saniei se calculează astfel:

$$L_t = \text{Forța de tracțiune} * l = \mu * m * g * l$$

Lucrul mecanic efectuat de forța de frecare se calculează astfel:

$$L_f = \text{Forța de frecare} * d * \cos 180^\circ = -\mu * m * g * d$$



**Rețineți**

Funcțiile trigonometrice în C++ se apelează puțin mai diferit decât probabil vă așteptați. Valoarea lui  $x$  reprezintă un unghi exprimat în *radiani*. Un *radian* este egal cu  $180/\text{PI}$  grade (unde  $\text{PI} = 3.14159265$ ). Astfel, pentru a-i cere programului să calculeze „ $\cos(x)$ ”, trebuie să scriem:  $\cos(x*\text{PI}/180)$ .

Codul programului este următorul (parcurgeți-l și analizați-l cu atenție, linie cu linie):

```
#include <iostream>
#include <cmath>           // <cmath> este libraria care contine functii matematice
                           // Avem nevoie de aceasta librarie pentru apelarea functiei cos()
using namespace std;
int main()
{
    // m = masa saniei
    // l = distanta parcursa de sanie
    // u = coeficientul de frecare
    // Ft - forta de tracțiune
    // Ff - forta de frecare
    // La - Lucrul mecanic efectuat de Alex asupra saniei
    // Lf - Lucrul mecanic al forței de frecare
    // g - acceleratia gravitationala (=9,80665 m/s2), dar o aproximam la 10

    // declarăm toate variabilele necesare:
    int m=20, l=100;
    float u, Ft, Ff, La, Lf, alfa=180;
    const float PI = 3.1415926;
    const float g = 10;

    cout<<"Introduceti coeficientul de frecare: ";
    cin>>u;
```

```

Ft=u*m*g;
La=Ft*l;
cout<<"Lucrul mecanic efectuat de Alex asupra saniei este: "<<endl;
cout<<"La = Ft*l = u*m*g*l= "<<La<<" J"<<endl;
cout<<endl;

// Cosinusul unui unghi de „alfa” grade se exprima astfel în C++: cos(alfa*PI/180.0)
Ff=u*m*g;
Lf=Ff*l*cos(alfa*PI/180.0);

cout<<"Lucrul mecanic al forței de frecare este: "<<endl ;
cout<<"Lf = Ff*l*cos(180) = "<<Lf<<" J"<<endl;

return 0;
}

```

## Aplicați și investigați

- În programul anterior, introduceți diferite valori ale coeficientului de frecare și observați rezultatul asupra valorii lucrului mecanic.
- Modificați programul anterior astfel încât să se introducă de la tastatură masa saniei și lungimea pe care o împinge Alex. Încercați și descoperiți cum anume depinde lucru mecanic al forței de frecare de fiecare parte (și de variația masei corpului împins, și de lungimea pe care acesta este împins).

## Știați că...?

Dacă sunteți pasionați de jocuri pe calculator și vă surâde ideea de a lucra în Game Development (Dezvoltare de jocuri), aflați că foarte multe jocuri au la bază limbajul C++, iar majoritatea companiilor IT cer viitorilor angajați să cunoască bine acest limbaj de programare. Dezvoltarea unui joc necesită timp, răbdare și ingeniozitate, dar rezultatele sunt pe măsura efortului depus.



## Exersați

- Faceți un program în C++ care desenează un personaj din litere. Faceți-i o animație de deplasare pe orizontală la dreapta.
- Scriptați un program în C++ care să rezolve problema de mai jos.

Distanța de la punctul **a** și timpul de la punctul **b** se vor introduce de la tastatură.

*Se dă o mașină care se deplasează rectiliniu uniform cu viteza de 40 km/h între două orașe A și B aflate la distanța **d** km, **d** este cel mult 500 km și este un număr natural.*

- Să se afișeze în cât timp mașina parcurge distanța **d**.
- Să se calculeze și să se afișeze ce distanță parcurge mașina într-un timp dat de **t** minute, dacă merge constant cu aceeași viteză, **t** număr natural mai mic decât 600.

*Indicații:* Intrare: **d** = 120 km, **t** = 90 min, Ieșire: a 3 ore, b 60 km.

*Restricții:*  $0 < d \leq 500$

Implementați și o validare a valorii variabilei **d**, introdusă de la tastatură.

## Portofoliu

Scriptați un program C++ în care un pătrat construit din caracterul „\*” să se depleteze în jos pe ecran. Programul va fi salvat în folderul **Proiecte C++**, cu numele **Animatie**.

Schimbați culoarea fundalului ferestrei consolă în albastru. Schimbați culoarea pătratului la fiecare reafisare a acestuia pe ecran.

*	*	*	*	*
*				*
*				*
*				*
*	*	*	*	*

# 9 Aplicații de geografie în C++

Am văzut cum putem utiliza limbajul C++ pentru a realiza probleme de matematică, de fizică. În continuare vom crea cu ajutorul C++ instrumente digitale pe care le putem utiliza la ora de geografie.

## Descoperiți

### Chestionar cu întrebări grilă

► Vom pregăti un program tip chestionar (Quiz) care propune trei întrebări de geografie, fiecare cu câte trei variante de răspuns, dintre care doar o variantă este corectă. Se solicită utilizatorului să introducă un număr corespunzător răspunsului ales. Programul verifică răspunsurile și afișează scorul final.

**Date de intrare:** răspuns = număr întreg, cu valori de la 1 la 3, stochează răspunsul utilizatorului.

**Date de ieșire:** scor = variabila care stochează scorul utilizatorului. La fiecare întrebare, valoarea acestei variabile crește cu 1 dacă răspunsul este corect; inițial se pleacă de la scor 0.

Programul C++ este următorul:

```
#include <iostream>
using namespace std;

int main()
{ int raspuns, scor = 0;

    // Intrebare 1
    cout << "Care este capitala Japoniei?" << endl;
    cout << "1. Beijing" << endl;
    cout << "2. Tokyo" << endl;
    cout << "3. Cairo" << endl;
    cout << "Raspunsul tau (introdu numarul corespunzator): ";

    cin >> raspuns;
    if (raspuns == 2)
    { cout << "Corect!" << endl;
        scor++;
    }
    else
        cout << "Raspuns gresit. Raspunsul corect este: 2. Tokyo" << endl<<endl;

    // Intrebare 2
    cout << "Care este cel mai mare continent al lumii?" << endl;
    cout << "1. Europa" << endl;
    cout << "2. Asia" << endl;
    cout << "3. Africa" << endl;
    cout << "Raspunsul tau (introdu numarul corespunzator): ";

    cin >> raspuns;
    if (raspuns == 2)
    { cout << "Corect!" << endl;
        scor++;
    }
    else
        cout << "Raspuns gresit. Raspunsul corect este: 2. Asia" << endl<<endl;
```



```
// Intrebare 3
cout << "Care este cel mai lung fluviu american?" << endl;
cout << "1. Amazon" << endl;
cout << "2. Colorado" << endl;
cout << "3. Rio Grande" << endl;
cout << "Raspunsul tau (introdu numarul corespunzator): ";

cin >> raspuns;
if (raspuns == 1)
{   cout << "Corect!" << endl;
    scor++;
}
else
    cout << "Raspuns gresit. Raspunsul corect este: 1. Amazon" << endl<<endl;

cout << "Scorul tau final este: " << scor << "/3" << endl;      // Afiseaza scorul final
return 0;
}
```

## Aplicații și investigații

1. La prima întrebare din chestionar, înlocuiți introducerea răspunsului și verificarea sa cu liniile de cod de mai jos și investigați care este în acest moment comportamentul programului în cazul primei întrebări. Cum se face acum validarea? Când se afișează răspunsul corect?

```
cin >> raspuns;
int nrIncercari=1;
while (raspuns != 2 && nrIncercari<2)
{
    cout << "ncorect. Mai introdu o data raspunsul: ";
    cin >> raspuns;
    nrIncercari++;
}
```

2. Modificați programul astfel încât validarea răspunsului și afișarea răspunsului corect să se facă și pentru întrebările 2 și 3, la fel ca la întrebarea 1.  
 3. Adăugați la acest chestionar alte două întrebări de geografie, din ceea ce ati învățat până la acest moment.

## Exersații

1. Care credeți că este avantajul pe care îl prezintă utilizarea unui chestionar digital realizat în C++ față de aplicarea unui chestionar tipărit pe hârtie?  
 2. Scrieți o secvență de cod în C++ care afișează câte un mesaj în funcție de valoarea unei variabile **scor**, număr natural, mai mic decât 10 astfel:

Valoare variabilă scor	Mesaj afișat
< 5	Insuficient
5 sau 6	Suficient
7 sau 8	Bine
9 sau 10	Foarte bine



## Portofoliu



Scrieți un program C++ care să conțină un chestionar cu zece întrebări pentru orice materie doriți voi. Întrebările trebuie să aibă câte trei răspunsuri, dintre care unul singur să fie corect. Utilizatorul introduce de la tastatură numărul răspunsului, iar acesta se validează imediat. La final se afișează scorul. Programul va avea numele **Chestionar** și va fi salvat în folderul **Proiecte C++**.

# Concurs: Micii programatori



În cadrul acestui concurs, veți realiza un proiect C++, cu ajutorul căruia veți recapitula noțiunile de programare dobândite. La final, toate proiectele vor fi evaluate și vor primi note.

## Organizzare

Formați echipe de câte trei-patru elevi și dați nume fiecărei echipe.

## Tema concursului

Fiecare echipă va realiza un proiect C++ cu tema: **Cine știe câștigă**. Proiectul va conține o aplicație de tip chestionar (Quiz), pentru una dintre materiile de la școală, cu minimum 20 de întrebări la care utilizatorii trebuie să răspundă. Răspunsul la fiecare întrebare este verificat imediat, iar la final se afișează scorul. La fiecare validare se poate afișa un mic desen realizat din caractere sau o mică animație, după cum dorește fiecare echipă. Aveti două exemple în *imaginea a.*

a

## **Lista de verificare a proiectului**

- 1.** Am respectat tema proiectului:
    - a.** În întregime
    - b.** Parțial
  - 2.** În timpul de lucru alocat am terminat proiectul:
    - a.** În întregime
    - b.** Parțial
  - 3.** Nivelul de documentare în vederea elaborării proiectului este:
    - a.** Foarte bun
    - b.** Bun
    - c.** Slab
  - 4.** Comunicarea cu colegii a fost:
    - a.** Foarte bună
    - b.** Buna
    - c.** Slabă
  - 5.** Nivelul de creativitate la realizarea proiectului a fost:
    - a.** Foarte bun
    - b.** Bun
    - c.** Slab
  - 6.** Prezentarea proiectului a avut impact la colegi:
    - a.** În mare măsură
    - b.** În măsură normală
    - c.** În mică măsură

## Timpul de lucru

## Alocat de cadrul didactic

## Cum veți lucra

- Stabiliti pentru ce materie creați conținutul jocului vostru.
  - Împărtiți sarcinile fiecărui membru al echipei.
  - Documentați-vă și căutați informații care să vă ajute să construiți întrebările. Respectați drepturile de autor privitoare la conținuturile pe care le luați drept repere documentare pentru proiect. Respectați regulile de securitate și prevenție la navigarea pe Internet, pentru a nu vă infecta calculatorul cu programe de tip malware.

- Realizați programul C++ utilizând conținutul pregătit anterior.

## Prezentarea proiectului și evaluarea

  - Realizați o scurtă prezentare PowerPoint a proiectului vostru, pornind de la idee și detaliind punerea sa în execuție.
  - Expuneți prezentarea în fața colegilor, cu rularea efectivă a proiectului, ca exemplificare. Răspundeți la întrebările acestora și, foarte important, încadrați-vă în timpul alocat echipei.
  - Fiecare echipă va acorda un număr de puncte de la 0 la 10 proiectelor celorlalte echipe; la final se face suma tuturor punctajelor. Va câștiga echipa care a câștigat cele mai multe puncte.
  - Faceți-vă fiecare o autoevaluare a acestei activități, folosind *Lista de verificare* a proiectului din caseta alăturată.

# 10 Programarea roboților virtuali. Utilizarea senzorilor

## Descoperiți

Este fascinant să te gândești la roboți. Dar cum anume îi programăm să facă ceea ce ne dorim noi?

Vom explora, în cele ce urmează, lumea minunată a programării roboților, chiar fără să deținem un robot fizic, adevărat. Vom crea simulări cu roboți virtuali utilizând un alt mediu de programare online, gratuit, **Open Roberta Lab** (<https://lab.open-roberta.org/>). Open Roberta Lab este conceput special să fie utilizat de către elevi și studenți și are integrat un editor grafic, intuitiv, bazat pe blocuri, în stilul Scratch. Limbajul de programare grafic utilizat în platforma Open Roberta Lab pentru a programa roboții se numește NEPO.

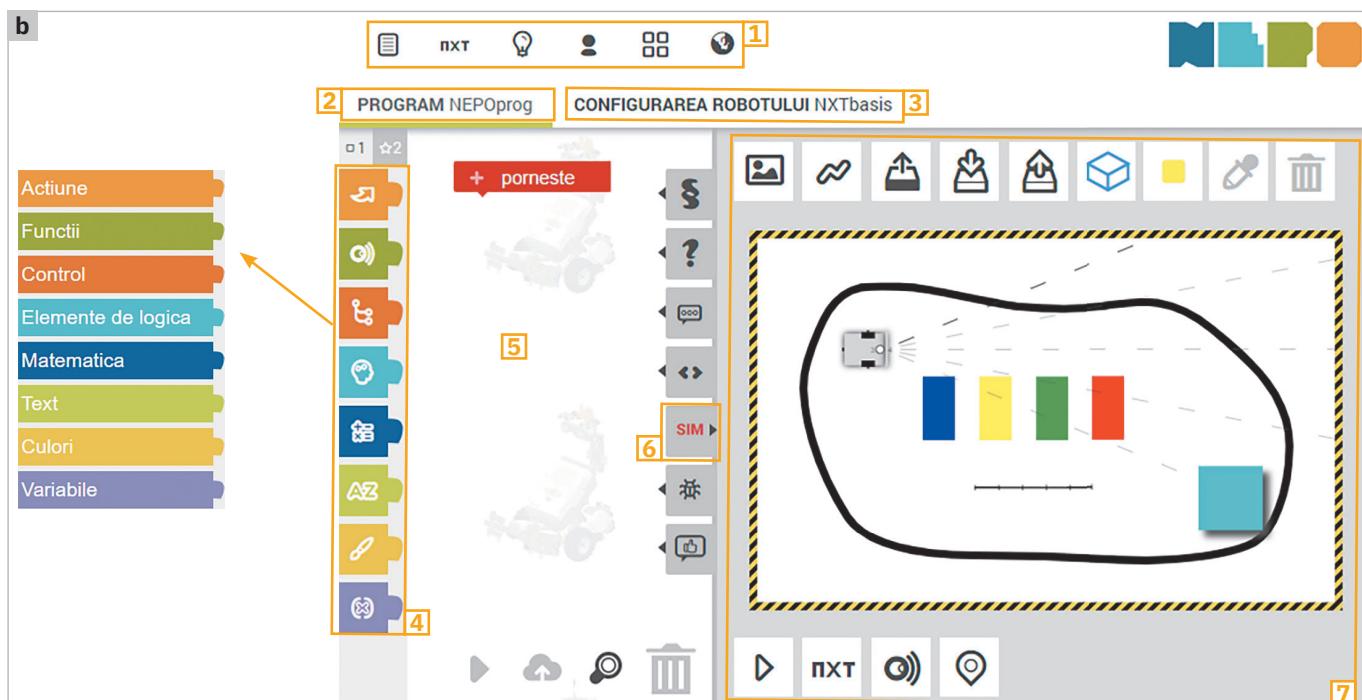


## Interfața Open Roberta Lab

La accesarea aplicației OpenRobertta Lab se deschide o fereastră ce are în partea de sus o bară derulantă din care trebuie să vă alegeti sistemul (robotul) pe care doriți să îl programați (*imaginea a*).



În cele ce urmează, vom utiliza robotul NXT, care este unul educațional realizat de către compania LEGO; alegeti din banda derulantă NXT.



Elementele de interfață sunt următoarele (*imaginea b*):

1. Bara de comenzi
2. Fila programului Nepo – aici se editează programul
3. Fila Configurarea Robotului – aici se configuraază senzorii și motoarele robotului
4. Categoriile de instrucțiuni
5. Zona de programare
6. Butonul de afișare a Zonei de simulare
7. Zona de simulare

Limba Română a fost selectată apăsând pe butonul din **Bara de comenzi** a interfeței.

## Mișcarea robotului

În cele ce urmează vom învăța cum să programăm roboțelul să se miște pe scenă. Îl vom face să meargă pe o traiectorie în formă de hexagon (figură geometrică plană, închisă, având șase laturi și șase unghiuri egale).

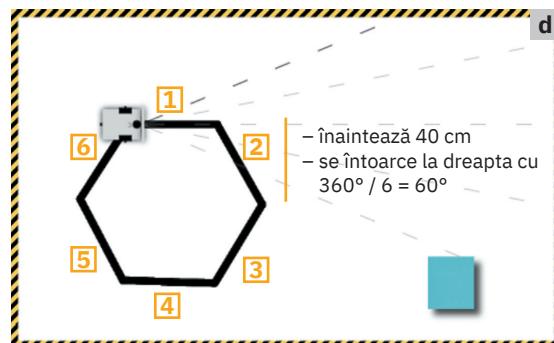
Roboțelul nostru poate fi ușor deplasat utilizând instrucțiunile de mișcare din categoria **ACTIONE** și instrucțiunea repetitivă cu număr cunoscut de pași din categoria **Control** (instrucțiunea **repetă de () ori**). În *imaginea c* găsiți codul programului nostru, explicat pas cu pas:

Program roboțel	Algoritm de funcționare
<pre>+ porneste repeta de 6 ori fă     actionare motor inainte viteza % 30     distanta cm 40     intoarce dreapta viteza % 30     grad 60 c</pre>	<p>Pornește robotul.</p> <p><b>Se repetă de 6 ori</b> cele 2 instrucțiuni:</p> <p>Roboțelul <b>se deplasează înainte</b>, cu 30% din viteza maximă, pe o distanță de 40 cm.</p> <p>Roboțelul <b>se întoarce la dreapta</b>, 60°, având o viteză de 30% din viteza maximă.</p>

Facem următoarele acțiuni în zona de simulare:

1. Alegem o scenă cât mai simplă, apăsând pe butonul .
2. Activăm opțiunea de desenarea a traiectoriei robotului la mișcarea acestuia, apăsând pe butonul .
3. Pornim simularea apăsând pe butonul .

► Rezultatul simulației este prezentat în *imaginea d*. La calcularea unghiului de întoarcere la dreapta, aplicăm formula: **360°/numărul de laturi = 360°/6 = 60°** pentru că hexagonul are șase laturi egale și trebuie să ne întoarcem de unde am plecat, în șase pași egali.



## Aplicați și investigați

1. În programul anterior măriți viteza de deplasare a roboțelului înainte și/sau la întoarceri și urmăriți cum se mișcă acesta în simulare.
2. Schimbați direcția de deplasare, apăsând pe săgeata mică de lângă „înainte” sau direcția de întoarcere apăsând pe săgeata din dreapta direcției „dreapta”. Urmăriți mișările în simulator.
3. Investigați cum ar trebui să modificați programul astfel încât roboțelul să se miște pe o traiectorie în formă de octogon (figură geometrică închisă, plană, cu opt laturi și opt unghiuri egale).

## Descoperiți

### Utilizarea senzorilor – Robotul urmărește o traiectorie trasată cu culoarea neagră

Robotul nostru NXT are două motoare și maximum patru senzori. De altfel, fiecărui robot fizic îi putem adăuga senzori. **Un senzor** este un dispozitiv electronic care detectează și măsoară schimbările în mediul înconjurător și convertește aceste informații în semnale pe care un robot le poate interpreta și utiliza într-un fel anume.

Exemple de senzori:

1. **Senzorul de culoare** (*light sensor*): detectează culorile, poate fi utilizat în aplicații precum urmărirea unei traiectorii sau pentru recunoașterea obiectelor colorate.
2. **Senzorul de distanță** (*ultrasonic sensor*): măsoară distanța până la obiectele din fața robotului, poate fi folosit pentru evitarea obstacolelor sau pentru alte aplicații care necesită detectarea distanței.
3. **Senzorul tactil** (*touch sensor*): detectează atingerea sau presiunea, poate fi utilizat pentru a detecta coliziunile sau pentru a activa anumite acțiuni când este atins.
4. **Senzor de sunet** (*sound sensor*): detectează nivelul de zgomot din mediul înconjurător, este utilizat pentru a reacționa la sunete specifice.

► Vom programa un robot pentru a se deplasa pe o traiectorie trasată de noi în prealabil cu culoarea neagră. Pentru aceasta, vom configura ca robotul nostru să aibă ca senzor 3 un **senzor de culoare** (din fila **Configurarea robotului**). Senzorul nostru va trebui să depisteze culoarea neagră.

## Programarea roboților virtuali. Utilizarea senzorilor

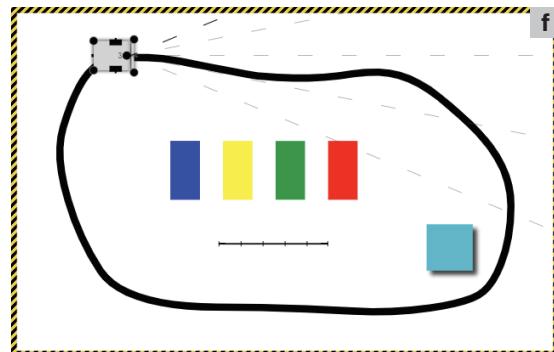
În imaginea e găsiți codul programului nostru, explicat pas cu pas.

Program roboțel	Algoritm de funcționare
<pre> + porneste repetă la infinit fă + dacă [ obtine culoare v = Senzor de culoare Port 3 = ]     fă conduce înainte viteza % stanga 10     viteza % dreapta 30     altfel conduce înainte viteza % stanga 30         viteza % dreapta 10     e   </pre>	<p><b>Pornește</b> robotul.</p> <p><b>Repetăm la infinit:</b></p> <ul style="list-style-type: none"> <li><b>Dacă</b> senzorul depisteașă culoarea neagră</li> <li><b>Atunci</b> robotul face o curbă la stânga.</li> <li><b>Altfel</b>, face o curbă la dreapta, similară ca amplitudine cu cea spre stânga.</li> </ul>

Instrucțiunea: **conduce (înainte) viteza % stanga ()**  
**viteza % dreapta ()**

configurează viteza celor două motoare ale robotului la încărcare. Robotul face curbă la dreapta dacă viteza motorului din stânga este mai mare, și spre stânga în caz contrar.

▶ Pentru simulare am ales o scenă cu un traseu închis (imaginea f). Se observă în cerculețul de pe robot care este culoarea sesizată de senzor. Robotul se va mișca pe acest traseu. Când sesizează culoarea neagră se mișcă spre stânga, apoi când sesizează altă culoare (alb), o ia la stânga, găsind iarăși culoarea neagră.

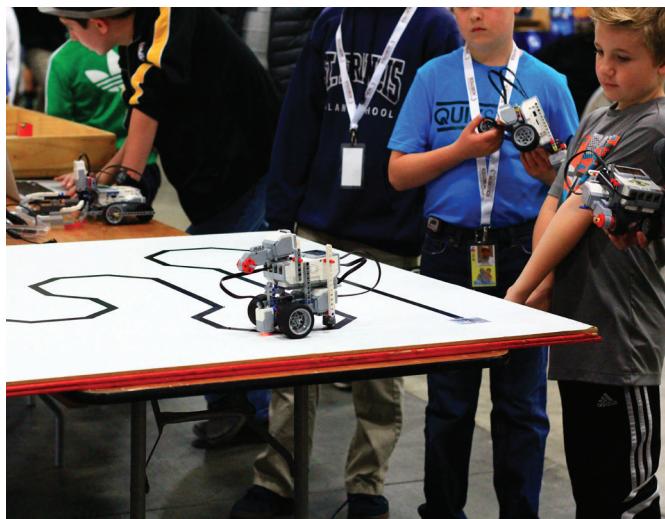


## Aplicații și investigații

- Modificați programul de mai sus astfel încât robotul să urmărească un traseu de culoare roșie.
- Investigați ce se întâmplă dacă modificați vitezele celor două motoare în programul de mai sus, astfel:
  - la depistarea culorii negre, modificați valorile vitezelor astfel încât viteza la stânga să fie mai mare decât viteza la dreapta. În caz contrar (la depistarea altelor culori), modificați valoarea vitezei la dreapta ca să fie mai mare decât cea din stânga, iar valorile de aici să fie aceleași cu cele din primul caz, dar interschimbate.
  - puneți valori total diferite pentru cele două viteze în ambele cazuri (și la depistarea culorii negre și la depistarea altelor culori). Încercați și varianta când valoarea vitezei la stânga e mai mare decât cea la dreapta, și invers, pentru ambele cazuri.

## Erexsații

- Definiți noțiunea de senzor al unui robot.
- Dați două exemple de senzori cunoscuți. La ce se pot utiliza?
- Dezbateră:** Organizați o dezbatere cu tema „Este sau nu utilă realizarea unei simulări a comportamentului unui robot?“.
- Asociați corect, în caiete:



- SIM**
  - 
  -
- Schimbarea scenei unei simulări
  - Pornirea unei simulări
  - Afișarea configurației robotului
  - Afișarea zonei de simulare

# Exerciții recapitative

Rezolvați pe caiete sarcinile de lucru de mai jos.

1. Variabilele  $x$  și  $y$  sunt de tip întreg. Lui  $x$  îi se atribuie valoarea 7, iar lui  $y$ , valoarea 10. Care dintre expresiile C++ următoare va avea valoarea 3?

- a.  $((x+y)+3)/2;$       b.  $(x+2*y)\%8;$       c.  $(7*x-4*y)/7;$       d.  $!(x>y).$

2. Un număr întreg pozitiv este numit numărul lui Armstrong dacă suma cuburilor cifrelor lui este egală cu numărul însuși. Faceți un program care să afișeze dacă un număr pozitiv de maximum 4 cifre introdus de la tastatură este număr Armstrong sau nu.

*Indicații:*

Intrare: 153, Ieșire: „153 este număr Armstrong.”

Intrare: 12, Ieșire: „12 nu este număr Armstrong.”

$153 = 1 * 1 * 1 + 5 * 5 * 5 + 3 * 3 * 3 // 153$  este număr Armstrong.

$12 != 1 * 1 * 1 + 2 * 2 * 2 // 12$  nu este număr Armstrong.

3. În cazul programului din caseta alăturată, ce rezultat se afișează pe ecran pentru  $x=4$ ?

- a. 10;      b. 4;      c. 1.

4. Faceți un program care citește  $n$  numere naturale de la tastatură și afișează pe ecran suma ultimelor lor cifre.

5. Spuneți în ce categorie de instrucțiuni găsiți în aplicația *Open Roberta Lab*, instrucțiunea din imaginea a:

- a. Control;      b. Acțiuni;      c. Elemente de logică.

repetă la infinit  
fă

a

```
#include <iostream>
using namespace std;
int main()
{
    int x;
    cout<<"Introduceti valoarea lui x: ";
    cin>>x;
    if(x==1 || x==0)
        cout<<1;
    if(x%2==0)
        cout<<(x/2)+2;
    else
        cout<<(x-1)+3;
    return 0;
}
```

## Autoevaluare

Timp de lucru: **15 minute**

Rezolvați în caiet exercițiile date. Evaluați-vă singuri rezultatele.

1. Variabilele  $x$  și  $y$  sunt de tip întreg,  $x$  memorând valoarea 3, iar  $y$  valoarea 4. Indicați expresia C++ care are valoarea de adevăr Adevărat:

- a.  $4*(x-3*x) == 0$   
 b.  $4*x-3*y == 0$   
 c.  $!x == y$

2. Ce se afișează pe ecran dacă este executat programul din caseta alăturată?

- a. 0123...89;  
 b. 0123...8910;  
 c. apare eroare la compilare.

3. Priviți scriptul din imaginea b realizat în aplicația *Open Roberta Lab* pentru un robot NXT. Analizați și spuneți cum se va deplasa robotul dacă pornim simularea mișcării sale:

- a. înainte;      b. la stânga;

```
#include <iostream>
using namespace std;
int main()
{
    int x=0;
    while(x<10)
    {
        cout<<x;
        x++;
    }
    return 0;
}
```

- c. la dreapta.

b + porneste  
repetă la infinit  
fă conduce înainte viteza % stanga 20  
viteza % dreapta 50

## CUM VĂ APRECIATI?

- 1 Desenați ● în dreptul sarcinilor pe care le-ați rezolvat corect.

●	●●	●●●
40 p.	70 p.	100 p.

- 2 Scrieți cum vă simțiți după ce ați rezolvat acest test, alegând unul dintre cuvintele de mai jos:

- nemulțumit(ă);  
 – mulțumit(ă);  
 – încântat(ă).



# Recapitulative finală

Toate cerințele se rezolvă în caiete sau pe calculator.

1. Indicați o expresie C++ care are valoarea *Adevărat* dacă și numai dacă numărul natural memorat în variabila întreagă *x* este multiplu de 3.

a.  $x/3 == 0$ ;      b.  $x \% 3 == 0$ ;      c.  $(x/3)\%10 == 0$ .

2. Considerăm două variabile întregi *a* și *b*. Asociați corect instrucțiunea de atribuire C++ din partea stângă cu instrucțiunile echivalente ei.

$a = ++b;$	$a=b; b=b+1;$
$a = --b;$	$b=b+1; a=b;$
$a = b++;$	$a=b; b=b-1;$
	$b=b-1; a=b;$



Scrieți câte un program C++ care să rezolve problemele 3-8.

3. Într-o curte sunt *g* găini, de două ori mai mulți curcani decât găini și cu 5 mai multe rațe decât curcani. Calculați și afișați câte păsări sunt în total în curte, dacă în curte sunt maximum 100 de găini.

*Indicație:* Intrare: *g*=10, Ieșire: 45 ( $10 + 2*10 + (2*10+5) = 45$  păsări)

4. Se citesc de la tastatură două numere naturale *a* și *b* mai mici decât 1000000. Afișați „DA” dacă ultima cifră a sumei acestora este egală cu 6 și „NU”, altfel.

*Indicație:* Intrare: *a*=2 *b*=4, Ieșire: DA ( $2 + 4 = 6$ ); Intrare: *a*=2 *b*=3, Ieșire: NU ( $2 + 3 = 5$ )

5. Se citesc de la tastatură două numere naturale *a* și *b* mai mici decât 1000000. Afișați „DA” dacă *a* și *b* au aceeași paritate (sunt ambele pare sau ambele impare) și „NU” altfel.

*Indicație:* Dacă două numere au aceeași paritate, atunci suma lor este întotdeauna un număr par. Calculați suma *a* + *b* și vedeți dacă aceasta este pară sau nu.

Intrare: *a*=2 *b*=4, Ieșire: DA ( $2 + 4 = 6$  par); Intrare: *a*=2 *b*=3, Ieșire: NU ( $2 + 3 = 5$  impar)

6. Se citesc de la tastatură numere naturale mai mici decât 2000, până la introducerea lui 0. Afișați câte dintre acestea sunt impare.

*Indicație:* Intrare: 2 3 4 5 0, Ieșire: 2 (3, 5 impare)

7. Se citesc numere naturale de la tastatură până la apariția lui zero. Numerele introduse sunt mai mari decât 0 și mai mici decât 500. Să se afișeze câte dintre ele sunt divizibile cu 7.

*Indicație:* Intrare: 2 3 7 14 22 21 0, Ieșire: 2 (7 și 14 sunt divizibile cu 7)

8. Se citește de la tastatură un număr *n* natural, cu maximum 9 cifre. Să se afișeze câte dintre cifrele sale sunt egale cu cifra sa minimă.

*Indicație:* Intrare: 232422, Ieșire: 4 (cifra sa minimă este 2, sunt 4 cifre de 2)

9. Se dă problema următoare:

*Se introduc de la tastatură *n* numere naturale mai mici decât 100 ( $n \leq 100$ ). Să se calculeze și să se afișeze suma lor.*

Care dintre variantele următoare sunt corecte pentru a completa instrucțiunea **for** din secvența de cod din caseta alăturată, astfel încât execuția acesteia să afișeze rezultatul cerut?

- a.  $i = 1; i \leq n; i = i + 1;$   
 b.  $i = 1; i < n; i = i + 1;$   
 c.  $i = n; i \geq 1; i = i - 1.$

```
int n, s=0, nr, i;
cin>>n;
for ( ... )
{
    cin>>nr;
    s=s+nr;
}
cout<<s;
```

10. Pentru aplicația *Open Roberta Lab* alegeti **A** (Adevărat) sau **F** (Fals), în funcție de valoarea de adevar a enunțurilor:

- a. Robotul NXT are două motoare. **A** **F**  
 b. În timpul unei simulări nu putem vedea ce culoarea a depistat senzorul de culoare al robotului. **A** **F**  
 c. În timpul unei simulări, robotul poate trasa traectoria pe care merge. **A** **F**

# Evaluare finală

Toate cerințele se rezolvă în caiete.

1. Adevărat (A) sau fals (F)? Verificați fiecare dintre afirmațiile de mai jos:

- a. O linie de cod C++ se termină cu „;”. A F
- b. Operatorul utilizat pentru compararea valorilor a două variabile este „=”.A F
- c. Folosim **for( ; ; )** pentru a realiza un ciclu infinit. A F

2. Care dintre senzorii care pot fi atașați unui robot este utilizat pentru a detecta coliziunile?

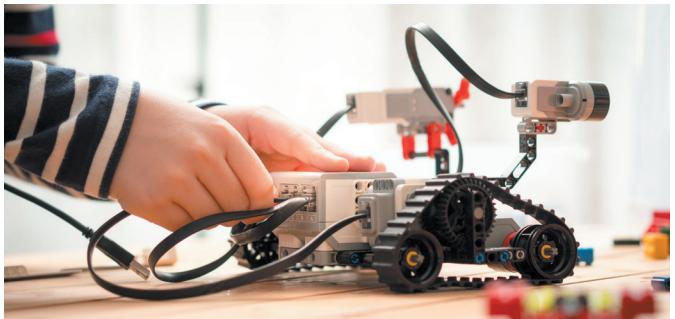
- a. de distanță;
- b. tactil;
- c. de culoare.

3. Dați un exemplu de o structură repetitivă în care acțiunile se repetă minimum o dată.

4. Asociați secvențele de cod cu rezultatele corespunzătoare:

a. 

```
int i = 3;
switch (i)
{ case 0+1: cout<<"Soare";
    break;
case 1+2: cout<<"Umbra";
    break;
default: cout<<"Optiune inexistentă";
}
```



b. 

```
int i, a=3;
for (i=1;i<7;i++)
    a=a+2;
cout<<a;
```

c. 

```
int i=1, a=5;
do
{
    a=a+2;
    i++;
} while (i<5);
cout<<a;
```

1. 13

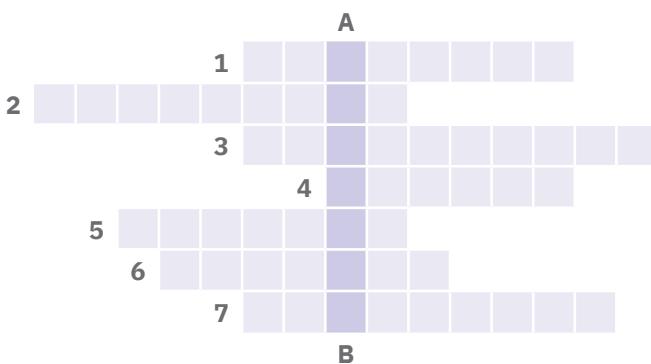
2. Umbra

3. 15

4. 21

5. Enumerați trei operatori aritmetici și doi operatori logici în C++.

6. Completați rebusul următor și descoperiți cuvântul ascuns pe coloana AB.



1. Acțiunea de reparare a erorilor găsite într-un program.
2. Pornirea sau rularea programului.
3. Etapă inițială din realizare a unui program, la care se face fie algoritmul în pseudocod, fie schema logică a programului.
4. De nedorit, atât în program, cât și în bucătărie (tradus în limba română).
5. Parcuregerea programului cu scopul de a identifica posibilele probleme.
6. În această etapă se scrie efectiv programul sursă.
7. Acțiune prin care se convertește codul sursă al programului într-un cod intermedian, numit cod obiect.

7. Scrieți un program în C++ care calculează și afișează aria și lungimea unui cerc cu raza  $r$ , număr natural, mai mic decât 1000. Pentru calcule, considerați  $\pi = 3.14$ .

*Indicații:* Intrare:  $r=2$ , Ieșire: Aria cercului este: 12.56; Lungimea cercului este: 12.56

## Punctaje

1 – 10 puncte	6 – 20 puncte
2 – 10 puncte	7 – 20 puncte
3 – 10 puncte	
4 – 10 puncte	
5 – 20 puncte	

## Punctaj maxim: 100 de puncte

- Insuficient: 0-49 de puncte
- Suficient: 50-69 de puncte
- Bine: 70-89 de puncte
- Foarte bine: 90-100 de puncte

**Timp de lucru: stabilit de cadrul didactic**

# Fișă de autoevaluare a activității și a comportamentului

Realizează o fișă în genul celei de mai jos, folosind culorile indicate la casete.

Numele și prenumele: .....

Clasa: .....

Cum te apreciezi? Scrie în caseta albastră numărul din dreptul variantei tale de răspuns.

- 1 Într-o foarte mare măsură
- 2 Într-o mare măsură
- 3 Într-o oarecare măsură
- 4 În mică măsură
- 5 Deloc

Doamna/domnul profesor va scrie în caseta verde varianta sa de răspuns.



Am participat cu plăcere la rezolvarea sarcinilor de învățare.


Am participat cu plăcere la discuții, dezbateri.


Răspunsurile mele au fost în legătură cu tema discuției.


Am fost atent/atentă la desfășurarea activităților din timpul orelor de curs.


Mi-a plăcut să particip la activități organizate în echipă.


Am cerut ajutor atunci când am avut nevoie.


Am dus la bun sfârșit sarcinile primite la școală, în cadrul orelor de *Informatică și TIC*.


Am împărtășit rezultatele investigațiilor sau proiectelor celorlalți colegi.



## CUM VĂ APRECIAȚI?

Scriți cum vă simțiți după ce ați completat această fișă, alegând unul dintre cuvintele de mai jos:

- nemulțumit(ă);
- mulțumit(ă);
- încântat(ă).



# Răspunsuri (selectiv)

## Autoevaluare (p. 26)

**1.** C – 1, A – 2, B – 3. **2.** Două avantaje ale stocării fișierelor document în *cloud* sunt: 1. puteți avea acces la fișierele voastre de oriunde din lume; 2. puteți utiliza și partaja ușor fișierele voastre cu oricine dorîți (vedeți pag. 10). **3.** a – F, b – A, c – A.

## Autoevaluare (p. 46)

**1.** a. Add Text; b. New Project; c. video. **2.** Oricare trei exemple dintre: Echo, Reverb, Telephone, Speed Change, Voice Change (vedeți pag. 34). **3.** 1 – VIDEOPAD; 2 – PRELUCRARI; 3 – NORMALIZE; 4 – AMPLIFY.

## Autoevaluare (p. 60)

**1.** d. **2.** Domenii de activitate dintre care se poate alege pentru construirea răspunsului: cercetare, managementul afacerilor (în luarea unor decizii), gestiunea bunurilor și a stocurilor, cartografie, managementul resurselor naturale, realizarea planurilor de evacuare în cazul unor dezastre, turism, agricultură, planificarea și conservarea resurselor naturale (de exemplu, în silvicultură), analiza poluărilor etc. (vedeți pag. 57).

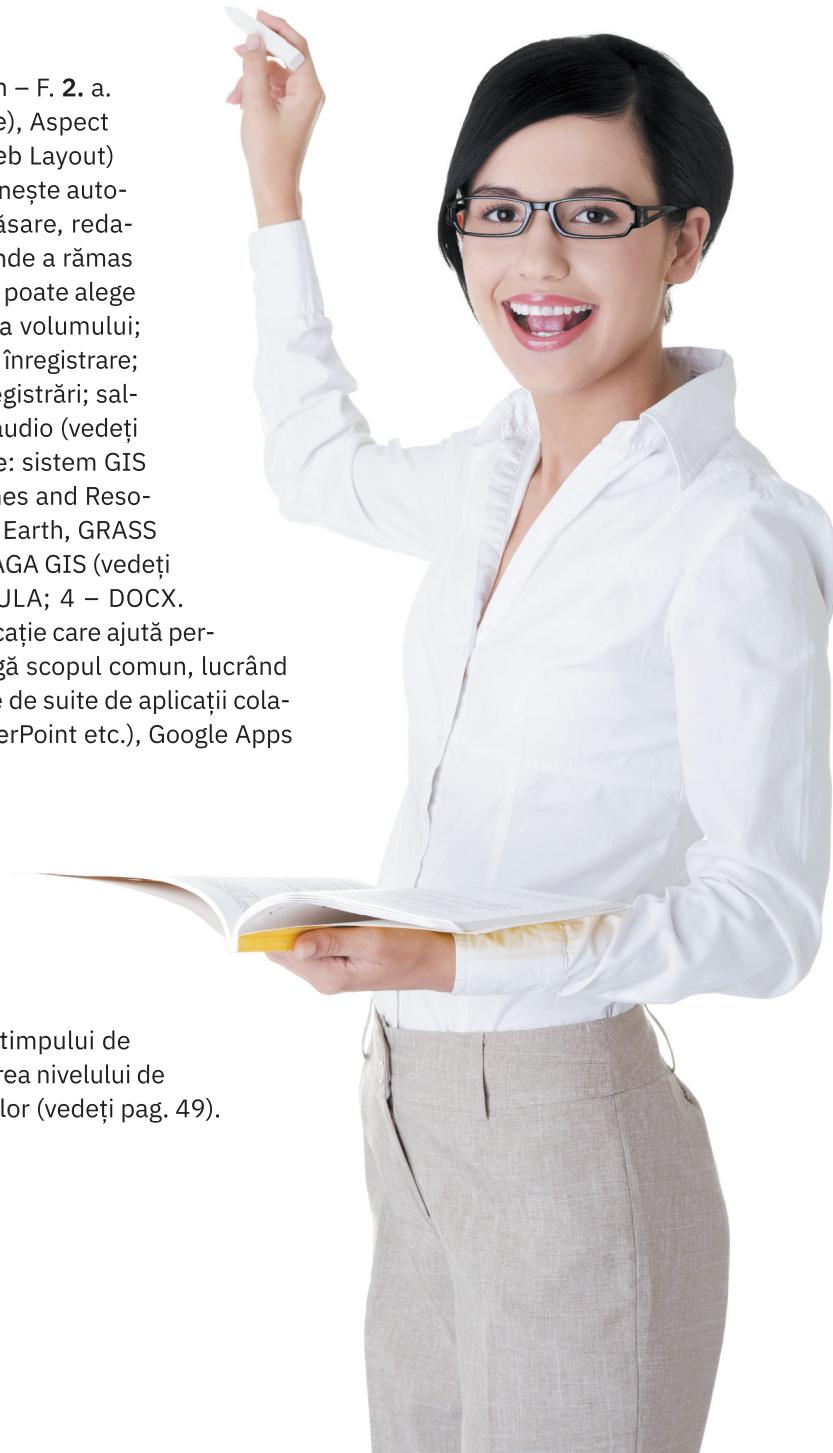
**2.** Folosiți-vă cunoștințele din anul anterior legate de crearea unei prezentări digitale; în plus vedeți informațiile de la pagina 54, legate de aplicația Google Slides.

## Evaluare sumativă TIC (p. 62)

**1.** a – A, b – A, c – A, d – A, e – A, f – A, g – F, h – F. **2.** a. **3.** Mod Focalizare (Focus), Mod citire (Read Mode), Aspect pagină imprimată (Print Layout), Aspect Web (Web Layout) (vedeți pag. 17). **4.** La prima apăsare, redarea pornește automat de la începutul fișierului. La următoarea apăsare, redarea se oprește. La o nouă apăsare, pornește de unde a rămas (vedeți pag. 31). **5.** Operații de bază dintre care se poate alege în construirea răspunsului: înregistrarea; reglarea volumului; egalizarea tonurilor; decuparea unor fragmente din înregistrare; lipirea mai multor fragmente; mixarea a două înregistrări; salvarea fișierelor și exportul lor în diferite formate audio (vedeți pag. 28). **6.** Da, există. Oricare trei exemple dintre: sistem GIS creat în 1967 de către Department of Energy, Mines and Resources, ESRI ARC/INFO 1.0, Google Maps, Google Earth, GRASS GIS, ILWIS, JUMP GIS, MapWindow GIS, QGIS, SAGA GIS (vedeți pag. 58). **7.** 1 – BULLETS; 2 – WORD; 3 – CELULA; 4 – DOCX. **8.** O aplicație software colaborativă este acea aplicație care ajută persoanele implicate în activități comune să își atingă scopul comun, lucrând împreună, în colaborare (vedeți pag. 48). Exemple de suite de aplicații collaborative Microsoft Office 365° (Word, Excel, PowerPoint etc.), Google Apps (Docs, Sheets, Slides etc.). **9.** Vedeți *Aplicați și investigați* pag. 59. **10.** Câteva dintre avantajele utilizării aplicațiilor collaborative sunt: creșterea productivității și atingerea mai rapidă a obiectivelor, comunicarea mai facilă, coordonarea persoanelor aflate oriunde în lume, facilitarea managementului de proiect, facilitarea lucrului în echipă și consolidarea relațiilor între membri, scăderea timpului de realizare a activităților, scăderea costurilor, creșterea nivelului de transparență al procesului de realizare a activităților (vedeți pag. 49).

## Autoevaluare (p. 106)

**1.** b. **2.** a. **3.** b.



### Evaluare finală (p. 108)

**1.** a – A, b – F, c – A. **2.** b. **3.** Structura repetitivă condiționată posterior. (vedeți pag. 86) **4.** a – 2, b – 3, c – 1. **5.** Operatori aritmetici din care se poate alege: +, -, \*, /, %; operatorii logici din care se poate alege: &&, ||, ! **6.** 1 – DEPANARE; 2 – EXECUTARE; 3 – PROIECTARE; 4 – GANDAC; 5 – TESTARE; 6 – EDITARE; 7 – COMPILEARE. **7.** Programul C++ este:

```
#include <iostream>

using namespace std;

int main()
{
    float a, c;
    const float PI=3.14;
    int r;

    cout<<"r=";
    cin>>r;

    a=PI*r*r; c=2*PI*r;

    cout<<"Aria cercului este: „<<a<<endl;
    cout<<"Lungimea cercului este: „<<c<<endl;

    return 0;
}
```



# Dicționar de termeni

**clipboard** – caracteristică a sistemelor de operare, care permite utilizatorilor să copieze și să stocheze temporar date, cum ar fi text sau imagini, într-un spațiu de memorie dedicat; aceste date pot fi apoi transferate sau lipite în altă parte a sistemului sau în altă aplicație (de exemplu, atunci când copiați text dintr-un document și îl lipiți într-un alt document sau într-un e-mail, datele sunt temporar stocate în clipboard).

**criptare** – procesul de transformare a datelor într-o formă ilizibilă sau într-un format codificat, numit text cifrat, folosind un algoritm matematic; scopul principal al criptării este acela de a asigura confidențialitatea și securitatea datelor în timpul transmiterii sau stocării acestora.

**drept de autor** – drept de proprietate intelectuală care conferă autorului unei opere (carte, poezie, articol, pictură, sculptură, desen, muzică, film, software etc.) dreptul exclusiv de a controla utilizarea și distribuția (inclusiv comercializarea) acelei opere.

**editare** – procesul de revizuire, modificare și îmbunătățire a unui text, a unei imagini, a unui videoclip sau a altui conținut, în scopul îmbunătățirii acestuia.

**emoticon** – reprezentare grafică stilizată a unei expresii faciale, utilizată pentru a transmite emoții sau stări în mesaje text sau în alte forme de comunicare electronică.

**ENTER** – este o tastă sau un buton ce se folosește pentru a confirma sau a trimite o comandă, un text sau o instrucțiune. Atunci când apeși tasta ENTER, de obicei, este echivalent cu apăsarea unui buton OK; de asemenea, ENTER este folosit pentru a indica trecerea la o nouă linie într-un document text.

**fereastră** – interfață grafică utilizată pentru a afișa conținutul unei aplicații software pe ecranul unui calculator sau unui telefon mobil; o fereastră este o zonă dreptunghiulară pe ecran care poate conține diverse elemente, precum texte, imagini, butoane, meniuri și alte elemente de interfață.

**font** – set de caractere tipografice care au o anumită formă și un anumit stil de scriere; fiecare font este definit de un design specific, care determină aspectul și stilul caracterelor din acel font.

**formatare** – procesul de a da un aspect specific sau de a organiza conținutul într-un mod particular într-un document, fișier sau alt tip de mediu digital.

**indentare** – aranjarea liniilor de text astfel încât lizibilitatea să fie crescută (să se poată urmări ușor).

**interactiv** – caracteristica unui sistem, a unei tehnologii sau a unei aplicații de a permite utilizatorilor să interacționeze activ cu conținutul sau cu funcționalitățile sale; o experiență interactivă implică participarea activă a utilizatorului, permitându-i să controleze sau să influențeze modul în care se desfășoară acțiunile sau cum este prezentat conținutul.

**lizibilitate** – ușurința cu care poate fi citit și înțeles un text sau alt material scris; este o măsură a gradului în care



textul este clar și accesibil cititorilor; factorii care afectează lizibilitatea includ dimensiunea și stilul fontului, spațierea între litere și rânduri etc.

**mixare** – amestecarea (suprapunerea) a două sau mai multe secvențe audio sau video, cu scopul de a obține un efect final deosebit.

**PDF** – tip de fișier utilizat pentru a prezenta și a schimba documente într-un mod independent de software și de hardware; fișierele PDF (prescurtare pentru „Portable Document Format”) păstrează aspectul original al documentului, inclusiv fonturile, imaginile, graficele și formatarea, astfel încât să arate la fel, indiferent de dispozitivele pe care sunt vizualizate.

**potențiometru** – termenul provine din electronică, unde înseamnă „rezistor căruia i se poate modifica rezistență”; în practică este cunoscut ca un buton de la care, prin învărtire, se poate modifica intensitatea luminii sau volumul sonor la un aparat audio, de exemplu; este utilizat și în programe, sub diferite forme (rotativ sau cursor liniar) pentru a varia anumiți parametri.

**prelucrare** – procesul de a transforma sau de a modifica un material, un obiect sau o substanță printr-o serie de operații sau tehnici specifice.

**tehnoredactare** – procesul de a aranja și structura documente sau texte, pentru a obține un aspect profesional și coerent; aceasta include aplicarea unor stiluri uniforme, formatarea fontului, alinierea textului, inserarea și aranjarea imaginilor sau a altor elemente grafice etc.

**wave** – tip de fișier de sunet, denumit astfel după forma de „val” pe care o are unda sonoră reprezentată grafic.



