

# METODE ALTERNATIVE DE ÎNVĂȚARE A PROGRAMĂRII

**Prof. Informatică, Jakab Irma-Tünde**  
**Liceul Teoretic „Bolyai Farkas”**  
**Tîrgu-Mureș, județul Mureș**

Cerința secolului al XXI-lea, de a se putea adapta la schimbările permanente, este valabilă nu numai pentru adulți, dar și pentru copii. Majoritatea tinerilor de azi, într-o formă sau alta, zilnic sunt în contact cu calculatorul, și aceste „mașini” sunt folosite din ce în ce mai des ca auxiliare în predare.

Adolescenții zilelor noastre sunt „nativi digitali”; ei se orientează cu ușurință în lumea digitală, trimit mesaje scrise instantaneu, se distrează cu jocuri on-line și navighează pe web fără niciun efort. Dar asta oare înseamnă că ei utilizează eficient tehnologia digitală? Puțin sunt printre ei, care sunt capabili să-și creeze jocuri, animații, simulații proprii.

Este importantă învățarea conceptelor matematice și a celor de calcul, dar în același timp, este util ca elevii să gândească creativ, sistematic și să poată colabora cu ceilalți. Există o mulțime de oportunități de învățare, care folosesc calculatorul nu doar pentru prezentarea unor materiale; de exemplu prin programare, se oferă posibilitatea unei învățări activizante, care ajută dezvoltarea competențelor digitale și matematice deopotrivă.

Pentru ca un elev, să poată proiecta și să creeze utilizând calculatorul, este necesară cunoașterea unui limbaj de programare.

Programarea dezvoltă gândirea algoritmică, disponibilitatea de rezolvare a problemelor, învățarea unor strategii de planificare, cum ar fi modularizarea și proiectarea iterativă. Există numeroase medii de programare dezvoltate special în scopuri educative, instrumentate corespunzător. Un exemplu bun este mediul de programare Scratch, proiectat de Lifelong Kindergarten: MIT Media Lab (Cambridge, Ma, SUA) în spiritul unor proiecte mai vechi, precum Logo și LEGO/Logo, conceput pentru copii cu vârsta între 8-16 ani. [1]

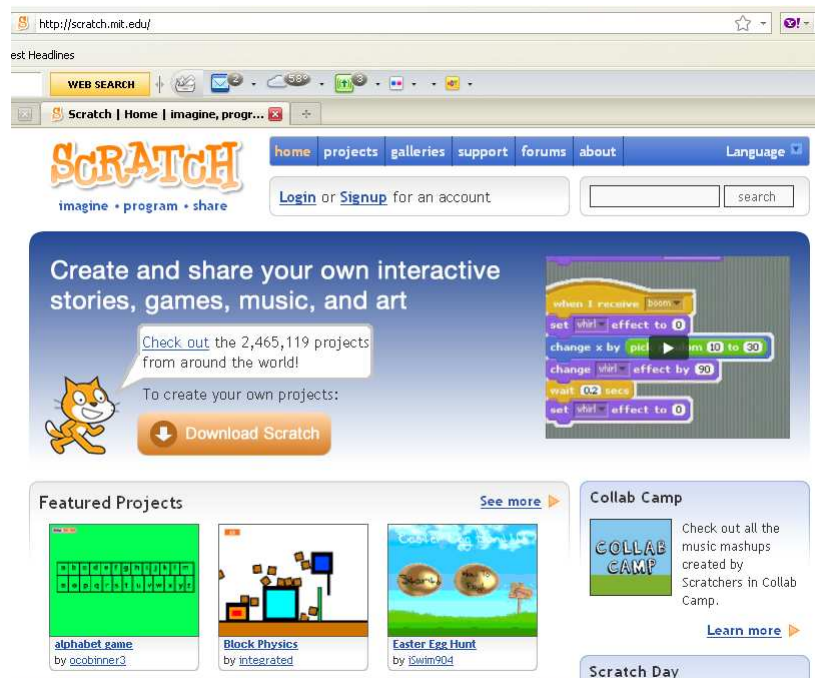
## **De ce Scratch?**

Am ales Scratch-ul, pentru că dezvoltatorii au încercat să creeze un mediu de programare care: [2]

- este vizual și interactiv, adică totul este la îndemâna programatorului, într-un loc vizibil și poate urmării pe loc rezultatul executării instrucțiunilor;
- este prietenos, fiind potrivit atât pentru rezolvarea unor probleme tradiționale, bazate pe matematică, cât și pentru rezolvarea unor probleme considerate interesante de către copii.

Aceștia știu întotdeauna exact ce-și doresc. În acest mediu, se pot integra în paralel imagini statice sau dinamice, elemente de sunet și muzică, elevii se pot aventura, se pot angrena în descoperiri independente, în timp ce ei captează cunoștințe și experiență de programare, într-un mod „invizibil”;

- are efecte educative. Paralel cu dezvoltarea limbajului de programare Scratch, s-a dezvoltat și site-ul Scratch (<http://scratch.mit.edu>), unde oricine poate încărca propriile „creații”, care ulterior sunt rulate de către alții. Astfel, a apărut o comunitate, unde se poate cere ajutor și suport, se pot exprima impresii despre munca altora, dar în același timp trebuie suportată părerea celorlalți, chiar dacă pot conține și critici negative.



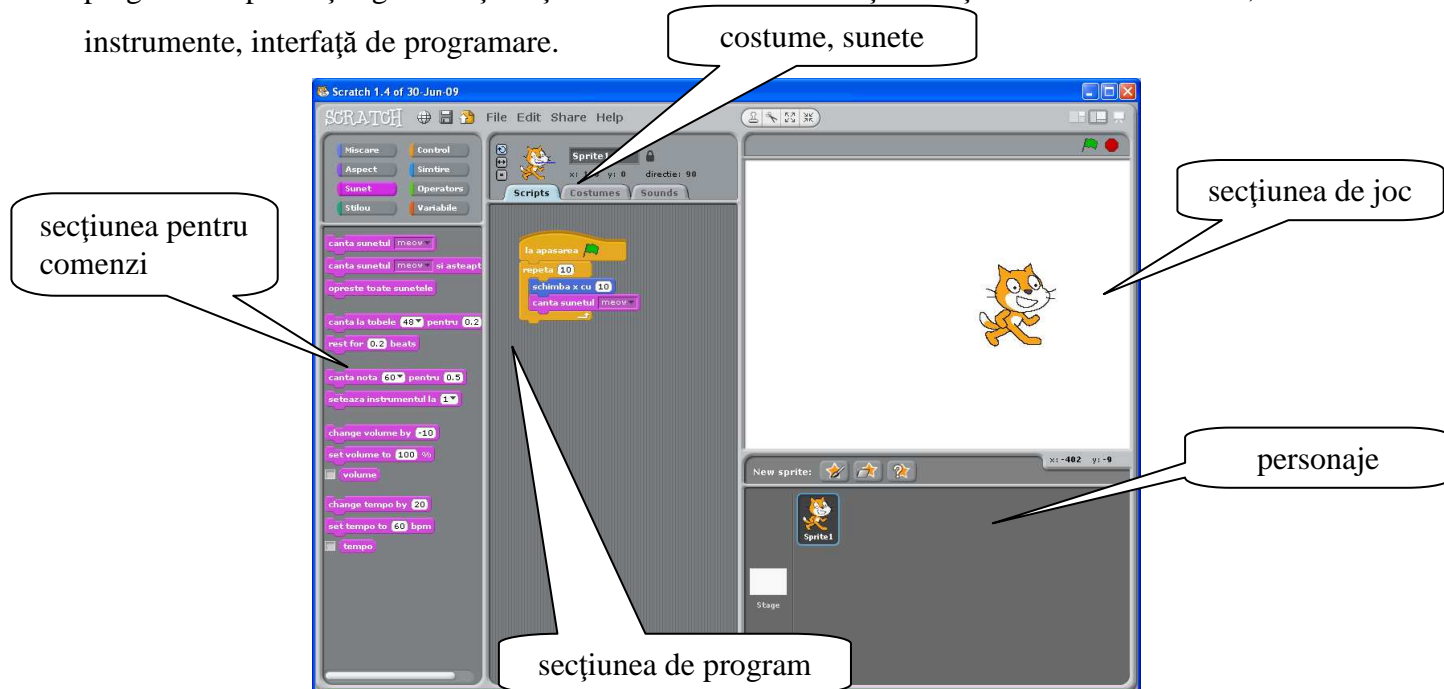
### Ce este Scratch-ul?

Scratch-ul este un mediu de programare vizual, care se poate descărca gratuit, dezvoltat special pentru copii, în scopul de a face programarea mai atractivă, interesantă și pusă la dispoziția oricui.

- În centru se află personajele, este dinamic și sprijină crearea aplicațiilor de tip joc și animație. Prin utilizarea alternativă a diferitelor elemente de media – imagini, sunete – se deschide imaginația programatorului și se nasc programe interactive în care se reflectă creativitatea. Programatorul poate utiliza elementele încorporate, dar, de asemenea, poate importa propriile creații, în conformitate cu nevoile și gustul personal.
- Este orientat pe obiecte. Elementele de bază a unui program Scratch sunt personajele. Comenzile și instrucțiunile de fapt descriu proprietățile și comportamentul personajelor.
- Mediul este multilingvistic. Dacă este încorporată traducerea corespunzătoare, fiecare utilizator poate edita sursa în limba proprie, mai mult, poate vizualiza sursa celorlalți în propria limbă.[3]

## Cum funcționează Scratch?

Interfața limbajului Scratch este prietenoasă, lizibilă, cu acces direct la comenzi. Utilizarea lor adecvată este sprijinită de fereastra *Ajutor*, care oferă asistență rapidă. Limba interfaței se poate seta în funcție de opțiunea programatorului. Astfel, utilizatorii care folosesc variante diferite pentru programare pot înțelege cu ușurință codul sursă a celorlalți. Conține o bară de meniu, bară de instrumente, interfață de programare.



Interfața de programare are trei părți principale: secțiunea pentru comenzi, secțiunea de program și secțiunea de joc.

Plăcile colorate ale comenzilor, acoperă câte un grup de comenzi (Mișcare, Aspect, ... etc.). La selectare uneia dintre ele, în partea inferioară a ferestrei sunt vizualizate elementele grupului. Comenzile sunt afișate folosind structuri grafice pe care le distingem în funcție de culoarea și forma lor.

Folosind tehnica „drag-and-drop”, cu ajutorul mouse-ului putem trece comenzile și structurile de control din secțiunea pentru comenzi în secțiunea de program, și ca o construcție Lego, alăturăm piesele programului. Structura programului va fi clar vizibilă, iar imbricarea blocurilor va fi bine evidențiată. Blocurile de comenzi se pot mișca cu ușurință, parametrii, condițiile se pot rescrie în mod simplu. Deoarece comenzile nu trebuiesc tastate, nu vor apărea greșeli de sintaxă în program. Unde totuși putem modifica manual, vom produce eventuale erori logice, deoarece se permite introducerea doar a unor valori de tip potrivit sau de expresii corespunzătoare. Feedback-ul este imediat, nu sunt mesaje de eroare, astfel experiența neplăcută a eșecului lipsește. [6]

Secțiunea de program pe lângă sarcini (*Scripts*), conține caracteristicile de costum (*Costumes*) și de sunet (*Sounds*) pentru personaje.

Executarea instrucțiunilor se poate urmări în secțiunea de joc, astfel sentimentul reușitei este imediată. Deoarece codul sursă al programului și rezultatul rulării programului sunt vizibile deodată, erorile logice se pot detecta și, implicit, corecta rapid.

Încărcarea personajelor se efectuează de asemenea pe secțiunea de joc. Se pot edita caracteristicile acestora și se pot personaliza în fereastra de pictură.

### **Aplicabilitate**

Mulțimea de aplicații, încărcate pe siteul Scratch de-a lungul anilor, dovedește că elevii învață cu drag acest limbaj de programare, iar profesorii îl folosesc cu succes în instruire.

Este un instrument simplu în predarea elementelor de bază al programării și în același timp, datorită caracteristicii de limbaj orientat pe obiecte, interpretat, cu sprijinirea executării paralele a instrucțiunilor, este potrivit pentru crearea unor aplicații complexe. Prin Scratch poate fi învățată rigurozitatea necesară dezvoltării programelor, și poate fi folosită eficient în transferul de cunoștințe precum: [8]

- Înțelegerea noțiunilor de font, grafică, în scopul creării unui produs finit atractiv;
- Prelucrarea intuitivă a elementelor de grafică și de sunet în aplicațiile multimedia;
- Transferarea informațiilor prin intermediul animațiilor și a muzicii;
- Elaborarea unor aplicații, care reflectă caracterul transdisciplinar al subiectului prelucrat;
- Proiectarea, testarea controlului evenimentelor prin modificarea secvențelor de instrucțiuni;
- Mișcarea personajelor pe ecran, și reacția acestora la mediu;
- Elaborarea unor secvențe de instrucțiuni;
- Utilizarea instrucțiunilor condiționale, în scopul determinării cursului unor acțiuni;
- Executarea repetată a unor secvențe de instrucțiuni;
- Tratarea tastaturii și a altor evenimente;
- Utilizarea variabilelor;
- Transmiterea cerințelor și a mesajelor în beneficiul verificării aplicațiilor.

Utilizarea Scratch, din perspectiva caracteristicii de vârstă a elevilor, prezintă două alternative. Cei mici (8-12 ani), sunt captivați în special de realizarea animațiilor. Elaborarea acestora necesită cunoștințe și experiență de programare sumară, însă permite experimentare iar rezultatul este spectaculos. Ei învață să utilizeze un alt editor pentru desene, gustă puțin din multimedia. Fac cunoștințe cu instrucțiunile simple, și înțeleg esența ciclurilor repetitive. Elevii mai mari (13-16 ani), se pot familiariza cu misterele elaborării programelor de tip joc. La acest nivel, este importantă cunoașterea exactă a funcționării instrucțiunilor. Particularitatea mediului este faptul că personajele își execută sarcinile în paralel. Condiția prealabilă a funcționării corecte a

programului este proiectarea corectă a algoritmului, recunoașterea relațiilor existente între personaje, planificarea și verificarea.[2]

În cazul în care elevii ar ajunge la nivelul de a fi capabili de abstractizare, de a descompune problema pe elemente simple, iar apoi să le construiască înapoi într-o structură logică, care rezultă un program ce funcționează corect, atunci efortul elevului și a profesorului deopotrivă, nu a fost în zadar.

### **Bibliografie:**

1. Răzvan Bologa, Scratch, Revista Informatica Economică nr. 1 (37), București, 2006
2. Takács Valéria, Tananyagkészítés a Scratch programozási környezethez,  
[http://scratch.inf.elte.hu/stuff/scratch\\_bev\\_modszertan.pdf](http://scratch.inf.elte.hu/stuff/scratch_bev_modszertan.pdf)
3. Mitchel Resnick, John Maloney, Andrés Monroy Hernández, Natalie Rusk, Evelyn Eastmond, Karen Brennan, Amon Millner, Eric Rosenbaum, Jay Silver, Brian Silverman, Yasmin Kafai, Scratch: Programming for Everyone, Accepted for publication in Communications of the ACM (CACM), <http://scratch.wiki.hoover.k12.al.us/file/view/scratch-cacm.pdf>
4. <http://scratch.mit.edu>
5. <http://scratch.inf.elte.hu>
6. [http://scratch.inf.elte.hu/stuff/scratch\\_infoera2007.pdf](http://scratch.inf.elte.hu/stuff/scratch_infoera2007.pdf)
7. <http://wik.ed.uiuc.edu/index.php/Scratch>
8. [http://wik.ed.uiuc.edu/index.php/Scratch\\_Student\\_Project](http://wik.ed.uiuc.edu/index.php/Scratch_Student_Project)