Acta et Commentationes, Sciences of Education, nr. 2(36), 2024 ISSN 1857-0623
p. 59-67 E-ISSN 2587-3636

CZU: 378:004.4 DOI: 10.36120/2587-3636.v36i2.59-67

ABORDAREA ÎNVĂȚĂRII BAZATE PE SARCINI ÎN PROGRAMAREA JAVA

Tatiana CHIRIAC, dr., conf. univ.
https://orcid.org/0000-0002-6122-1937
Universitatea Pedagogică de Stat „Ion Creanga” din mun. Chișinău

Rezumat. În lucrare se analizează abordarea învățării bazate pe sarcini în contextul limbajului de programare Java. Se evidențiază ideea că abordarea bazată pe sarcini aplicată în studierea programării ajută studenții să stabilească conexiuni clare între dezvoltarea de competențe în cadrul unui curs și probleme din viața reală. Întrebările cercetării date urmăresc elucidarea aspectelor legate de abordarea bazată pe sarcini în literatura de specialitate și cum poate fi aplicată aceasta în elaborarea unui program bazat pe sarcini în studierea limbajului de programare Java. Dezvoltarea unui program bazat pe sarcini implică stabilirea de obiective clar declarate, selectarea sarcinilor și furnizarea de formare integrată, ținând cont de alte cerințe ale cursanților.
Cuvinte-cheie: învățare activă, învățarea bazată pe sarcini, elementele cheie ale unui program bazat pe sarcini.

TASK-BASED LEARNING APPROACH IN JAVA PROGRAMMING

Abstract. The task-based learning methodology is examined in this study within the framework of the Java programming language. It emphasizes the idea that students are better able to draw connections between the skills they are learning in a course and real-world issues when they use a task-based approach when studying programming. The purpose of the provided research questions is to clarify various features of the task-based approach found in specialized literature and to demonstrate how it may be used when learning Java programming to construct a task-based program. Choosing activities, integrating training, and establishing explicit objectives are all necessary when creating a task-based program that takes into consideration the needs of the learners.
Keywords: active learning, Task-Based Learning, key elements of a task-based program.

Introducere

Învățarea unui limbaj de programare este o parte esențială a formării fiecărui student din domeniul tehnologiilor informaționale și informatică. Dincolo de aplicațiile software, programarea reprezintă primul pas în înțelegerea naturii impactului indiscutabil al informaticii asupra lumii moderne. Scopul principal în procesul de predare a unui limbaj de programare este de a oferi studenților experiență și tehnici de bază necesare pentru a utiliza eficient elementele de programare. În prezent, cele mai reușite practici de predare-învățare a programării sunt cele care cultivă studenților o abordare pragmatică a scrierii unui program, care poate fi o experiență naturală, satisfăcătoare și creativă și nu doar o sarcină rezervată experților în domeniu. În învățarea unui limbaj de programare este important să introducem progresiv concepte esențiale, folosind metode active de predare-învățare, să utilizăm aplicații clasice de matematică aplicată și științe, treptat trecând la concepte ce oferă studenților oportunități de a scrie programe în mod individual pentru a rezolva probleme complexe.

În ultimele decenii, tehnicile de învățare activă în cadrul orelor universitare tradiționale se utilizează tot mai intens. Ideea principală este că învățarea tradițională (profesorul predă și studenții ascultă) este foarte pasivă, de aceea pentru a depăși anumite probleme, profesorul încearcă o varietate de tehnici, sintetizate sub conceptul de „învățare activă", pentru a implica studenții în propria formare.

Învățarea activă este o abordare largă care include mai multe forme interactive de predare și învățare, concentrându-ne pe modul în care studenții asimilează informația și învață (cum?), nu doar pe ceea ce învață (ce?). Această abordare urmărește implicarea activă a studenților în dobândirea de competențe și încurajează procese de gândire mai complexe. Oportunitățile oferite de învățare activă, cum ar fi învățarea bazată pe sarcini, învățarea bazată pe probleme, investigația, studiu de caz și alte metode, provoacă studenții și îi sprijină să-și dezvolte cunoștințele și propria înțelegere a unui domeniu. În studiul realizat de grupul Adam et al, tehnicile de învățare activă sunt grupate sub denumiri generice precum „învățare activă", „strategii de învățare pentru studenții activi", „învățare activă la clasă", „activități de învățare activă", menționându-se necesitatea atragerii studenților în activități semnificative care le cer să se gândească la „cea ce fac" și „de ce o fac" [1]. Pentru a crește și a menține motivația studenților în procesul de învățare al cursului „Elemente de programare", autorii Mironova et al, specifică că este necesar ca procesul de învățare de rutină să fie mai atractiv și mai dinamic, evidențiind metode de tip e-learning, prelegeri interactive, aplicații interactive de tip „games", lucrul în grup asupra sarcinilor, utilizarea programelor gata făcute pentru a prezenta unele subiecte, în care, în prealabil, se conțin greșeli în cod pentru a fi discutate și corectate împreună cu studenții – metodele aplicate având ca scop creșterea interesului studenților pentru disciplina de programare [2].

Idea învățării limbajelor de programare prin abordarea învățării centrată pe student, a fost promovată de autor și într-o lucrare anterioară (Chiriac, 2023), în care se specifică necesitatea utilizării proiectelor bazate pe sarcini și proiectelor individuale, sau utilizarea interacțiunii complexe între învățarea practicii și învățarea teoriei [3].

Metodologia cercetării

Acest articol oferă o privire de ansamblu asupra modului în care tehnicile de învățare activă, precum învățarea bazată pe sarcini, pot fi aplicate în cadrul cursurilor introductive de programare. Învățarea bazată pe sarcini în contextul limbajului de programare Java poate fi un model și pentru alte limbaje de programare, subliniind ideea că abordarea bazată pe sarcini a programării ajută studenții să stabilească conexiuni clare între dezvoltarea de competențe în cadrul unui curs și probleme din viața reală. În acest context se analizează:

- Cum este interpretată abordarea bazată pe sarcini în literatura de specialitate;
- Cum se aplică abordarea bazată pe sarcini în programare.

Orice unitate de curs de programare conține un set de materiale didactice și tehnologii care urmăresc formarea conexiunilor dintre dezvoltarea de competențe, aplicarea componentei de evaluare și asigurarea tranziției viitorilor absolvenți către angajare și formare continuă. În condițiile în care abordarea bazată pe sarcini se aplică în învățarea limbajului de programare Java, studentul devine capabil să scrie programe individual, care ulterior îl ajută la rezolvarea diferitor probleme complexe din viața reală. Luând în considerație că limbajul de programare Java conține câteva zeci de cuvinte de vocabular și reguli de sintaxă, o mare parte fiind exprimate și în limbajul C sau C++, învățarea programării devine o provocare doar în momentul când trebuie de introdus detalii relevante într-o situație dată de programare.

Abordarea bazată pe sarcini: paradigmă de învățare

Învățarea bazată pe sarcini (din engleză Task-Based Learning) este un procedeu de secvențiere a activităților din cadrul lecțiilor, care, conform literaturii de specialitate, mai des se utilizează în predarea limbilor străine (TBLT, cea ce subînțelege „predarea bazată pe sarcini a limbilor"). În opinia autoarei Kawasaki, predarea bazată pe sarcini este una dintre multele metode moderne de predare a limbilor ce se concentrează pe stabilirea unui obiectiv pentru studenți – fie un raport, un videoclip sau o prezentare – urmând trei pași principali pentru a atinge acest obiectiv: sarcina prealabilă, sarcina de bază, post-sarcina [4]. Cu referire la site-ul centrului de formare Europass Teacher Academy (https://www.teacheracademy.eu/blog/task-based-learning/), „învățarea bazată pe sarcini a fost (mai mult sau mai puțin) definită de cercetătorii lingvistici ca: lucruri pe care oamenii le fac în viața de zi cu zi (Long, 1985); o activitate orientată spre un scop care duce la un rezultat (Willis, 1996); un plan de lucru finalizat care poate fi evaluat (Ellis, 2003)". În acord cu autorii Weidinger și Borer „învățarea bazată pe sarcini reprezintă combinația ideală dintre învățarea constructivistă și învățarea cu ajutorul instrucțiunilor", iar „în procesul rezolvării sarcinilor, cursanții explorează mai multe căi de găsire a soluției și, astfel, dobândesc competențe și abilități utile" [5, p.10]. Autoarele Bulat și Ursu concluzionează că „ÎBSL (învățare bazată pe sarcini de lucru) se bazează pe ideea că învățarea este mai eficientă atunci când este centrată pe experiență, iar formabilii sunt implicați activ în procesul de învățare, educație, formare și dezvoltare" [6, p. 251].

Într-o analiza mai amplă privind designului instrucțional, autorii Kirschner și van Merriënboer analizează că sarcinile de învățare sunt una dintre cele 4 componente ale învățării complexe (sarcini de învățare, informații de suport, informații procedurale și exersarea sarcinilor parțiale). În cazul acestui studiu sarcinile de învățare pot fi studii de caz, proiecte, probleme etc. Acestea reprezintă experiențe autentice, bazate pe sarcini din viața reală, care vizează integrarea abilităților, cunoștințelor și atitudinilor [7, p. 246]. Informațiile de suport ajută studenții să realizeze activități ale sarcinilor de învățare, care

implică adesea rezolvarea problemelor și raționamentul. Acestea oferă o punte între cunoștințele anterioare și cele viitoare pentru a realiza sarcinile de învățare. Informațiile procedurale permit studenților să execute aspecte de rutină ale sarcinilor de învățare care sunt întotdeauna efectuate similar. Aceste aspecte se estompează rapid pe măsură ce studenții dobândesc mai multă experiență. În cele din urmă, exersarea sarcinilor parțiale se referă la exersarea suplimentară a aspectelor de rutină, astfel încât studenții să poată dezvolta un nivel ridicat de automatism în realizarea de sarcini [8]. Modelul analizat de autorii menționați subliniază interdependența elementelor care constituie un sistem de învățare, recunoscând natura dinamică a acestei interdependențe. O astfel de abordare este sistematică prin paradigma „input-proces-output", în care ieșirile anumitor elemente ale sistemului de învățare servesc ca intrări pentru următoarele elemente, iar ieșirile anumitor activități de proiectare servesc ca intrări pentru alte activități [7].

În cele din urmă, se va specifica că abordarea învățării bazată pe sarcini ajută studenții să stabilească conexiuni clare între cunoștințele anterioare și noile experiențe și concepte. Analiza și proiectarea sarcinilor și a conținutului, bazate pe strategiile de învățare activă, în mod obișnuit, implică un amestec de sarcini individuale și colaborative, oferind studenților atât experiențe individuale de învățare, dar și oportunități de a împărtăși și discuta ideile lor cu colegii. Activitățile pot fi organizate în cadrul unei lecții, cât și planificate în segmente mari ale unei perioade de curs (temă, modul), scopul fiind de a activa procesele cognitive ale studenților și de a valora cunoștințele anterioare pentru a servi drept pentru înțelegerea noilor informații.

Abordarea bazată pe sarcini în programarea Java

Modelul învățării active a limbajelor de programare poate fi transpus în contextul abordării *învățare bazate pe sarcini (ÎBS)*. Conform cercetării realizate de autorii Figas, Bartel & Hagel, un limbaj de programare este un limbaj prin definiție, de aceea învățarea bazată pe sarcini poate fi utilizată și pentru învățarea programării, nu doar pentru studierea limbilor străine [9]. Învățarea bazată pe sarcini este, de asemenea, o abordare ce contribuie la reținerea eficientă a conceptelor fundamentale în programare, la formarea de abilități de bază în rezolvarea problemelor de programare care sunt aplicabile în diverse medii de programare, și poate fi o paradigma eficientă de învățare destinată studenților fără experiență anterioară în programare. Într-un context general, prin învățarea bazată pe sarcini studenții învață prin cercetarea metodelor de rezolvare a unei probleme, stabilind, de fapt, pașii/algoritmul către soluția problemei. Prin asociere putem spune că multe situații din viața reală urmăresc găsirea de soluții pentru probleme.

Conform ghidului „Practitioner Guide to Task-Based Programming", un program care este organizat în jurul sarcinilor este important pentru cursanți și alți actori în acord cu următoarele trei motive [10, p. 2]:

1. *O sarcină poate fi înțeleasă în termeni reali*, ceea ce presupune că sarcinile formulate în acord cu obiectivele învățării, permit studenților să vadă legătura dintre învățarea propriu zisă și integrarea rezultatelor învățării în activitățile de zi cu zi.
2. *O sarcină poate fi standardizată* în acord cu descriptorii de sarcină, indicatorii și exemplele de sarcini din cadrul curriculumului.
3. *O sarcină poate fi evaluată*, procesul fiind foarte important pentru înregistrarea progresului studenților și al succesului general al unui program de studiu.

Abordarea învățării bazate pe sarcini, ca și orice altă strategie didactică, necesită contextualizarea algoritmului de implementare a acesteia. Proiectarea și dezvoltarea adecvată a sarcinilor pentru studenți va asigura realizarea scopului și obiectivelor de învățare în cadrul oricărei unități de curs. Astfel, etapele de bază în planificarea unor sarcini de învățare vizează:

- etapa premergătoare de introducere tematică și explicarea sarcinilor de lucru;
- etapa de realizare a sarcinilor, care poate fi una ciclică;
- etapa de raportare a sarcinilor realizate.

Șablonul pentru a planifica o activitate de învățare bazată pe sarcini este foarte simplu: „proiectați/creați/implementați un program care ... ”. Totodată, elaborarea unui proiect eficient al unei activități de învățare bazată pe sarcini depinde de stabilirea exactă a obiectivelor educaționale și de elaborarea unui set de sarcini adecvate atingerii acestor obiective. Acest proiect sau plan de lecție va specifică tema și obiectivele specifice, informații cheie pentru înțelegerea noțiunilor și acumularea de cunoștințe, exemple de programe necesare în detalierea elementelor unui programul (cuvinte-cheie, instrucțiuni, metode etc.), activități de exersare independentă a unor programe, prezentarea rezultatelor și strategia de evaluare.

De exemplu, pentru a prezenta studenților dezvoltarea programelor Java, la prima lecție studenții trebuie să cunoască pașii de bază necesari pentru a rula un program simplu. Sistemul sau platforma Java este o colecție de aplicații, care ajută utilizatorii să dezvolte și să ruleze eficient aplicații de programare Java. Spre deosebire de alte limbaje, sistemul Java include un motor de execuție, un compilator și un set de biblioteci incorporat [11]. După analiza noțiunilor legate de instalarea mediului de programare Java pe computer (care ar putea fi și o sarcină în sine individuală pentru utilizatorii mai experimentați), pentru a programa în Java, putem planifica următoarele sarcini:

- Creați un program într-un editor de text salvând fișierul java (de exemplu, PrimCode.java);
- Compilați fișierul java tastând javac PrimCode.java într-o fereastră Windows terminal;
- Executați fișierul java tastând java numele clasei (.class) într-o fereastră Windows terminal;

– Creați primul proiect Java HelloWorld folosind IDE-ul Eclipse.

După ce studenții fac cunoștință cu noțiunile de bază: class, main() și succesiunea de afirmații cuprinse între acolade System.out.print("Hello, World"), System.out.println(), se pot propune alte sarcini precum: Creați un proiect Java care să afișeze în consolă o strofă din poezia lui Gr. Vieru, „Mi-e dor de tine, mamă” sau altele, pentru a exersa utilizarea parametrilor pe care studenții doresc să le imprime pe ecranul de ieșire.

Totodată, unele studii atenționează că conceptul de „sarcină” deseori este confundat cu conceptul „exercițiu”, cât și „activitate” [12]. Deși sarcinile și exercițiile pot părea similare, ele servesc unor scopuri diferite. Un exercițiu este o procedură de predare care implică practica ghidată și controlată în studierea unor concepte. Conform autorului Shehadeh (apud Richards & Schmidt), exercițiul este „o activitate care este concepută pentru a exersa un element de învățare” [ib], pe când sarcinile sunt de obicei efectuate pentru a atinge un obiectiv specific și se referă la o anumită lucrare. Sarcinile pot fi simple sau complexe și pot fi îndeplinite individual sau ca parte a unei echipe. De asemenea, sarcinile sunt realizate de studenți folosind materialele existente sau pe cele care au fost furnizate la etapa premergătoare sarcinii. Sarcinile în activități de programare oferă oportunități de reflecție asupra înțelegerii tehnicilor de programare și însușirii elementelor de sintaxă a unei limbi. Termenul activitate este general și se referă la orice procedeu didactic care implică studenții în atingerea obiectivelor educaționale.

În scopul de a ghida studenții prin înțelegerea structurii unui program Java, bazele Java, sintaxa, operatori, matrice, șiruri de caractere și altele, până la programarea orientată pe obiecte explorând clase și metode, profesorul va utiliza abordarea bazată pe sarcini în programarea Java pentru a aprofunda înțelegerea și a perfecționa abilitățile de programare. Sarcinile de învățare aplicate în programare oferă experiență practică în rezolvarea problemelor, contribuie la consolidarea noțiunilor de bază și elementelor fundamentale de programare Java. Exemple de sarcini în programarea Java pot fi: elaborați un program Java, folosind clasa *Scanner*, care va aduna două numere introduse de la tastieră; elaborați un program Java, folosind clasa *String*, care va compara două șiruri de caractere; elaborați un program Java, folosind clasa *Scanner* pentru a prelua intrarea de la utilizator, care va elimina spațiile dintre cuvintele introduse etc. În tabelul de mai jos este prezentat un exemplu o activitate de învățare bazată pe sarcini.

Tabel 1. Exemplu de abordare a învățării bazate pe sarcini

Sarcina premergătoare: Elaborați programul Java care va ilustra utilizarea instrucțiunii *if* în jocul „Ghicește numărul”	

`import static java.lang.System.*;`
`import java.util.Scanner;`
`import java.util.Random;`
`public class ghicesteNumar1 {`
`public static void main(String args[]) {`
	Introduceti un numar de 1 pana la 10:
	5
	Nu este corect.
	Numarul generat aleator este: 7.

```
Scanner tastiera = new Scanner(System.in);
out.print("Introduceti un numar de 1 pana la
10: ");
int inputNumar = tastiera.nextInt();
int randomNumar = new Random().nextInt(10) +
1;
if (inputNumar == randomNumar) {
out.println("*Ati ghicit numarul.*"); }
else {
out.println("Nu este corect.");
out.print("Numarul generat aleator este: ");
out.println(randomNumar + ".");
}
out.println("Sfarsit!");
} }
```

Sfarsit!

Sarcina: Modificați programul Java care va ilustra utilizarea instrucțiunii *if* în jocul „Ghicește numărul”. Creați o metodă *ghiceste()*, declarată în cadrul clasei *ghicesteNumar2*.
Abordarea este de a genera un număr aliator folosind metoda Math.random() în Java.
Folosind o buclă, specificați intrarea K de la tastieră și pentru fiecare intrare afișați dacă numărul este mai mic sau mai mare decât numărul randomizat de program.
Dacă în 3 încercări utilizatorul a ghicit corect numărul, afișați mesajul că utilizatorul a ghicit numărul. În caz contrat, imprimați că nu a putut să ghicească și apoi imprimați randomizat de program.

```
import java.util.Scanner;
public class ghicesteNumar2 {
    public static void ghiceste()     {
Scanner sc = new Scanner(System.in);
        int number = 1 + (int)(10*
Math.random());
        int K = 3;
        int i, a;
System.out.println("Ghiceste numarul de la 1
pana la 10," + " avand la dispozitie 3
incercari. ");
        for (i = 0; i < K; i++) {
System.out.println("Ghiceste numarul:");
        a = sc.nextInt();
        if (number == a) {
System.out.println("Felicitari!" + " Ati
ghicit numarul.");
        break;            }
       else if (number > a && i != K - 1) {
System.out.println("Numarul este " + "mai
mare ca " + a);             }
       else if (number < a && i != K - 1) {
System.out.println("Numarul este" + "mai mic
ca " + a);
```

Ghiceste numarul de la 1 pana la 10, avand la dispozitie 3 incercari.
Ghiceste numarul:
5
Numarul este mai mare ca 5
Ghiceste numarul:
8
Felicitari! Ati ghicit numarul.

Sau:
Ghiceste numarul de la 1 pana la 10, avand la dispozitie 3 incercari.
Ghiceste numarul:
7
Numarul este mai mic ca 7
Ghiceste numarul:
6
Numarul este mai mic ca 6
Ghiceste numarul:
5

`} }`
`if (i == K) {`
`System.out.println("Ati depasit numarul de incercari admise." );`
`System.out.println("Numarul generat aleatoriu: " + number);`
`} }`
`public static void main(String arg[])`
`{`
`ghiceste(); } }`
	Ati depasit numarul de incercari admise.
	Numarul generat aleatoriu: 3

Într-o interpretare foarte generală, învățarea bazată pe sarcini reprezintă un set de activități de învățare care conduc la dezvoltarea competențelor și formare continuă în scop de a echipa studenții pentru cerințele unui mediu de învățare academică. Aceste sarcini se concentrează pe activități care includ acumularea de note periodice în cadrul lecțiilor, pregătirea pentru teste sumative și examene, analiza literaturii de specialitate, rezolvarea de probleme complexe realizarea cercetărilor, elaborarea de proiecte individuale, lucrul în grup, realizarea practicii de specialitate etc.

Concluzii

În contextul cercetării date, se poate concluziona că învățarea activă aplicată sub forma abordării bazate pe sarcini în programarea Java, ajută nu numai studenții să înțeleagă noi cunoștințe și să obțină rezultate mai bune, ci îi ajută pe profesori să-și dezvolte, să-și îmbunătățească și să-și sporească abilitățile profesionale. În cadrul abordării bazate pe sarcini, profesorul se concentrează, în special, pe algoritmul programului, înțelegerea acestuia, decât pe predarea noțiunilor de sintaxă.

Bibliografie

1. ADAMS, M.E.; HAUTE, T.; RAY, I.P. Active Learning Strategies for Middle and Secondary School Teachers. Indiana State University, Northridge Middle School, Crawfordsville. 2016. https://www.in.gov/che/cte/files/active-learning-strategies-final.pdf (vizitat 20.04.2024).
2. MIRONOVA, O., AMITAN, I., VILIPÕLD, I., and SAAR, M. Active learning methods in programming for non-IT students. În: *International Conference e-Learning*, 2016. https://files.eric.ed.gov/fulltext/ED571498.pdf (vizitat 20.04.2024).
3. CHIRIAC, T. Proiectarea și implementarea cursului de programare Java: strategii, instrumente și provocări în predare. In: *Acta et commentationes (Ştiinţe ale Educaţiei)*, 2023, nr. 2(32), pp. 27-34. ISSN 1857-0623. DOI: https://doi.org/10.36120/2587-3636.v32i2.27-34 (vizitat 10.04.2024).

4. KAWASAKI, J. What Is Task-Based Learning? A Guide to the Popular Teaching Method. *BridgeUniverse™*. 2021. https://bridge.edu/tefl/blog/what-is-task-based-learning/ (vizitat 22.04.2024).
5. WEIDINGER, W.; BORER C. Cum sprijinim învățarea activă – Broșură pentru profesori. În: *International Projects in Education,* 2017. Zurich University of Teacher Education. www.phzh.ch/ipe. https://ipe-textbooks.phzh.ch/globalassets/ipe-textbooks.phzh.ch/romanian/jobs-romanian/jobs-students/teacher-brochure/jobsface_teacher_training_ro_gzd3_171030_double_page.pdf (vizitat 26.04.2024).
6. BULAT (IURCISIN), A.; URSU, L. Evoluția conceptului de învățare bazată pe sarcini de lucru în științele educației. In: *Science and education: new approaches and perspectives*, Ed. 25, 24-25 martie 2023, Chişinău, Vol.2, pp. 247-254. ISBN 978-9975-46-783-4. DOI: https://doi.org/10.46727/c.v2.24-25-03-2023.p247-254 (vizitat 12.04.2024).
7. KIRSCHNER, P.; VAN MERRIËNBOER, J. Ten Steps to Complex Learning. A New Approach to Instruction and Instructional Design. https://web.mit.edu/xtalks/TenStepsToComplexLearning-Kirschner-VanMerrienboer.pdf(vizitat 12.05.2024).
8. FIGAS, P., BARTEL, A., HAGEL, G. Task-based programming learning in higher education. În: *IEEE Global Engineering Education Conference (EDUCON)*, 2015, 648-653. https://www.researchgate.net/publication/271137026_Task-based_Programming_Learning_in_Higher_Education (vizitat 10.05.2024).
9. UTRECHT P. K.; VAN MERRIËNBOER, J. J. G. Ten Steps to Complex Learning A New Approach to Instruction and Instructional Design. În: *TechTrends* 2018. nr. 62, pp. 204–205. https://doi.org/10.1007/s11528-018-0254-0 (vizitat 09.05.2024).
10. Practitioner Guide to Task-Based Programming. Ontario Adult Literacy Curriculum Framework. Ontario Ministry of Training, Colleges and Universities, March 2011. https://cesba.com/wp-content/uploads/2020/10/OALCF-Guide-to-Task-Basked-Programming.pdf (vizitat 13.05.2024).
11. HARTMAN, J. Ce este Java? Definiția, semnificația și caracteristicile platformelor Java. Site-ul oficial Guru99. 2023. https://www.guru99.com/ro/java-platform.html#1 (vizitat 14.05.2024).
12. SHEHADEH, A. It is a task, not an exercise: What is the difference? În: *System*. 2024, Vol. 123, ISSN 0346-251X, https://doi.org/10.1016/j.system.2024.103299. https://www.sciencedirect.com/science/article/pii/S0346251X24000812 (vizitat 15.05.2024).