

POSSIBILITĂȚILE DIDACTICE ALE VIZUALIZATORULUI DE ALGORITMI ÎN PROCESUL STUDIERII DISCIPLINEI „BAZELE PROGRAMĂRII”

Diana MOGLAN, dr., conf. univ.,
Facultatea de Științe Reale, Economice și ale Mediului,
Universitatea de Stat „Alec Russo” din Bălți

Abstract: *The article is devoted to the didactic potential on the use of the visualizer of algorithms as software, which graphically demonstrates the work of algorithms for processing input data. The author proposed to use the algorithm visualizer, as an auxiliary visual means of teaching algorithmization, for more successful mastering the mechanism of the algorithms at lectures.*

Keywords: *programming, algorithm, visualization of algorithms, visualizer, educational process.*

În cadrul Universității de Stat „Alec Russo” din Bălți la studiile superioare de licență este realizată formarea specialiștilor la specialitatea „Informatică”. Conform planului de învățământ al specialității 444.1 „Informatică” și Cadrului Național al Calificărilor profesionale din Republica Moldova [1], un viitor specialist în domeniul informaticii și tehnologiilor informaționale ar trebui să aibă următoarele competențe profesionale la finalizarea instruirii:

- capacitatea de a opera cu fundamentele științifice ale informaticii și matematicii și de utilizare a acestor noțiuni în comunicarea profesională;
- capacitatea de a elabora modele pentru descrierea fenomenelor și proceselor reale;
- capacitatea de a proiecta, elabora și analiza algoritmi pentru soluționarea problemelor din activitatea profesională;
- capacitatea de a programa, dezvolta și întreține aplicațiile informatice în limbaje de nivel înalt;
- capacitatea de a integra tehnologiile informaționale în diferite domenii de activitate profesională;
- capacitatea de a prelucra, analiza și interpreta date.

Pentru activitatea profesională de succes a viitorilor absolvenți la specialitatea „Informatică”, un rol important îl are studierea bazelor algoritmicizării și metodelor de prelucrare a structurilor de date, care creează fundamentul (cunoștințele de bază, abilitățile și deprinderile) pentru formarea competențelor profesionale ale absolvenților. Studiarea programării poate juca un rol decisiv în îmbunătățirea abilităților cognitive ale studenților prin predarea algoritmilor fundamentali și a tehnicilor de programare, deoarece contribuie la dezvoltarea abilităților studenților de a întocmi un plan de soluționare a unei probleme specifice, de a descrie într-o formă specială succesiunea acțiunilor pentru obținerea rezultatului dorit, de a prognoza rezultatul acțiunilor proiectate și de a face corectări în funcție de datele finale obținute. Prin crearea algoritmilor studenții își dezvoltă abilitățile de a analiza, compara, generaliza, trage concluzii logice, descrie un plan de activitate și de a-și exprima propriile gânduri într-o secvență logică strictă [2].

În cadrul curriculumului pentru specialitatea 444.1 „Informatică”, una dintre disciplinele fundamentale este „Bazele programării”, studiată de studenții anului I. Această disciplină este dedicată studierii formelor de bază de organizare și stocare a datelor într-un computer, structurilor de control utilizate în programarea structurală (structură liniară, structură alternativă, structură repetitivă), abstracției în programare și metodelor standarde pentru prelucrarea structurilor de date [2].

Activitatea de succes a unui specialist în domeniul tehnologiilor informaționale depinde nu numai de cunoașterea limbajului de programare, pe care îl folosește atunci când dezvoltă aplicații pentru rezolvarea problemelor, ci și, într-o măsură mai mare, depinde de „capitalul” cunoștințelor în domeniul proiectării algoritmilor bine structurați de prelucrare a informațiilor reprezentate prin diverse structuri de date [3, 4] și experiența acumulată în rezolvarea problemelor aplicative.

În timpul studierii programării, studenții întâmpină destul de des dificultăți semnificative în a găsi răspunsul la întrebarea de ce un anumit algoritm este potrivit pentru rezolvarea problemei și că acesta va permite obținerea datelor finale necesare. Justificarea corectitudinii algoritmului pentru studenți este destul de dificilă dacă nu pot înțelege proprietățile obiectului, să evidențieze conexiunile logice dintre obiectele separate și să se abată de la alte proprietăți și conexiuni neesențiale. De exemplu, cunoașterea structurii unui ciclu cu contor nu înseamnă că există abilitatea de construire în realitate a unui algoritm pentru calculul temperaturii medii lunare.

În opinia noastră, pentru stăpânirea cu succes a mecanismului de funcționare a algoritmilor, o condiție necesară reprezintă descrierea vizuală a algoritmului la fiecare etapă de execuție a acestuia. Vizualizarea este procesul de prezentare a informațiilor, datelor, cunoștințelor sub forma unor imagini, cu scopul de a maximiza comoditatea percepției, înțelegerii și analizei lor [5].

Reprezentarea vizuală a algoritmului permite studentului să înțeleagă și să vadă mai bine ce elemente structurale sunt utilizate în proiectarea algoritmului, cum este descrisă logica interacțiunii elementelor structurale ale algoritmului, ce obiecte reale corespund anumitor elemente din problema rezolvată. Pentru a înțelege structura algoritmului și acțiunile care au loc în interiorul acestuia, este necesar să fie văzut clar fiecare pas al algoritmului, ceea ce va permite studenților să construiască un model mental eficient prin simularea comportamentului algoritmului executat.

În calitate de instrument de instruire vizuală, care ar permite implementarea acestei abordări, poate fi utilizat *vizualizatorul de algoritmi*. Vizualizatorii simulează procesele de lucru al algoritmilor, permițând studenților, care utilizează interfața intuitivă de utilizator a vizualizatorilor, să parcurgă algoritmul pas cu pas pentru un set de date definit [6].

Unul dintre punctele forte ale vizualizatorului algoritmilor este posibilitatea de a utiliza canalul auditiv și vizual pentru îmbunătățirea percepției informațiilor metodice în procesul de instruire. Studiile făcute de Z. Katai [7] au confirmat că, dacă metodele și instrumentele de predare influențează asupra diferitor simțuri ale studenților, aceștia din urmă vor fi capabili să asimileze mai multe informații și să mențină eficient instruirea în cadrul studierii algoritmilor.

Vom menționa caracteristicile distinctive ale vizualizatorilor [8]:

1. Interactivitatea și caracterul intuitiv în gestionarea procesului de vizualizare de către utilizator;
2. Simplitatea de utilizare a interfeței vizualizatorului;
3. Afișarea mersului execuției algoritmului. Vizualizatorul afișează atât modificări ale valorilor variabilelor algoritmului, cât și acțiunile asupra acestora;
4. Disponibilitatea explicațiilor. Pe parcursul lucrului vizualizatorului sunt afișate comentarii text, care explică fiecare pas al algoritmului;
5. Susținerea a două moduri de vizualizare: pas cu pas și automat.

În procesul de învățare, vizualizatorii algoritmilor pot fi utilizați în conformitate cu următoarele cerințe [9]:

- Afișarea datelor de intrare și ieșire într-o formă intuitivă prin care se demonstrează execuția algoritmului – studenților ar trebui să li se ofere posibilitatea de a defini seturi arbitrare de date de intrare și de a vizualiza execuția algoritmului pentru acestea.
- Vizualizarea acțiunilor algoritmului pentru toate elementele vizualizate – elevilor trebuie să li se ofere posibilitatea de a vizualiza nu numai modificările aduse de vizualizator în setul de date, dar și alte acțiuni, ca de exemplu, comparații atunci când se caută elementul minim din tabel.
- Afișarea comentariilor la execuția algoritmului – în timpul funcționării algoritmului trebuie afișate comentarii, care să explice toate acțiunile efectuate la fiecare pas.
- Afișarea pas cu pas a lucrului algoritmului – studenții, bazându-se pe propriile abilități în cunoașterea materialului, ar trebui să poată urmări pașii algoritmului de la început până la sfârșit, pas cu pas.

În calitate de mediu de elaborare a vizualizatorului de algoritmi a fost ales mediul de programare Delphi, care include diverse instrumente pentru crearea de interfețe pentru vizualizatorii de algoritmi. Un exemplu de produs software elaborat pentru vizualizarea funcționării algoritmilor din compartimentul „Structuri de date statice. Tabele unidimensionale” este prezentat în figura 1.

Interfața grafică a vizualizatorului este realizată sub forma unei aplicații, care conține trei zone funcționale (vizualizare, cod algoritm, comentarii) și un panou de control.

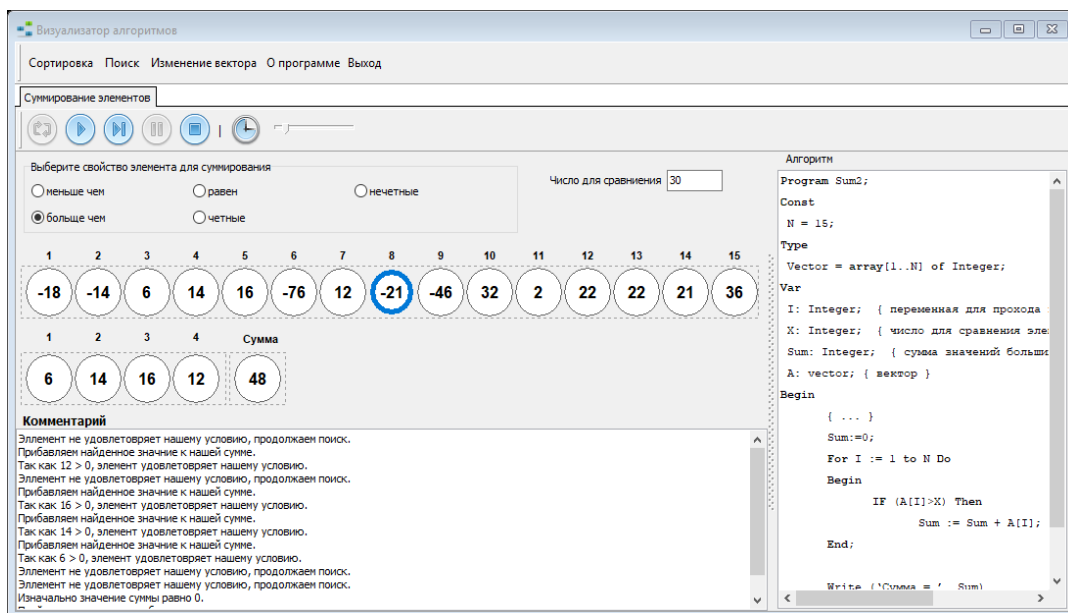


Fig. 1. Vizualizarea algoritmului de sumare a elementelor unui tabel

Fereastra de vizualizare este utilizată pentru a plasa elemente pe ecran și pentru a vizualiza funcționarea algoritmului. Fereastra cod este utilizată pentru a vizualiza codul algoritmului. Fereastra comentariilor este utilizată pentru a afișa explicații la acțiunea curentă de funcționare a algoritmului.

Procesul de vizualizare a algoritmilor este dirijat cu ajutorul panoului de control, pe care se află următoarele elemente de interfață:

- *Inițializare* – buton pentru inițializarea lucrului vizualizatorului, la pornirea vizualizatorului, seturile de date de intrare sunt generate aleatoriu;
- *Următor* – buton pentru executarea pas cu pas a algoritmului, adică de fiecare dată când este activat butonul se va executa un pas al algoritmului;
- *Automat* – buton pentru executarea automată a algoritmului fără intervenția în procesul de lucru al algoritmului;
- *Pauză* – buton pentru suspendarea temporară a lucrului vizualizatorului;
- *Stop* – buton pentru a suspenda complet activitatea vizualizatorului de algoritmi;
- *Viteza de mișcare* – buton pentru controlul vitezei de vizualizare a procesului de execuție a algoritmului (indicat în milisecunde).

Vom cerceta aplicarea vizualizatorului elaborat în procesul studierii disciplinei „Bazele programării” pe exemplul implementării algoritmului de determinare a elementului minim într-un tablou unidimensional de n numere întregi. Problema determinării elementului minim în tablou se rezolvă folosind următorul algoritmi:

```

Min:=A[1];
For i:=2 to n do
If A[i] < Min then
Min:=A[i];

```

În acest algoritmi, A este un tablou unidimensional, în care este căutat elementul minim, Min este valoarea curentă a minimului (printre primele i elemente ale tabloului). Inițial, valoarea minimă este inițializată cu valoarea primului element al tabloului.

Principalele date în vizualizator sunt elementele tabloului și valoarea elementului minim curent (Fig. 2). Pentru o reprezentare vizuală a funcționării algoritmului, valorile elementelor tabloului și ale elementului minim trebuie afișate constant în fereastra de vizualizare. Atunci când se caută elementul minim, cele mai „importante” acțiuni din algoritmi sunt operațiile de comparare a valorii elementului curent cu valoarea minimă, utilizând operatorul condiționat și operația asociată cu actualizarea valorii minime curente.

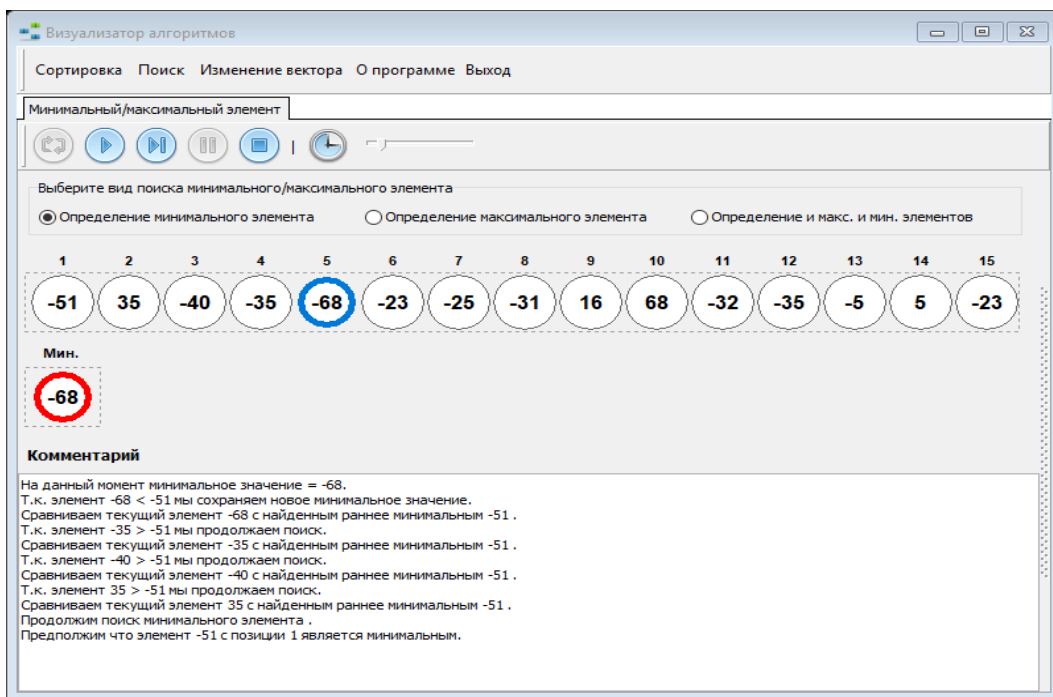


Fig. 2. Vizualizarea algoritmului căutării elementului minim în tablou

La fiecare pas al algoritmului, elementul curent al tabloului este evidențiat cu albastru, iar dacă se găsește o nouă valoare minimă, elementul corespunzător este evidențiat cu roșu. Pentru a explica funcționarea algoritmului, în fereastra de comentarii este afișat un mesaj despre operația efectuată la pasul curent al algoritmului. Menționăm că, la etapa de verificare a actualizării valorii minime, care corespunde operatorului condițional, sunt posibile două cazuri: pentru condiție falsă și adevărată. La fiecare etapă a algoritmului, valorile parametrilor sunt calculate și înlocuite în timpul executării algoritmului.

În comparație cu alte metode de desfășurare a unei lecții de prelegeri pentru explicarea algoritmilor, se pot distinge cele mai semnificative avantaje în utilizarea vizualizatoarelor:

- vizualizarea transmite conceptul dinamic al algoritmului;
- reprezentarea vizuală a lucrului algoritmului poate ajuta profesorii la orele de prelegeri să cheltuiască mai puțin timp pentru explicarea materialului teoretic și pentru demonstrarea studenților a funcționării algoritmilor în practică;
- vizualizarea dinamică a execuției algoritmului permite să fie afișată clar o caracteristică semnificativă a algoritmului, precum complexitatea acestuia, mai ales în timpul unei demonstrații pas cu pas, similar modulului de trasare (debug) al programelor.

În concluzie, sistematizăm rezultatele obținute în procesul de asimilare de către studenți a conținutului compartimentului „Structuri statice de date” din cadrul disciplinei „Bazele programării”, folosind un vizualizator de algoritmi. Experiența promovării orelor cu ajutorul vizualizatorului de algoritmi a arătat că, atunci când materialul teoretic este studiat folosind vizualizatorul la orele de prelegeri, este recomandabil să fie utilizate următoarele forme și metode de predare:

- cunoașterea preliminară independentă a studenților cu algoritmul studiat folosind vizualizatorul;
- analiza colectivă a diferitor algoritmi pentru un set dat de date de intrare folosind un computer și un proiector multimedia, explicând într-o formă vizuală conceptele de bază ale algoritmului studiat;
- sondaj sistematic în timpul orelor de prelegeri;
- oferirea studenților a tuturor vizualizatorilor în format electronic după prelegere;
- determinarea rezultatelor lucrului vizualizatorului pentru un algoritm specific (trasare pas cu pas).

Vizualizarea algoritmului oferă o modalitate alternativă de a explica studenților temele referitoare la elementele de bază ale algoritmizării și programării, care sunt mai ușor însușite prin modul vizual în comparație cu modul verbal. În rezultatul orei de prelegeri studenții vor profita de infor-

mații mai utile, atunci când la explicarea lucrului algoritmilor vor fi folosiți vizualizatorii de algoritmi în calitate de instrument auxiliar de instruire.

Bibliografie:

1. Cadrul național al calificărilor: învățământul superior. Chișinău: Bons Offices, 2015, 493 p. [online] [citată 21.09.2019]. Disponibil: https://mecc.gov.md/sites/default/files/cnc_22_31_32_33_34_38_42_44_55_85.pdf
2. DEINEGO, Nona, CABAC, Valeriu. *Bazele programării: Curs fundamental*. Bălți: Presa universitară bălțeană, 2013. 234 p.
3. КНУТ, Д. *Искусство программирования. Том 1. Основные алгоритмы*. Москва: Вильямс, 2002. 720 p.
4. КОРМЕН, Т., ЛЕЙЗЕРСОН, Ч., РИВЕС, Р. *Алгоритмы. Построение и анализ*. Москва: МЦНМО, 2001. 960 p.
5. БАЖЕНОВА, И. В. Визуальное моделирование в обучении программированию. In: *Вестник Челябинского государственного педагогического университета. Педагогические науки*, 2017, nr. 6, p. 15-20.
6. КАЗАКОВ, М.А., ШАЛЫТО, А.А. Использование автоматного программирования для реализации автоматов. In: *Компьютерные инструменты в образовании*, 2004, nr. 2, p. 19-33.
7. KATAI, Z., JUHASZ, K., ADORJANI, A.K. On the role of senses in education. In: *Computers & Education*, 2008, nr. 51 (4), P. 1707-1717.
8. AVANCENA, A.T., NISHIHARA, A. Usability and pedagogical assessment of an algorithm learning tool: A case study for an introductory programming course for highschool. In: *Issues in Informing Science and Information Technology*, 2015, nr. 12, p. 21-43. [online] [citată 21.09.2019]. Disponibil: <http://iisit.org/Vol12/IISITv12p021-043Avancena1817.pdf>
9. HUNDHAUSEN, C., DOUGLAS, S.A., STASKO, J.T. A meta-study of algorithm visualization effectiveness. In: *Journal of Visual Languages and Computing*, 2002, nr. 13, p. 259-290. [online] [citată 21.12.2019]. Disponibil: <https://www.cc.gatech.edu/~john.stasko/papers/jvlc.02.pdf>