

ИЗПОЛЗВАНЕ НА ДИГИТАЛНИ ИНСТРУМЕНТИ ЗА ВИЗУАЛИЗАЦИЯ НА АЛГОРИТМИ В ОБУЧЕНИЕТО ПО ПРОГРАМИРАНЕ

Елена Тодорова, Стефка Анева

Резюме. В настоящата статия са предложени педагогически подходи за използване и интегриране на визуализацията като средство за нагледност в учебния процес по програмиране в училище. Представени са различни дигитални инструменти и платформи, позволяващи динамичното представяне на алгоритми. Подчертава се значението на нагледността като средство за повишаване на мотивацията и ефективността в обучението по програмиране.

Ключови думи: дигитални инструменти, динамична визуализация, интерактивност, програмиране, методика на обучение.

Въведение

Нагледността е основен дидактически принцип, който има следните основни функции: улеснява възприемането и облекчава процеса на усвояване на знания; подобрява запаметяването; повишава интереса и мотивацията; съдейства за развитие на аналитично мислене у учениците; подпомага за разбиране на връзката между научните знания и житейската практика. При този принцип ефективността на обучението зависи до голяма степен от подходящото ангажиране на сетивата при възприемане и обработка на учебното съдържание. За тази цел се използват различни нагледни методи, при които източникът на информация се явява нагледния образ, който възниква в процеса на усещанията и възприятията [1, 3].

Съществуват четири основни форми на нагледност [2], които могат да бъдат използвани в образователния процес:

- предметнообразна – използва реални обекти, изображения, картини, модели, представящи конкретни явления или предмети от действителността;
- словеснообразна – постига се чрез езикови средства, които създават образни представи за характеристиката на обекти, явления и процеси, като се използват описания, сравнения и др.;
- условноизобразителна – реализира се чрез схеми, чертежи, таблици,

графики и диаграми, които представят информацията в структуриран и обобщен вид;

- динамична – отразява процесите и явленията в тяхното развитие и движение, чрез анимации, симулации, видео и интерактивни визуализации.

В съвременната образователна практика, в която информационните и комуникационни технологии предлагат широк спектър от възможности, нагледността се развива в нови посоки и измерения – чрез интерактивни симулации, динамични визуализации, виртуална и добавена реалност.

Подходи за визуализация на алгоритми в обучението по програмиране

Визуализацията на алгоритми може да бъде разгледана като процес на графично или анимирано представяне на отделните стъпки, през които преминава самото изпълнение. Този процес позволява: проследяване на изпълнението на даден алгоритъм; превръщане на абстрактен код в нагледна форма; по-достъпно разбиране на сложни алгоритмични структури.

Информационните и комуникационни технологии предоставят богат набор от дигитални инструменти и платформи, позволяващи:

- динамично представяне на алгоритми чрез анимации и симулации;
- представяне на алгоритми чрез използване на визуални и интерактивни среди за програмиране;
- използване на диаграми и графики при представяне на структури от данни;
- реализиране на блок-схеми, мисловни карти;
- реализиране на видео уроци и екранни записи за демонстрация на програмни процеси; мултимедийни презентации; използване на интерактивни уеб приложения.

Използване на динамични визуализации и симулации

Динамичните визуализации чрез симулации и анимации, проследяващи отделните стъпки и изменения на процесите в развитие, подпомагат обучението по програмиране при обяснение и анализ на алгоритми и обуславят тяхното по-ефективно разбиране и запомняне.

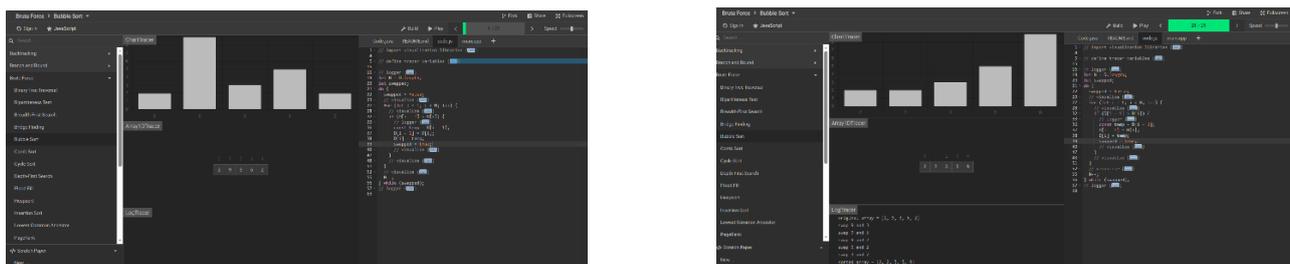
Категоризирането на средствата за динамична визуализация на алгоритми може да бъде разгледано в следните аспекти: инструменти за визуализация, представящи структурата и хода на изпълнение на програмен код

[5, 6]; инструменти за анимиране на алгоритми, при които се скрива изцяло кода [7, 9], или се показва само псевдокод [8].

Някои дигитални инструменти за динамични визуализации на алгоритми са: Algorithm Visualizer, VisuAlgo, Algomations и др.

1. Algorithm Visualizer (<https://algorithm-visualizer.org> [5])

Algorithm Visualizer е онлайн платформа за интерактивни визуализации на алгоритми в реално време, чрез представяне и постъпково проследяване на действията на даден алгоритъм. Предоставя възможности за визуализиране и проследяване на код, написан на различни програмни езици (Java, JavaScript, C++). Поддържа структури от данни, сортиране, търсене и графови алгоритми. Позволява писане на собствен код и наблюдение на изпълнението му стъпка по стъпка. Платформата предоставя ангажиращ начин за изследване, ефективна среда за анализ и по-добро възприемане и разбиране на действието на различни алгоритми (Фиг. 1).



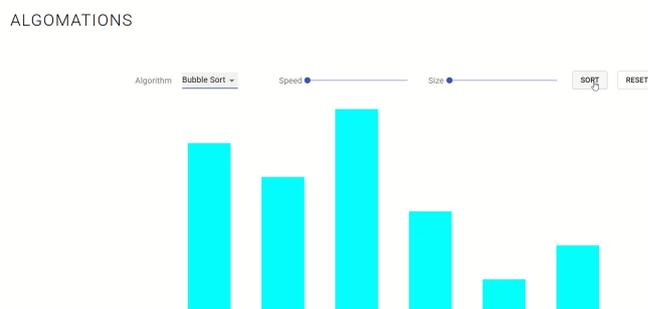
Фигура 1. Визуализиране и постъпково проследяване на алгоритъма за сортиране по метода на мехурчето (JavaScript)

2. VisuAlgo (<https://visualgo.net> [6]) – предоставя интерактивни визуализации на алгоритми и структури от данни чрез анимации и проследяване на свързан към тях програмен код. В платформата могат да се задават и собствени данни за всеки алгоритъм, вместо да се използват само автоматично генерирани примерни данни (Фиг. 2).



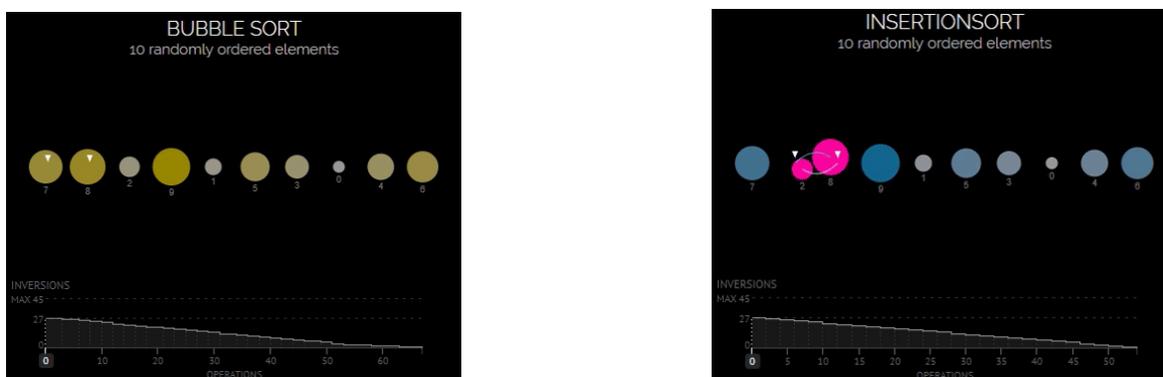
Фигура 2. Визуализация на алгоритъм за сортиране по метода на мехурчето чрез VisuAlgo

3. Algomations (<https://www.algomations.com>[7]) – web-базиран инструмент за представяне на анимирани визуализации на различни алгоритми за сортиране. Позволява на потребителите да настройват само брой елементи и скорост на анимация (Фиг. 3).



Фигура 3. Визуализация на алгоритъм за сортиране по метода на мехурчето чрез Algomations

4. Анимирани визуализации на алгоритми за сортиране (<https://tproger.ru/digest/sorting-algorithms-visualized> [4, 9])



Фигура 4. Gif анимации, илюстриращи различни алгоритми за сортиране

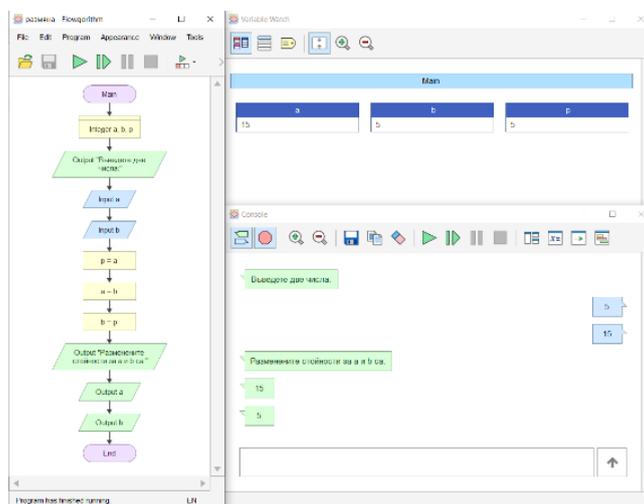
Използване на блок-схеми и създаване на мисловни карти

Блок-схемата е графично (визуално) представяне на даден алгоритъм чрез стандартизирани форми, които изобразяват отделните стъпки, съставляващи алгоритъма. Чрез тях се улеснява преобразуването на абстрактни идеи и концепции в ясни и разбираеми визуални модели. Дигиталните инструменти за създаване на блок-схеми могат да бъдат обособени в два аспекта:

- инструменти, позволяващи визуализация на алгоритъм чрез блок-схема, паралелно генериране на програмен код и изпълнението му

в самата среда. Например, софтуерът **Flowgorithm** (достъпен на <http://www.flowgorithm.org> [10]) позволява визуализация на алгоритми чрез създаване на блок-схеми, автоматично генериране на програмен код на различни езици и неговото директно изпълнение (Фиг. 5). Акцентът е насочен към усвояване на основни концепции в програмирането, а не върху разглеждането на специфични аспекти на даден език за програмиране;

- инструменти, предоставящи възможности за визуализация на алгоритъм чрез блок-схема преди неговата програмна реализация на избран език за програмиране; както и създаване на блок-схема на алгоритъм на база предоставен готов програмен код (Фиг. 6). Например MindOnMap, Smartdraw, Draw.io и др. са дигитални web-базирани инструменти за създаване на блок-схеми.

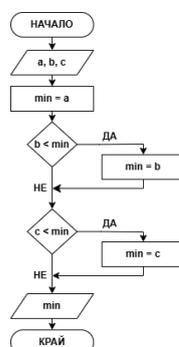


Фигура 5. Блок-схема на алгоритъм за размяна на стойности на две променливи чрез Flowgorithm

```

1 print("Въведи три числа:")
2 a=int(input())
3 b=int(input())
4 c=int(input())
5 min=a
6 if b<min:
7     min=b
8 if c<min:
9     min=c
10 print("Минимален елемент:",min)

```



Фигура 6. Блок-схема на алгоритъм за намиране на минимален елемент на три числа

Мисловните карти са визуален инструмент за организиране на информация, генериране на идеи и подпомагане на разбирането на сложни и абс-

трактни структури чрез използване на визуални и асоциативни връзки. Те насърчават креативното мислене у учениците, като превръщат абстрактните понятия в ясни визуални модели. В обучението по програмиране те могат да се използват както в процеса на преподаване, така и за активно учене от страна на обучаемите. Създаването на мисловни карти в обучението може да бъде свързано с планиране на проекти, обобщаване на учебно съдържание, визуално представяне на логическите стъпки в алгоритъм. Някои дигитални web-базирани инструменти за създаване на мисловни карти са: **MindOnMap**, **Coggle**, **MindMeister** и др.

Практически примери

Пример 1. Реализиране на алгоритъм за размяна на стойности на две променливи.

Основни методически акценти:

- Запознаване със същността и особеностите на алгоритъма за размяна на стойности на две променливи (с използване на помощна променлива);
- Визуализация на анимиран вариант на алгоритъма чрез използване на интерактивен проект на Scratch (Фиг. 7).



Фигура 7. Проект на Scratch, представящ постъпково изпълнение на алгоритъм за размяна на стойности на две променливи

Пример 2. Реализиране на алгоритъм за размяна на стойности на две променливи по два начина: с използване на помощна променлива; без използване на помощна променлива.

Основни методически акценти:

- Запознаване с различни подходи за решаване на един и същ проблем; Развиване на алгоритмично мислене и умение за сравнение на варианти;
- Визуализация на съответните алгоритми чрез блок-схеми; Използване на мисловни карти за обобщаване, сравнение и анализ на различните варианти (Фиг. 8).



Фигура 8. Блок-схеми и мисловна карта за зад. 1

Пример 3. Реализиране на алгоритми за сортиране.

Основни методически акценти:

- Запознаване с различни алгоритми за сортиране; сравнение и анализ;
- Използване на динамични визуализации, представящи постъпково изпълнение на различни алгоритми за сортиране чрез прилагане на интерактивни платформи: Algorithm Visualizer, VisuAlgo, Algomations и др. (виж Фиг. 1–4).

Пример 4. Изчертаване на съставна композиция от триъгълници.

Основни методически акценти:

- Разглеждане и анализ на два подхода за създаване на комплексна композиция от триъгълници според избрания образуващ елемент и реда на конструиране;
- Визуализация на изчертаването чрез видео демонстрации.

Таблица 1. Програмна реализация на зад. 4

Вариант 1: Програмен код на Python

```
import turtle
pen=turtle.Turtle()
pen.shape("triangle")
pen.pensize(3)
pen.speed(5)
pen.color("orange")
colors=["yellow","light green","pink","red"]
def draw_triangle(a):
    for i in range(3):
        pen.forward(a)
        pen.left(120)
br=4
pen.left(60)
for m in range(br):
    a=100 #страна на триъгълника
    b=a/(2*len(colors)) #стъпка
    while a>10:
        for c in colors:
            pen.fillcolor(c)
            pen.begin_fill()
            draw_triangle(a)
            pen.end_fill()
            a=a-b
        pen.left(360/br)
```

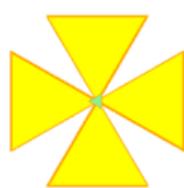
Вариант 2: Програмен код на Python

```
import turtle
pen=turtle.Turtle()
pen.shape("triangle")
pen.pensize(3)
pen.speed(5)
pen.color("orange")
colors=["yellow","light green","pink","red"]
def draw_triangle(a):
    for i in range(3):
        pen.forward(a)
        pen.left(120)
a=100 #страна на триъгълника
b=a/(2*len(colors)) #стъпка
br=4
pen.left(60)
while a>10:
    for c in colors:
        pen.fillcolor(c)
        for i in range(br):
            pen.begin_fill()
            draw_triangle(a)
            pen.end_fill()
            pen.left(360/br)
        a=a-b
```

Образуващ елемент и краен резултат



Образуващ елемент и краен резултат



Заклучение

Нагледността в обучението играе ключова роля, като превръща абстрактното знание в достъпни и разбираеми образи, улесняващи възприемането, запомнянето и прилагането на учебното съдържание. Комбинирането на различни видове нагледност с активни методи на учене води до по-дълбоко разбиране, по-висока мотивация и трайно усвояване на знанията в обучението по програмиране.

В статията предложихме примерна концепция за използване на различни дигитални инструменти и платформи за визуализация на алгоритми в обучението по програмиране. Бяха представени практически задачи, при които в процеса на усвояване на знанията се приложиха различни възможности за онагледяване на съответните алгоритми, както и тяхното изпълнение чрез използване на: блок-схеми и мисловни карти; интерактивни среди за програмиране; динамични визуализации, представящи постъпково изпълнение в реално време. Демонстрираният подход може успешно да бъде приложен в процеса на обучение по програмиране както в прогимназиален, така и в гимназиален етап.

Благодарности

Тази работа е финансирана от проект ФП25-ФМИ-010 „Иновативни интердисциплинарни изследвания по информатика, математика и педагогика на обучението“ към НПД на ПУ.

Литература

- [1] М. Андреев, *Процесът на обучението. Дидактика*, Университетско издателство „Св. Климент Охридски“, София, 1996, ISBN: 954-07-0861-3
- [2] Д. Дурева-Тупарова, *Проблеми от методиката на обучение по информатика и информационни технологии*, Университетско издателство ЮЗУ „Неофит Рилски“, Благоевград, 2003, ISBN: 954-680-254-9
- [3] И. Осмоловская, *Наглядные методы обучения*, Издательский центр „Академия“, Москва, 2009, ISBN: 978-5-7695-5799-6
- [4] С. Гроздев, Т. Терзиева, *Визуализация методов сортировки массивов*,

Электронный журнал Российской Академии Образования „Информационная среда образования и науки“, Выпуск 5, Москва, 2011

- [5] <https://algorithm-visualizer.org> (последно посетен на 25.11.2025 г.)
- [6] <https://visualgo.net> (последно посетен на 25.11.2025 г.)
- [7] <https://www.algomations.com> (последно посетен на 25.11.2025 г.)
- [8] <https://algorithm.enhar.net> (последно посетен на 25.11.2025 г.)
- [9] <https://tproger.ru/digest/sorting-algorithms-visualized> (последно посетен на 25.11.2025 г.)
- [10] <http://www.flowgorithm.org> (последно посетен на 25.11.2025 г.)

Елена Тодорова¹, Стефка Анева¹

¹ Пловдивски университет „Паисий Хилендарски“

Факултет по математика и информатика

бул. „България“ № 236, 4027 Пловдив, България

Автор за кореспонденция etodorova@uni-plovdiv.bg

USE OF DIGITAL TOOLS FOR ALGORITHM VISUALIZATION IN PROGRAMMING EDUCATION

Elena Todorova, Stefka Aneva

Abstract. *This article proposes pedagogical approaches for the use and integration of visualization as a means of visual support in programming education at school. Various digital tools and platforms enabling the dynamic representation of algorithms are presented. The impact of visual support as a means of enhancing motivation and effectiveness in programming education is emphasized.*

Key words: Digital Tools, Dynamic Visualization, Interactivity, Programming, Teaching Methodology.