

Алгоритмичното мислене като универсално умение

Стелиана Маринова

Algorithmic thinking as a universal skill

Steliana Marinova

Abstract:

At its core, the concept of algorithmic thinking is the ability of a person to break down tasks set before him into steps, to arrange these steps in the correct sequence and to check whether the achieved result matches expectations. In modern society, one of the most important skills that a student must possess is precisely this. The key component of algorithmic thinking is the ability to determine the steps for solving a problem clearly and consistently. It can be learned and, when well developed, builds personal, communicative qualities and skills.

Without claiming to be exhaustive, this article examines the evolution of the concept of algorithmic thinking and the need to teach students from the initial stage of education.

Keywords: algorithmic thinking; computational thinking; primary education; Scratch; visual programming; decomposition; abstraction; algorithmization; 21st century skills

For contacts: Steliana Marinova, University of Ruse, spmarinova@uni-ruse.bg

ВЪВЕДЕНИЕ

В условията на цифрова трансформация и нарастващо проникване на изкуствения интелект във всички сфери на човешкия живот уменията, свързани с алгоритмичното мислене, придобиват все по-голямо значение. Днес те се разглеждат не просто като компонент на компютърните науки, а като фундаментална интелектуална компетентност, сравнима по важност с четенето, писането и аритметиката. Тази идея, формулирана в различни варианти от учени в различни точки на света, преминава през няколко вълни на развитие – от пионерските работи на Сеймур Пейпърт и акад. Андрей Ершов през 1980-те до знаковата статия на Джанет Уинг от 2006 г. и съвременните образователни реформи в десетки държави.

ИЗЛОЖЕНИЕ

Една от ранните формулировки на идеята за алгоритмичното мислене като общочовешко умение принадлежи на акад. Андрей Петрович Ершов (1931–1988) – създател на Сибирската школа по информатика и пионер на теоретичното програмиране в СССР. На Третата световна конференция на IFIP и ЮНЕСКО в Лозана през 1981 г. той изнася доклада „Програмирането – втора грамотност“ [22], в който разглежда програмирането не като тясно-професионална дейност, а като нов тип грамотност, формиращ умения за планиране на действия, структуриране на информация и систематично решаване на проблеми. Четири години по-късно, през 1985 г., под негово ръководство в съветските училища се въвежда курсът „Основи на информатиката и изчислителната техника“ [23] и за пръв път в световната образователна практика алгоритмичното мислене се интегрира като задължителен компонент на общообразователната подготовка. Руските изследователи Кушниренко, Лебедев и Първин определят алгоритмичното мислене като „умение да се планира структурата на действията, необходими за

постигане на дадена цел, посредством фиксиран набор от средства“ [23]. И до днес руските учени (Бирка и Сущенко, 2024 [2]; Ковалчук, 2018 [7]) продължават да разширяват полето на алгоритмичното мислене отвъд компютърните науки, като го разглеждат като условие за формиране на общокултурни компетентности.

Четвърт век по-късно, през 2006 г., Джанет Уинг – ръководител на катедрата по компютърни науки в Университета Карнеги Мелън – публикува в статия [15], в която формулира изчислителното мислене като многокомпонентно, което включва както алгоритмичното мислене, така и декомпозицията, абстракцията, разпознаването на шаблони, дебъгването, моделирането, генерализацията. Тя определя това като много важно умение за всеки човек, а не само за компютърните специалисти, и поставя амбициозната цел то да стане толкова базово, колкото четенето, писането и смятането, до средата на XXI век. Уинг подчертава, че в основата на изчислителното мислене стои абстрактното – мисленето на няколко нива за решаване на проблеми, проектиране на системи и разбиране на човешкото поведение. В по-късните си работи (2008 [16], 2017 [17]) тя уточнява, че изчислителното мислене включва формулирането на проблеми по начин, допускащ изчислително решение, т.е. решение, което може да бъде изпълнено както от човек, така и от машина. Статията провокира глобална дискусия и катализира образователни реформи в десетки държави.

Както отбелязват Тедре и Денинг (2021) [13], самият термин изчислително мислене е въведен още през 1980 г. от Сеймур Пейпърт, макар и с различен акцент. Докато Уинг поставя фокус върху компютърните науки като централна дисциплина, Пейпърт акцентира върху идеята, че само социалното и емоционалното включване на учениците в техническото съдържание ще превърне програмирането в инструмент за учене, приложим в различни дисциплини.

В литературата се среща значително разнообразие от определения за алгоритмичното мислене, но повечето изследователи се обединяват около четири ключови когнитивни компонента, обобщени от Леман (2023) [9] и подкрепени от множество емпирични изследвания. Декомпозицията е умението за разбиване на сложен проблем на по-прости подзадачи. Ритър и Щандл (2023) [10] показват, че тя е „прецизиращо умение“, използвано в различни фази на алгоритмичния процес – от изграждането на абстрактни представяния до създаването на алгоритми. В руската научна традиция декомпозицията се разглежда като мисловна операция, лежаща в основата на структурния анализ на задачата. Абстракцията включва способността да се идентифицират съществените характеристики на проблема и да се игнорират несъществените детайли – Уинг [15] я нарича „ядрото на изчислителното мислене“, а китайските изследователи (Чжан Дечен, 2013 [20]; Харбински технологичен университет) също я поставят в центъра, разглеждайки изчислителното мислене като комбинация от програмно, системно и алгоритмично мислене. Алгоритмизацията е процесът на проектиране на решение стъпка по стъпка. Футшек (2006) [4] я дефинира като „пул от умения“ за разбиране и създаване на алгоритми, а Сари и колеги (2022) [11] разработват рамка с три показателя: разбиране на проблема, определяне на решение и създаване на алгоритъм за решението. Четвъртият компонент – дебъгването – изисква рефлексивно мислене и критичен анализ на собствените решения. Китайските образователни стандарти (2017) [26] формулират изчислителното мислене като

компетентност, обхващаща формализацията, моделирането, автоматизацията и систематизацията.

Великобритания е сред първите държави, интегрирали изчислителното мислене в националния учебен план. Разработва се оперативна дефиниция, включваща формулиране на проблеми, логическо организиране и анализ на данни, представяне чрез абстракции, автоматизиране на решения и идентифициране на трансферни възможности. Европейската комисия също промотира изчислителното мислене като ключова компетентност, а рамката DigComp 2.0 съдържа множество пресечни точки с него. В Швейцария се работи по универсални инструменти за оценяване на алгоритмичното мислене.

В постсъветското пространство съвременните изследвания надграждат наследството на Ершов. Бирка и колеги (2021 [1], 2024 [2]) анализират наблюдаеми и измерими компоненти на алгоритмичното мислене, формулирани като знания, умения и способности (KSAs). Ковалчук и колеги (2020) [8] го разглеждат като съдържателен компонент на когнитивните компетентности на бъдещия инженер с подчертана междудисциплинарна значимост, а в началното училище проблемът се решава чрез поетапно формиране на логически похвати с постепенен преход към алгоритмизацията.

Китайските образователни стандарти по информационни технологии (ревизирани 2017 г.) [26] включват изчислителното мислене като едно от четирите ядра на предметната компетентност. Джанет Уинг [15] е широко цитирана в китайската академична среда, а изследванията, публикувани на платформата ICTEDU, третираат изчислителното мислене като кръстовище на математика, наука, инженерство и технологии. Особено внимание се отделя на стратегиите за развитие в ерата на изкуствения интелект. Сингапур определя изчислителното мислене като „национална способност“, а Австралия, Корея, Нова Зеландия и Япония също активно го интегрират в учебните си програми.

В арабскоезичната научна литература алгоритмичното мислене се разглежда в контекста на по-широкото понятие „изчислително мислене“. Арабските изследователи подчертават историческата символика: самият термин „алгоритъм“ произлиза от името на Мохамед ибн Муса ал-Хорезми (780–850 г.), чийто труд „Китаб ал-джабр уал-мукабала“ полага основите на алгебрата. Платформата Hsoub Academy и изследователски центрове в Саудитска Арабия и ОАЕ активно развиват ресурси за формиране на алгоритмично мислене, третирано като ключово умение за XXI век.

Един от най-спорните въпроси в научната литература е доколко алгоритмичното мислене, развито в контекста на програмирането, може да се „пренесе“ към други области. Уинг (2017) [17] твърди, че изчислителното мислене вече трансформира биологията, статистиката и дори хуманитарните науки. Тедре и Денинг (2021) [13] обаче предупреждават, че твърдението за автоматичен трансфер е „често цитирано и рядко проверено“, а Вугт и колеги (2015) [14] отбелязват, че прекалено широкото тълкуване рискува да го размие и да го направи неразлично от други общи умения на XXI век. Въпреки тези резерви, практическите изследвания дават обнадеждаващи резултати: Грутър и Ядав (2016) [5] показват, че интегрирането на изчислително мислене с медийна и информационна грамотност е ефективен подход; Уонг и колеги (2024) [18]

доказват повишаване на поведенческата, когнитивната и емоционалната ангажираност при деца, работещи със Scratch; Гюлер (2021) [6] демонстрира, че курс по алгоритмично мислене без компютри помага на бъдещи учители да развият не само алгоритмични, но и по-широки академични компетентности. В ежедневието алгоритмичното мислене се проявява като умение за планиране, разбиване на сложни задачи, оценка на ефективността и предвиждане на резултати – по думите на руските изследователи, когато казваме, че даден човек „умее да мисли“, обикновено имаме предвид именно това.

Педагогическите подходи за формиране на алгоритмичното мислене варират от дейности, свързани с блоково и текстово програмиране, до дейности без компютър и хибридни комбинации. Блокното програмиране се утвърждава като ефективен инструмент при ученици от 8 до 12 години, а Юсуф и колеги (2024) [19] установяват, че комбинацията от текстово и блоково програмиране дава по-добри резултати от използването само на един от двата подхода. Дейностите без компютър също доказват своята стойност – международното състезание Бобър е ярък пример за оценяване на алгоритмичното мислене чрез задачи без програмиране. Обърнатата класна стая, проектно-базираното учене и колаборативното учене допълнително обогатяват методическата палитра.

Сред инструментите за формиране на алгоритмично мислене в начален етап средата за визуално програмиране Scratch, разработена от групата Lifelong Kindergarten при MIT Media Lab, заема централно място. Систематичните прегледи (Fagerlund et al., 2021 [3]; Zhang & Nouri, 2019 [21]; Stewart & Baek, 2023 [12]) обхващат стотици емпирични изследвания, потвърждаващи ефективността на Scratch за развитие на изчислителното мислене при деца от начален етап. Три различни научни традиции – руската, китайската и арабската – предлагат интересни и специфични перспективи върху употребата на Scratch при ученици на 9–10 години.

Руската научна традиция разглежда Scratch преди всичко като пропедевтичен инструмент – мост между елементарните алгоритмични познания и „истинското“ програмиране. Изследвания от КиберЛенинка [24] разглеждат организацията на проектна научно-познавателна извънучебна дейност в средата Scratch и обсъждат включването на курса не само като извънучебна дейност, но и в учебната програма по информатика в 3–4 клас [25]. Установено е, че за самостоятелен творчески проект (например мултфилм) учениците достигат след не по-малко от 20–25 учебни часа, като развиват логическо мислене, креативност и комуникативни качества. Особено ценно е, че Scratch е облекчена среда без синтактични грешки – готовите блокове позволяват свързване само в правилни конструкции, което е критично за деца, при които концентрацията все още не е силна страна, а интересът бързо угасва при среща с абстрактен текстов код. Опитът от Беларус (scratch.by) показва, че Scratch „обединява всички стилове на мислене“, като създаването на проекти винаги започва с разработка на сценарий – по същество упражнение в планиране и структуриране. Учители споделят, че са преминали от традиционни езици (КуМир, Pascal) към Scratch заради загубата на интерес при деца с художествен или хуманитарен профил, но предупреждават, че Scratch трябва да се използва целенасочено и методически обосновано, а не само като забавна игра.

Китайската перспектива се отличава с най-силната междупредметна интеграция. Scratch е най-широко прилаганият визуален програмен софтуер в началното училище, а учениците не трябва да запомнят сложен код – вместо това се концентрират върху анализа на проблема и превръщането му в блокови решения. Китайските автори предлагат тристъпков дидактически модел „изследване → имитация → реализация“: учениците първо самостоятелно проучват Scratch, после имитират под ръководството на учителя чрез блок-схеми и накрая реализират самостоятелна творба. Описва се как ученици рисуват правилни многоъгълници чрез Scratch и по пътя откриват, че с увеличаване на броя на страните фигурата се приближава до кръг – абстрактни математически знания се превръщат в нагледна визуализация. Програмирането на интерактивни театрални постановки (например „Заекът и костенурката“) изисква от учениците да проектират персонажи, диалози, последователност на сцените и причинно-следствени връзки, с което се развиват едновременно алгоритмичното и наративното мислене. Друг модел предлага пет стъпки: създаване на проблемна ситуация, описание на проблема, декомпозиция и абстрактно моделиране, писане на скрипт с оптимизация и дебъгване, оценка и изграждане на обобщен модел. Мисловните карти широко се използват за визуализиране на декомпозицията, а учебникът за провинция Хъбей структурира всеки урок с учебна цел, мисловна карта, задача за изпълнение и допълнително задание, обвързани с житейския опит на учениците.

Арабскоезичната литература за Scratch в началното образование все още е по-малобройна, но темата се развива активно в Саудитска Арабия, ОАЕ и Катар. Характерно е позоваването на историческата връзка между алгоритмите и арабската цивилизация, което придава културна легитимност на обучението. Образователните платформи предлагат устойчив четиристъпков път: моделиране, дефиниране на ограничения, избор на стратегия и проверка. Scratch се използва предимно чрез частни платформи и STEAM центрове, като националните учебни програми постепенно интегрират блоково програмиране. Обучението съчетава практически проекти с игрови елементи и е насочено към формиране на систематичен подход към решаването на проблеми.

И трите традиции разглеждат Scratch като ефективен инструмент за формиране на алгоритмично мислене при деца на 9–10 години, който премахва бариерата на синтаксиса и позволява фокусиране върху логиката. Руската традиция акцентира върху пропедевтичната функция и прехода към „истинско“

програмиране в наследството на алгоритмичната култура на Ершов. Китайската се отличава с междупредметната интеграция и систематичните дидактически модели с мисловни карти. Арабската привнася културно-историческа перспектива и STEAM подход. Общото е преходът от разбирането на Scratch като „игра“ към осъзнаването му като сериозен педагогически инструмент, чиято ефективност зависи от методическата рамка, в която е вграден – проектно-базирано учене, колаборативни задачи и обвързване с реални проблемни ситуации превръщат Scratch от среда за кодиране в среда за мислене.

ЗАКЛЮЧЕНИЕ

Международният преглед на научните изследвания разкрива забележително сближаване на идеи, родили се независимо в различни културни и академични контексти - алгоритмичното мислене е универсално когнитивно умение, необходимо за всеки гражданин на XXI век, независимо от професията му.

Образователните политики в световен мащаб вече интегрират алгоритмичното мислене в задължителното образование, като предизвикателствата се изместват от въпроса “дали” към въпросите “как” и “с какво качество”. За съвременния учител-методист основната задача е да намери подходящи пътища за формиране на алгоритмично мислене, адаптирани към възрастовите особености на учениците, с баланс между “включени” и “изключени” подходи и с акцент върху автентичното решаване на проблеми.

ЛИТЕРАТУРА

1. Byrka, M., Sushchenko, A., Svatiev, A., Mazin, V. & Veritov, O. (2021). A new dimension of learning in higher education: Algorithmic thinking. *Propósitos y Representaciones*, 9(SPE2), e990.
2. Byrka, M., Sushchenko, A., Luchko, V., Perun, G. & Luchko, V. (2024). Algorithmic thinking in higher education: Determining observable and measurable content. *Information Technologies and Learning Tools*, 104(6), 1–13.
3. Fagerlund, J., Häkkinen, P. & Vesisenaho, M. (2021). Computational thinking in programming with Scratch in primary schools: A systematic review. *Computer Applications in Engineering Education*, 29(1), 12–28.
4. Futschek, G. (2006). Algorithmic thinking. In *Lecture Notes in Computer Science: Vol. 4226*. Springer.
5. Gretter, S. & Yadav, A. (2016). Computational thinking and media & information literacy: An integrated approach to teaching twenty-first century skills. *TechTrends*, 60, 510–516.
6. Güler, Ç. (2021). Algorithmic thinking skills without computers for prospective computer science teachers. *Journal of Theoretical Educational Science*, 14(4), 570–585.
7. Kovalchuk, M. (2018). Semantic aspects of algorithmic thinking. *Physical and Mathematical Education*, 3(17), 61–66. [in Ukrainian]
8. Kovalchuk, M., Voievoda, A. & Prozor, E. (2020). Algorithmic thinking as the meaningful component of cognitive competencies of the future engineer. *Universal Journal of Educational Research*, 8(11B), 6248–6255.

9. Lehmann, B. (2023). A unified framework for algorithmic thinking. In Proceedings of the International Conference on Informatics in Schools.
10. Ritter, S. & Standl, B. (2023). A five-stage algorithmic thinking process. *Computers & Education*, 195.
11. Sari, U. et al. (2022). A framework to measure student algorithmic thinking skills. *Education and Information Technologies*, 27.
12. Stewart, W. & Baek, K. (2023). Analyzing computational thinking studies in Scratch programming: A review of elementary education literature. *International Journal of Computer Science Education in Schools*, 6(1), 35–58.
13. Tedre, M. & Denning, P.J. (2021). Computational thinking, between Papert and Wing. *Science & Education*, 30, 781–799.
14. Voogt, J., Fisser, P., Good, J., Mishra, P. & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies*, 20(4), 715–728.
15. Wing, J.M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35.
16. Wing, J.M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A*, 366, 3717–3725.
17. Wing, J.M. (2017). Computational thinking's influence on research and education for all. *Italian Journal of Educational Technology*, 25(2), 7–14.
18. Wong, G.K.W. et al. (2024). Development of children's algorithmic thinking skills using Scratch. *Computers & Education*, 210.
19. Yusuf, B.N. et al. (2024). Modeling students' algorithmic thinking growth trajectories in different programming environments. *Smart Learning Environments*, 11.
20. Zhan, D. (2013). 大学计算机——计算思维导论 [University Computing: Introduction to Computational Thinking]. Harbin Institute of Technology Press. [изд. на китайски]
21. Zhang, L. & Nouri, J. (2019). A systematic review of learning computational thinking through Scratch in K-9. *Computers & Education*, 141, 103607.
22. Ершов, А.П. (1981). Программирование – вторая грамотность. Доклад на 3-тата световна конференция на IFIP и ЮНЕСКО, Лозана, Швейцария.
23. Ершов, А.П., Кушниренко, А.Г., Лебедев, Г.В. и др. (1985). Основы информатики и вычислительной техники. Москва: Просвещение.
24. Дженжер, В.О. и Денисова, Л.В. (2013). Среда программирования Scratch как способ организации проектной деятельности младшего школьника. КиберЛенинка.
25. Перевалова, Н.А. и Рожина, И.В. (2020). Пропедевтика программирования с использованием среды Scratch. КиберЛенинка.
26. 教育部. (2017). 普通高中信息技术课程标准 [Курикулумни стандарти по информационни технологии за средното общообразователно училище]. Пекин.