

ПОНЯТТЯ АЛГОРИТМУ В МАТЕМАТИЦІ ТА ПРОГРАМУВАННІ

Вінницький національний технічний університет

Анотація:

У доповіді розглянуто поняття алгоритму як фундаментальної категорії математики та програмування. Проаналізовано історичні передумови формування алгоритмічного підходу, основні властивості алгоритмів та їх роль у формалізації процесу розв'язання задач. Особливу увагу приділено значенню алгоритмів у сучасних інформаційних технологіях і підготовці майбутніх фахівців.

Ключові слова: алгоритм, алгоритмізація, програмування, обчислювальний процес, алгоритмічне мислення.

Abstract:

The paper considers the concept of an algorithm as a fundamental category in mathematics and programming. The historical background of algorithmic approaches, main properties of algorithms, and their role in formal problem solving are analyzed. The importance of algorithms in modern information technologies and professional training is emphasized.

Keywords: algorithm, algorithmization, programming, computational process.

Поняття алгоритму є одним із ключових у математиці, інформатиці та програмуванні. Алгоритм визначається як скінченна, впорядкована та однозначно визначена послідовність дій, виконання яких забезпечує розв'язання певної задачі. Формування алгоритмічного підходу стало основою розвитку обчислювальної техніки та сучасних інформаційних систем. Алгоритми дозволяють формалізувати процес розв'язання задач і забезпечують можливість його автоматизації. Завдяки алгоритмам складні процеси можуть бути подані у вигляді простих і чітких кроків, що робить їх придатними для виконання як людиною, так і комп'ютером. Універсальність алгоритмічного підходу полягає в можливості застосування одного й того самого алгоритму до широкого класу однотипних задач. Крім того, алгоритми є основою побудови програмних моделей та інформаційних систем різного рівня складності. Саме тому вивчення алгоритмів є необхідною складовою підготовки фахівців у галузі інформаційних технологій [4].

Історично поняття алгоритму виникло задовго до появи комп'ютерів. Алгоритмічні процедури застосовувалися в арифметиці, геометрії та логіці. У математиці алгоритм розглядається як формальний метод обчислень або доведення, який не залежить від конкретного виконавця. Саме математичні алгоритми заклали теоретичні основи подальшого розвитку обчислювальних методів і формалізованих процедур розв'язання задач [5].

Важливе місце в математиці посідають чисельні алгоритми, які застосовуються для розв'язання задач, у яких аналітичні методи є складними або неможливими для практичного використання. Типовим прикладом є задачі обчислення інтегральних функціоналів для яких використовується апарат чисельного інтегрування. Загальний підхід до таких задач полягає у заміні неперервного функціонала його дискретною апроксимацією на скінченній множині вузлів. Такий підхід дозволяє звести неперервну математичну задачу до скінченної послідовності обчислювальних операцій, придатних для реалізації на комп'ютері.

$$I(f) = \int_a^b f(x) dx$$

де $I(f)$ — інтегральний функціонал, який позначає точне значення визначеного інтеграла від підінтегральної функції $f(x)$ на відрізку $[a, b]$ — і відповідає аналітичному (неперервному) розв'язанню задачі.

У теорії чисельних методів чисельне інтегрування описується за допомогою квадратурних формул загального вигляду:

$$I(f) \approx Q_n(f) = \sum_{k=1}^n w_k f(x_k)$$

де $x_k \in [a, b]$ — вузли квадратури, а w_k — відповідні вагові коефіцієнти. Величина $Q_n(f)$ є квадратурною формулою, тобто чисельною апроксимацією інтегрального функціонала $I(f)$, яка використовується для практичного обчислення інтегралів. Вибір вузлів та ваг визначає порядок точності алгоритму, його збіжність і обчислювальну ефективність.

Важливою характеристикою чисельного алгоритму є залишковий член, який дозволяє оцінити похибку апроксимації залежно від гладкості підінтегральної функції.

$$R_n(f) = I(f) - Q_n(f)$$

Залишковий член $R_n(f)$ характеризує похибку чисельного інтегрування та показує різницю між точним значенням інтегрального функціонала $I(f)$ і його чисельною апроксимацією $Q_n(f)$.

Для функцій, що належать до класів $C^m[a, b]$, похибка квадратурних алгоритмів має асимптотичну оцінку:

$$R_n(f) = O(h^m)$$

Асимптотична оцінка означає, що величина залишкового члена $R_n(f)$ за модулем не перевищує деяку сталу, помножену на h^m , при достатньо малих значеннях кроку розбиття. Тут $R_n(f)$ — похибка квадратурної формули, h — максимальна відстань між сусідніми вузлами квадратури, а показник m характеризує порядок точності чисельного алгоритму та пов'язаний із класом гладкості підінтегральної функції $f(x)$. Зі зменшенням кроку h похибка чисельного інтегрування зменшується, що свідчить про збіжність квадратурного алгоритму до точного значення інтегрального функціонала $I(f)$. Такий підхід дає змогу досліджувати точність і збіжність алгоритмів при згущенні обчислювальної сітки.

Наведені чисельні алгоритми є прикладом формалізованих математичних процедур, які безпосередньо реалізуються засобами програмування. Вони демонструють перехід від неперервної математичної моделі до дискретного алгоритму, придатного для автоматизованих обчислень у прикладних задачах математичного моделювання, інженерних розрахунках та аналізі даних. [1, 2].

У програмуванні алгоритм є концептуальною основою створення програмного забезпечення. Кожна програма реалізує певний алгоритм, поданий мовою програмування з урахуванням обмежень апаратних ресурсів, швидкодії та коректності обробки даних. На відміну від математичного опису, програмний алгоритм повинен бути адаптований до конкретної обчислювальної системи та враховувати практичні аспекти виконання [4]. Окрім цього, програмні алгоритми мають бути зрозумілими для подальшого супроводу та модифікації, що зумовлює вимоги до їх структурованості та читабельності. Важливим етапом реалізації алгоритму є його налагодження та тестування, які дозволяють виявити логічні та технічні помилки. У процесі програмування алгоритм часто зазнає оптимізації з метою підвищення продуктивності програмного продукту. Усе це визначає тісний зв'язок між алгоритмом як теоретичною моделлю та програмою як практичним результатом його реалізації.

Важливим прикладом застосування алгоритмів є розв'язання систем лінійних рівнянь. Зокрема, метод Гауса використовується для знаходження розв'язків систем лінійних алгебраїчних рівнянь і широко застосовується в задачах обробки даних, комп'ютерної графіки та машинного навчання [5].

До основних властивостей алгоритму належать дискретність, визначеність, скінченність, результативність та масовість. Наявність цих властивостей забезпечує можливість реалізації алгоритму у вигляді програм та його застосування до широкого класу задач.

Порівнюючи алгоритми в математиці та програмуванні, слід зазначити, що в математиці алгоритм орієнтований насамперед на логічну строгість і теоретичну коректність результату. У програмуванні ж алгоритм додатково оцінюється за ефективністю використання ресурсів, швидкістю виконання та стійкістю до помилок вхідних даних [2, 3]. Таким чином, програмні алгоритми є практичною реалізацією математичних алгоритмів з урахуванням реальних умов обчислення. Крім того, математичні алгоритми зазвичай описуються у вигляді формул, правил або словесних інструкцій, тоді як програмні алгоритми

потребують точного синтаксичного оформлення мовою програмування. Важливою відмінністю є також необхідність тестування програмних алгоритмів, оскільки навіть теоретично коректний алгоритм може містити реалізаційні помилки. У програмуванні значну увагу приділяють оптимізації алгоритмів з метою зменшення часових і просторових витрат. Водночас математичні алгоритми слугують основою для доведення правильності програмних рішень та забезпечення їх надійності.

Алгоритмізація задач передбачає аналіз умови, виділення вхідних і вихідних даних, визначення послідовності обчислень та перевірку правильності отриманого результату. Цей процес є важливою складовою інженерного та програмного мислення. У математичних задачах цей процес зазвичай завершується отриманням аналітичного або чисельного розв'язку, тоді як у програмуванні результатом є програмний код, здатний автоматично виконувати обчислення для множини вхідних даних.

Особливе значення алгоритмічне мислення має у підготовці фахівців з інформаційних технологій. Воно формує здатність до логічного аналізу, абстрагування та системного підходу до розв'язання складних практичних задач. Крім того, алгоритмічне мислення сприяє розвитку навичок декомпозиції задач, що дозволяє розбивати складні проблеми на простіші та більш керовані етапи. Воно є основою для ефективного програмування, проектування програмних систем і оптимізації процесів обробки даних. Сформоване алгоритмічне мислення допомагає майбутнім фахівцям з ІТ швидко адаптуватися до нових технологій і мов програмування, а також приймати обґрунтовані рішення в умовах невизначеності. У результаті це підвищує професійну компетентність та конкурентоспроможність спеціалістів на сучасному ринку праці.

Поєднання математичної строгості та програмної реалізації сприяє формуванню універсальних професійних компетентностей майбутніх фахівців. Крім того, сучасні алгоритми широко застосовуються в галузях обробки великих даних, штучного інтелекту та інформаційної безпеки, що підкреслює їх міждисциплінарний характер і практичну значущість. Використання алгоритмічного підходу дозволяє ефективно розв'язувати складні задачі, пов'язані з аналізом, прогнозуванням та оптимізацією процесів. У сфері штучного інтелекту алгоритми забезпечують навчання моделей і прийняття рішень на основі великих обсягів даних. В інформаційній безпеці алгоритми відіграють ключову роль у шифруванні, захисті та перевірці цілісності інформації. Алгоритмічні методи також активно застосовуються в інженерії, економіці та природничих науках. Таким чином, знання та вміння застосовувати алгоритми є необхідною умовою професійної діяльності сучасного ІТ-фахівця [1-3].

Таким чином, алгоритм виступає універсальним інструментом, що поєднує математичну теорію та практику програмування. Гармонійна взаємодія математичних і програмних алгоритмів забезпечує створення ефективних, надійних та масштабованих інформаційних систем, що є ключовим завданням сучасної інформатики.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Л. О. Волонтир, Л. В. Зелінська, Н. А. Потапова, І. А. Чіков. Чисельні методи. Навчальний посібник. URL: <https://socrates.vsau.org/repository/getfile.php/27703.pdf> (дата звернення: 30.12.2025).
2. Ляшенко О. І. Чисельні методи розв'язання прикладних задач. Навчальний посібник. URL: https://pdf.lib.vntu.edu.ua/books/2015/Lyashenko_1996_288.pdf (дата звернення: 30.12.2025).
3. А. Л. Литвинов, М. Ю. Воеводіна. Практикум з чисельних методів. Навчальний посібник. URL: <https://files.znu.edu.ua/files/Bibliobooks/Inshi80/0060477.pdf> (дата звернення: 30.12.2025).
4. Р. Н. Кветний, Я. В. Іванчук, І. В. Богач, О. Ю. Софіна, М. В. Барабан. Методи та алгоритми комп'ютерних обчислень: теорія і практика. URL: https://pdf.lib.vntu.edu.ua/books/2023/Kvetniy_2023_280.pdf (дата звернення: 30.12.2025).
5. О. О. Коваленко, О. М. Ткаченко, Р. Ю. Чехмиструк. Алгоритми та структури даних. URL: https://pdf.lib.vntu.edu.ua/books/2025/Kovalenko_2025_113.pdf (дата звернення: 30.12.2025)

Кирилячук Світлана Анатоліївна – доцент кафедри вищої математики, Вінницький національний технічний університет, м. Вінниця, e-mail: kyrylashchuk@vntu.edu.ua.

Ільчук Маріна Русланівна – студент групи 2ПІ-246, Вінницький національний технічний університет, м. Вінниця, e-mail: marinailchuk05@gmail.com

Kyrylashchuk Svitlana A. – Associate Professor of the Department of Higher Mathematics, Vinnytsia National Technical University, Vinnytsia, e-mail: kyrylashchuk@vntu.edu.ua.

Ilchuk Marina R. – student of 2BC-24b group, Vinnytsia National Technical University, Vinnytsya, e-mail: marinailchuk05@gmail.com