

УДК 004.43:004.9

DOI [https://doi.org/10.15589/znп2026.1\(503\).2.2](https://doi.org/10.15589/znп2026.1(503).2.2)**ANALYSIS OF THE DYNAMICS OF THE POPULARITY
OF PROGRAMMING LANGUAGES ACCORDING TO OPEN INDEX DATA****АНАЛІЗ ДИНАМІКИ ПОПУЛЯРНOSTІ МОВ ПРОГРАМУВАННЯ
ЗА ДАНИМИ ВІДКРИТИХ ІНДЕКСІВ****Yurii V. Kuchma**

krabatua@gmail.com

ORCID: 0009-0002-5498-4271

Alina S. Yanko

al9_yanko@ukr.net

ORCID: 0000-0003-2876-9316

Eduard Yu. Rubin

edw.rubin@gmail.com

ORCID: 0009-0005-4447-4413

Viacheslav V. Polinovskyi

v.v.polin@gmail.com

ORCID: 0009-0008-1271-5528

Ю. В. Кучма,

канд. техн. наук, доцент

А. С. Янко,

канд. техн. наук, доцент

Е. Ю. Рубін,

канд. техн. наук, доцент

В. В. Поліновський,

канд. техн. наук, доцент

Admiral Makarov National University of Shipbuilding, Mykolaiv*Національний університет кораблебудування імені адмірала Макарова, м. Миколаїв*

Abstract. The article conducts a comprehensive study of the dynamics of the popularity of programming languages in the period 2005–2025 based on open analytical sources, including the TIOBE, GitHub Octoverse and RedMonk indices. The relevance of the research is due not only to the rapid development of information technologies, but also to the critical dependence of modern digital systems on the effective choice of development tools. It is important to note that errors in the selection of the technological stack lead to increased costs, reduced performance and risks of scaling systems. At the same time, in the conditions of the digital transformation of the economy, programming languages are becoming a strategic resource, and their popularity is an indicator of the direction of development of the industry.

In addition, relevance is strengthened by the dynamics of the labor market. The demand for IT professionals is directly correlated with the prevalence of specific languages. This affects educational programs, training and investment decisions of companies. Therefore, the lack of systematic analysis of trends makes it difficult to make informed decisions both at the business level and at the level of state policy in the field of digitalization. Thus, the study of the popularity of programming languages has applied significance and forms the basis for strategic planning.

The presented work summarizes the main approaches to assessing the popularity of programming languages. The TIOBE index was found to be based on search query analysis and to reflect the level of user interest. At the same time, GitHub Octoverse evaluates the real use of languages through activity in repositories.

At the same time, the RedMonk rating integrates both approaches and provides a more balanced rating. The analysis showed that each of the approaches has limitations, so none of them gives a complete picture separately.

This determines the need for comprehensive use of various data sources. That is why the combination of indices allows you to reduce the impact of errors and increase the reliability of the results. This approach provides a more accurate reflection of real technological trends and creates a basis for further forecasting of the development of programming languages.

Purpose. The purpose of the study is a comparative analysis of the dynamics of the popularity of programming languages in 2005–2025 based on the integration of TIOBE, RedMonk and GitHub Octoverse data in order to identify key trends and changes in technological leadership. The study also aims to improve the validity of the evaluation by combining search and repository data.

Method. The methodological basis of the study is the system collection, normalization and comparative analysis of time series of the popularity of programming languages, formed according to the TIOBE, RedMonk and GitHub Octoverse indexes. A multifactorial approach integrating indicators of interest (search data) and actual use (repository activity) is applied. Unification of scales and time synchronization of data were performed to ensure comparability. A

correlation analysis between indices was carried out in order to assess the consistency of the results. On the basis of the processed data, long-term trends were identified and key points of structural changes in the dynamics of the popularity of programming languages were determined.

Results. The obtained results confirm a significant transformation of the popularity structure of Python-dominated programming languages, which is due to the versatility and developed ecosystem for data processing and AI. A decrease in Java's share while maintaining its role in corporate systems and stable leadership of JavaScript in web-oriented environments were recorded. An increase in the importance of the Go and Rust languages was revealed, which is due to the requirements for performance, parallelism and memory security. At the same time, the conducted comparative analysis of the indices confirmed their moderate correlation and the expediency of integrated use to increase the objectivity of the assessment.

Scientific novelty. The key factors affecting the dynamics of the popularity of programming languages are identified, in particular the development of ecosystems, the level of community support, compliance with modern technological trends and demand on the labor market. The relationship between the intensity of development of the instrumental environment and the rate of growth of the popularity of languages has been established. Languages with a high level of integration and an active community have been shown to show consistently higher rates of distribution. The obtained results expand approaches to the assessment of technological leadership in software engineering.

Practical importance. The obtained results can be used to substantiate the content of educational programs, plan the professional development of IT specialists and select technological stacks in software projects. Python orientation as a basic language, JavaScript as a key web development technology, and the inclusion of Go and Rust are recommended, taking into account modern performance and security requirements. The expediency of forming multidisciplinary competencies of developers is emphasized. The proposed approaches contribute to increasing adaptability to dynamic changes in the technological environment.

Key words: programming languages; IT education; development trends; Python; JavaScript; Java; Go; Rust; TIOBE; GitHub Octoverse; RedMonk.

Анотація. У статті проведено комплексне дослідження динаміки популярності мов програмування у період 2005–2025 років на основі відкритих аналітичних джерел, зокрема індексів TIOBE, GitHub Octoverse та RedMonk. Актуальність дослідження зумовлена не лише швидким розвитком інформаційних технологій, але й критичною залежністю сучасних цифрових систем від ефективного вибору інструментів розробки. Важливо зазначити, що помилки у виборі технологічного стеку призводять до зростання витрат, зниження продуктивності та ризиків масштабування систем. При цьому в умовах цифрової трансформації економіки мови програмування стають стратегічним ресурсом, а їх популярність – індикатором напрямів розвитку галузі.

Додатково актуальність підсилюється динамікою ринку праці. Попит на IT-фахівців безпосередньо корелює з поширеністю конкретних мов. Це впливає на освітні програми, підготовку кадрів та інвестиційні рішення компаній. Тому відсутність системного аналізу трендів ускладнює прийняття обґрунтованих рішень як на рівні бізнесу, так і на рівні державної політики у сфері цифровізації. Таким чином, дослідження популярності мов програмування має прикладне значення та формує основу для стратегічного планування.

В представленій роботі узагальнено основні підходи до оцінювання популярності мов програмування. Встановлено, що індекс TIOBE базується на аналізі пошукових запитів і відображає рівень інтересу користувачів. Водночас GitHub Octoverse оцінює реальне використання мов через активність у репозиторіях. При цьому рейтинг RedMonk інтегрує обидва підходи і забезпечує більш збалансовану оцінку. Проведений аналіз показав, що кожен із підходів має обмеження, тож, жоден із них не дає повної картини окремо.

Зазначене обумовлює необхідність комплексного використання різних джерел даних. Саме тому, поєднання індексів дозволяє знизити вплив похибок та підвищити достовірність результатів. Такий підхід забезпечує більш точне відображення реальних технологічних трендів і створює основу для подальшого прогнозування розвитку мов програмування.

Мета. Метою дослідження є порівняльний аналіз динаміки популярності мов програмування у 2005–2025 рр. на основі інтеграції даних TIOBE, RedMonk і GitHub Octoverse з метою виявлення ключових трендів та змін технологічного лідерства. Дослідження також спрямоване на підвищення достовірності оцінювання шляхом поєднання пошукових і репозиторних даних.

Методика. Методологічну основу дослідження становить системний збір, нормалізація та порівняльний аналіз часових рядів популярності мов програмування, сформованих за даними індексів TIOBE, RedMonk і GitHub Octoverse. Застосовано багатофакторний підхід, що інтегрує показники інтересу (пошукові дані) та фактичного використання (репозиторна активність). Для забезпечення зіставності виконано уніфікацію шкал і часову синхронізацію даних. Проведено кореляційний аналіз між індексами з метою оцінювання узгодженості результатів.

На основі оброблених даних ідентифіковано довгострокові тренди та визначено ключові точки структурних змін у динаміці популярності мов програмування.

Результати. Отримані результати підтверджують суттєву трансформацію структури популярності мов програмування з домінуванням Python, що зумовлено універсальністю та розвинутою екосистемою для обробки даних і ШІ. Зафіксовано зниження частки Java при збереженні її ролі в корпоративних системах та стабільне лідерство JavaScript у веб-орієнтованих середовищах. Виявлено зростання значущості мов Go і Rust, що обумовлено вимогами до продуктивності, паралелізму та безпеки пам'яті. При цьому проведений порівняльний аналіз індексів підтвердив їх помірну кореляцію та доцільність інтегрованого використання для підвищення об'єктивності оцінювання.

Наукова новизна. Визначено ключові фактори, що впливають на динаміку популярності мов програмування, зокрема розвиток екосистем, рівень підтримки спільноти, відповідність сучасним технологічним трендам і попит на ринку праці. Встановлено залежність між інтенсивністю розвитку інструментального середовища та темпами зростання популярності мов. Доведено, що мови з високим рівнем інтеграції та активною спільнотою демонструють стабільно вищі темпи поширення. Отримані результати розширюють підходи до оцінювання технологічного лідерства у програмній інженерії.

Практична значимість. Отримані результати можуть бути використані для обґрунтування змісту освітніх програм, планування професійного розвитку IT-фахівців і вибору технологічних стеків у програмних проєктах. Рекомендовано орієнтацію на Python як базову мову, JavaScript як ключову технологію веб-розробки, а також включення Go і Rust з урахуванням сучасних вимог до продуктивності та безпеки. Підкреслено доцільність формування мультидисциплінарних компетенцій розробників. Запропоновані підходи сприяють підвищенню адаптивності до динамічних змін технологічного середовища.

Ключові слова: мови програмування; аналіз популярності; Python; JavaScript; GitHub; TIOBE; RedMonk; інформаційні технології; тренди розвитку.

ПОСТАНОВКА ЗАДАЧІ

Стрімка еволюція IT зумовлює необхідність об'єктивного оцінювання динаміки популярності мов програмування як індикатора змін у програмній інженерії. Існуючі підходи до ранжування мов базуються на різноманітних джерелах даних і характеризуються методологічними обмеженнями, що ускладнює формування узгодженої оцінки технологічного лідерства [1]. Відсутність інтегрованого підходу призводить до фрагментарності результатів і зниження їх прикладної цінності.

У зв'язку з цим постає задача формування узгодженої методики аналізу популярності мов програмування на основі поєднання пошукових, репозиторних і аналітичних індексів. Вагомим завданням постають питання забезпечення нормалізації та синхронізації даних, виконання порівняльного аналізу часових рядів і визначення ступіню узгодженості між різними джерелами. Іншим важливим аспектом є ідентифікація стійких трендів та ключових факторів, що впливають на зміну позицій мов у довгостроковій перспективі.

Саме розв'язання поставленого завдання дозволить підвищити достовірність оцінювання популярності мов програмування та сформулювати аналітичну основу для прийняття обґрунтованих рішень у сфері освіти, професійного розвитку та вибору технологічних стеків.

АНАЛІЗ ОСТАННІХ ДОСЛІДЖЕНЬ І ПУБЛІКАЦІЙ

Мови програмування еволюціонують у тісному зв'язку з розвитком апаратного забезпечення, методологій розробки та бізнес-потреб IT-галузі. У першій

декаді XXI століття технологічний ландшафт залишався відносно стабільним, Java, C і C++ формували основу корпоративних і системних рішень, тоді як PHP домінував у сегменті динамічних веб-застосунків [2]. Проте подальший розвиток хмарних технологій, зростання обсягів даних і поширення методів машинного навчання суттєво змінили вимоги до мов програмування. На перший план вийшли характеристики, пов'язані з простотою використання, швидкістю розробки та наявністю розвинутої екосистеми бібліотек.

У цьому контексті значної популярності набула мова Python, яка поєднує простоту синтаксису з потужним інструментарієм для аналізу даних, автоматизації та серверної розробки. Паралельно загострилися вимоги до продуктивності та безпеки програмних систем, що стимулювало появу нових системних мов, зокрема Go та Rust. Вони орієнтовані на ефективну роботу в умовах високих навантажень і забезпечують покращене управління ресурсами та паралелізмом.

Аналіз наукових публікацій [3, 4] свідчить, що дослідження популярності мов програмування здебільшого базуються на окремих джерелах даних, таких як пошукові запити або опитування розробників. Такий підхід обмежує можливість отримання об'єктивних висновків, оскільки не враховує довгострокову динаміку та реальну практику використання мов. Для подолання цих обмежень доцільним є використання довготермінових часових рядів і порівняльний аналіз різних аналітичних індексів [5].

Найбільш репрезентативним джерелом довгострокових даних є індекс TIOBE, який акумулює

інформацію про популярність мов на основі пошукових запитів, починаючи з 2001 року. Водночас для підвищення достовірності результатів у сучасних дослідженнях активно використовуються репозиторні метрики GitHub Octoverse, що відображають реальну активність розробників, а також рейтинг RedMonk, який інтегрує дані GitHub і Stack Overflow [6]. Поєднання зазначених підходів дозволяє здійснити багатовимірний аналіз і оцінити узгодженість між інтересом до мов та їх практичним використанням.

Таким чином, сучасні аналітичні дослідження демонструють необхідність переходу від фрагментарних оцінок до комплексного аналізу популярності мов програмування на основі інтеграції різномірних джерел даних, що і визначає напрям даної роботи.

ВІДОКРЕМЛЕННЯ НЕВИРІШЕНИХ РАНІШЕ ЧАСТИН ЗАГАЛЬНОЇ ПРОБЛЕМИ

Незважаючи на наявність значної кількості досліджень, проблема об'єктивного оцінювання динаміки популярності мов програмування залишається недостатньо вирішеною. Існуючі підходи переважно базуються на окремих джерелах даних, що призводить до фрагментарності результатів і обмежує можливість їх узагальнення. Зокрема, пошукові індекси відображають рівень інтересу, але не враховують реальне використання мов, тоді як репозиторні метрики не завжди коректно відображають освітні та дослідницькі тренди [7].

Додатковою проблемою є відсутність уніфікованих методів нормалізації та синхронізації даних із різних джерел, що ускладнює їх порівняння у довгостроковій перспективі. Недостатньо дослідженими залишаються питання виявлення стійких трендів та точок перелому у зміні технологічного лідерства, а також кількісного оцінювання впливу зовнішніх факторів, таких як розвиток хмарних технологій, штучного інтелекту та ринку праці [8].

Таким чином, невирішеною частиною проблеми є розробка інтегрованого підходу до аналізу популярності мов програмування, який забезпечує поєднання різномірних джерел даних, підвищення достовірності результатів і можливість формування обґрунтованих висновків щодо розвитку технологічних трендів.

МЕТА ДОСЛІДЖЕННЯ

Метою дослідження є побудова об'єктивної моделі оцінювання популярності мов програмування шляхом інтегрованого аналізу даних TIOBE, RedMonk та GitHub Octoverse. Робота спрямована на виявлення стійких технологічних трендів для формування аналітичної бази при прийнятті рішень у сфері IT-освіти та вибору стеку розробки.

МЕТОДИ, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

Об'єктом дослідження є процеси формування та зміни популярності мов програмування в умовах

розвитку інформаційних технологій. Предметом дослідження виступають закономірності динаміки популярності мов програмування, а також фактори, що впливають на їх технологічне позиціонування.

У дослідженні використано методи збору та нормалізації даних, порівняльного аналізу часових рядів, кореляційного аналізу та узагальнення результатів. Інформаційну основу становлять дані індексів TIOBE, RedMonk і GitHub Octoverse. Застосовано інтегрований підхід до обробки різномірних джерел, що забезпечує підвищення достовірності та об'єктивності отриманих результатів.

ОСНОВНИЙ МАТЕРІАЛ

Еволюція мов програмування (Рис. 1) відображає постійне зростання рівня абстракції та ускладнення програмних систем, що зумовлює трансформацію підходів до розробки та виконання програмного коду. Перший етап цієї еволюції характеризується переходом від машинних кодів і асемблера до мов високого рівня, які забезпечують декларативне програмування та автоматизовану компіляцію. Так, Fortran запровадив декларативний підхід і компіляцію, що значно підвищило продуктивність наукових обчислень [9]. ALGOL 60 формалізував синтаксис та структуру програм, задавши стандарти для майбутніх мов [10]. C забезпечила баланс між продуктивністю та переносимістю, заклавши основу для розвитку системного програмування та створення операційних систем і низькорівневих бібліотек [11].

Згодом у 1990-х роках сформувалася концепція платформної незалежності, що радикально змінила підходи до виконання програм. Так Java реалізувала виконання програм у віртуальному середовищі, забезпечуючи переносимість та безпеку коду. Паралельно, JavaScript започаткував інтерактивний веб, а подальший розвиток його екосистеми, включно з Node.js, React, Vue та Angular, перетворив мову на універсальний інструмент розробки для клієнтських та серверних застосунків [12].

У 2000-х роках відбулося зміщення фокусу розвитку програмування у бік обробки даних і наукових обчислень, у межах якого Python здобув широку популярність завдяки потужним науковим бібліотекам (NumPy, SciPy, Pandas) та фреймворкам машинного навчання (TensorFlow, PyTorch), що забезпечило його домінування у сферах Data Science та ШІ [13]. Водночас зростання вимог до продуктивності та безпеки програмних систем зумовило появу нових підходів. Зокрема, Go запропонував ефективну конкурентність на основі легковагової моделі паралельного виконання з низькими накладними витратами середовища виконання, тоді як Rust забезпечив суворий контроль пам'яті на етапі компіляції, мінімізуючи ризики типових помилок керування ресурсами.

Подальший розвиток IT-інфраструктури, зокрема впровадження технологій контейнеризації та хмарних

сервісів (Docker, Kubernetes, serverless-архітектури), зумовив зміну технологічних пріоритетів і підвищив роль Go у розробці високонавантажених мікросервісних систем. Водночас стандартизація ECMAScript стимулювала розвиток TypeScript як типобезпечної надбудови для масштабних вебзастосунків, що сприяє зниженню ризиків помилок на етапі розробки.

Сучасний етап розвитку мов програмування характеризується активною експансією технологій ШІ та обробки великих даних, у межах якої Python закріпив лідерські позиції завдяки розвиненій екосистемі інструментів для машинного навчання й аналітики. Водночас Rust і Go утвердилися як ключові рішення для створення високопродуктивних і ресурсомістких систем. У результаті сформувався мультиполярний ландшафт мов програмування, у межах якого технології диференціюються за функціональними нішами, зокрема веброзробкою, наукою про дані, системним програмуванням і високопродуктивними інфраструктурними сервісами.

Логічним продовженням сформованого мультиполярного бачення розвитку мов програмування є кількісна інтерпретація виявлених тенденцій на основі довготермінових емпіричних даних.

Джерела даних. Довготермінову основу дослідження становить секція Very Long Term History індексу TIOBE Index, у якій за кожен непарний рік у період з 2001 до 2025 року наведено рангові позиції двадцяти провідних мов програмування [14]. Індекс розраховується на основі частки результатів пошукових запитів за формулою:

$$TIOBE(L) = \frac{Hits(L)}{Hits_{total}} \times 100\% \quad (1)$$

де $Hits(L)$ – кількість результатів пошуку для мови L , $Hits_{total}$ – сумарна кількість результатів для всіх мов у вибірці.

Попри те, що починаючи з 2023 року детальні помісячні значення індексу стали частково обмеженими комерційним доступом, історичні рангові ряди залишаються репрезентативними для аналізу довгострокових трендів. Саме на їх основі було побудовано лінійні залежності динаміки популярності. Для оцінки поточного стану використано рейтинг Top-20 за квітень 2025 року, який містить відсоткові оцінки пошукової активності, що інтерпретуються як індикатори інформаційного інтересу до мов програмування.

З метою верифікації відповідності між інформаційним інтересом і реальною практикою розробки до аналізу залучено звіт GitHub Octoverse 2024, який відображає частку новостворених публічних репозиторіїв за мовами [15]. Його перевага полягає у безпосередньому відображенні активності розробників у межах платформи GitHub. Додатково використано комбінований індекс RedMonk Programming Language Rankings (06/2024), що інтегрує дані про кількість репозиторіїв на GitHub і активність обговорень на Stack Overflow [16], поєднуючи таким чином продуктивну діяльність і взаємодію спільноти.

Методика нормалізації та аналізу даних. Для оцінки популярності мов програмування використовувались дані індексу TIOBE, що відображає позиції мов у рейтингу за активністю спільноти, та GitHub Octoverse, що фіксує частки репозиторіїв у відкритих проєктах.

Нормалізація рангових позицій TIOBE. Рангові позиції TIOBE перетворювались у псевдодискретний інтервал $[0;1]$ за формулою:

$$x' = 1 - \frac{x-1}{x_{max}-1} \quad (2)$$

де x – позиція мови, $(x=1) = 20$

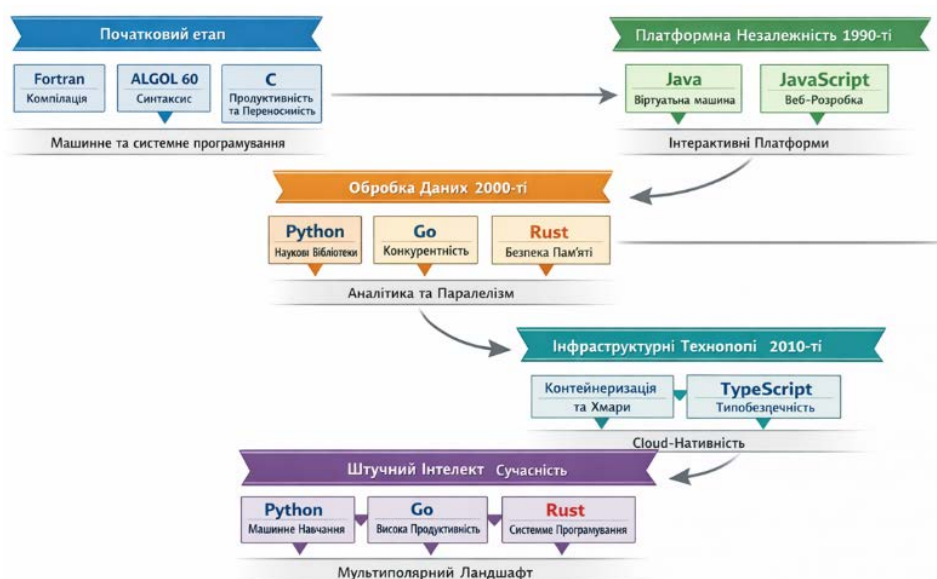


Рис. 1. Хронологія еволюції мов програмування

Отже, мова Python займала 1-шу позицію ($x=1$) у рейтингу ТІОБЕ, тому нормалізоване значення становить:

$$x' = 1 - \frac{1-1}{20-1} = 1$$

Для мови Java, яка займала 3-тю позицію ($x=3$).

$$x' = 1 - \frac{3-1}{20-1} \approx 0.895$$

Таким чином, найпопулярніші мови мають значення, близькі до 1, а менш популярні – ближче до нуля.

Нормалізація відсоткових часток GitHub. Частки мов у репозиторіях GitHub нормалізуються методом мінімакс-перетворення:

$$\text{Share}_{\text{norm}} = \frac{\text{Share} - \text{Share}_{\min}}{\text{Share}_{\max} - \text{Share}_{\min}} \quad (3)$$

Отже, якщо Python має частку 15%, JavaScript – 20%, а мінімальна та максимальна частки в наборі даних – 5% та 25% відповідно, то нормалізовані значення:

$$\text{Python}_{\text{norm}} = \frac{15-5}{25-5} = 0.5$$

$$\text{JavaScript}_{\text{norm}} = \frac{20-5}{25-5} = 0.75$$

Кореляційний аналіз. Для оцінки лінійного взаємозв'язку між нормалізованими даними ТІОБЕ та GitHub застосовується коефіцієнт кореляції Пірсона:

$$r = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^n (X_i - \bar{X})^2 \sum_{i=1}^n (Y_i - \bar{Y})^2}} \quad (4)$$

де x_i та y_i – нормалізовані значення рангу та частки відповідно, $\bar{x}$ та $\bar{y}$ – середні значення у вибірці. Такий підхід дозволяє кількісно оцінити узгодженість різних джерел даних та виявити довгострокові тенденції популярності мов.

Отже, для Python, Java та JavaScript нормалізовані значення $x = [1, 0.895, 0.842]$, $y = [0.5, 0.6, 0.75]$. Коефіцієнт Пірсона розраховується як $r \approx 0.98$, що свідчить про високу узгодженість між рейтингом ТІОБЕ та частками на GitHub.

Середньорічний темп приросту (CAGR). Динаміку змін часток мов у часі оцінювали за показником середньорічного темпу приросту, що визначається як:

$$\text{CAGR} = \left(\frac{\text{Share}_{\text{final}}}{\text{Share}_{\text{initial}}} \right)^{1/n} - 1 \quad (5)$$

де n – кількість років у періоді спостереження. CAGR дозволяє порівнювати середньорічні темпи приросту незалежно від коливань на окремі роки.

Отже, якщо частка Python зросла з 10% до 15% за 5 років, то:

$$\text{CAGR} = \left(\frac{15}{10} \right)^{1/5} - 1 \approx 0.084 \text{ або } 8.4\% \text{ на рік}$$

Таким чином, поєднання нормалізації рангових позицій, мін-макс-перетворення часток та кореляційного аналізу дозволяє отримати цілісну картину тенденцій розвитку популярності мов програмування та забезпечує надійну основу для подальших статистичних досліджень.

Аналіз результатів. Аналітичні розрахунки базуються на відкритих даних ТІОБЕ, звітах GitHub Octoverse та дослідженнях RedMonk.

Динаміка рангових позицій за індексом ТІОБЕ. Ретроспективний аналіз рангових позицій індексу ТІОБЕ (рис. 2) виявляє чітко виражену зміну технологічного лідерства. У період 2005–2018 рр. Так, мова Java стабільно утримувала першу позицію, що відображало її домінування у корпоративному секторі. Водночас Python демонстрував стійку висхідну динаміку, послідовно покращуючи свої позиції. Перетин трендових кривих відбувся наприкінці 2018 р., після чого Python закріпився як лідер, починаючи з 2021 р. Зазначене свідчить про структурний зсув від enterprise-орієнтованих рішень до data-driven та AI-орієнтованих технологій.

Подальший аналіз показує перерозподіл позицій у групі провідних мов. Java поступово втратила домінування і змістилася на четверту позицію, поступившись Python, C та C++. Водночас C++ після періоду стагнації відновила зростання, що пов'язано з оновленням стандарту та зростанням попиту на високопродуктивні обчислення. Така динаміка підтверджує збереження ролі системних мов у критичних обчислювальних середовищах.

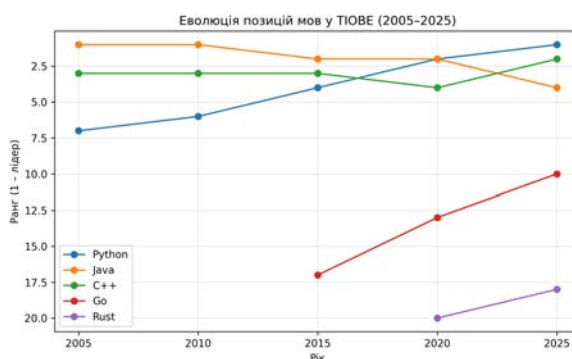


Рис. 2. Еволюція рангових позицій мов програмування за індексом ТІОБЕ (2005–2025)

Окремої уваги заслуговують нові мови програмування. Go та Rust, з'явившись у рейтингу відносно нещодавно, демонструють стабільне зростання. Go активно просувається у сфері хмарних сервісів і мікросервісної архітектури, досягнувши середини рейтингу. Rust, хоча й залишається у нижній частині списку, характеризується стійким приростом, що свідчить про поступове впровадження у нішах, критичних до безпеки пам'яті.

Актуальний розподіл пошукової уваги станом на 2025 рік підтверджує домінування Python із часткою 15,16%. Формується група найближчих конкурентів, до якої входять C (10,97%), C++ (10,95%) і Java (8,93%). Водночас спостерігається суттєва невідповідність між позиціями окремих мов у різних

індексах. Зокрема, JavaScript займає лише шосту позицію (3,17%) в TIOBE, що обумовлено специфікою методології, орієнтованої на пошукові запити. Це підтверджує, що пошукові метрики не повною мірою відображають реальну практику розробки та потребують доповнення репозиторними даними.

Таким чином, отримані результати кількісно підтверджують формування мультиполярного технологічного ландшафту та узгоджуються з попереднім історичним аналізом. Виявлені тренди відображають глибинні трансформації у структурі сучасної програмної інженерії.

Аналіз репозиторної активності GitHub Octoverse. Виявлені розбіжності у пошукових метриках зумовлюють необхідність їх верифікації на основі даних, що безпосередньо відображають практику розробки. У цьому контексті доцільним є залучення репозиторних показників, які дозволяють оцінити фактичну активність розробників та структуру створеного програмного забезпечення.

Аналіз репозиторної активності за даними GitHub Octoverse 2024 (рис. 3) демонструє суттєво відмінний розподіл популярності мов програмування. Лідерські позиції впевнено утримує JavaScript із часткою 28,0% нових публічних репозиторіїв. Python займає другу позицію з показником 22,9%, скорочуючи відставання, але не досягаючи домінування. Третє місце належить Java (9,7%), тоді як наступні позиції займають TypeScript (9,0%), Go (5,8%) та C++ (5,5%). Така структура відображає реальну архітектуру сучасних програмних систем, у якій веб-орієнтовані та хмарні технології відіграють визначальну роль.

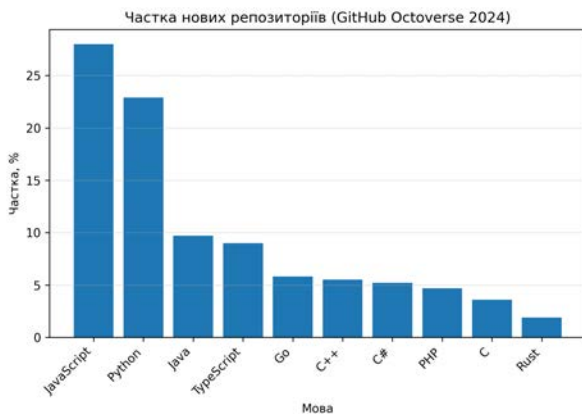


Рис. 3. Частка нових репозиторіїв за даними GitHub Octoverse 2024

Порівняння з результатами індексу TIOBE виявляє принципові відмінності у підходах до оцінювання популярності. Пошукові індекси фіксують рівень інтересу та освітньої активності, тоді як репозиторні метрики відображають безпосередню участь мов у створенні програмного коду. Зазначене пояснює домінування JavaScript і TypeScript у GitHub Octoverse, оскільки ці мови є базовими для

веб-розробки та активно використовуються у клієнтських і серверних застосунках.

Водночас мови C та C++, які займають високі позиції в TIOBE, демонструють нижчу частку у репозиторних даних. Це зумовлено їх специфікою використання у системному програмуванні та наявністю значної частки закритого корпоративного коду. Таким чином, спостерігається системна невідповідність між інформаційною увагою та реальною практикою розробки.

Підсумовуючи, можна зазначити, що отримані результати підтверджують методологічну обмеженість окремих індексів і обґрунтовують необхідність їх інтегрованого використання. При цьому поєднання пошукових і репозиторних даних дозволяє сформувати більш об'єктивне уявлення про сучасний стан і тенденції розвитку мов програмування.

Порівняльний аналіз індексів. Виявлені відмінності між пошуковими та репозиторними метриками обумовлюють необхідність їх узагальнення в межах єдиного аналітичного підходу. З цієї метою виконано інтеграцію даних індексів TIOBE, GitHub Octoverse та RedMonk, що дозволяє здійснити комплексний порівняльний аналіз популярності мов програмування.

Зведені результати (табл. 1) формують узгоджену інформаційну базу для подальшої інтерпретації.

Таблиця 1. Порівняння показників популярності мов програмування за даними TIOBE, GitHub Octoverse та RedMonk у 2024–2025 pp.

Мова	TIOBE Rank 2025	TIOBE Share 2025 (%)	Octoverse 2024 (%)	RedMonk Rank 2024
Python	1	15,16	22,9	2
C	2	10,97	3,6	9
C++	3	10,95	5,5	7
Java	4	8,93	9,7	3
C#	5	7,58	5,2	5
JavaScript	6	3,17	28,0	1
PHP	7	2,47	4,7	4
Visual Basic	8	2,26	-	-
SQL	9	1,86	-	-
Go	11	1,28	5,8	12
TypeScript	14	0,93	9,0	6
Rust	20	0,73	1,9	19

Оскільки вихідні показники мають різну природу та масштаб, для забезпечення коректності зіставлення застосовано процедуру нормалізації. Зазначене дозволило перейти до уніфікованого простору оцінювання та побудувати порівняльну діаграму (рис. 3), яка наочно відображає розбіжності між індексами.

Оцінка узгодженості між джерелами свідчить про помірний позитивний зв'язок між показниками TIOBE та GitHub Octoverse. Такий рівень кореляції є статистично значущим, проте недостатнім для взаємозамінності індексів. Це підтверджує, що пошукова активність і репозиторна присутність відображають різні аспекти використання мов програмування: когнітивний інтерес та практичну реалізацію відповідно.

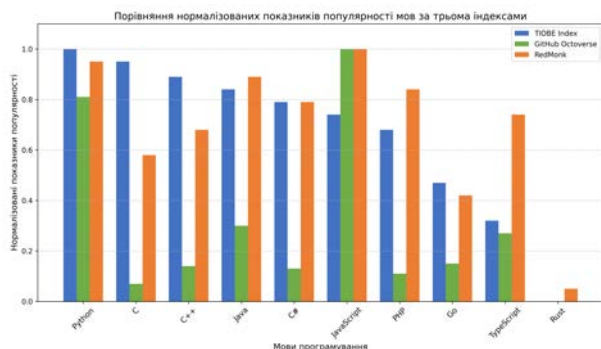


Рис. 4. Порівняння нормалізованих показників популярності мов за трьома індексами

Додатковий аналіз динаміки розвитку мов виконано на основі середньорічних темпів приросту. За даними TIOBE, Python демонструє стійке зростання з інтенсивністю близько 0,68% на рік, що підтверджує його системне посилення у технологічному ландшафті. Мови Go (0,22% рік⁻¹) та Rust (0,14% рік⁻¹) також характеризуються позитивною динамікою, хоча їхній вплив наразі обмежений нішевими сегментами. Водночас C та C++ залишаються відносно стабільними ($\pm 0,05\%$ рік⁻¹), що свідчить про зрілість і насиченість відповідного ринку. JavaScript у межах пошукових метрик демонструє квазістаціонарну поведінку.

На відміну від цього, репозиторні дані GitHub Octoverse виявляють інші закономірності. Найвищі темпи зростання демонструє TypeScript (1,2% рік⁻¹), що зумовлено переходом індустрії до типобезпечних рішень у веб-розробці. Rust також демонструє прискорене зростання (0,36% рік⁻¹), що відображає підвищення вимог до безпеки програмного забезпечення. Таким чином, спостерігається зміщення практичних переваг розробників у бік сучасних мов із розширеними можливостями типізації та контролю ресурсів.

Отримані результати підтверджують доцільність інтегрованого підходу до аналізу популярності мов програмування. Поєднання різних індексів дозволяє не лише підвищити достовірність оцінювання, але й виявити глибинні структурні зміни у сучасній програмній інженерії.

ОБГОВОРЕННЯ ОТРИМАНИХ РЕЗУЛЬТАТІВ

Аналіз факторів, що вплинули на зміну розстановки сил у рейтингах мов програмування, демонструє комплексність сучасного програмного ландшафту. Порівняння даних TIOBE, GitHub Octoverse та RedMonk дозволяє ідентифікувати низку ключових драйверів трансформації, що визначають популярність і практичне застосування мов у різних сегментах індустрії. Наведені фактори тісно взаємодіють, формуючи нову мультиполярну екосистему, де лідерські позиції традиційних мов співіснують із швидкозростаючими платформами для веб-розробки, аналітики даних та системного програмування.

1. *Експансія веб-розробки та JavaScript-екосистеми.* Домінування JavaScript у репозиторних даних (28,0% GitHub Octoverse) безпосередньо пов'язане з поширенням веб-застосунків та концепцією Single Page Application. Розвиток фреймворків React, Vue, Angular і середовища Node.js дозволив мові вийти за межі браузера й стати платформою для повного стеку розробки. Паралельно зростання TypeScript (9,0% GitHub Octoverse, 6 місце RedMonk) відображає прагнення індустрії до типобезпечного програмування в масштабних проєктах, що суттєво підвищує стабільність та передбачуваність коду.

2. *Злет Python у науці про дані та машинному навчанні.* Ключовим фактором популярності Python став стрімкий розвиток аналітики даних і штучного інтелекту. Поява бібліотек NumPy (2006), SciPy (2008), а особливо TensorFlow (2015) та PyTorch (2016), закріпила його статус мови №1 для обробки даних. Додатковий імпульс надали великі мовні моделі (2022+), де Python залишається основною мовою для експериментів та розгортання. За даними AI Index Report 2024, понад 83% нових проєктів у сфері машинного навчання розробляються на Python, створюючи мультиплікативний ефект для його екосистеми.

3. *Інтеграція хмарних технологій та контейнеризації.* Поява AWS Lambda та Google Cloud Functions (2014-2016) інтегрувала Python у serverless-ландшафт. Одночасно розвиток контейнеризації стимулював поширення Go, мова якого спочатку розроблялася для конкурентних сервісів із мінімальними залежностями. Частка Go у GitHub Octoverse (5,8%) значно перевищує його позицію в TIOBE (1,28%), що демонструє активне використання в нових проєктах мікросервісної архітектури.

4. *Підвищення вимог до безпеки програмного забезпечення.* Сфера системного програмування зазнала трансформації під впливом двох векторів: поява «cloud-native» Go з легкою моделлю конкурентності та Rust, що забезпечує гарантії безпеки пам'яті без накладних витрат збирача сміття. Звіт Google Security Blog (2022) зафіксував, що 70% критичних вразливостей Android пов'язані з помилками управління пам'яттю, що легітимізує Rust у кореневих компонентах ОС. Незважаючи на скромну частку в GitHub Octoverse (1,9%), Rust демонструє найвищі темпи зростання серед системних мов.

У сукупності наведені фактори формують новий ландшафт популярності мов програмування. Традиційні лідери, такі як Python, C та C++, зберігають стабільні позиції у рейтингу TIOBE, який оцінює популярність мов на основі пошукових запитів у глобальному інтернет-просторі. Водночас сучасні тенденції розробки більш точно відображаються у репозиторних даних, таких як GitHub Octoverse та RedMonk, де значення має безпосереднє використання мов у реальних проєктах.

Наприклад, JavaScript у TIOBE займає шосте місце (3,17%), тоді як у GitHub Octoverse він посідає першу позицію (28,0%). Ця диспропорція пояснюється специфікою пошукових запитів де розробники активно користуються внутрішньою документацією та пакетним менеджером `npm`, а не загальними пошуковими системами. Аналогічна ситуація спостерігається для TypeScript, який у TIOBE посідає 14-те місце (0,93%), тоді як у RedMonk він входить до топ-6, а в GitHub Octoverse займає четверте місце (9,0%).

Таке порівняння демонструє, що класичні рейтинги на основі пошукових запитів відображають історичну та академічну популярність мов, тоді як репозиторні дані фіксують практичне використання у сучасних проєктах. Це підкреслює необхідність комплексного підходу до аналізу тенденцій розвитку мов програмування та формує основу для побудови багатовимірних порівняльних моделей.

5. *Обмеження дослідження.* Серед головних обмежень слід зазначити часткову закритість щомісячних даних TIOBE після 2023 року, що ускладнює побудову повної динаміки. Частково це компенсовано використанням таблиці Very Long Term History та відкритих джерел, таких як GitHub Octoverse і RedMonk.

Структура вибірки GitHub Octoverse також має обмеження. Враховуються переважно публічні репозиторії, що може занижувати частку мов, широко застосованих у корпоративному секторі (Java, C#). Gartner оцінює [14], що до 70% критичного корпоративного коду залишається у закритих системах, що впливає на точність вимірювання популярності.

Рейтинг RedMonk може переоцінювати мови, популярні серед початківців та активних учасників спільнот, і недооцінювати нішеві мови, що використовуються в академічних або вбудованих системах. Врахування цих обмежень є обов'язковим при інтерпретації результатів та формуванні практичних рекомендацій.

Наведені обмеження не нівелюють значущості отриманих результатів, однак потребують врахування під час їх інтерпретації, зокрема при формулюванні практичних рекомендацій або екстраполяції на вузькоспеціалізовані сегменти IT-ринку.

6. *Практичні рекомендації.* Результати дослідження мають прикладне значення для освіти та професійного розвитку. Програми навчання доцільно будувати на мультипарадигмальній моделі, що поєднує різні сегменти ландшафту програмування.

Дані TIOBE, GitHub Octoverse та RedMonk співпадають у тому, що Python та JavaScript/TypeScript формують ядро сучасної розробки програмного забезпечення [17]. Відповідно, освітні програми доцільно будувати навколо цих мов як фундаментальних інструментів для формування базових навичок програмування, алгоритмічного мислення та розуміння сучасної веб-екосистеми. При цьому для системного рівня залишаються важливими C/C++, а Rust та Go

є перспективними інструментами для безпечного та продуктивного програмування.

З огляду на потреби ринку праці, базова комбінація знань Python і JavaScript/TypeScript забезпечує конкурентну перевагу для junior-розробників, тоді як володіння Go або Rust стає важливим для middle та senior фахівців у сферах хмарних сервісів, контейнеризації та високонавантажених систем. Поява Go у ключових модулях Kubernetes та Rust у ядрах браузерів і криптографічних бібліотеках підтверджує їх зростаючу роль у галузі.

На рівні індустрійних стандартів ці зміни також поступово фіксуються. Зокрема, у 2024 році робоча група MISRA анонсувала ініціативу «MISRA Next», у межах якої Rust визначено як кандидата для наступного покоління стандартів безпечного програмування в галузі automotive. Паралельно Go вже демонструє відповідність вимогам FIPS 140-3 у складі криптографічної бібліотеки Google Tink, що підсилює його позиції як надійного інструмента для розробки безпечних інфраструктурних сервісів.

Таким чином, співставлення трьох незалежних індексів підтверджує формування мультиполярної екосистеми, де JavaScript домінує у веб-розробці, Python утримує лідерство в аналітиці даних, а системний рівень програмування розподіляється між традиційними C/C++ та новими Go й Rust у відповідних нішах. Ця диференціація має бути врахована при формуванні освітніх та кадрових стратегій для забезпечення відповідності навичок розробників актуальним вимогам ринку.

ВИСНОВКИ

За результатами проведеного дослідження сформовано інтегроване бачення популярності мов програмування на основі комплексного аналізу рейтингів TIOBE, RedMonk та GitHub Octoverse. Застосування комплементарного підходу дозволило зменшити вплив методологічних обмежень окремих індексів і отримати більш об'єктивну оцінку сучасного технологічного ландшафту.

Встановлено, що ринок мов програмування зазнає системної трансформації, яка проявляється у функціональній диференціації мов. Python закріпився як універсальна мова з домінуванням у сфері аналізу даних і штучного інтелекту, поєднуючи високі позиції у всіх досліджуваних індексах. JavaScript демонструє лідерство у практичній розробці, що підтверджується його домінуванням у GitHub Octoverse та RedMonk, попри відносно нижчу позицію в TIOBE.

Показано, що C/C++ зберігають значущість у системному програмуванні, однак їхня роль у нових проєктах поступово зменшується. Натомість Go та Rust демонструють стійке зростання як інструменти для створення високопродуктивних, безпечних та cloud-native систем, що відповідає сучасним вимогам до програмної інфраструктури.

В цілому отримані результати формують аналітичну основу для прийняття рішень у сфері ІТ-освіти та вибору технологічного стеку. Обґрунтовано доцільність використання базового набору мов Python–JavaScript–SQL для формування фундаментальних компетенцій, а також необхідність включення Go і Rust для підготовки фахівців у сфері хмарних та високонавантажених систем.

Підтверджено формування мультиполярного програмного ландшафту, в якому мови програмування спеціалізуються за функціональними нішами. Запропонована інтегрована модель оцінювання може бути використана як інструмент для подальших досліджень, а також для стратегічного планування розвитку освітніх програм і професійних компетенцій у галузі інформаційних технологій.

REFERENCES

- [1] Rockstar Developer University. (2026). Programming language statistics 2026: Popularity, salary, and growth data. <https://rockstardeveloperuniversity.com/programming-language-statistics/>
- [2] Spinellis, D., & Gousios, G. (2009). Beautiful architecture: Leading thinkers reveal the hidden beauty in software design. O'Reilly Media. <https://books.google.com.ua/books?id=h34pwy005nYC>
- [3] Stack Overflow. (2024). Stack Overflow Developer Survey 2024. <https://survey.stackoverflow.co/2024/>
- [4] TIOBE. (2025). TIOBE Programming Community Index. <https://www.tiobe.com/tiobe-index/>
- [5] Ray, B., Posnett, D., Devanbu, P., & Filkov, V. (2017). A large-scale study of programming languages and code quality in GitHub. Communications of the ACM, 60(10), 91–100. <https://doi.org/10.1145/3126905>
- [6] Cherny-Shahar, T., & Yehudai, A. (2024). Multi-lingual development & programming languages interoperability: An empirical study. *arXiv preprint arXiv:2411.08388*, 1–26. <https://doi.org/10.48550/arXiv.2411.08388>
- [7] Đurdev, D. (2024). Popularity of programming languages (PYPL, TIOBE, Stack Overflow indices comparison). AIDASCO Reviews, 2(2), 24–29. <https://doi.org/10.59783/aire.2024.59>
- [8] Hindle, A., Barr, E. T., Su, Z., Gabel, M., & Devanbu, P. (2012). On the naturalness of software. In Proceedings of the 34th International Conference on Software Engineering (ICSE). <https://doi.org/10.1109/ICSE.2012.6227135>
- [9] Backus, J. W., Beeber, R. J., Best, S., Goldberg, R., Haibt, L. M., Herrick, H. L., ... Nutt, R. (1957). The FORTRAN automatic coding system. IBM Journal of Research and Development, 1(1), 3–17. <https://doi.org/10.1145/1455567.1455599>
- [10] Naur, P., & Backus, J. W. (1960). Revised report on the algorithmic language ALGOL 60. Communications of the ACM, 3(5), 299–314.
- [11] Patel, M. H. (2025). The enduring relevance of C programming: Analysing its efficiency, portability, and role in modern computing. GAP Interdisciplinarity: A Global Journal of Interdisciplinary Studies, VIII(Special Issue), 546–550. [https://www.gapinterdisciplinarity.org/res/articles/\(546-550\)-THE-ENDURING-RELEVANCE-OF-C-PROGRAMMING-ANALYSING-ITS-EFFICIENCY-PORTABILITY-AND-ROLE-IN-MODERN-COMPUTING-20250818035718.pdf](https://www.gapinterdisciplinarity.org/res/articles/(546-550)-THE-ENDURING-RELEVANCE-OF-C-PROGRAMMING-ANALYSING-ITS-EFFICIENCY-PORTABILITY-AND-ROLE-IN-MODERN-COMPUTING-20250818035718.pdf)
- [12] Shukla, A. (2023). Modern JavaScript frameworks and JavaScript's future as a full stack programming language. Journal of Artificial Intelligence & Cloud Computing, 2(4), 1–5. [https://doi.org/10.47363/JAICC/2023\(2\)144](https://doi.org/10.47363/JAICC/2023(2)144)
- [13] Python Software Foundation. (2025). Python language reference (Version 3.12). Python Software Foundation. <https://docs.python.org/3/reference/>
- [14] TIOBE Software. (2025). TIOBE index. TIOBE Software. <https://www.tiobe.com>
- [15] GitHub. (2024, October 29). Octoverse 2024 report: AI leads JavaScript and Python to top languages as the number of global developers surges. <https://github.blog/news-insights/octoverse/octoverse-2024/>
- [16] O'Grady, S. (2024, September 12). The RedMonk programming language rankings: June 2024. <https://redmonk.com/sograde/2024/09/12/language-rankings-6-24/>
- [17] Boriak, B. R., Yanko, A. S., & Mykhailichenko, O. V. (2025). Validation of control system software for a ground robotic complex. *Taurian Scientific Bulletin. Series: Technical Sciences*, 1(5), 56–64. <https://doi.org/10.32782/tnv-tech.2025.5.1.7>

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

- [1] Rockstar Developer University. (2026). Programming language statistics 2026: Popularity, salary, and growth data. <https://rockstardeveloperuniversity.com/programming-language-statistics/>
- [2] Spinellis, D., & Gousios, G. (2009). Beautiful architecture: Leading thinkers reveal the hidden beauty in software design. O'Reilly Media. <https://books.google.com.ua/books?id=h34pwy005nYC>
- [3] Stack Overflow. (2024). Stack Overflow Developer Survey 2024. <https://survey.stackoverflow.co/2024/>
- [4] TIOBE. (2025). TIOBE Programming Community Index. <https://www.tiobe.com/tiobe-index/>
- [5] Baishakhi Ray, Daryl Posnett, Premkumar Devanbu, & Vladimir Filkov. (2017). A large-scale study of programming languages and code quality in GitHub. Communications of the ACM, 60 (10), 91–100. <https://doi.org/10.1145/3126905>
- [6] Cherny-Shahar, T., & Yehudai, A. (2024). Multi-Lingual Development & Programming Languages Interoperability: An Empirical Study. *arXiv preprint arXiv:2411.08388*, 1–26. <https://doi.org/10.48550/arXiv.2411.08388>
- [7] Baishakhi Ray, Daryl Posnett, Premkumar Devanbu, & Vladimir Filkov. (2017). A large-scale study of programming languages and code quality in GitHub. Communications of the ACM, 60(10), 91–100. <https://doi.org/10.1145/3126905>

- [8] Abram Hindle, Earl T. Barr, Zhendong Su, Mark Gabel, & Premkumar Devanbu. (2012). On the naturalness of software. Proceedings of the 34th International Conference on Software Engineering (ICSE). <https://doi.org/10.1109/ICSE.2012.6227135>
- [9] Backus, J. W., Beeber, R. J., Best, S., Goldberg, R., Haibt, L. M., Herrick, H. L., ... & Nutt, R. (1957). The FORTRAN automatic coding system. IBM Journal of Research and Development, 1(1), 3–17. <https://doi.org/10.1145/1455567.1455599>
- [10] Naur, P., & Backus, J. W. (1960). Revised report on the algorithmic language ALGOL 60. Communications of the ACM, 3(5), 299–314.
- [11] Patel, M. H. (2025). The enduring relevance of C programming: Analysing its efficiency, portability, and role in modern computing. GAP Interdisciplinarity: A Global Journal of Interdisciplinary Studies, VIII(Special Issue), 546–550. [https://www.gapinterdisciplinarity.org/res/articles/\(546-550\)-THE-ENDURING-RELEVANCE-OF-C-PROGRAMMING-ANALYSING-ITS-EFFICIENCY-PORTABILITY-AND-ROLE-IN-MODERN-COMPUTING-20250818035718.pdf](https://www.gapinterdisciplinarity.org/res/articles/(546-550)-THE-ENDURING-RELEVANCE-OF-C-PROGRAMMING-ANALYSING-ITS-EFFICIENCY-PORTABILITY-AND-ROLE-IN-MODERN-COMPUTING-20250818035718.pdf)
- [12] Shukla, A. (2023). Modern JavaScript frameworks and JavaScript's future as a full-stack programming language. Journal of Artificial Intelligence & Cloud Computing, 2(4), 1–5. [https://doi.org/10.47363/JAICC/2023\(2\)144](https://doi.org/10.47363/JAICC/2023(2)144)
- [13] Python Software Foundation. (2025). Python language reference (Version 3.12). Python Software Foundation. <https://docs.python.org/3/reference/>
- [14] TIOBE Software. (2025). TIOBE index. TIOBE Software. <https://www.tiobe.com>
- [15] GitHub. (2024, October 29). Octoverse 2024 report: AI leads JavaScript and Python to top languages as the number of global developers surges. <https://github.blog/news-insights/octoverse/octoverse-2024/>
- [16] O'Grady, S. (2024, September 12). The RedMonk programming language rankings: June 2024. <https://redmonk.com/sograde/2024/09/12/language-rankings-6-24/>
- [17] Боряк, Б.Р., Янко, А.С., Михайліченко О.В. (2025) Валідація програмного забезпечення системи керування наземного роботизованого комплексу. *Таврійський науковий вісник. Серія: Технічні науки*, 1(5), 56–64. <https://doi.org/10.32782/tnv-tech.2025.5.1.7>

© Кучма Ю. В., Янко А. С., Рубін Е. Ю., Поліновський В. В.

Дата першого надходження статті до видання: 11.02.2026

Дата прийняття статті до друку після рецензування: 06.03.2026

Дата публікації (оприлюднення) статті: 22.05.2026



Стаття поширюється на умовах ліцензії відкритого доступу (CC BY 4.0)