

УДК: 004.4

*Просянніков А. В., здобувач вищої освіти,
Антонов Ю. С., канд. фіз.-мат. наук, доцент,
доцент кафедри інформаційних технологій,
Донецький національний університет імені Василя Стуса*

АНАЛІЗ ВИБОРУ МОВИ ПРОГРАМУВАННЯ ЗАЛЕЖНО ВІД СФЕРИ ЗАСТОСУВАННЯ ТА ХАРАКТЕРИСТИК ПРОДУКТИВНОСТІ

Анотація. У роботі проведено аналіз вибору мови програмування залежно від сфери застосування, характеристик продуктивності та популярності серед розробників. Досліджено взаємозв'язок між швидкодією виконання коду, можливостями компіляторів і складністю переходу між мовами у разі зміни напряму діяльності програміста.

Ключові слова: мови програмування, сфера застосування, продуктивність, компілятори, швидкодія, перехід між мовами, популярність.

Вступ. Сучасна ІТ-індустрія характеризується високою динамікою розвитку та різноманіттям мов програмування, кожна з яких має власні переваги й обмеження. Вибір мови залежить не лише від зручності синтаксису чи особистих уподобань розробника, а насамперед від сфери застосування, вимог до швидкодії, ефективності компіляції та підтримки певної архітектури.

У зв'язку з розширенням спектра завдань, які вирішуються програмними засобами, розробники часто переходять з однієї мови на іншу. Такі переходи обумовлені зміною технологічного напряму: наприклад, розроблення системного або вбудованого програмного забезпечення потребує володіння C чи Rust, тоді як у сфері веброзробки переважають JavaScript, TypeScript або Java. У наукових і аналітичних обчисленнях найчастіше застосовують Python, Julia чи R, що забезпечують зручність реалізації алгоритмів машинного навчання, проте поступаються в швидкодії компільованим мовам.

Проблема раціонального вибору мови програмування набуває особливої актуальності в умовах інтеграції штучного інтелекту, інтернету речей і високопродуктивних обчислень. Вона постає не лише перед початківцями, а й перед досвідченими фахівцями, які змінюють сферу діяльності – наприклад, переходять із корпоративного програмування до системного або наукового.

Метою роботи є дослідження факторів, що визначають доцільність використання певних мов програмування залежно від сфери застосування, а також оцінка впливу швидкодії, можливостей компіляторів і складності переходу між мовами на ефективність розроблення програмних систем.

Основний текст. Для дослідження було використано відкриті аналітичні джерела, зокрема Stack Overflow Developer Survey 2024 [1], GitHub Octoverse 2024 [2], Computer Language Benchmarks Game [3] та офіційну документацію до компіляторів GCC, LLVM і JIT-середовищ [4]. Для аналізу попиту на мови програмування використано відкриті статистичні джерела, зокрема Stack Overflow Developer Survey 2024 [1], GitHub Octoverse 2024 [2] та TIOBE Index 2025 [5]. Отримані дані дали змогу виявити основні закономірності у виборі мов програмування залежно від сфери їх застосування та технічних характеристик (табл. 1) [3].

Таблиця 1 – Порівняльна характеристика мов програмування за сферами застосування та продуктивністю

Сфера застосування	Типові мови	Компілятор	Особливості продуктивності
Системне ПЗ, драйвери	C, C++	GCC, Clang	Висока швидкодія, контроль пам'яті, низькорівневі оптимізації
Вбудовані системи (embedded)	C, Rust	GCC, LLVM	Мінімальні ресурси, безпечна робота з пам'яттю
Веброзроблення	JavaScript, TypeScript, Java, C#	Node.js, JIT CLR/JVM	Баланс між швидкістю і масштабованістю
Ігрова індустрія	C++, C#, UnrealScript	Visual C++, Mono, Unity IL2CPP	Висока FPS-швидкість, оптимізація під GPU
Штучний інтелект і наука	Python, R, Julia	CPython, PyPy, LLVM	Простота реалізації алгоритмів, велика кількість бібліотек
Безпека, криптографія	Go, Rust, C, Python	GCC, LLVM	Безпечність і ефективне керування потоками

Різні галузі програмування мають власні вимоги до продуктивності, безпеки, масштабованості й зручності розроблення. У системному програмуванні перевага надається компільованим мовам, як-от C та C++, які забезпечують високий рівень контролю над апаратними ресурсами й мінімальні витрати пам'яті. У сфері вбудованих систем активно розвиваються Rust і C, що поєднують ефективність із підвищеною безпечністю виконання коду.

У веброботці, навпаки, ключовими критеріями є швидкість розробки та можливість створення кросплатформних рішень. Тут домінують JavaScript, TypeScript, Java та C#, що працюють у середовищах із динамічною компіляцією (JIT) і мають розвинуті екосистеми бібліотек. У мобільних і ігрових додатках використовуються C++, C#, UnrealScript, які забезпечують високу продуктивність графіки та взаємодію з апаратним забезпеченням.

Наукові обчислення, штучний інтелект і аналітика даних переважно реалізуються мовами C++, Fortran, Python, R та Julia. Для безпекових систем і криптографії застосовують Go, Rust і C, які забезпечують стабільність і ефективне керування потоками виконання.

Крім сфери застосування, важливим фактором вибору є можливості компілятора. Традиційні інструменти, такі як GCC і Clang, орієнтовані на глибоку оптимізацію під конкретну архітектуру. LLVM підтримує модульність і міжмовну оптимізацію [4], тоді як JIT-компілятори у Java та C# забезпечують динамічне покращення продуктивності під час виконання. Натомість IL2CPP у Unity поєднує швидкість C++ із гнучкістю мов на базі .NET, що особливо важливо для ігрової індустрії [5].

Окремої уваги заслуговує питання переходу між мовами. Розробники часто змінюють мову не через її синтаксис, а через вимоги нової сфери діяльності. Наприклад, перехід із Python на C++ або Rust зумовлений потребою у вищій швидкодії, тоді як перехід з C# на Go пов'язаний із переходом від монолітних до мікросервісних архітектур. Така міграція потребує врахування не лише технічних відмінностей, але й екосистеми, доступних фреймворків і рівня автоматизації інструментів. Для оцінки адаптаційної складності у разі зміни мов програмування

проведено порівняння їх популярності та складності опанування серед розробників (табл. 2).

Таблиця 2 – Оцінка популярності та складності мов програмування

Мова програмування	Популярність (1–5)	Складність опанування (1–5)	Узагальнений показник «попит / вхід»
Python	5	1	5,0
JavaScript	5	2	4,5
C#	4	2	4,0
Java	4	3	3,5
C++	3	5	2,0
Go	3	3	3,0
Rust	2	5	1,5

Отже, вибір мови програмування – це багатофакторне завдання, у якому поєднуються три групи параметрів: сфера застосування, технічні характеристики (продуктивність, тип компіляції, оптимізація) та зручність освоєння. Представлені результати створюють підґрунтя для розроблення рекомендаційної системи, здатної формувати пропозиції щодо вибору або переходу між мовами відповідно до цілей, контексту й технічних вимог розробника.

Висновки. Проведений аналіз показав, що вибір мови програмування визначається поєднанням сфери застосування, технічних характеристик і зручності освоєння. Компільовані мови (C, C++, Rust) забезпечують високу продуктивність у системному та вбудованому програмуванні, тоді як Python, JavaScript і C# надають гнучкість і швидкість розроблення у веб- та наукових проєктах. Отримані результати можуть бути використані для побудови рекомендаційної системи, яка автоматично пропонуватиме оптимальну мову залежно від цілей і вимог розробника [6].

Список використаних джерел

1. Stack Overflow Developer Survey 2024. URL: <https://survey.stackoverflow.co/2024> (дата звернення: 22.10.2025).
2. GitHub Octoverse 2024. URL: <https://octoverse.github.com> (дата звернення: 22.10.2025).
3. The Computer Language Benchmarks Game. URL: <https://benchmarksgame-team.pages.debian.net> (дата звернення: 22.10.2025).
4. LLVM Project. Official Documentation. URL: <https://llvm.org/docs> (дата звернення: 22.10.2025).
5. TIOBE Index 2025. URL: <https://www.tiobe.com/tiobe-index> (дата звернення: 22.10.2025).
6. Антонов Ю. С. Залежність швидкодії програм від інструментальних засобів розробки та синтаксичних конструкцій. *Ukrainian Journal of Information Systems and Data Science*. 2024. Т. 2, № 1. С. 27–38.