

DOI: <https://doi.org/10.36910/6775-2524-0560-2024-57-04>

УДК: 004.42

Бойко Лев Степанович, к.т.н.

<https://orcid.org/0009-0001-0117-8551>

Ліщина Наталія Миколаївна, к.т.н., доцент

<https://orcid.org/0000-0002-5200-536X>

Луцький національний технічний університет, м. Луцьк, Україна

ДОСЛІДЖЕННЯ МОЖЛИВОСТІ ВИКОРИСТАННЯ GOLANG У ЯКОСТІ ПЕРШОЇ МОВИ ПРОГРАМУВАННЯ У НАВЧАЛЬНОМУ ПРОЦЕСІ

Бойко Л.С., Ліщина Н.М. Дослідження можливості використання Golang у якості першої мови програмування у навчальному процесі. У статті висвітлено переваги використання Golang у якості першої мови програмування. Зокрема розглянуто такі аспекти як простота та зрозумілість синтаксису, переваги строгої типізації, особливості мови, які дозволяють зменшити кількість потенційних помилок а також важливість передбачуваності результатів виконання коду, можливість написання програм на Go з використанням різних парадигм розробки програмного забезпечення та наявність додаткової інформації і активної спільноти. Виконано порівняння Go з іншими мовами програмування (наприклад, C++, JavaScript, PHP та Python) з точки зору застосування у навчальному процесі. Наведено приклади коли Golang має суттєві переваги щодо використання у процесі навчання майбутніх спеціалістів та випадки у яких дещо поступається.

Ключові слова: мова програмування, Golang, Go, розробка, простота.

Boiko L., Lishchina N. Studying the possibility of using golang as the first programming language in the educational process. This article highlights the advantages of using Golang as a first programming language. In particular, the article considers such aspects as simplicity and clarity of syntax, advantages of strict typing, language features that reduce the number of potential errors, the importance of predictability of code execution results, the ability to write programs in Go using different software development paradigms and the availability of additional information and an active community. In the article, Go has been compared with other programming languages (e.g., C++, JavaScript, PHP, and Python) in terms of its application in the educational process. Examples are given where Golang has significant advantages in terms of use in the process of training future specialists and cases where it is somewhat inferior.

Key words: programming language, Golang, Go, software development, simplicity.

Постановка проблеми. Сфера розробки програмного забезпечення – це динамічне середовище, у якому доволі часто відбуваються зміни у трендах та технологіях. Те, що ще донедавна могло вважатися актуальним, перестає бути таким із-за появи нових «гравців на ринку» або ж створення альтернатив існуючим інструментам розробки. Це у свою чергу призводить до необхідності актуалізації навчальної програми аби відповідати вимогам сьогодення та готувати спеціалістів, які будуть затребувані на ринку праці.

Аналіз останніх досліджень і публікацій. На основі аналізу робочих програм деяких закладів вищої освіти України можна зробити висновок, що здебільшого першою мовою програмування (далі МП) обирають C++ [1], що є цілком очікуваним з однієї сторони в силу певних «усталених традицій» вищої школи, з іншої – в силу практичності, адже протягом викладання курсу «Об'єктно-орієнтованого програмування» не потрібно буде вивчати ще одну МП.

Проте, варто зазначити що у якості першої МП вищі деякі навчальні заклади роблять інший вибір та обирають, наприклад, Object Pascal [2] або ж Visual Basic [3] що лише підкреслює різноманітність точок зору щодо цього питання.

Виділення невирішених раніше частин загальної проблеми. Динаміка з якою відбуваються зміни у сфері розробки програмного забезпечення, потребує перегляду усталених концепцій формування навчальних програм з метою надання студентам актуальних знань та інструментів щодо створення ПЗ.

Нині все більше компаній при розробці бекенд-рішень використовують мікро-сервісну архітектуру, для задач якої мова Golang (Go) пасує чи не найкраще, оскільки дозволяє у стислі терміни створювати надійні рішення.

Також варто пам'ятати, що питання вибору першої МП важливе у формуванні майбутньої кар'єри студента як розробника, оскільки занадто складна мова може призвести до втрати бажання вчитися, адже сам процес навчання не буде приносити задоволення. У той же час обрана МП повинна демонструвати сучасні можливості та готувати до виконання реальних задач.

Мета статті. Основною метою даної статті є висвітлення можливості та переваг застосування Golang у якості першої мови програмування для студентів.

Виклад основного матеріалу дослідження. На нашу думку, перша МП повинна мати наступні характеристики:

- *строга типізація.* Це необхідний критерій саме в процесі навчання аби студенти на практиці зрозуміли що таке «тип», як це працює та як це використовувати.
- *простий та зрозумілий для початківців синтаксис.* Наприклад, Pascal з його `if ... then ... else, begin ... end` конструкціями дозволяє дуже легко описувати початок та кінець відповідного блоку коду;
- *мінімізувати кількість можливих помилок.* На практиці, важливо (хоча і не завжди) аби у розробника було якомога більше можливостей та варіантів виконати завдання. Проте під час навчання занадто велика «свобода дій», яку надають деякі МП (наприклад, PHP) може зіграти злий жарт у майбутньому, зокрема призвести до відсутності навичок у написанні «чистого» та зрозумілого коду, що критично важливо на реальних проєктах. Хорошим прикладом, є Python у якому деякі аспекти оформлення коду по-суті є частиною синтаксису мови.
- *можливість розробки програм з використанням різних парадигм.* Наприклад, функціональне та процедурне програмування; розробка як одно- так і багатопотокових застосунків тощо. Це дозволяє вивчати різні підходи до написання коду без необхідності зміни МП.
- *наявність додаткової інформації та активної спільноти розробників.* У сучасному світі немає сенсу вивчати застарілі або вже не використовувані технології (приклад, **Pascal**). Натомість варто обирати те, що активно розвивається, має достатню кількість актуальної та якісної інформації.
- *підтримка сучасними інструментами розробки* (або наявність таких для конкретної МП) аби студенти відразу починали опановувати те з чим працюють діючі спеціалісти.
- *передбачуваність результатів виконання коду.* Безумовно, майбутній спеціаліст повинен розуміти внутрішні механізми МП, чому певні речі працюють так як вони працюють. Проте на початкових етапах навчання, на нашу думку, варто уникати таких ситуацій.

Так JavaScript знала своє «неоднозначною поведінкою». Наприклад:

- автоматичне приведення типів може призводити до неочікуваних результатів:

```
console.log(1 + "2"); // "12"  
console.log(1 - "2"); // -1  
console.log(('b' + 'a' + 'a' + 'a').toLowerCase()); // "banana"
```

- результат порівняння `null` та `0` може здивувати:

```
console.log(null == 0); // false  
console.log(null > 0); // false  
console.log(null >= 0); // true
```

- результат роботи вбудованого механізму сортування інколи дивує

```
console.log([10, 1, 3].sort()) // [1, 10, 3]
```

У контексті зазначених вище критеріїв розглянемо МП go, її переваги та недоліки.

Строга типізація.

У випадку з Go строгість типів, на перший погляд, виходить на новий рівень. Наприклад, якщо є мінлива типу `int32` і її значення необхідно присвоїти мінливій типу `int64`, то це буде неможливо зробити без приведення значення першої мінливої до типу `int64`.

```
var i32 int32 = 1  
var i64 int64 = 0  
i64 = i32 // cannot use i32 (variable of type int32) as type int64 in assignment  
.  
.  
.  
.  
.  
.  
var i32 int32 = 1  
var i64 int64 = 0  
i64 = int64(i32)
```

Простота та зрозумілість синтаксису.

Go має C-подібний синтаксис. Так, він не такий простий та виразний як у Pascal, проте зважаючи на те що більшість сучасних МП має саме C-подібний синтаксис (Dart, JavaScript, Rust, PHP та ін.) це не є суттєвим недоліком, а навіть навпаки може бути розцінено як перевага, оскільки дозволить швидше та легше опановувати інші мови програмування.

Іншою важливою характеристикою синтаксису Go є те, що у цій МП лише 25 [4] ключових слів. У той час як, наприклад, у Python їх 35 [5], JavaScript - 35-39 (залежно від режиму) [6], Java – 50 [7], PHP – 71 [8] та C++ 23 – 97 [9]. Така порівняно невелика кількість ключових слів дозволяє значно легше опанувати основи синтаксису та загалом швидше почати використовувати усі можливості МП на повну силу.

Мінімізація кількості можливих помилок.

Як зазначалося вище, чим більшу свободу дій надає МП тим краще, але у навчальному процесі замість того аби бути перевагою це може стати недоліком. І саме у Go є чудові вбудовані механізми які майже «змушують» писати «чистий код» та допомагають підтримувати цей код. Зокрема це:

- заборона на імпорт пакетів без їх подальшого використання. У разі якщо таке відбудеться, застосунок не буде скомпільовано:

```
package main
import "fmt" // пакет імпортовано, але він не використовується

func main() {
    _ = 1
}

//помилка компіляції
//"fmt" imported and not used
```

- заборона на оголошення мінливих без їх подальшого використання

```
package main
func main() {
    var i int // оголошення мінливої без наступного використання
}

//помилка компіляції
//i declared and not used
```

- заборона на циклічний імпорт пакетів. Наприклад якщо пакет `foo` імпортує пакет `bar`, а той у свою чергу імпортує пакет `foo`:

```
package demo
import demo/foo
import demo/bar
import demo/foo: import cycle not allowed
```

На перший погляд це може видатися суттєвим недоліком, проте насправді це вчить студента правильній структуризації коду та умінню планувати розробку програмного рішення.

Передбачуваність результатів виконання коду.

За рахунок строгої типізації, у Go неможливо здійснювати такі операції як, наприклад, у Python:

```
print("1" * 10) // у консоль буде виведено "1111111111"
```

чи PHP:

```
echo 10 * "2"; // у консоль буде виведено 20
```

За рахунок строгої типізації результати завжди передбачувані та, що важливо, очевидні: арифметичні операції виконуються лише з числовими типами, операції на стрічках з типами `string` або `[]byte` і т.д.

Можливість розробки програм з використанням різних парадигм.

Go дозволяє розробляти програмні застосунки використовуючи як процедурний підхід (коли код розбито на блоки, які послідовно викликаються), так і функціональний (передавати функції як значення та використовувати їх як аргументи). За потреби, на Go можна розглядати основи ООП, проте на нашу думку для вивчення цього розділу програмування, все ж ліпше використовувати “класичні рішення” на кшталт C++, Java чи навіть PHP.

Наявність додаткової інформації та активної спільноти розробників.

Згідно індексу ТІОБЕ станом на листопад 2024 [10], Go знаходиться на 7-му місці рейтингу по популярності серед інших МП. Також згідно індексу вподобання серед розробників на сайті dou.ua, Golang знаходиться у трійці лідерів [11].

Таке ставлення до мови серед спеціалістів призводить до того, що у мережі з'являється велика кількість матеріалів (як курсів так інформаційних статей) по Go, розробляються нові open-source рішень на Golang. Як наслідок компанії вирішують реалізовувати нові або переписувати старі проекти саме Go що у свою чергу збільшує попит на спеціалістів зі знанням цієї МП.

Підтримка сучасними інструментами розробки.

Чи не найкращим рішенням для розробки на Go є GoLand від компанії JetBrains. Дана IDE (integrated development environment) забезпечує не тільки зручну роботу з кодом (рефакторинг, автодоповнення, генерація коду та ін), а також містить досить об'ємний перелік допоміжних інструментів та функцій, серед яких можна виділити наступні:

- взаємодія із системами контролю версій (git, maven)
- робота з базами даних (PostgreSQL, MySQL, MongoDB та ін)
- підтримка роботи з Docker-контейнерами
- HTTP-клієнт та ін.

Також на Go можна писати у популярному безкоштовному редакторі VSCode. Для цього необхідно завантажити лише деякі розширення (плагіни), після чого можна розпочинати розробку на Go.

Висновки та перспективи. Як можна зрозуміти з наведеної вище інформації, Go є досить перспективним варіантом для застосування у якості першої МП.

Її простота дозволяє швидше зрозуміти основні поняття мови програмування такі як типи даних, мінливі та константи, цикли, функції тощо. За допомогою вбудованих механізмів та сторонніх інструментів (наприклад, golangci-lint) студенти можуть навчатися та привчатися створювати зрозумілий та підтримуваний код, що є важливою навичкою для майбутніх спеціалістів з розробки програмного забезпечення.

Тренд на зростання зацікавленості Golang не тільки зі сторони розробників, а й компаній, які вирішують використовувати цю мову для створення програмних продуктів вказує на те, що студенти із досвідом розробки на Go матимуть високі шанси на знаходження роботи за фахом. А це, у свою чергу, матиме виключно позитивний вплив на репутацію навчального закладу зокрема та економіку країни загалом.

Проте перед тим як здійснювати зміни у навчальній програмі щодо переходу на іншу мову програмування, варто розуміти той факт що це може бути доволі трудомістким процесом. Для викладацького складу це буде означати необхідність повного оновлення навчальних матеріалів: лекцій, практичних та/або лабораторних робіт, тестів тощо. Окрім того, викладання нової мови вимагає переосмислення підходу до пояснення базових концепцій, оскільки різні мови мають свої особливості та парадигми.

Список бібліографічного опису.

1. Горчинський С., Борисов Д. Обґрунтування вибору мови програмування для початкових курсів програмування. *Вісник Національного університету "Чернігівський колегіум" імені Т. Г. Шевченка*. 2024. Т. 180, № 24. С. 100–108.
2. Робоча програма навчальної дисципліни «Інформатика та програмування». Ужгородський національний університет. URL: <https://www.uzhnu.edu.ua/en/infocentre/get/31334>
3. Робоча програма навчальної дисципліни «Основи програмування». Ізмаїльський державний гуманітарний університет. URL: http://idgu.edu.ua/wp-content/uploads/2019/02/dvv_osnovy_prohramuvannja_abrosimov.pdf
4. The Go Programming Language Specification. URL: <https://go.dev/ref/spec#Keywords>
5. The Python Language Reference. URL: https://docs.python.org/3.8/reference/lexical_analysis.html#keywords
6. JavaScript reference. Lexical grammar. URL: <http://surl.li/uknjlv>
7. Java Language Keywords. URL: https://docs.oracle.com/javase/tutorial/java/nutsandbolts/_keywords.html
8. PHP Manual. List of Keywords. URL: <https://www.php.net/manual/en/reserved.keywords.php>
9. C++ keywords. URL: <https://en.cppreference.com/w/cpp/keyword>
10. ТІОБЕ Index for November 2024. URL: <https://www.tiobe.com/tiobe-index/>
11. Рейтинг мов програмування 2024. URL <https://dou.ua/lenta/articles/language-rating-2024/>

References

1. Horchynskyi S., Borysov D. Rationale for choosing a programming language for initial programming courses. *Visnyk Natsionalnoho universytetu "Chernihivskiy kolehium" imeni T. H. Shevchenka*. 2024. V. 180, № 24. S. 100–108.
2. Robocha prohrama navchalno dystsypliny «Informatyka ta prohramuvannia». Uzhhorodskiy natsionalnyi universytet. URL: <https://www.uzhnu.edu.ua/en/infocentre/get/31334>

3. Robocha prohrama navchalno dystsypliny «Osnovy prohramuvannia». Izmail'skyi derzhavnyi humanitarnyi universytet. URL: http://idgu.edu.ua/wp-content/uploads/2019/02/dvv_osnovy_prohramuvannja_abrosimov.pdf
4. The Go Programming Language Specification. URL: <https://go.dev/ref/spec#Keywords>
5. The Python Language Reference. URL: https://docs.python.org/3.8/reference/lexical_analysis.html#keywords
6. JavaScript reference. Lexical grammar. URL: <http://surl.li/uknjlv>
7. Java Language Keywords. URL: https://docs.oracle.com/javase/tutorial/java/nutsandbolts/_keywords.html
8. PHP Manual. List of Keywords. URL: <https://www.php.net/manual/en/reserved.keywords.php>
9. C++ keywords. URL: <https://en.cppreference.com/w/cpp/keyword>
10. TIOBE Index for November 2024. URL: <https://www.tiobe.com/tiobe-index/>
11. Reitynh mov prohramuvannia 2024. URL <https://dou.ua/lenta/articles/language-rating-2024/>