

Обзор языка программирования Go

Ленкин Алексей Викторович

*Приамурский государственный университет имени Шолом-Алейхема
Студент*

Аннотация

В данной статье описан новый язык программирования Go, выделены его преимущества и недостатки, описаны сферы применения и возможности.

Ключевые слова: Go, Google, статически типизированный, компилируемый

Overview of the Go Programming Language

Lenkin Aleksei Viktorovich

*Sholom-Aleichem Priamursky State University
Student*

Abstract

This article describes the new language capacity Go, highlights its advantages and disadvantages, describes the scope of application and capabilities.

Keywords: Go, Google, statically typed, compiled

На сегодняшний день активно начинают активно использоваться и развиваться новые языки программирования. Это связано с тем, что многим пользователям кажутся сложными старые языки программирования либо им недостаточно функционала в них. Также существует огромное число сфер их применения, одной из таких является создание веб-сервисов и клиент-серверных приложений, для этих нужд существует язык программирования Go.

Go представляет компилируемый статически типизированный язык программирования от компании Google. Язык Go предназначен для создания различного рода приложений, но прежде всего это веб-сервисы и клиент-серверные приложения [1].

Исследованиями в данной теме занимались следующие авторы. Ю.В. Адаменко описал подходы к обучению студентов основам языка программирования Go. Язык программирования Go был рассмотрена у И. Орещенкова [3]. Т.Р. Файзрахманов осуществил разработку комплекса инструментов для создания кросс-платформенных и кросс-браузерных приложений на основе веб-технологий и языка Go [4].

Первоначальная разработка Go началась в компании Google в 2007 году, официально его представили в 2009. Был создан для создания высокоэффективных программ. Существует мнение, что он был разработан как попытка заменить C и C++, но, по словам одного из разработчиков Роба

Пайка, язык Go был создан для решения реальных проблем при разработке программного обеспечения Google [5].

Приведем пример простейшей программы на языке Go.

```
packagemain
import"fmt"

funcmain() {
    fmt.Println("HelloWorld!")
}
```

Так как язык Go не является объектно-ориентированным, здесь нет классов, поэтому программа на нем определяется в виде пакетов. В данном случае пакет «main» (packagemain). Причем пакет должен называться именно «main», так как именно данный пакет определяет исполняемый файл. Для вывода «HelloWorld!» функцией (fmt.Println), необходимо подключить пакет, в котором она определена (import "fmt").

Основные преимущества языка Go[6]:

1. Простой и понятный синтаксис. Большинство функций языка написано на значащем английском слове. Это упрощает понимание кода. Также язык был разработан с низким порогом вхождения, что позволяет быстро внедряться в разработку проектов на нём достаточно быстро.

2. Статическая типизация. Означает, что функция связывается с типом в момент объявления и не может измениться позже. Это позволяет избежать ошибок при написании кода, так как приходится обдумывать конструкцию функции заранее.

3. Скорость и компиляция. Скорость у Go в десятки раз быстрее, чем у скриптовых языков, при меньшем потреблении памяти. При этом, компиляция практически мгновенна. После компиляции вся программа со всеми библиотеками собирается в один файл, который позже можно запускать без компилятора.

4. Отход от ООП. В языке нет классов, но есть структуры данных с методами. Наследование заменяется механизмом встраивания.

5. Параллелизм. Параллельные вычисления в языке делаются просто, для этого используется технология Goroutines, которая работает по аналогии с многопоточностью.

На данный момент не происходит значительного увеличения частоты процессора из-за отсутствия технологических прорывов в этой области уже почти десять лет и частым проявлением квантовых свойств объекта на больших частотах. Производители придумали как увеличить мощность процессора обходя эти ограничения добавлять больше ядер в процессор, открыв начало гиперпоточности. Поэтому, так как большинство языков было создано ещё в момент однопоточности, они неэффективно используют эти возможности. Но язык Go был создан уже с учетом многозадачности и использует её почти максимально в технологии Goroutines (схематично изображено на рисунке 1).



Рисунок 1. Использование Goroutines

6. Богатая стандартная библиотека. В языке присутствует огромное число функций, поэтому возможна разработка любых проектов разной сложности, как веб-сервисов, так и клиент-серверных приложений. Возможно использование библиотек из других языков, таких как С и С++.

7. Возможность писать в функциональном стиле. В языке есть замыкания (closures) и анонимные функции. Функции являются объектами первого порядка, их можно передавать в качестве аргументов и использовать в качестве типов данных.

8. Авторитетные отцы-основатели и сильное комьюнити. Роб Пайк, Кен Томпсон, Роберт Гризмер стояли у истоков. Сейчас у языка более 300 контрибьюторов. Язык имеет сильное сообщество и постоянно развивается. Активно продвигается компанией Google, также используется у Abode, BBC, IBM, Intel и Medium [7].

9. OpenSource. Означает, что исходный код языка открыт и доступен для модификации или просмотра, это позволяет убедиться в отсутствии уязвимостей или неприемлимого функционала.

Так как компания Google вкладывает огромные силы и средства в разработку языка Go, то у него отсутствуют существенные минусы, но некоторые всё же есть [8]:

1. Малая распространенность языка. Из-за того, что до Go уже существовали языки с похожим функционалом, то не каждый стремится изучать его, ведь есть более распространенные.

2. Излишняя легкость и низкий порог вхождения. Означают, что при дальнейшем развитии количество программистов на Go будет расти, что приведет к появлению большого числа плохого кода, так как большинство будет искать в нём лишь прибыль.

Таким образом, можно сказать, что язык программирования Go является очень перспективным для изучения и дальнейшей разработки на нём, так как очень легок в освоении и имеет мощную функциональную базу. Но ко всему прочему следует помнить, что необходимо тщательней писать свой код, чтобы быть действительно хорошим специалистом на нём.

Библиографический список

1. Введение в язык Go [Электронный ресурс] URL:

- <https://metanit.com/go/tutorial/1.1.php>(дата обращения 03.09.2018)
2. Адаменко Ю.В. Подходы к обучению студентов основам языка программирования Go // В сборнике: наука XXI века: технологии, управление, безопасность. Сборник материалов I международной научно-практической конференции. 2017. С. 514-520.
 3. Орещенков И. Язык программирования Go. Подготовка среды для разработки под Windows // Системный администратор. 2015. № 4 (149). С. 58-62.
 4. Файзрахманов Т.Р. Разработка комплекса инструментов для создания кросс-платформенных и кросс-браузерных приложений на основе веб-технологий и языка Go // В сборнике: Информатика, математическое моделирование, экономика сборник научных статей по итогам пятой международной научно-практической конференции. 2015. С. 343-347.
 5. Go at Google: Language Design in the Service of Software Engineering [Электронный ресурс] URL: <https://talks.golang.org/2012/splash.article> (дата обращения 03.09.2018)
 6. Язык Go для начинающих [Электронный ресурс] URL: <https://habr.com/post/219459/> (дата обращения 03.09.2018)
 7. Почему вам стоит изучить Go? [Электронный ресурс] URL: <https://habr.com/post/326798/> (дата обращения 03.09.2018)
 8. Язык программирования Go: мнения и перспектива [Электронный ресурс] URL: <http://timeweb.com/ru/community/articles/yazyk-programmirovaniya-go-mneniya-i-perspektiva-1> (дата обращения 03.09.2018)