***Go: Что если мы создадим C второй раз?***

**004**

Чугаев А. В.

*Данная статья посвящена объяснению шутки про язык программирования Go: «Что если мы создадим Си второй раз?» В данной статье рассматриваются языки программирования Go и Си, основные концепции, на которые опирались при создании первого, а также их сходства и различия, в следствии которых появилась эта шутка.*

Ключевые слова: Go, C, C++, Go lang, программирование, язык программирования

Создатели Go позиционируют свое детище как системный язык, сочетающий в себе эффективность и скорость исполнения кода, написанного на Си, с простотой разработки на более высокоуровневых скриптовых языках, да еще и со встроенными средствами параллельного программирования. При этом внешне Go напоминает какую-то странную солянку из синтаксисов языков Си, Pascal и ADA. Он может рассматриваться как попытка создать замену языкам [Си](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8_(%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)) и [C++](https://ru.wikipedia.org/wiki/C%2B%2B)[[1]](#_Интернет-ресурсы:).

В основу Go положено три фундаментальных идеи:

1. Гарантия высокой скорости компиляции и производительности приложений.
2. Простота разработки и поддержки приложений, свойственная высокоуровневым скриптовым языкам.
3. Встроенные средства параллельного программирования, позволяющие задействовать все имеющиеся ядра современных процессоров.

# Основные преимущества языка Go[[2]](#_Интернет-ресурсы:):

* Простой и понятный синтаксис. Это делает написание кода приятным занятием.
* Статическая типизация. Позволяет избежать ошибок, допущенных по невнимательности, упрощает чтение и понимание кода, делает код однозначным.
* Скорость и компиляция. Скорость у Go в десятки раз быстрее, чем у скриптовых языков, при меньшем потреблении памяти. При этом, компиляция практически мгновенна. Весь проект компилируется в один бинарный файл, без зависимостей. Как говорится, «просто добавь воды». И вам не надо заботиться о памяти, есть сборщик мусора.
* Отход от ООП. В языке нет классов, но есть структуры данных с методами. Наследование заменяется механизмом встраивания. Существуют интерфейсы, которые не нужно явно имплементировать, а лишь достаточно реализовать методы интерфейса.
* Параллелизм. Параллельные вычисления в языке делаются просто, изящно и без головной боли. Горутины (что-то типа потоков) легковесны, потребляют мало памяти.
* Богатая стандартная библиотека. В языке есть все необходимое для веб-разработки и не только. Количество сторонних библиотек постоянно растет. Кроме того, есть возможность использовать библиотеки C и C++.
* 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.
* 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.
* Open Source

|  |
| --- |
| **package** main  */\* Импорт \*/*  **import** (  "fmt" *// Стандартный пакет для форматированного вывода*  "database/sql" *// Импорт вложенного пакета*  w "os" *// Импорт с псевдонимом*  . "math" *// Импорт без квалификации при использовании*  \_ "gopkg.in/goracle.v2" *// Пакет не имеет явных обращений в коде*  )  **func** main() {  **for** \_, arg := **range** w.Args { *// Обращение к массиву Args, объявленному в пакете "os", через псевдоним*  fmt.Println(arg) *// Обращение к функции Println(), объявленной в пакете "fmt", с именем пакета*  }  db \*sql.DB = sql.Open(driver, dataSource) *// Имена из вложенного пакета квалифицируются*  *// только именем самого пакета (sql)*  x := Sin(1.0) *// вызов math.Sin() - квалификация именем пакета math не нужна,*  *// так как он импортирован без имени*  *// Обращений к пакету "goracle.v2" в коде нет, но он будет импортирован.*  }  Пример кода на Go[[3]](#_Интернет-ресурсы:) |
| #include *<iostream> // для использования std::cout*  #include *<vector> // содержит динамический массив*  #include *<map> // содержит тип данных словарь*  #include *<string>*  int main()  {  *// импортируем все объявления в пространстве имён "std" в глобальное пространство имён.*  **using** **namespace** std;  *// Объявляем ассоциативный контейнер со строковыми ключами и данными в виде векторов строк*  map< string, vector<string> > items;  *// Добавим в этот ассоциативный контейнер пару человек и дадим им несколько предметов*  items["Anya"].push\_back("scarf");  items["Dmitry"].push\_back("tickets");  items["Anya"].push\_back("puppy");  *// Переберём все объекты в контейнере*  **for**(**const** **auto**& person : items) {  *// person - это пара двух объектов: person.first - это его имя,*  *// person.second - это список его предметов (вектор строк)*  cout << person.first << " is carrying " << person.second.size() << " items" << endl;  }  }  Пример кода на C++[[4]](#_Интернет-ресурсы:) |

Табица 1. Сравнение языков Go и C++

# Сравнение Go с другими языками:

На первый взгляд листинг очень напоминает код программы, написанной на языке Си, C++ или даже Java[[1]](#_Интернет-ресурсы:), но при более детальном изучении становятся видны различия — Go унаследовал от Си только базовый синтаксис, в то время как большинство ключевых слов и лексика изменились. Исходный код начинается с ключевого слова package, следом за которым идет имя пакета, к которому этот код относится. Все запускаемые пользователем программы должны иметь имя main, тогда как библиотеки могут иметь произвольное имя, которое будет использовано для доступа к ее функциям и переменным после импортирования. При этом для пометки, должна ли функция или переменная быть экспортируемой, используется верхний регистр: все объекты, имена которых начинаются с большой буквы, будут экспортированы, остальные останутся приватными.

Несмотря эти отличия, язык Go является Си-подобным и обычно воспринимается как Си с измененным синтаксисом, а не полноценная замена ему.

# Источники:

### Интернет-ресурсы:

1. [Журнал «Хакер»](https://xakep.ru/author/jonsnow/): [Электронный ресурс]. 23.05.2011. URL: <https://xakep.ru/2011/05/23/55752/> (Дата обращения: 30.10.2019).
2. «Хабр»: [Электронный ресурс]. 23.05.2011. URL: <https://habr.com/ru/post/219459/> (Дата обращения: 30.10.2019).
3. Материал из Википедии — свободной энциклопедии: [Электронный ресурс]. URL: <https://ru.wikipedia.org/wiki/Go> (Дата обращения: 30.10.2019).
4. Материал из Википедии — свободной энциклопедии: [Электронный ресурс]. URL: <https://ru.wikipedia.org/wiki/C%2B%2B> (Дата обращения: 30.10.2019).