

Почему golang?

Василий Романов

v.romanov@corp.mail.ru

telegram: rvasily

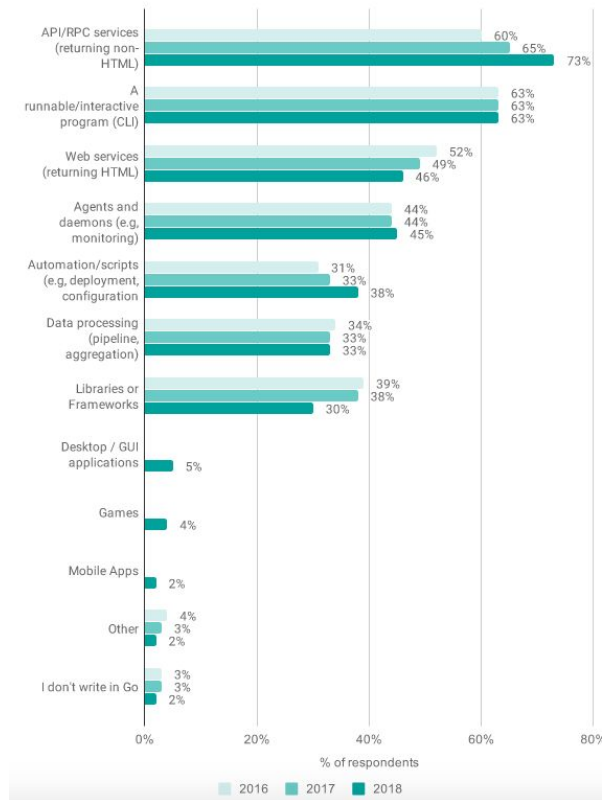
Включить скринкаст

Организационное

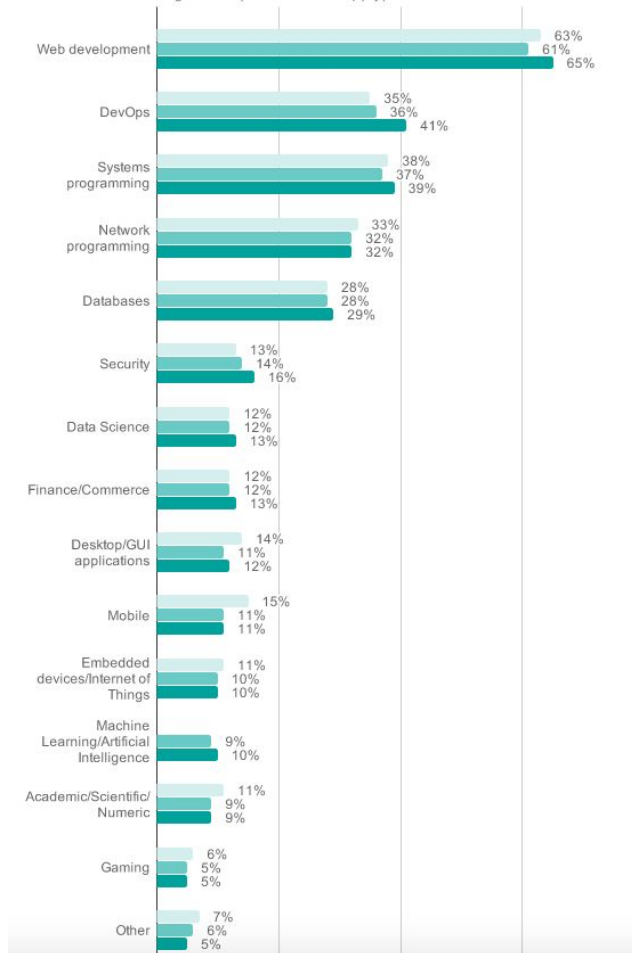
- Это факультатив, ваше решение ходить на него
- Можно просто ходить и не делать домашки
- Можно не делать домашку если она вам не нравится
- Если вы делаете домашку - делайте сами
- За списывание отчисляем с курса



Сферы применения



I work in the following areas: (select all that apply)



Кто использует

Mail.ru Group

- Почта
- Реклама
- Юла
- Delivery Club
- Облако
- Внутренняя разработка
- Медиапроекты
- Процессинг
- VK
- Ситимобил

Кто использует

Mail.ru Group

- Почта
- Реклама
- Юла
- Delivery Club
- Облако
- Внутренняя разработка
- Медиапроекты
- Процессинг
- VK
- Ситимобил

- Яндекс (Яндекс.Еда)
- Ozon
- Avito
- Badoo
- Lamoda
- Wildberries
- Gett
- Wheely
- ivi
- Tutu
- ДомКлик (Сбербанк)
- Ростелеком
- МТС
- Альфа-банк
- Тинькофф

Краткая история golang

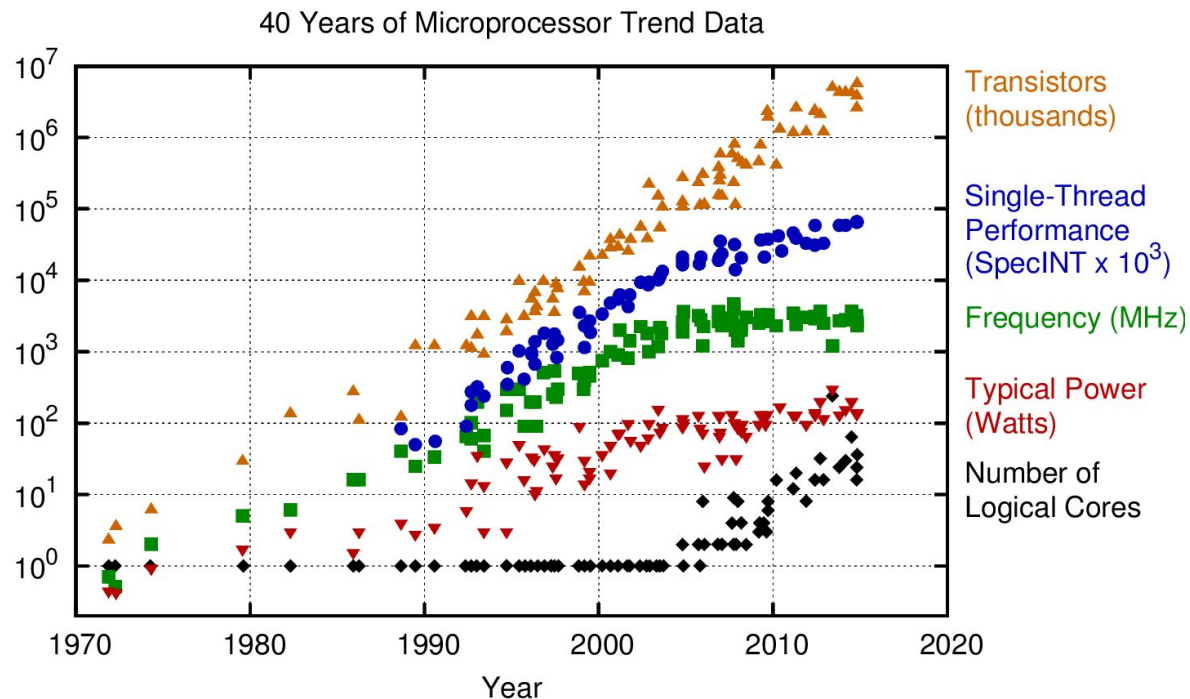
Разрабатывался ветеранами индустрии

- Кен Томпсон (UNIX, UTF-8)
- Роб Пайк (UTF-8, Plan 9, Inferno)
- Роберт Гризмер (Java HotSpot, Sawzall, распределённые системы Google)

В условиях большой компании:

- Много кода
- Много программистов
- Много серверов (а на них много ядер)
- И есть легаси код

Про сервера и ядра



Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2015 by K. Rupp

Реалии разработки*

- В целом, меньше кода проще для понимания чем больше кода. Однако, есть грань, когда код настолько минималистичен, что его трудно читать
- Код читается чаще чем пишется
- Код часто живёт дольше чем мы предполагаем
- Человек, который тестирует или поддерживает код чаще всего не его первоначальный автор
- В масштабе, уровень разработчика, который читает/пишет/поддерживает/тестирует код имеет нормальное распределение со средним "не эксперт"

* <https://www.quora.com/Scala-vs-Go-Could-people-help-compare-contrast-these-on-relative-merits-demerits>

Много кода, много программистов, legacy

- Простой и компактный синтаксис
- Мало магии и синтаксического сахара
 - Нет даже тернарного оператора
- Ориентация на простоту и читабельность кода
- Жесткий стиль кода и инструмент для авто-форматирования
- Множество синтаксических анализаторов

Много кода, много программистов, legacy

- Быстрая компиляция
 - Веб-апи почты собирается за 3 мин
 - Микросервисы за 1 мин
- Тесты из коробки
 - Вместе с покрытием и отчетом
- Профилировщик из коробки

Много серверов

- Статический бинарь *
- Нет dependency hell
- Удобная работа с зависимостями
- Просто кладём их в репозиторий
- Кросс-компиляция *

* До тех пор пока вы не используете sgo

Много ядер в процессоре

- Асинхронный i/o на уровне языка (по модели CSP)
 - Нет callback-hell
 - Весь код стандартной библиотеки и внешних либ тоже!
- Приложение масштабируется на все ядра процессора

10 кубиков программиста

- Бизнесу не нужно самовыражение программиста и его игры в песочнице

язык	язык	язык	магия	рантайм	стиль	отладка	задача	задача	задача
------	------	------	-------	---------	-------	---------	--------	--------	--------

- Бизнесу нужен продукт
- На го скучно заниматься чем-то кроме продукта
 - Потому что там не так прикольно делать всякие классные штуки с новым синтаксисом

язык	язык	язык	магия	задача	задача	задача	задача	задача	задача
------	------	------	-------	--------	--------	--------	--------	--------	--------

