

Занятие 1

# Программирование на Golang

Василий Романов

# Пожалуйста, отметьтесь на лекции

---



# Давайте знакомиться

---



Василий  
Романов



Илья  
Ожерельев



Дмитрий  
Дорофеев

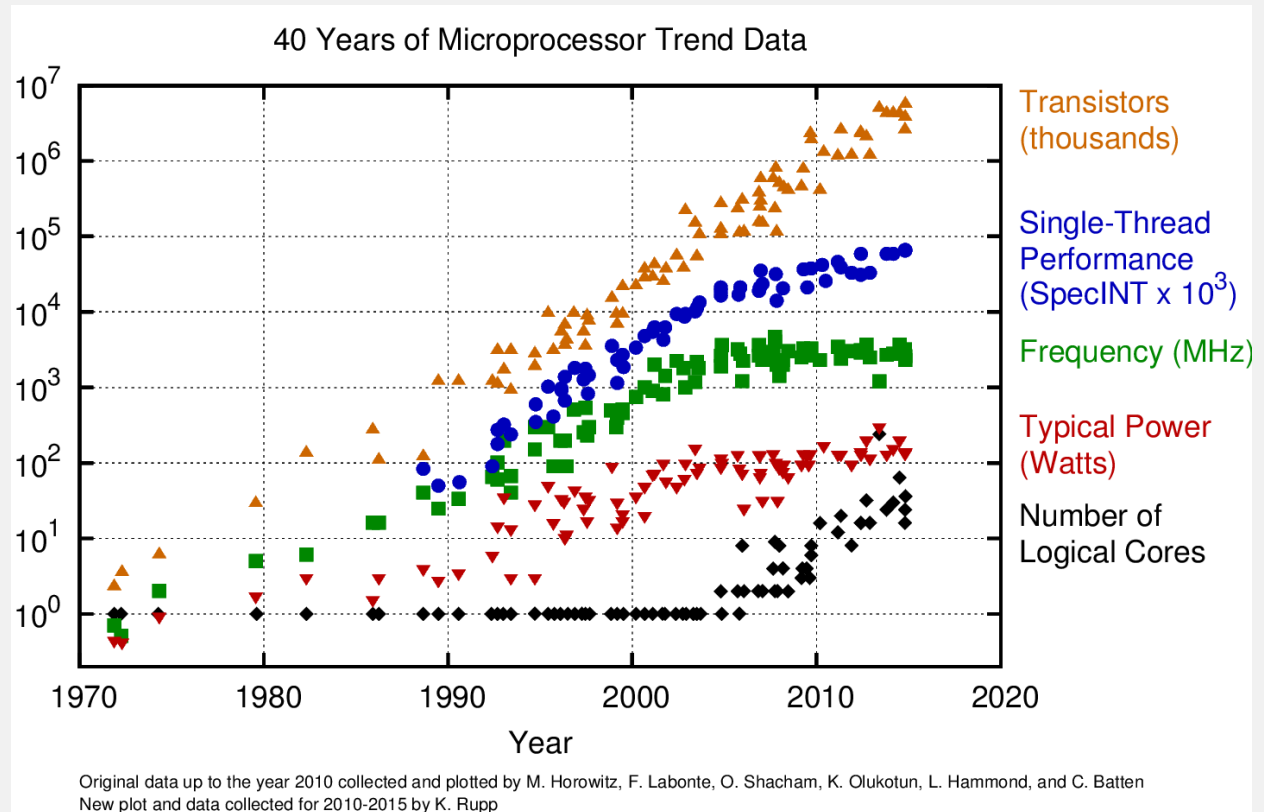


1. Вести с полей современного ИТ
2. История Golang и его ключевые идеи
3. Основы языка
  1. Установка и настройка
  2. Первая программа
  3. Переменные
  4. Управляющие структуры

# Зачем нужен ещё один язык программирования?



1972 – C  
1980 – C++  
1987 – Perl  
1987 – Erlang  
1991 – Python  
1993 – Ruby  
1993 – Lua  
1995 – PHP  
1995 – Java  
1995 – JavaScript  
2001 – C#  
2003 – Scala  
2009 – Go  
2010 – Rust  
2014 – Swift





1. 2007 - Начало разработки
  1. Кен Томпсон ( UNIX, UTF-8 )
  2. Роб Пайк ( UTF-8, Plan 9, Inferno )
  3. Роберт Гризмер (Java HotSpot, Sawzall, распределённые системы Google )
2. 2009 – Первая доступная версия
3. 2012 – Версия 1.0
4. 2014 – Первый сервис в Почте@Mail.ru
5. 2016 – Первое API в Почте@Mail.ru



- Компилируемый язык
  - Нет накладных расходов на интерпретатор
- Сборщик мусора
  - Есть пауза на сборку мусора
- CSP-style асинхронная модель
  - Хорошо подходит для сетевых сервисов
- Сравнение производительности
  - Веб-сервер - <https://habr.ru/company/mailru/blog/273341/>
  - Числодробилка - <http://benchmarksgame.alioth.debian.org>



# Ключевые идеи

---



- Простота
- Читабельность
- Конкурентность (concurrency)





- Код читается больше чем пишется
- Код часто живёт дольше чем мы думаем
- Если вы занимаетесь поддержкой кода – большая вероятность что его автор кто-то другой
- Со временем медиана работающего над проектам человека останавливается на уровне «средний»
- Сложность языка: 85 страниц спецификации + 55 страниц основного тайного знания

1. C++ - 1300 страниц
2. Java – 780 страниц
3. C – 550 страниц
4. ECMA-262 – 580 страниц (мелким шрифтом)
5. Scala – 191 страница

# Читабельность



Жестко зафиксированный стиль кода

Инструмент для авто-форматирования ( `go fmt` )

Мало магии и синтаксического сахара

# Конкурентность ( concurrency )



20xx – эпоха бурного развития интернета

Бурный рост = много работы на сервере

Много работы на сервере = много походов по сети\*

Много походов по сети = много ждем, время ожидания сильно больше времени работы

\* 8 запросов во внешние хранилища в самом частом запросе Почты

# Установка и настройка



## Windows

Next – Next – Next – Finish - Relogin

## Linux

```
wget --no-check-certificate https://storage.googleapis.com/golang/go1.8.linux-amd64.tar.gz  
tar -C /usr/local -xzf go1.8.linux-amd64.tar.gz  
echo 'export PATH=$PATH:/usr/local/go/bin  
export GOPATH=$HOME/GO  
export PATH=$PATH:$GOPATH/bin' >> /etc/profile
```

# Hello World



```
package main

import "fmt"

func main() {
    fmt.Println("Hello, World")
}
```

go run main.go

# Типы данных – базовые переменные

---



Int & co

Double, float

Bool

String

Array

Slice

Map

# Управляющие структуры

---



If

For

Switch

Select

Range

✦Slice

✦Array

✦Map

# Система пакетов



Пакет – это папка

Разные файлы в папке – один пакет

Импорт работает на уровне файла



# Импорт внешних зависимостей



Установка\*:

```
go get -u https://github.com/username/reponame
```

Требуется установленный git

Использование:

```
import "github.com/rvasily/exemplerepo"  
Name, exist := exemplerepo.GiveName(map[string][string]{“FirstName”: “Vasily”})
```

\* Требуется установленный git

# Структура проекта – default GOPATH



```
$GOPATH/
  bin/
    hello          # command executable
    outyet         # command executable
  pkg/
    linux_amd64/
      github.com/golang/example/
        stringutil.a    # package object
  src/
    github.com/golang/example/
      .git/             # Git repository metadata
    hello/
      hello.go          # command source
    outyet/
      main.go           # command source
      main_test.go      # test source
    stringutil/
      reverse.go        # package source
      reverse_test.go   # test source
```

# Обработка ошибок

---



Исключений ( Exceptions ) нет  
Ошибки обрабатываются явно

# Контроль за качеством кода

---



Go fmt

Go vet

Go doc

# Материалы для ознакомления

---



<https://golang.org/ref/spec>

<https://golang.org/doc/code.html>

<https://golang.org/cmd/go/>

<https://blog.golang.org/strings>

<https://blog.golang.org/slices>

<https://blog.golang.org/go-slices-usage-and-internals>

<https://blog.golang.org/go-maps-in-action>

<https://blog.golang.org/organizing-go-code>



# Код, презентация, канал в Telegram

---



<https://gitlab.com/rvasily/msu-go-11>

<https://t.me/joinchat/AAAAAAkdkEa-k2OvTYCnEA>



# Домашнее задание № 1

---



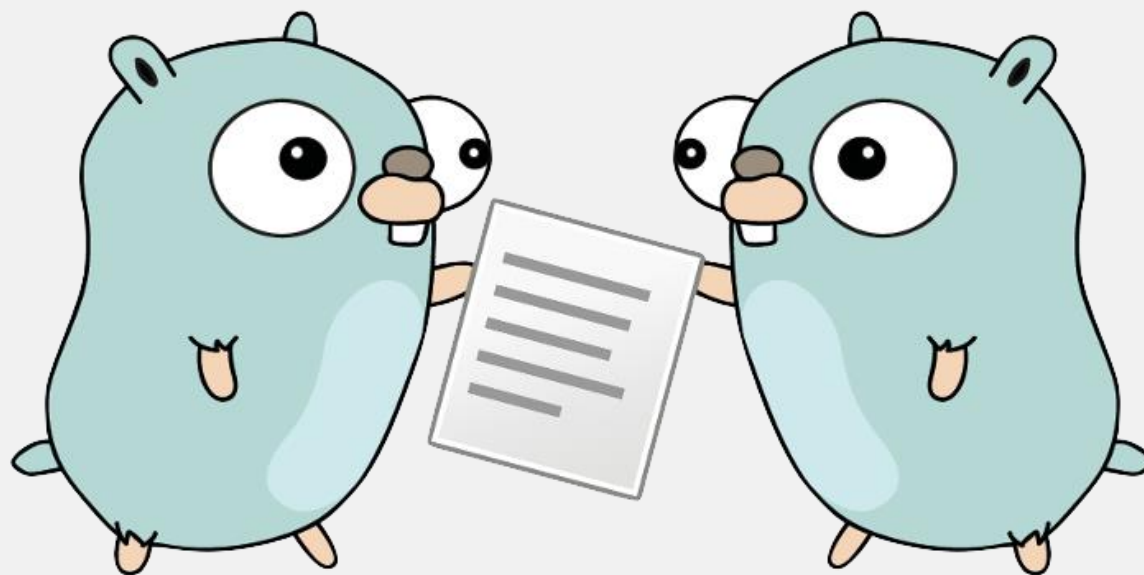
- Пройти все тесты в [https://gitlab.com/rvasily/msu-go-11/tree/master/1/99\\_homework](https://gitlab.com/rvasily/msu-go-11/tree/master/1/99_homework)

**Срок сдачи**

*1 апреля 2017*

# Оставьте нам обратную связь

---



Это поможет сделать курс лучше





**Спасибо за  
внимание!**