

Покоряем golang
Лекция 1. Знакомство

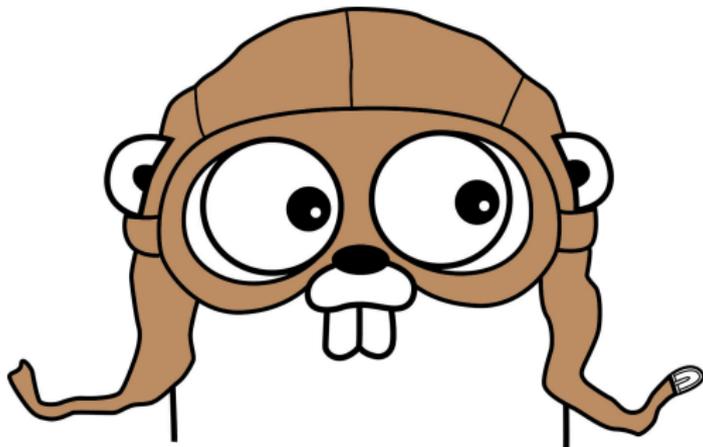
Григорий Базилевич

20 ноября 2021 года

Кто такой golang?

Golang — язык программирования с открытым исходным кодом, который позволяет легко делать простые, надежные и эффективные программы.

<https://golang.org/>



Экскурс в историю

Go начал свое существование в 2007 году в компании Google. Разработчики языка хотели собрать в языке как можно больше приятных характеристик:

Экскурс в историю

Go начал свое существование в 2007 году в компании Google. Разработчики языка хотели собрать в языке как можно больше приятных характеристик:

- ▶ Статическую типизацию и эффективность во время выполнения (как C / C++),

Экскурс в историю

Go начал свое существование в 2007 году в компании Google. Разработчики языка хотели собрать в языке как можно больше приятных характеристик:

- ▶ Статическую типизацию и эффективность во время выполнения (как C / C++),
- ▶ Читательность и удобство использование (как Python и JavaScript),

Экскурс в историю

Go начал свое существование в 2007 году в компании Google. Разработчики языка хотели собрать в языке как можно больше приятных характеристик:

- ▶ Статическую типизацию и эффективность во время выполнения (как C / C++),
- ▶ Читабельность и удобство использования (как Python и JavaScript),
- ▶ Высокую производительность в сетевых задачах и многопоточность.

Экскурс в историю

Go начал свое существование в 2007 году в компании Google. Разработчики языка хотели собрать в языке как можно больше приятных характеристик:

- ▶ Статическую типизацию и эффективность во время выполнения (как C / C++),
- ▶ Читабельность и удобство использования (как Python и JavaScript),
- ▶ Высокую производительность в сетевых задачах и многопоточность.

10 ноября 2009 года — анонс языка.

Экскурс в историю

Go начал свое существование в 2007 году в компании Google. Разработчики языка хотели собрать в языке как можно больше приятных характеристик:

- ▶ Статическую типизацию и эффективность во время выполнения (как C / C++),
- ▶ Читабельность и удобство использования (как Python и JavaScript),
- ▶ Высокую производительность в сетевых задачах и многопоточность.

10 ноября 2009 года — анонс языка.

28 марта 2012 года — выход версии 1.0.

Важные личности

Важные личности



Роберт Гриземер

Важные личности



Роберт Гриземер



Роберт Пайк

Важные личности



Роберт Гриземер



Роберт Пайк



Кен Томпсон

Чем же хорош этот ваш go lang?

Чем же хорош этот ваш golang?

1. Быстрая компиляция в бинарный код,

Чем же хорош этот ваш golang?

1. Быстрая компиляция в бинарный код,
2. статическая типизация и отсутствие неявных преобразований,

Чем же хорош этот ваш golang?

1. Быстрая компиляция в бинарный код,
2. статическая типизация и отсутствие неявных преобразований,
3. простой синтаксис и минимум ключевых слов,

Чем же хорош этот ваш golang?

1. Быстрая компиляция в бинарный код,
2. статическая типизация и отсутствие неявных преобразований,
3. простой синтаксис и минимум ключевых слов,
4. жесткий стиль кода и автоматическое форматирование,

Чем же хорош этот ваш golang?

1. Быстрая компиляция в бинарный код,
2. статическая типизация и отсутствие неявных преобразований,
3. простой синтаксис и минимум ключевых слов,
4. жесткий стиль кода и автоматическое форматирование,
5. встроенный сборщик мусора,

Чем же хорош этот ваш golang?

1. Быстрая компиляция в бинарный код,
2. статическая типизация и отсутствие неявных преобразований,
3. простой синтаксис и минимум ключевых слов,
4. жесткий стиль кода и автоматическое форматирование,
5. встроенный сборщик мусора,
6. асинхронность из коробки,

Чем же хорош этот ваш golang?

1. Быстрая компиляция в бинарный код,
2. статическая типизация и отсутствие неявных преобразований,
3. простой синтаксис и минимум ключевых слов,
4. жесткий стиль кода и автоматическое форматирование,
5. встроенный сборщик мусора,
6. асинхронность из коробки,
7. удобная и эффективная система управления зависимостями.

А все ли так хорошо?

А все ли так хорошо?

1. Отсутствуют обработчики исключений,

А все ли так хорошо?

1. Отсутствуют обработчики исключений,
2. нет прямого наследования и ООП в целом,

А все ли так хорошо?

1. Отсутствуют обработчики исключений,
2. нет прямого наследования и ООП в целом,
3. отсутствует возможность переопределять методы и функции,

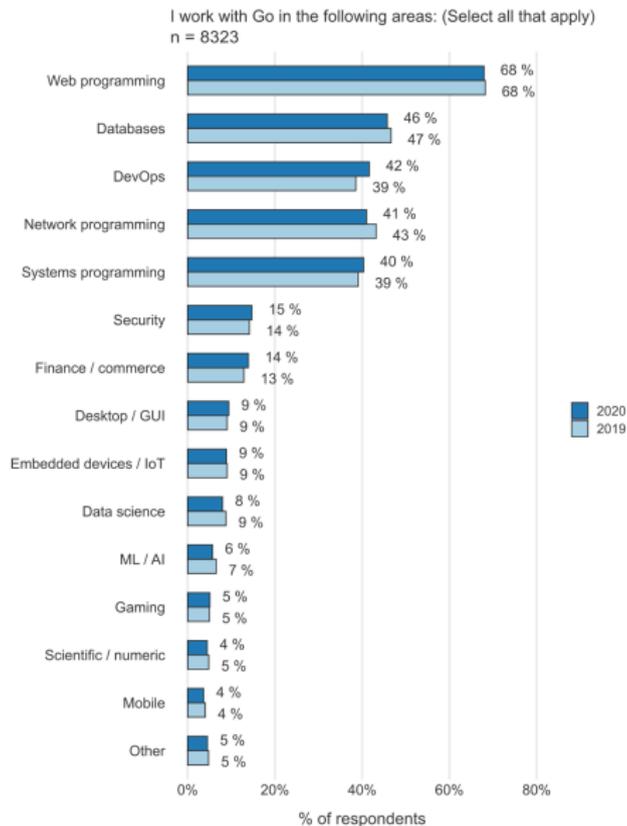
А все ли так хорошо?

1. Отсутствуют обработчики исключений,
2. нет прямого наследования и ООП в целом,
3. отсутствует возможность переопределять методы и функции,
4. нет отрицательных индексов,

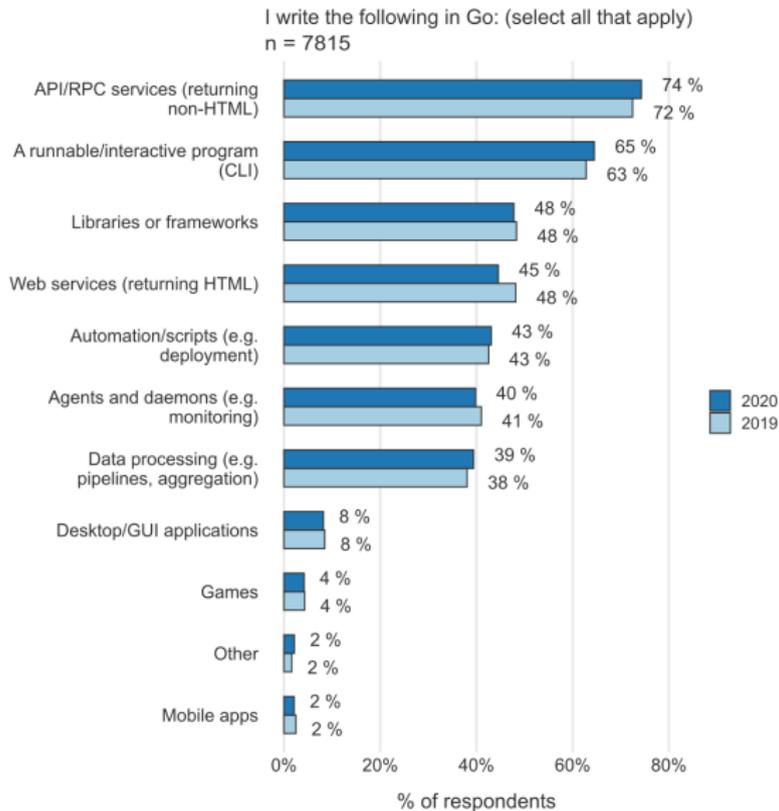
А все ли так хорошо?

1. Отсутствуют обработчики исключений,
2. нет прямого наследования и ООП в целом,
3. отсутствует возможность переопределять методы и функции,
4. нет отрицательных индексов,
5. неоднозначная стандартная библиотека.

Где применяют golang?



Что пишут на go lang?



Давайте поприветствуем мир!

```
package main
```

Давайте поприветствуем мир!

```
package main
```

```
import "fmt"
```

Давайте поприветствуем мир!

```
package main  
  
import "fmt"  
  
func main() {
```

Давайте поприветствуем мир!

```
package main

import "fmt"

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

Давайте поприветствуем мир!

```
package main

import "fmt"

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

<https://play.golang.org/> — golang playground.

<https://go.dev/> — скачать golang без смс и регистрации.

Что нас ждет?

В задумке этого курса звучали такие моменты:

- ▶ знакомство с синтаксисом языка,
- ▶ решение задачек на `golang`,
- ▶ лекции по инструментам языка (`go test`, `go bench`),

Что нас ждет?

В задумке этого курса звучали такие моменты:

- ▶ знакомство с синтаксисом языка,
- ▶ решение задачек на `golang`,
- ▶ лекции по инструментам языка (`go test`, `go bench`),
- ▶ лекция по `git`,

Что нас ждет?

В задумке этого курса звучали такие моменты:

- ▶ знакомство с синтаксисом языка,
- ▶ решение задачек на `golang`,
- ▶ лекции по инструментам языка (`go test`, `go bench`),
- ▶ лекция по `git`,
- ▶ работа в команде

Что нас ждет?

В задумке этого курса звучали такие моменты:

- ▶ знакомство с синтаксисом языка,
- ▶ решение задачек на golang,
- ▶ лекции по инструментам языка (go test, go bench),
- ▶ лекция по git,
- ▶ работа в команде с git

Что нас ждет?

В задумке этого курса звучали такие моменты:

- ▶ знакомство с синтаксисом языка,
- ▶ решение задачек на `golang`,
- ▶ лекции по инструментам языка (`go test`, `go bench`),
- ▶ лекция по `git`,
- ▶ работа в команде с `git` над финальным проектом.

Насколько это все воплотится в жизнь узнаем в ближайшем будущем.