

3. Golang приложения

Go модули

- <https://go.dev/doc/tutorial/create-module>
- <https://go.dev/ref/mod#introduction>
- легкие программы можно написать в main.go
 - может быть много кода
 - а если нужна какая-то сторонняя, не стандартная библиотека?
 - или много библиотек
 - надо как-то фиксировать зависимости
- код приложения обычно пишется в одном Go модуле
 - код внутри модуля структурируется в пакеты (package)
 - функции для работы с файлами – в отдельный пакет
 - функции для работы с БД – в отдельный пакет
 - package main – <https://go.dev/play/>
 - *A module is a collection of packages that are released, versioned, and distributed together.*

go.mod

- создается в результате `go mod init`
- название модуля, версия Go, версии зависимостей




The screenshot shows a GitHub code editor interface. At the top, there are tabs for 'Code' and 'Blame'. To the right of the tabs, it says '14 lines (11 loc) · 311 Bytes'. Further right, there is a GitHub Copilot logo and the text 'Code 55% faster with GitHub Copilot'. The main area displays the content of a `go.mod` file with line numbers 1 through 14 on the left. The code is as follows:

```
1  module goroscope
2
3  go 1.20
4
5  require (
6      github.com/go-telegram-bot-api/telegram-bot-api/v5 v5.5.1
7      github.com/joho/godotenv v1.5.1
8      github.com/redis/go-redis/v9 v9.0.3
9  )
10
11  require (
12      github.com/cespare/xxhash/v2 v2.2.0 // indirect
13      github.com/dgryski/go-rendezvous v0.0.0-20200823014737-9f7001d12a5f // indirect
14  )
```

go.sum

- создается в результате go mod tidy
- содержит названия, версии, хеш-суммы зависимостей

Code Blame 12 lines (12 loc) · 1.11 KB  Code 55% faster with GitHub Copilot

```
1  github.com/bsm/ginkgo/v2 v2.7.0 h1:ItPMPH90RbmZJt5GtkcNvIRuGEdwlBItdNVoyzaNQao=
2  github.com/bsm/gomega v1.26.0 h1:LhQm+AFcgV2M0WyKroMASzAzCAJVpAxQXv4SaI9a69Y=
3  github.com/cespare/xxhash/v2 v2.2.0 h1:DC2CZ1Ep5Y4k3ZQ899DldepgrayRUGE6BBZ/cd9Cj44=
4  github.com/cespare/xxhash/v2 v2.2.0/go.mod h1:VGX0DQ3Q6kWi7AoAeZDth3/j3BFt0ZR5XLFGcrjCOs=
5  github.com/dgryski/go-rendezvous v0.0.0-20200823014737-9f7001d12a5f h1:l04WD4F/rVNCu3HqELle0ji
6  github.com/dgryski/go-rendezvous v0.0.0-20200823014737-9f7001d12a5f/go.mod h1:cuUVRXasLTGF7a8h
7  github.com/go-telegram-bot-api/telegram-bot-api/v5 v5.5.1 h1:wG8n/XJQ07TmjbITcGiUaOtXxdrINDz1b
8  github.com/go-telegram-bot-api/telegram-bot-api/v5 v5.5.1/go.mod h1:A2S0CWkNylc2phvKXWBBdD3K0i
9  github.com/joho/godotenv v1.5.1 h1:7eLL/+HRGLY0ldzfGMeQkb7vMd0as4CFYvUVzLqw0N0=
10 github.com/joho/godotenv v1.5.1/go.mod h1:f4LDn5Voq0i2e/R5DDN0oa2zzDfwtkZa6DnEwAbqwq4=
11 github.com/redis/go-redis/v9 v9.0.3 h1:+7mmR26M0IvyLxGZUHxu4GiBkJkVDId0Un+j4ScYu4k=
12 github.com/redis/go-redis/v9 v9.0.3/go.mod h1:WqMKv5vnQbRuZstUwxQI195wHy+t4PuXD0jzMvcuQHk=
```

Импорт кода из другого пакета

- <https://go.dev/doc/tutorial/call-module-code>
- название модуля + путь до файла
- пример: <https://github.com/katevi/goroscope/tree/master>
- код в папке internal
 - https://pkg.go.dev/cmd/go#hdr-Internal_Directories
 - можно импортировать только из тех директорий, которые содержат эту папку (родительских директорий поддерева с internal папкой)
 - можно импортировать только из этого же модуля
- все, что нужно импортировать – с большой буквы (функции, структуры)

Стандартная библиотека

- <https://pkg.go.dev/std>
 - io
 - интерфейсы: Writer, Reader
 - io/fs – интерфейсы, описывающие работу с файлами
 - path/filepath – для работы с путями файлов
 - bufio
 - для I/O операций (ввода / вывода)
 - os
 - работа с файловой системой
 - работа с процессами ОС
 - работа с переменными окружения (ENV)
 - os/exec - для исполнения внешних команд (exe файлов)
 - strings, sort, slices...

bufio

- <https://pkg.go.dev/bufio@go1.24.1#pkg-overview>
- buffered I/O – input/output
 - Reader / Writer с буфером
 - Scanner – упрощенный Reader, который читает по токенам
- Scanner читает откуда угодно в буфер
 - <https://pkg.go.dev/bufio@go1.24.1#Scanner> – много примеров
 - из stdin (из консоли) <https://go.dev/play/p/fjmy9hxvAOI>
 - из массива байтов <https://go.dev/play/p/SGQ35l5cnzr>

OS

- <https://pkg.go.dev/os@go1.24.1>
- низкоуровневые операции с файлами
 - создание, открытие файла
 - удаление файла
 - выставление прав доступа к файлу/директории
 - создание директории
- модификация/чтение окружения, в котором запущена программа в целом
 - <https://go.dev/play/p/-Sb6fS6YT31>
 - https://go.dev/play/p/24_yYUmXq-l

io/fs, path/filepath

- io/fs - <https://pkg.go.dev/io/fs@go1.24.1#FS>
 - интерфейсы, описывающие работу с файлами
 - интерфейс ФС <https://pkg.go.dev/io/fs@go1.24.1#FS>
 - интерфейс файла <https://pkg.go.dev/io/fs@go1.24.1#File>
 - интерфейс информации о файле <https://pkg.go.dev/io/fs@go1.24.1#FileInfo>
- path/filepath <https://pkg.go.dev/path/filepath@go1.24.1>
 - для работы с путями файлов (абсолютный путь, относительный, ...)
- пример: <https://go.dev/play/p/DWPXEoV9C6O>
 - конкретную реализацию файла создали с помощью os
 - читаем с помощью абстракций из io/fs
- пример:
 - обход директории <https://go.dev/play/p/mezhmklUiz9>

strings, strconv

- <https://pkg.go.dev/strings>
 - умеет разделять строки по символам, склеивать строки, заменять символы в строках
 - и так далее
 - <https://go.dev/play/p/pRXu555SinI>
- <https://pkg.go.dev/strconv>
 - чтобы строки конвертировать в нужные типы
 - удобно, если есть какие-то входные данные (они обычно текстовые)
 - <https://go.dev/play/p/rAutFokZPou>
- если хочется сделать что-то со строками – надо посмотреть, нет ли этого в стандартном пакете

slices

- <https://pkg.go.dev/slices@go1.24.1>
- много разных функций
 - https://go.dev/play/p/dPmca_Y4Bpn
- если хочется сделать что-то со слайсом — надо посмотреть, нет ли этого в стандартном пакете

sort

- <https://pkg.go.dev/sort@go1.24.1>
- <https://go.dev/play/p/KFOHPBh9UDF>
- `sort.Slice` – сортирует слайсы любых типов
 - числа
 - строки
 - свои собственные структуры