1. Golang intro

Golang

- Изначально создавался как альтернатива C
  - решить проблему зависимостей
  - простой язык
  - сборка мусора
- Язык для написания инфраструктуры:
  - Docker, Kubernetes, Prometheus, VictoriaMetrics etc.
  - но сейчас везде и активно на нем пишут микросервисы
- Процедурный язык программирования
- Основные возможности:
  - строгая статическая типизация
  - есть указатели
  - конкуррентность
  - богатая стандартная библиотека
  - обработка ошибок
  - сборка мусора
  - можно вызывать C из Go (CGO)

Golang standard tools

Все базовые вещи из коробки:

- go version – версия Go
- go build – собрать приложение (бинарник)
- go run – собрать и запустить
- go test – запустить тесты на приложение
- go mod tidy – поставить зависимости приложения
- go fmt – отформатировать исходники приложения под стандартный формат
    - в Go стайлгайд один и контролируется средствами языка

Golang tools

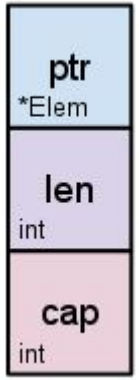
- go playground – онлайн редактор для Go
  - можно писать простые программы
  - можно делиться своими плейграундами
  - удобно, если нужно разобрать/показать кусочек кода
- golangci-lint
  - дополнительные проверки качества кода помимо go fmt
    - контроль длины функции
    - контроль
    - можно написать свой линтер
- стандартная библиотека предоставляет много всего:
  - даже пакет ast для построения синтаксического дерева программы

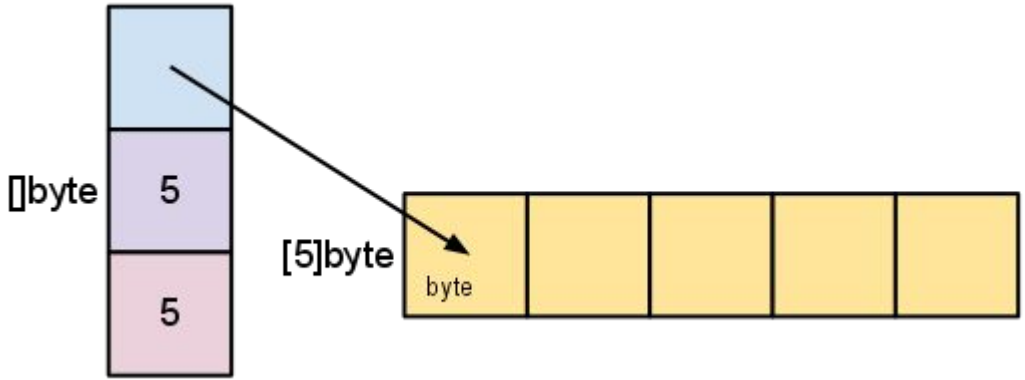
Базовые конструкции

- переменные: https://gobyexample.com/variables
  - var, :=
- константы: https://gobyexample.com/constants
  - const
- циклы: https://gobyexample.com/for
- if/else: https://gobyexample.com/if-else
- switch: https://gobyexample.com/switch
  - выполняет только сматчившуюся ветку case
- массивы: https://gobyexample.com/arrays

Слайсы

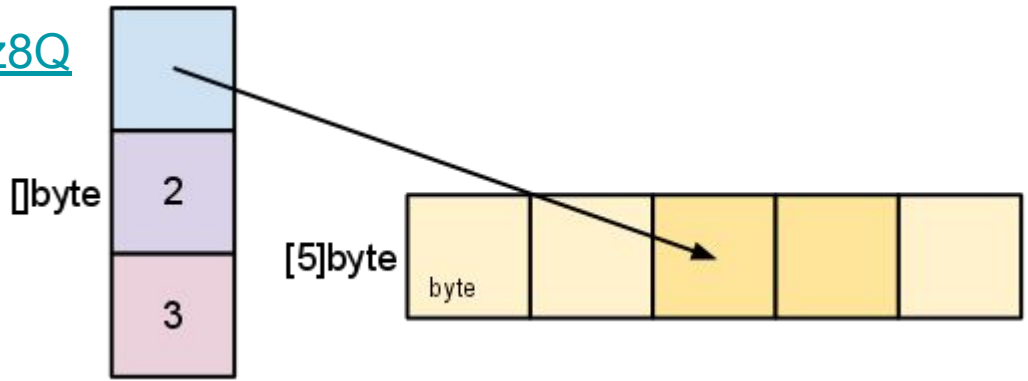
- динамически переаллоцируемый массив
  - структура из указателя на голову, len, cap
  - len – текущая длина слайса
  - cap – максимальная длина слайса (до переаллокации)
- слайсы можно создать:
  - через конструктор `func make([]T, len, cap) []T`
  - или объявить явно
  - `make([]byte, 5)`





Операции со слайсами

- можно взять элемент по индексу
- можно взять кусочек слайса `s = s[2:4]`
- операции над слайсами
  - append – добавить к слайсу
  - copy – скопировать слайс в новый слайс
- https://gobyexample.com/slices
- https://go.dev/play/p/yo6v_FPcz8Q



Мапы

- словарь
- раньше была реализована как хеш-таблица (массив корзин)
  - в Go1.24 все переписали :)
- `map[KeyType]ValueType`
- мапу можно создать:
  - `make(map[key-type]val-type, 5)`
  - явно
- ключи в мапе должны реализовывать интерфейс Comparable
  - чтобы их можно было сравнить между собой

Операции с мапами

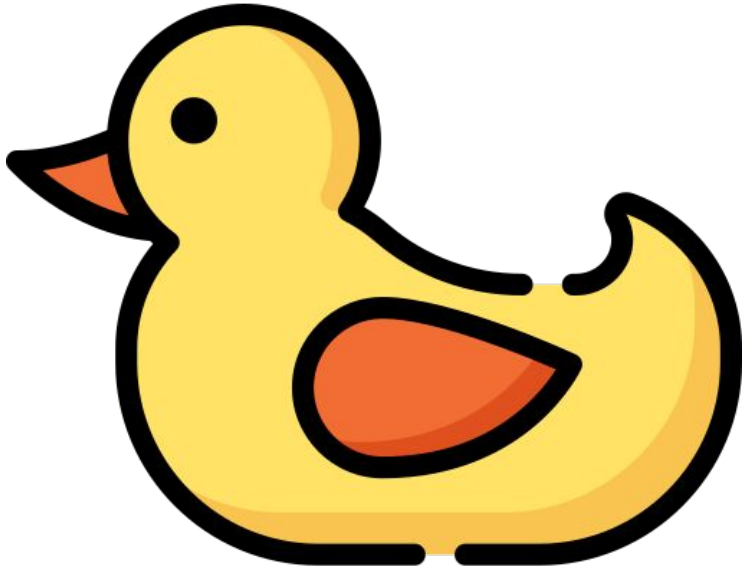
- можно получить элемент по ключу
- операции над мапами
  - delete – удалить ключ-значение
- обойти значения мапы
  - порядок не определен
- https://gobyexample.com/maps
- https://go.dev/play/p/IOCFPE67ixx

Структуры и интерфейсы

- структуры https://gobyexample.com/methods
  - у структур есть поля
  - у структур есть методы
    - методы могут быть по значению и по указателю
- интерфейсы: https://gobyexample.com/interfaces
  - объект реализует интерфейс, если он реализует все его методы

Duck typing

- если что-то выглядит как утка и крякает как утка, то это – утка
- если объект реализует все методы интерфейса, то его можно подставить везде, где ожидается интерфейс
- например есть интерфейс io.Writer
  - файл реализует этот интерфейс
  - массив байт реализует этот интерфейс
  - а можно самому реализовать этот интерфейс и использовать везде, где ждут io.Writer



Ошибки

- Ошибки создаются и обрабатываются явно
  - ошибка считается нормальной ситуацией
  - не Exception как в Java
  - https://go.dev/blog/errors-are-values
- Ошибки имеют тип **error**
- тип **error** – это интерфейс
  - https://go.dev/tour/methods/19
  - пример возврата ошибки https://gobyexample.com/errors

Задача: телефонная книга

- https://go.dev/play/p/amYzhfYDU_5
  - завели структуру телефонной книги
    - на основе мапы
  - реализовали методы для этой структуры
    - удаление, добавление пользователей
    - листинг через печать в stdout
  - добавили нескольких пользователей

Ссылки

- Материалы к курсу: https://github.com/katevi/golang-course
- Пример реализации интерфейса
  - io.Writer: https://pkg.go.dev/io#Writer
  - File: https://pkg.go.dev/os#File.Write
  - Bytes: https://pkg.go.dev/bytes#Buffer.Write
- Go Playground https://go.dev/play/
- Go Tour https://go.dev/tour/list
- https://go.dev/blog/slices-intro
- https://go.dev/blog/maps