# ТОП-50 ВОПРОСОВ НА СОБЕСЕДОВАНИЯХ GO-РАЗРАБОТЧИКОВ

585 586 587 588 589

587

реальных собеседований проанализировано

Ozon Яндекс Avito WB Lamoda

VK MTS X5 Kaspersky и др.

### В чём разница между массивами и слайсами в Go?

~85 раз

Go основы

Ozon, X5, WB, Lamoda, Магнит, МТС, Совкомбанк, Basis, РСХБ, ВсеИнструменты

### Что происходит при вызове append для слайса? Когда создаётся новый массив, а когда используется существующий?

~70 раз

Go основы

Ozon, X5, WB, Lamoda, Магнит, Совкомбанк, МТС, Basis, CloudX

### Как устроена map в Go под капотом? Бакеты, коллизии, эвакуация?

~65 раз

Go основы

Ozon, X5, WB, Lamoda, Магнит, МТС, ВсеИнструменты, Basis, Лента, CRTEX

### Какие примитивы синхронизации в Go ты знаешь и с какими работал?

~65 раз

Go конкурентность

Ozon, X5, WB, Lamoda, Магнит, VK, Совкомбанк, CloudX, InDrive, Лента, Островок

### Что будет если писать в закрытый канал? А если читать из закрытого канала?

~60 раз

Go каналы

Ozon, X5, WB, МТС, Совкомбанк, Lamoda, РСХБ, ЛинАйти, VK, MascotGames

### Какие уровни изоляции транзакций ты знаешь? В чём разница между ними?

~60 раз

Базы данных

Ozon, X5, Lamoda, InfoWatch, Совкомбанк, VK, Avito, Сараффан, Astra, МТС, Островок

### Чем горутина отличается от потока ОС? За счёт чего горутины считаются легковесными?

~60 раз

Go конкурентность

Ozon, X5, Lamoda, Магнит, МТС, VK, InfoWatch, Совкомбанк, Kaspersky, Лента

## Чем отличается буферизированный канал от небуферизированного?

~55 раз

Go каналы

Ozon, X5, Lamoda, МТС, MascotGames, ВсеИнструменты, ЛинАйти, Ximilabs, БФТ

## Расскажи про GMP-модель планировщика Go. Как устроены очереди горутин?

~55 раз

Go runtime

X5, InfoWatch, CloudX, Ozon, VK, Магнит, Лента, Совкомбанк, МТС, Сараффан

## Что из себя представляют строки в Go? Какие подводные камни есть?

~55 раз

Go основы

Ozon, X5, Магнит, ВсеИнструменты, МТС, Совкомбанк, ZeBrains, РСХБ

## Какие типы индексов в PostgreSQL ты знаешь? Чем B-Tree отличается от Hash?

~50 раз

Базы данных

Ozon, X5, Lamoda, Avito, a7, Купер, МТС, CRTEX, Сараффан, Лента, VK

## Что такое интерфейс в Go? Для чего нужен пустой интерфейс?

~50 раз

Go основы

X5, VK, Ozon, Совкомбанк, РСХБ, a7, ВсеИнструменты, InDrive, Сараффан, ISpring

## Что такое ACID? Расскажи про каждую букву.

~45 раз

Базы данных

InfoWatch, X5, AMTEX, РСХБ, ЛинАйти, VK, Лента, Ozon

## Как передаются слайсы в функцию? Почему изменения иногда не видны снаружи?

~45 раз

Go основы

Ozon, X5, МТС, Совкомбанк, РСХБ, Магнит, Basis, VK

## Что такое defer? В каком порядке выполняются несколько defer?

~45 раз

Go основы

Ozon, X5, МТС, Совкомбанк, РСХБ, IT_One, Basis, ДИТМосквы

### Что такое data race? Как его обнаружить и как с ним бороться?

~45 раз

Go конкурентность

MascotGames, VK, Совкомбанк, X5, Форвард, техзор, БФТ, Ozon

### Почему map в Go не является потокобезопасной? Как организовать конкурентный доступ?

~45 раз

Go конкурентность

Ozon, ВсеИнструменты, MascotGames, X5, WB, CRTEX, Astra, БФТ

### Когда интерфейс в Go равен nil? Что будет если сравнить интерфейс с nil указателем внутри?

~40 раз

Go основы

X5, Ximilabs, IT_One, цифровые_привычки, ДИТМосквы, ТочкаБанк, VK

### Как работает garbage collector в Go? Можно ли на него влиять?

~40 раз

Go runtime

X5, Ozon, CRTEX, Лента, МТС, Совкомбанк

### Что будет если читать или писать в nil канал?

~40 раз

Go каналы

Совкомбанк, InfoWatch, X5, VK, РСХБ, Ozon

### Как len() работает со строкой на русском языке? Как посчитать количество символов?

~35 раз

Go основы

Ozon, X5, системнаяинтеграция, IT_One, ZeBrains, ВсеИнструменты, Basis

### Как работает select в Go?

~35 раз

Go каналы

Ozon, X5, Совкомбанк, VK, РСХБ, Kaspersky

### Чем отличается конкурентность от параллельности?

~35 раз

Go конкурентность

Совкомбанк, X5, InDrive, туту, ЛигаЦифровойЭкономики, Форвард

### Что такое контекст в Go? Для чего используется?

~35 раз

Go основы

РСХБ, DDL, системнаяинтеграция, туту, цифровые_привычки, ТочкаБанк

### Чем процесс отличается от потока?

~35 раз

ОС и сети

Совкомбанк, Avito, VK, техзор, f6, hrp

### Расскажи про паттерн Outbox. Зачем он нужен?

~35 раз

Архитектура

Ozon, Купер, Lamoda, РСХБ, ТочкаБанк, Лента

### Что такое EXPLAIN ANALYZE? На что обращать внимание в его выводе?

~35 раз

Базы данных

Ozon, Lamoda, Купер, Ximilabs, Лента, БФТ, DDL

### Что такое шардирование? Какие проблемы могут возникнуть?

~35 раз

Базы данных

X5, Avito, a7, Ozon, РСХБ, Lamoda, CRTEX

### Можно ли взять указатель на элемент map в Go? Почему?

~30 раз

Go основы

X5, МТС, VK, Ozon

### Чем отличается REST от gRPC? Когда что выбирать?

~30 раз

Сети/API

ЛигаЦифровойЭкономики, InDrive, IT_One, VK, РСХБ, ZeBrains

### Что такое deadlock в БД? Когда он возникает?

~30 раз

Базы данных

Lamoda, Совкомбанк, Форвард, МТС, Astra

### Расскажи про SOLID. Как эти принципы ложатся на Go?

~30 раз

Проектирование

Avito, РСХБ, IT_One, ISpring, ЛигаЦифровойЭкономики

### Что такое идемпотентность? Как обеспечить идемпотентность POST запроса?

~30 раз

Архитектура

Avito, MascotGames, ZeBrains, ТочкаБанк

### Чем отличается Kafka от RabbitMQ? Когда что использовать?

~30 раз

Брокеры

Lamoda, AMTEX, Ozon, ЛигаЦифровойЭкономики, VK

### Что такое graceful shutdown? Как его реализовать?

~30 раз

Архитектура

Ozon, MascotGames, employcity

### В чём разница между LEFT JOIN и INNER JOIN?

~30 раз

SQL

Ozon, МТС, Lamoda, X5

### Что будет если в горутине произойдёт паника? Можно ли её перехватить из родительской горутины?

~30 раз

Go конкурентность

InfoWatch, РСХБ, Совкомбанк, Kaspersky, Basis

### Что такое кэширование? Какие стратегии кэширования существуют?

~30 раз

Архитектура

Ozon, MascotGames, Совкомбанк, ЛигаЦифровойЭкономики, VK

### Гарантирует ли Kafka порядок сообщений? Как обеспечить порядок для одной сущности?

~25 раз

Брокеры

Магнит, Lamoda, X5, Лента

### Как устроен мьютекс под капотом? Что такое голодание мьютекса?

~25 раз

Go конкурентность

X5, VK, Островок, Ozon

### Как эффективно склеить тысячу строк в Go?

~25 раз

Go основы

ВсеИнструменты, Ozon, системнаяинтеграция, ZeBrains

### Что такое WAL (Write-Ahead Log) и для чего он нужен?

~25 раз

Базы данных

X5, Лента

### Как работает VACUUM и autovacuum в PostgreSQL?

~25 раз

Базы данных

Лента, X5, МТС, DDL

### Чем отличается HTTP/1.1 от HTTP/2?

~25 раз

Сети/API

Avito, IT_One, Совкомбанк, Лента

### Что такое NetPoller в Go и для чего он нужен?

~25 раз

Go runtime

X5, InfoWatch, CloudX, Ozon, Лента

### Как создать индекс без блокировки таблицы на проде?

~20 раз

Базы данных

X5, МТС, системнаяинтеграция

### Что такое sync.Pool и для чего он нужен?

~20 раз

Go конкурентность

X5, РСХБ, Ozon

### Как работает escape analysis в Go? Когда переменная попадает на кучу?

~20 раз

Go runtime

X5, Тинькофф, МТС

**Расскажи про структуру канала (hchan). Как устроен буфер?**

~20 раз

Go каналы

X5, Лента, AMTEX

**Что такое выравнивание (padding) в структурах Go? Сколько памяти занимает структура?**

~20 раз

Go runtime

WB, X5, Лента, ZeBrains

# ТОП-10 ЗАДАЧ НА ЛАЙВКОДИНГ

Ниже — реальные задачи с собеседований в крупнейших компаниях. Для каждой задачи приведено условие и код, который дают кандидату. Попробуйте решить их самостоятельно за 15–20 минут.

{ ЗАДАЧА 1 }

## Что выведет код со слайсами и append

Встречается ~50 раз | Ozon, X5, Lamoda, Basis, МТС, Магнит, Тинькофф, Совкомбанк, VK

Самая популярная задача на собеседованиях. Вам дают фрагмент кода и просят предсказать вывод, а затем объяснить почему. Ключевые моменты: слайс — это заголовок (ptr, len, cap); append может как изменить существующий underlying array, так и создать новый; передача слайса в функцию копирует заголовок, но не данные.

```go
func appendToSlice(s []int) {
   s = append(s, 4)
}

func addToMap(m map[string]int) {
   m["b"] = 2
}

func main() {
   s := []int{1, 2, 3}
   appendToSlice(s)
   fmt.Println(s) // Что выведет?

   m := map[string]int{"a": 1}
   addToMap(m)
   fmt.Println(m) // Что выведет?
}
```

Подсказка: Слайс: [1 2 3] (append не виден снаружи — заголовок скопирован).
Map: map[a:1 b:2] (map передаётся по ссылке).

{ ЗАДАЧА 2 }

## Параллельный Worker Pool с каналами

Встречается ~45 раз | Ozon, Lamoda, Магнит, МТС, DDL, Astra, NDA

Дан входной канал данных и выходной канал результатов. Нужно реализовать параллельную обработку в N воркерах. Каждый воркер читает из in, обрабатывает данные и пишет в out. Закрыть out после завершения всех воркеров. Усложнение: добавить поддержку context для отмены.

```go
func processInParallel(in <-chan int, out chan<- int,
             workers int) {
   // Необходимо написать параллельную обработку
   // в кол-ве горутин равном workers.
   // Получаем данные из канала in,
   // над каждым элементом выполняем processData,
   // передаём результат в out.
   panic("not implemented")
}

func processData(data int) int {
   time.Sleep(100 * time.Millisecond)
   return data * 2
}
```

Подсказка: sync.WaitGroup + N горутин, каждая читает range in.
Отдельная горутина ждёт wg.Wait() и закрывает out.

{ ЗАДАЧА 3 }

## Merge N каналов в один (fan-in)

Встречается ~35 раз | Ozon, Островок, ТочкаБанк, InfoWatch

Написать функцию merge, которая принимает произвольное количество входных каналов и объединяет их в один выходной. Выходной канал должен закрыться, когда все входные каналы закрыты. Не допускать утечек горутин.

```go
func merge(chans ...<-chan int) <-chan int {
   out := make(chan int)
   // Реализуйте объединение каналов.
   // Закройте out когда ВСЕ входные каналы
   // будут закрыты.
   return out
}
```

Подсказка: По горутине на каждый канал + sync.WaitGroup.
Отдельная горутина: wg.Wait(); close(out).

{ ЗАДАЧА 4 }

## Code Review: кэширующий прокси (МойОфис)

Встречается ~30 раз | Ozon, МойОфис, Astra, Островок, БФТ, VK, Купер

В проекте используется key-value БД (latency ~100мс). Нужно написать кэширующий прокси. Требования: потокобезопасность, пользователи кэша не должны знать про БД за ним, данные в памяти. Ниже — заготовка. Найдите проблемы и улучшите код.

```go
type KVStore interface {
   Get(key string) (string, error) // часто
   Keys() ([]string, error) // редко
}

type AddressStore struct {
   store KVStore
   cache map[string]string
   mu sync.RWMutex
}

func (a *AddressStore) Get(key string) (string, error) {
   a.mu.RLock()
   ip, found := a.cache[key]
   a.mu.RUnlock()
   if found {
      return ip, nil
   }
   ip, err := a.store.Get(key)
   if err != nil { return "", err }
   a.mu.Lock()
   a.cache[key] = ip
   a.mu.Unlock()
   return ip, nil
}
```

Подсказка: Проблемы: cache stampede (два потока одновременно идут в store), нет TTL, нет инвалидации, нет ограничения размера кэша. Решение: singleflight для дедупликации запросов.

{ ЗАДАЧА 5 }

## Code Review: корзина с ошибками (tutu)

Встречается ~25 раз | tutu, Ozon

Дан код корзины интернет-магазина. Нужно провести Code Review и найти как можно больше проблем. Затрагиваются темы: SOLID, конкурентность, безопасность, архитектура.

```
var items []Product
var vatRate = 10.0
var removalChannel chan bool

func init() {
   items = make([]Product, 100) // 100 пустых!
   removalChannel = make(chan bool, 1)
}

func AddItemToShoppingCart(item Product) {
   items = append(items, item)
}

func RemoveItemFromShoppingCart(name string) {
   go func() {
     for i, item := range items {
       if item.NameOfProduct == name {
         items = append(items[:i],
           items[i+1:]...)
         removalChannel <- true
         return
       }
     }
   }()
   removalChannel <- false // deadlock!
}
```

Подсказка: Ошибки: глобальное состояние, make([]Product, 100) создаёт 100 пустых, data race на items, deadlock в RemoveItem, SRP нарушен, хардкод пароля в SendEmail, нет DI.

{ ЗАДАЧА 6 }

## Слайсы и shared underlying array (Ozon)

Встречается ~45 раз | Ozon, X5, Тинькофф, WB

Вариация той же задачи, но с акцентом на общий underlying array. Когда cap слайса позволяет — append не аллоцирует новый массив, а перезаписывает данные, которые «видны» через другой слайс.

```go
func main() {
   nums := []int{1, 2, 3}

   addNum(nums[0:2])
   fmt.Println(nums) // ?

   addNums(nums[0:2])
   fmt.Println(nums) // ?
}

func addNum(nums []int) {
   nums = append(nums, 4)
}

func addNums(nums []int) {
   nums = append(nums, 5, 6)
}
```

Подсказка: addNum: cap(nums[0:2])=3 — append перезаписывает 3-й элемент -> [1 2 4].
addNums: нужно 4 элемента при cap=3 — новый массив, nums не меняется -> [1 2 4].

## { ЗАДАЧА 7 } Горутины в цикле: замыкание на переменную

Встречается ~25 раз | Магнит, X5, МТС, WB, Ozon, БФТ

Классическая задача на понимание замыканий в Go. Что выведет этот код и почему?
Как исправить? Примечание: в Go 1.22+ поведение изменилось — переменная цикла теперь per-iteration. Но на собесах по-прежнему спрашивают.

```go
func main() {
  for i := 0; i < 5; i++ {
    go func() {
      fmt.Println(i)
    }()
  }
  time.Sleep(time.Second)
}
```

Подсказка: До Go 1.22: выведет 5 5 5 5 5 (замыкание захватывает переменную i, а не значение).
Исправление: go func(i int) {...}(i). С Go 1.22+: 0-4 в случайном порядке.

## { ЗАДАЧА 8 } defer и значение аргумента (Ozon, WB)

Встречается ~25 раз | Ozon, WB, МТС, Совкомбанк

Задача на понимание того, когда вычисляются аргументы defer.
Два варианта: прямой вызов метода и вызов через замыкание. Что выведет каждый вариант?

```go
type X struct { V int }

func (x X) S() { fmt.Println(x.V) }

// Вариант 1
func main() {
  x := X{123}
  defer x.S()
  x.V = 456
}

// Вариант 2
func main() {
  x := X{123}
  defer func() { x.S() }()
  x.V = 456
}
```

Подсказка: Вариант 1: выведет 123 — аргументы defer (receiver x) вычисляются в момент регистрации.
Вариант 2: выведет 456 — замыкание захватывает переменную x.

{ ЗАДАЧА 9 }

## SQL: покупки до бана + агрегация

Встречается ~25 раз | Ozon, Lamoda, X5

Даны три таблицы: users (id, name), purchases (id, user_id, product, amount, created_at) и ban_list (user_id, banned_at). 1) Вывести уникальные пары (пользователь, товар) для покупок, совершённых ДО бана. 2) Вывести пользователей с суммой покупок > 5000.

```sql
-- Запрос 1: покупки до бана
SELECT DISTINCT u.name, p.product
FROM users u
JOIN purchases p ON p.user_id = u.id
JOIN ban_list b ON b.user_id = u.id
WHERE p.created_at < b.banned_at;

-- Запрос 2: сумма покупок > 5000
SELECT u.name, SUM(p.amount) AS total
FROM users u
JOIN purchases p ON p.user_id = u.id
GROUP BY u.name
HAVING SUM(p.amount) > 5000
ORDER BY total DESC;
```

Подсказка: Обратите внимание на LEFT JOIN vs INNER JOIN. Усложнение: что если пользователь не в ban_list?

{ ЗАДАЧА 10 }

## In-Memory Cache с TTL и потокобезопасностью

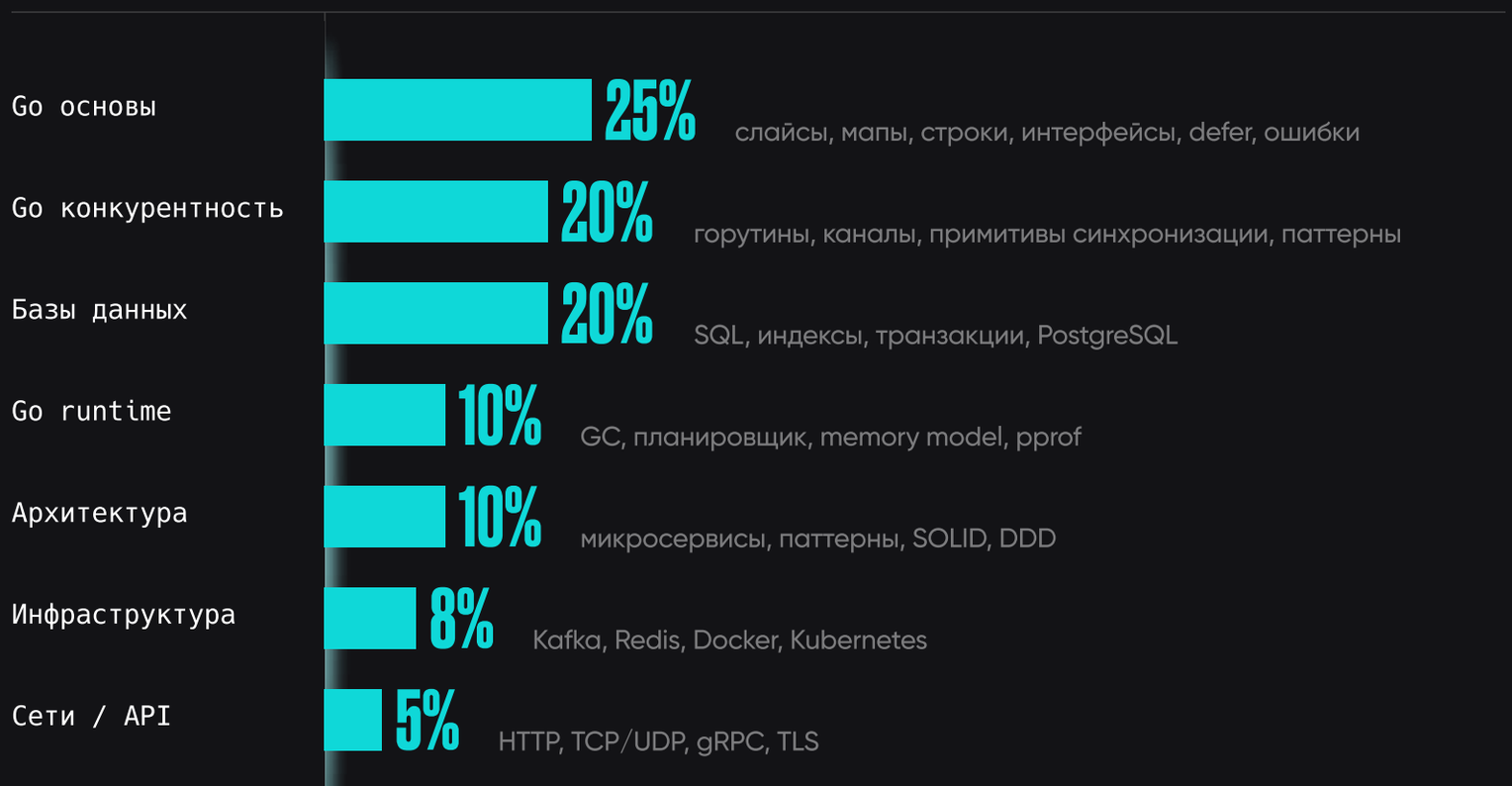
Встречается ~20 раз | Магнит, Astra, ВсеИнструменты

Реализовать потокобезопасный In-Memory кэш с поддержкой TTL.
Методы: Get(key) и Set(key, value, ttl). Элементы должны автоматически удаляться по истечении TTL.
Усложнение: шардирование для уменьшения contention на мьютексе.

```go
type Cache struct {
  mu sync.RWMutex
  items map[string]cacheItem
}

type cacheItem struct {
  value interface{}
  expiresAt time.Time
}

func (c *Cache) Set(key string,
  value interface{}, ttl time.Duration) {
  // реализуйте
}

func (c *Cache) Get(key string)
  (interface{}, bool) {
  // реализуйте
}
```

Подсказка: Set: Lock + записать с expiresAt = time.Now().Add(ttl). Get: RLock + проверить expiresAt.
Фоновая горутина для очистки просроченных элементов.

# РАСПРЕДЕЛЕНИЕ ПО КАТЕГОРИЯМ



System Design 2% проектирование систем

# РЕКОМЕНДАЦИИ ПО ПОДГОТОВКЕ

## 1 Знать наизусть

- Слайсы: append, capacity, shared underlying array
- Map: устройство, потокобезопасность, ключи
- Каналы: закрытый/nil канал, буферизованный/нет
- defer: порядок, вычисление аргументов, именованные return

## 2 Уметь написать за 15 минут

- Merge каналов (fan-in)
- Worker Pool
- Семафор для ограничения горутин
- In-Memory Cache с TTL
- Параллельный HTTP-краулер

## 3 SQL – уверенно владеть

- JOIN + GROUP BY + HAVING
- Составные индексы: порядок колонок, selectivity
- EXPLAIN ANALYZE: Seq Scan vs Index Scan, cost, rows
- Уровни изоляции: phantom read, non-repeatable read

## 4 Теория – уметь объяснить

- GMP-модель планировщика Go
- Garbage Collector: tri-color marking, STW phases
- ACID, CAP-теорема
- Outbox Pattern, Saga Pattern
- SOLID в контексте Go

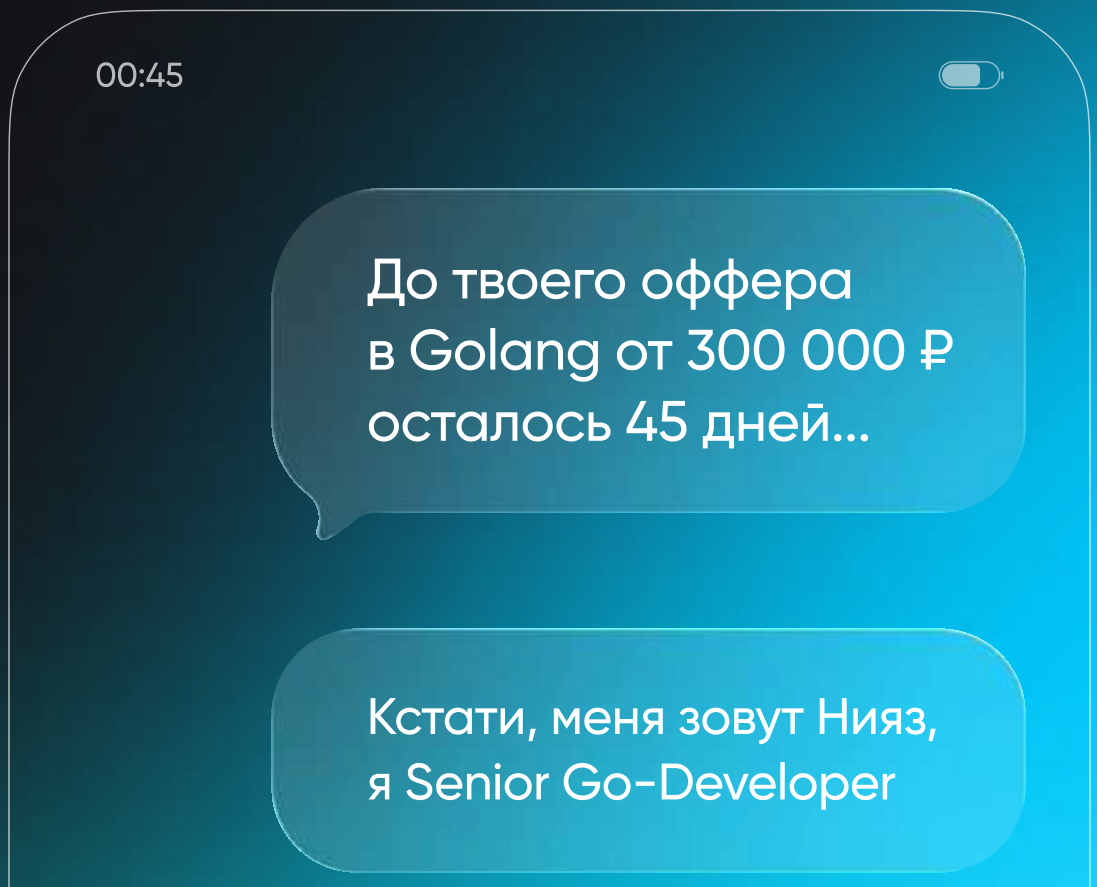
## 5 Архитектура (для Senior)

- Чистая / Гексагональная архитектура
- Микросервисные паттерны: Circuit Breaker, Retry, Bulkhead
- DDD: агрегаты, bounded context
- System Design: проектирование с нуля

# БОЛЬШЕ ПОЛЕЗНОГО В МОИХ СОЦИАЛЬНЫХ СЕТЯХ

Подписывайтесь

TELEGRAM YOUTUBE



Уже больше 150 моих учеников получили 130+ офферов в ТОП-компании VK, Авито, Ozon, Wildberries и т.п.

Запишись на бесплатную консультацию
и получи видеодемонстрацию о менторстве
и полезные материалы по Go

**Записаться на консультацию**