



Занятие 2

Тесты





Структура занятия

- 1. Разбор домашнего задания
- 2. Тесты
- 3. Бенчмарки
- 4. Библиотека testify
- 5. Интеграция тестов в проект
- 6. Обсуждение нового домашнего задания





Зачем вообще нужны тесты?

• Неожидаемое поведение выявляется быстрее

• Написание лучшего кода

 Проектировании лучшей архитектуры

- Проще проводить Bug Regression
- Рефакторинг становится безопаснее
- Деплой происходит увереннее

Testing





Как писать тесты в Golang?

- Сигнатуры функций должны быть вида func TestXxxx(t *testing.T)
- Название всегда начинается с Test
- Принимает единственный параметр *testing.T

- Файлы с тестами обычно называются так: sth_test.go
- Если происходит ошибка, тест считается зафейленным
- Фейлить можно вручную с помощью t.Error

Testing





Зачем вообще нужны бенчмарки?

- Оптимизация кода ради скорости/потребления памяти
- Поиск боттлнеков

- Сравнение эффективности различных реализаций функционала
- Победить в споре с коллегами

Benchmarking





Как писать бенчмарки в Golang?

• Сигнатуры функций должны быть вида BenchmarkXxxx(b *testing.B)

• Принимает единственный параметр *testing.B

• Бенчмарки обычно хранятся в одних файлах с тестами • Название всегда начинается с Benchmark

Benchmarking





CPU Bound

• Слабый процессор не позволяет выполняться задачам быстрее

I/O Bound

• Медленные диски, оперативная память или сеть не позволяют задачам выполняться быстрее

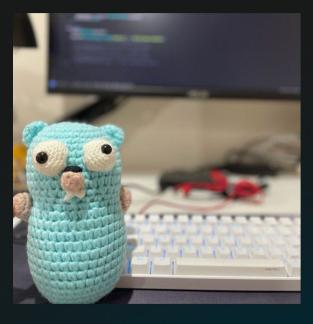
Benchmarking

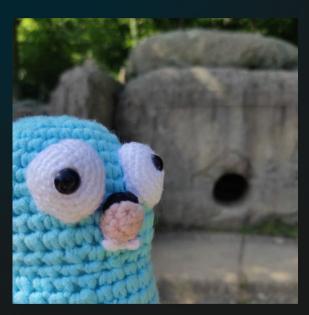














Форма для фидбека

