

Разработчик Golang

Разработка сетевых приложений и микросервисов на Go

Длительность курса: 146 академических часов

1 Начало работы с Go

1 Инструментарий и начало работы с Go

разобраться с базовым инструментарием Go.
после занятия вы сможете:
приступить к написанию программ на Go.

Домашние задания

1 Hello now

Цель: В результате выполнения ДЗ вы напишете первую «hello-world» программу на языке Go.
В данном задании тренируются навыки:
- работы с git, GitHub и Travis CI;
- работы с модулями в Go.

Описание и критерии оценки:

https://github.com/OtusGolang/home_work/tree/master/hw01_hello_now

Процесс сдачи домашнего задания:

[https://github.com/OtusGolang/home_work/wiki/\[Студентам\]-Процесс-сдачи-ДЗ](https://github.com/OtusGolang/home_work/wiki/[Студентам]-Процесс-сдачи-ДЗ)

2 Тестирование в Go. Часть 1

после занятия вы сможете:
писать юнит-тесты к программам на Go.

3 Элементарные типы данных в Go

узнать элементарные типа языка.
после занятия вы сможете:
использовать основные типы языка;
избегать ошибок при работе со строками и указателями.

Домашние задания

1 Распаковка строки

Цель: В результате выполнения ДЗ вы напишете функцию, работающую со строками.

В данном задании тренируются навыки:

- работы со строками и рунами;
- работы с пакетами "strconv", "strings" и "unicode".

Описание и критерии оценки:

https://github.com/OtusGolang/home_work/tree/master/hw02_unpack_string

Процесс сдачи домашнего задания:

[https://github.com/OtusGolang/home_work/wiki/\[Студентам\]-Процесс-сдачи-ДЗ](https://github.com/OtusGolang/home_work/wiki/[Студентам]-Процесс-сдачи-ДЗ)

4 Массивы, слайсы и словари

узнать про основные структуры данных (СД) языка.
после занятия вы сможете:
пользоваться основными СД языка.
не совершать ошибок при работе с ссылочными типами Go.

5 Структуры

после занятия вы сможете:
создавать пользовательские типы данных и комбинировать их между собой.

Домашние задания

1 Частотный анализ

Цель: В результате выполнения ДЗ вы напишете функцию, работающую с текстом.

В данном задании тренируются навыки:

- работы со строками;
- работы со слайсами и словарями.

Описание и критерии оценки:

https://github.com/OtusGolang/home_work/tree/master/hw03_frequency_analysis

Процесс сдачи домашнего задания:

[https://github.com/OtusGolang/home_work/wiki/\[Студентам\]-Процесс-сдачи-ДЗ](https://github.com/OtusGolang/home_work/wiki/[Студентам]-Процесс-сдачи-ДЗ)

6 Функции и методы

после занятия вы сможете:
объявлять и определять функции;
избегать ошибок, связанных с областью видимости.

7 Интерфейсы. Часть 1

узнать, что такое интерфейсы, как они устроены внутри и как их использовать.

8 Интерфейсы.

рассмотреть интерфейсы с более практической стороны;

Часть 2

после занятия вы сможете:
пользоваться type assertion и type switch.

Домашние задания

1 LRU-кэш

Цель: В результате выполнения ДЗ вы реализуете LRU-кэш на основе собственного двусвязного списка :)

В данном задании тренируются навыки:

- работы с интерфейсами;
- работы со стандартными структурами данных Go;
- реализации алгоритмов, полезных для разработчика.

Описание и критерии оценки:

https://github.com/OtusGolang/home_work/tree/master/hw04_lru_cache

Процесс сдачи домашнего задания:

[https://github.com/OtusGolang/home_work/wiki/\[Студентам\]-Процесс-сдачи-ДЗ](https://github.com/OtusGolang/home_work/wiki/[Студентам]-Процесс-сдачи-ДЗ)

9 **Обработка ошибок. Понятие паники**

поговорить про ошибки.
узнать, что такое паника, почему не стоит паниковать и как пользоваться отложенным вызовом функции.
после занятия вы сможете:
правильно обрабатывать ошибки;
создавать собственные типы ошибок.

10 **Тестирование в Go. Часть 2**

после занятия вы сможете:
писать юнит-тесты к программам на Go.

1 Горутины и каналы

начать работу с горутинами.
после занятия вы сможете:
реализовать передачу данных между горутинами с помощью канала.

2 Примитивы синхронизации. Часть 1

после занятия вы сможете:
пользоваться частью механизмов синхронизации в Go;
бороться с «гонками» в Go.

Домашние задания

1 Параллельное исполнение

Цель: В результате выполнения ДЗ вы реализуете функцию, конкурентно выполняющую заданный список задач.

В данном задании тренируются навыки:

- работы с горутинами и каналами;
- работы с оператором `select`;
- использования структур синхронизации данных.

Описание и критерии оценки:

https://github.com/OtusGolang/home_work/tree/master/hw05_parallel_execution

Процесс сдачи домашнего задания:

[https://github.com/OtusGolang/home_work/wiki/\[Студентам\]-Процесс-сдачи-ДЗ](https://github.com/OtusGolang/home_work/wiki/[Студентам]-Процесс-сдачи-ДЗ)

3 Примитивы синхронизации. Часть 2

после занятия вы сможете:
пользоваться полным набором механизмов синхронизации.

4 Concurrency patterns

узнать больше о конкурентности в Go.
после занятия вы сможете:
применять concurrency-паттерны на практике.

Домашние задания

1 Пайплайн

Цель: В результате выполнения ДЗ вы реализуете функцию, конкурентно выполняющую пайплайн заданных функций.

В данном задании тренируются навыки:

- работы с горутинами и каналами;
- работы с оператором `select`;
- работы с паттернами конкурентного выполнения программы.

Описание и критерии оценки:

https://github.com/OtusGolang/home_work/tree/master/hw06_parallel_execution

ipeline_execution

Процесс сдачи домашнего задания:

[https://github.com/OtusGolang/home_work/wiki/\[Студентам\]-](https://github.com/OtusGolang/home_work/wiki/[Студентам]-)

Процесс-сдачи-ДЗ

- 5 **Go внутри. Планировщик** познакомиться с устройством планировщика Go; узнать, когда именно происходит «переключение» горутин.
-
- 6 **Go внутри. Память и сборка мусора** получить базовую информацию об устройстве памяти в Go; узнать о механизме GC в Go.

1 Работа с вводом/ выводом в Go

узнать тип Buffer;
рассмотреть стандартные интерфейсы: Reader, Scanner, Writer, Closer;
узнать блочные устройства, Seeker;
узнать форматированный ввод и вывод: fmt.

Домашние задания

1 Утилита для копирования файлов

Цель: В результате выполнения ДЗ вы реализуете упрощенный аналог dd (man dd).
В данном задании тренируются навыки работы с файловой системой,
происходит первое знакомство с обработкой аргументов командной строки.

Описание и критерии оценки:

https://github.com/OtusGolang/home_work/tree/master/hw07_file_copying

Процесс сдачи домашнего задания:

[https://github.com/OtusGolang/home_work/wiki/\[Студентам\]-Процесс-сдачи-ДЗ](https://github.com/OtusGolang/home_work/wiki/[Студентам]-Процесс-сдачи-ДЗ)

2 Форматирование данных

после занятия вы сможете:
сериализовывать и десериализовывать данные различных форматов стандартными средствами языка и сторонними библиотеками.

3 Взаимодействие с OS

после занятия вы сможете:
работать с операционной системой из программы на Go.

Домашние задания

1 Утилита envdir

Цель: В результате выполнения ДЗ вы реализуете утилиту envdir (man envdir) на Go.

В данном задании тренируются навыки:

- работы с переменными окружения;
- запуска других процессов из программы на Go.

Описание и критерии оценки:

https://github.com/OtusGolang/home_work/tree/master/hw08_envdir_tool

Процесс сдачи домашнего задания:

[https://github.com/OtusGolang/home_work/wiki/\[Студентам\]-Процесс-сдачи-ДЗ](https://github.com/OtusGolang/home_work/wiki/[Студентам]-Процесс-сдачи-ДЗ)

4 Рефлексия

познакомиться с механизмом рефлексии в Go.

5 Кодогенерация в Go	<p>познакомиться с механизмом кодогенерации в Go; после занятия вы сможете: не бояться библиотек, использующих кодогенерацию.</p> <p>Домашние задания</p> <p>1 Генератор валидаторов</p> <p>Цель: В результате выполнения ДЗ вы реализуете утилиту для кодогенерации валидаторов на Go. В данном задании тренируются навыки: - работы с синтаксическим деревом программы (AST); - не использования, а реализации своей собственной кодогенерации.</p> <p>Описание и критерии оценки: https://github.com/OtusGolang/home_work/tree/master/hw09_generator_of_validators</p> <p>Процесс сдачи домашнего задания: https://github.com/OtusGolang/home_work/wiki/[Студентам]-Процесс-сдачи-ДЗ</p> <hr/>
6 Файлы конфигурации и логирование	<p>познакомиться с механизмами логирования и конфигурирования в Go.</p> <hr/>
7 Профилирование и оптимизация Go программ	<p>познакомиться с инструментами профилирования в Go. после занятия вы сможете: писать бенчмарки в Go; профилировать программы на Go; использовать практики оптимизации кода на Go.</p> <p>Домашние задания</p> <p>1 Оптимизация программы</p> <p>Цель: В результате выполнения ДЗ вы оптимизируете предоставленную вам программу. В данном задании тренируются навыки профилирования и оптимизации кода.</p> <p>Описание и критерии оценки: https://github.com/OtusGolang/home_work/tree/master/hw10_program_optimization</p> <p>Процесс сдачи домашнего задания: https://github.com/OtusGolang/home_work/wiki/[Студентам]-Процесс-сдачи-ДЗ</p>

1 Контекст и низкоуровневые сетевые протоколы

после занятия вы сможете:
работать с контекстом в Go;
использовать инструменты отладки сетевых проблем.

Домашние задания

1 Клиент TELNET

Цель: В результате выполнения ДЗ вы реализуете крайне примитивный клиент TELNET.

В данном задании тренируются навыки:

- работы с сетью;
- работы с читателями (`io.Reader`) и писателями (`io.Writer`).

Описание и критерии оценки:

https://github.com/OtusGolang/home_work/tree/master/hw11_telnet_client

Процесс сдачи домашнего задания:

[https://github.com/OtusGolang/home_work/wiki/\[Студентам\]-Процесс-сдачи-ДЗ](https://github.com/OtusGolang/home_work/wiki/[Студентам]-Процесс-сдачи-ДЗ)

2 Работа с SQL

после занятия вы сможете:
работать с SQL в Go.

Домашние задания

1 Заготовка сервиса Календарь

Цель: В результате выполнения ДЗ вы получите каркас сервиса Календарь.

В данном задании тренируются навыки:

- декомпозиции предметной области;
- построения элементарной архитектуры проекта;
- работы с СУБД и SQL.

Описание и критерии оценки:

https://github.com/OtusGolang/home_work/blob/master/hw12_13_14_15_calendar/docs/12_README.md

Процесс сдачи домашнего задания:

[https://github.com/OtusGolang/home_work/wiki/\[Студентам\]-Процесс-сдачи-ДЗ](https://github.com/OtusGolang/home_work/wiki/[Студентам]-Процесс-сдачи-ДЗ)

3 Работа с HTTP

после занятия вы сможете:
работать с HTTP в Go.

4 Работа с gRPC

после занятия вы сможете:
писать обратно совместимые Protobuf схемы;
писать gRPC сервисы.

Домашние задания

1 «API к Календарю»

Цель: В результате выполнения ДЗ вы разработаете API к созданному ранее Календарю.

В данном задании тренируются навыки работы с GRPC и HTTP - построения современного API.

Описание и критерии оценки:

https://github.com/OtusGolang/home_work/blob/master/hw12_13_14_15_calendar/docs/13_README.md

Процесс сдачи домашнего задания:

[https://github.com/OtusGolang/home_work/wiki/\[Студентам\]-Процесс-сдачи-ДЗ](https://github.com/OtusGolang/home_work/wiki/[Студентам]-Процесс-сдачи-ДЗ)

- 1 Монолит и микросервисы**

получить представление о популярных архитектурах сложных систем.

- 2 Очереди сообщений**

получить представление об очередях сообщений и событийно-ориентированной архитектуре.
после занятия вы сможете:
работать с RabbitMQ из Go.

Домашние задания

 - 1 Кроликизация Календаря**

Цель: В результате выполнения ДЗ вы получите процессы Рассыльщик и Планировщик (в составе сервиса Календарь), связанные между собой очередью сообщений.
В данном задании тренируются навыки работы с RabbitMQ и очередями в принципе.

Описание и критерии оценки:
https://github.com/OtusGolang/home_work/blob/master/hw12_13_14_15_calendar/docs/14_README.md

Процесс сдачи домашнего задания:
[https://github.com/OtusGolang/home_work/wiki/\[Студентам\]-Процесс-сдачи-ДЗ](https://github.com/OtusGolang/home_work/wiki/[Студентам]-Процесс-сдачи-ДЗ)

- 3 Docker**

получить представление о Docker контейнере и образе.
после занятия вы сможете:
докеризировать своё приложение.

- 4 Тестирование микросервисов**

после занятия вы сможете:
писать интеграционные тесты для своего приложения.

Домашние задания

 - 1 Докеризация и интеграционное тестирование Календаря**

Цель: В результате выполнения ДЗ вы получите полноценный сервис Календарь, способный запускаться вместе со своим окружением и покрытый интеграционными тестами.

В данном задании тренируются навыки:

 - работы с docker, написания Dockerfile'ов;
 - работы с docker-compose, написания compose-файлов;
 - написания интеграционных тестов к web-сервисам;
 - работы с BDD, языком Gherkin, библиотекой github.com/DATA-DOG/godog.

Описание и критерии оценки:
https://github.com/OtusGolang/home_work/blob/master/hw12_13_14_15_calendar/docs/15_README.md

5 **Мониторинг**

после занятия вы сможете:
мониторить свои приложения.

- 1 **Проектная работа** выбрать тему проектной работы, обсудить и начать выполнять её;
спланировать работу над проектом;
получить ответы на возможные вопросы о проекте;
ознакомиться с регламентом работы над проектом.

Домашние задания

1 Проект

Цель: закрепить знания и навыки, полученные в течение курса;
пополнить своё портфолио качественным проектом.

выбрать тему;
подтвердить тему в чате с преподавателем;
предоставить промежуточные наработки на промежуточное ревью;
предоставить финальную версию проекта к дедлайну.

-
- 2 **Консультация по проектам** получить ответы на вопросы, возникшие во время выполнения проектной работы;
получить ответы на вопросы по ДЗ и курсу в целом.

-
- 3 **Итоговое занятие** узнать, как получить сертификат об окончании курса, как взаимодействовать после окончания курса с OTUS и преподавателями, какие вакансии и позиции есть для выпускников (опционально - в России и за рубежом) и на какие компании стоит обратить внимание.