otus
онлайн-образование

Полная программа

Golang Developer. Professional

Разработка сетевых приложений и микросервисов на Go

Длительность курса: 156 часов

Модуль 1. Начало работы с Go

Тема 1

Знакомство с курсом / ДЗ

Цель занятия

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

Краткое содержание

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

Домашние задания

Hello, OTUS!

Цель

написать первую «hello-world» программу на языке Go.

Тема 2

Синтаксис языка

Цель занятия

узнать про основные конструкции, типы и структуры данных (СД) языка.
после занятия вы сможете:
использовать основные типы языка;
пользоваться основными СД языка.

Краткое содержание

основные синтаксические конструкции языка, структуры и теги;
функции и методы;
указатели;
слайсы и мапы;
строки;
«zero value».

Тема 3

Типы данных

Цель занятия

использовать простые и составные встроенные типы языка;
создавать и использовать свои типы;
использовать указатели.

Краткое содержание

массивы, слайсы и мапы;
строки;
указатели;
структуры, теги, методы;
указатели на функции и анонимные функции.

Тема 4

Особенности языка и типовые ошибки / ДЗ

Цель занятия

узнаете внутреннее устройство слайсов и мап, поговорим о распространенных ошибках.
после занятия вы сможете:
избегать ошибок при работе со строками и указателями;
не совершать ошибок при работе с ссылочными типами Go;
избегать ошибок, связанных с областью видимости.

Краткое содержание

устройство строк и руны;
стандартные функции для работы со строками и Unicode;
преобразование и присвоение типов;
ошибки, связанные с областью видимости;
ошибки при использовании слайсов и мап;
передача аргументов по ссылке и по значению.

Домашние задания

Распаковка строки

Цель

написать функцию, работающую со строками.

Тема 5

Лучшие практики работы с ошибками / ДЗ

Цель занятия

поговорим про ошибки и практики работы с ними;
вы узнаете, что такое паника, почему не стоит паниковать и как пользоваться отложенным вызовом функции.
после занятия вы сможете:
правильно обрабатывать ошибки;
создавать собственные типы ошибок.

Краткое содержание

интерфейс error;

лучшие и худшие практики обработки ошибок;
defer;
функции panic и recover.

Домашние задания

Частотный анализ

Цель

написать функцию, работающую с текстом.

Тема 6

Тестирование в Go

Цель занятия

после занятия вы сможете:
писать юнит-тесты к программам на Go, узнаете о “золотых” файлах.

Краткое содержание

тестирование программ на Go;
понятие табличных тестов;
пакет testing и библиотека testify;

Тема 7

Продвинутое тестирование в Go

Цель занятия

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

Краткое содержание

моки (для интерфейсов, времени, фс);
faker (для генерации тестовых данных);
тестирование мутированием;
golden files.

Тема 8

Интерфейсы изнутри / ДЗ

Цель занятия

узнать как интерфейсы устроены внутри и как их эффективно использовать;
пользоваться type assertion и type switch.

Краткое содержание

определение и реализация интерфейсов;
внутреннее устройство интерфейса;
влияние использования интерфейсов на производительность программы;
композиция интерфейсов;
пустой интерфейс (interface {});

значение типа интерфейс и ошибки, связанные с nil;
правила присваивания значений переменным типа интерфейс;
опасное и безопасное приведение типов (type cast);
использование switch в контексте интерфейсов;
слайсы и словари с интерфейсами.

Домашние задания

LRU-кэш

Цель

реализовать LRU-кэш на основе собственного двусвязного списка.

Модуль 2. Concurrency в Go

Тема 1

Горутины и каналы

Цель занятия

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

Краткое содержание

горутины и как их запускать;
канал и как он устроен внутри;
сравнение буферизированных и небуферизованных каналов;
использование каналов для передачи данных и синхронизации;
оператор select;
таймеры в Go.

Тема 2

Примитивы синхронизации в деталях / ДЗ

Цель занятия

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

Краткое содержание

группа ожидания (sync.WaitGroup);
гарантировано одноразовое выполнение (sync.Once);
«простой» мьютекс (sync.Mutex);
детектор гонок (race detector).

Домашние задания

Параллельное исполнение

Цель

реализовать функцию, конкурентно выполняющую заданный список задач.

Тема 3

Дополнительные примитивы синхронизации

Цель занятия

после занятия вы сможете:
пользоваться полным набором механизмов синхронизации.

Краткое содержание

пул объектов (sync.Pool);
read-write мьютекс (sync.RWMutex);
concurrent-safe словарь (sync.Map);
условные переменные (sync.Cond);
атомарные операции (пакет atomic);
модель памяти в Go.

Тема 4

Concurrency patterns / ДЗ

Цель занятия

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

Краткое содержание

паттерны синхронизации данных;
функции-генераторы и пайплайн;
работа с многими каналами: or, fanin, fanout, etc.

Домашние задания

Пайплайн

Цель

реализовать функцию, конкурентно выполняющую пайплайн заданных функций.

Тема 5

Go внутри. Планировщик

Цель занятия

познакомиться с устройством планировщика Go;
узнать, когда именно происходит «переключение» горутин.

Краткое содержание

основные структуры планировщика: P, M, G;
механизм переключения горутин;
обработка системных и сетевых вызовов планировщиком;
Net Poller.

Тема 6

Go внутри. Память и сборка мусора

Цель занятия

получить базовую информацию об устройстве памяти в Go;
узнать о механизме GC в Go.

Краткое содержание

структура памяти в Linux-процессе;
особенности памяти программы на Go;
выделение и освобождение памяти на стеке и куче;
понятие «escape analysis»;
механизм сборки мусора в Go.

Модуль 3. Стандартные библиотеки и практики

Тема 1

Работа с вводом/ выводом в Go / ДЗ

Цель занятия

узнать тип Buffer;
рассмотреть стандартные интерфейсы: Reader, Scanner, Writer, Closer;
узнать блочные устройства, Seeker;
работа с data intensive apps;
работа с Regex.

Краткое содержание

стандартные интерфейсы io.Reader, io.Writer и io.Closer;
работа с файлом;
последовательные и произвольные доступы и интерфейс io.Seeker;
буферизация ввода/вывода и оптимизация копирования;
Regex.

Домашние задания

Утилита для копирования файлов

Цель

реализовать упрощенный аналог dd (man dd).

Тема 2

Конфигурация и логирование

Цель занятия

познакомиться с механизмами логирования и конфигурирования в Go.

Краткое содержание

различные варианты конфигурации программы;
использование простых форматов конфигурации: .ini, .yaml, .json и т.п.;
чтение конфигурации из окружения;
библиотека для работы с конфигурацией: viper и confita;
стандартная библиотека для логирования;
расширенное логирование с помощью Zap.

Тема 3

CLI / ДЗ

Цель занятия

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

Краткое содержание

обработка аргументов командной строки: flags, pflag, cobra;
работа с переменными окружения;
запуск внешних программ;
создание временных файлов;
обработка сигналов.

Домашние задания

Утилита envdir

Цель

реализовать утилиту envdir (man envdir) на Go.

Тема 4

Рефлексия

Цель занятия

познакомиться с механизмом рефлексии в Go.

Краткое содержание

использование пакета reflect;
плюсы и минусы рефлексии;
reflect.Type и reflect.Value;
примеры использования рефлексии;
пакет unsafe и тип unsafe.Pointer.

Тема 5

Кодогенерация и AST

Цель занятия

познакомиться с механизмом кодогенерации в Go;
разобрать использование AST;
после занятия вы сможете:
использовать библиотеки, использующие кодогенерацию, писать собственные.

Краткое содержание

понятие кодогенерации;
инструмент `go generate`;
полезные библиотеки, использующие кодогенерацию: impl, stringer, jsonenums, easyjson и пр.;
решение;
работа с AST, практическая кодогенерация.

Тема 6

Дженерики в Go / ДЗ

Цель занятия

узнать механизмы обобщенного программирования с использованием дженериков;
рассмотреть внутренние механизмы работы, примеры использования.

Краткое содержание

дженерики в Go;
выведение типов;
практическое использование.

Домашние задания

Валидатор структур

Цель

реализовать функцию-валидатор структур на языке Go.

Тема 7

Профилирование и оптимизация Go программ / ДЗ

Цель занятия

познакомиться с инструментами профилирования в Go.
после занятия вы сможете:
писать бенчмарки в Go;
профилировать программы на Go;
использовать практики оптимизации кода на Go.

Краткое содержание

бенчмарки в Go;
различные оптимизации в Go: преаллокации, переиспользование объектов, работа со строками, регулярными выражениями и пр.;
профилирование в Go;
`go tool proof` и `go tool trace`: профилирование CPU и памяти, flame-диаграммы, дизассемблирование и пр.

Домашние задания

Оптимизация программы

Цель

оптимизировать предоставленную вам программу.

Модуль 4. Работа с сетью и БД

Тема 1

Контекст и низкоуровневые сетевые протоколы / ДЗ

Цель занятия

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

Краткое содержание

контекст в Go (пакет context);
сравнение протоколов TCP и UDP;
стандартные типы net.Dialer и net.Conn;
возможные сетевые проблемы и их отладка;
обеспечение сетевых таймаутов.

Домашние задания

Клиент TELNET

Цель

реализовать крайне примитивный клиент TELNET.

Тема 2

Работа с SQL / ДЗ

Цель занятия

после занятия вы сможете:
работать с SQL в Go.

Краткое содержание

подключение к СУБД и настройка пула соединений;
выполнение SQL-запросов и получение результатов;
обработка ошибок при работе с SQL;
стандартные интерфейсы sql.DB, sql.Rows и sql.Tx;
использование транзакций;
SQL-инъекции и борьба с ними.

Домашние задания

Заготовка сервиса Календарь

Цель

реализовать каркас сервиса «Календарь».

Тема 3

Работа с NoSQL

Цель занятия

на занятии вы познакомитесь с распространенными NoSQL БД, узнаете области их применения;
после занятия вы сможете:
работать с NoSQL в Go.

Краткое содержание

NoSQL БД (Mongo, Redis, Aerospile, Clickhouse);
LUA и отладка скриптов.

Тема 4

Работа с HTTP

Цель занятия

после занятия вы сможете:
работать с HTTP в Go.

Краткое содержание

использование HTTP-клиента;
создание HTTP-сервера;
декораторы и middleware;
REST фреймворки, RPC и OpenAPI;
тестирование HTTP-хэндлеров.

Тема 5

Работа с gRPC

Цель занятия

после занятия вы будете знать внутреннее устройство Protobuf, сможете писать обратно совместимые Protobuf схемы;
генерировать gRPC сервисы.

Краткое содержание

HTTP и gRPC;
Protocol buffers;
прямая и обратная совместимость в Protobuf;
описание API с помощью Protobuf.

Тема 6

Работа с gRPC ч.2 / ДЗ

Цель занятия

углубите знания по работе с gRPC в Go.

Краткое содержание

создание gRPC клиента и сервера;
лучшие практики gRPC;
интерсепторы, надёжность (ретри, задержки), TLS при работе с gRPC.

Домашние задания

API к Календарю

Цель

разработать API к созданному ранее Календарю.

Модуль 5. Микросервисы

Тема 1

Монолит и микросервисы

Цель занятия

получить представление о популярных архитектурах сложных систем.

Краткое содержание

сравнение монолитной и микросервисной архитектур;
плюсы и минусы микросервисов;
понятие 12-факторного приложения;
serverless.

Тема 2

Кэширование

Цель занятия

получить представление о кэшировании, разобрать часто встречающиеся механизмы и стратегии кэширования в Go.

Краткое содержание

cache miss;
cache hit;
прогрев кэша;
LRU, LFU кэш;
потокобезопасности;
инвалидация кэша;
ленивый кэш;
вытеснение данных.

Тема 3

Очереди сообщений / ДЗ

Цель занятия

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

Краткое содержание

очереди сообщений;
событийно-ориентированные архитектуры;
Apache Kafka;
RabbitMQ;
примеры работы с RabbitMQ из Go;
возможные проблемы с очередями: перегрузка, падение обработчиков, сбойные сообщения и пр.

Домашние задания

Кроликизация Календаря

Цель

получить процессы Рассыльщик и Планировщик (в составе сервиса Календарь), связанные между собой очередью сообщений.

Тема 4

Docker

Цель занятия

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

Краткое содержание

контейнеризация;
компоненты Docker: engine, cli, registry;написание Dockerfile и сборка собственных контейнеров;
запуск и управление контейнерами;
передача конфигурации и получение логов из контейнеров;
команды Docker: build, run, up, down, pull, push и пр.;сети в Docker;
docker compose.

Тема 5

K8s / ДЗ

Цель занятия

получить представление об оркестраторе Kubernetes, его архитектуре и внутренних компонентах;
разобраться с ресурсами K8s и манифестами.

Краткое содержание

K8s, minikube, deployment, ingress, service, probes, manifests.

Домашние задания

Докеризация и тестирование календаря

Цель

подготовить микросервисы к запуску в контейнерах и автоматизировать проверку их совместной работы через интеграционные тесты.

Тема 6

Helm

Цель занятия

познакомиться с шаблонизатором Helm и его синтаксисом.

Краткое содержание

helm, шаблоны, встроенные функции.

Тема 7

Мониторинг / ДЗ

Цель занятия

после занятия вы сможете:
мониторить свои приложения.

Краткое содержание

виды мониторинга;
мониторинг Linux-серверов;
системные метрики: LA, CPU, MEM, IO;
мониторинг web-серверов и баз данных;
«health check» в Docker;
количественный мониторинг, система Prometheus и работа с ней из Go.

Домашние задания

Шаблонизация манифестов сервиса календаря

Цель

подготовить манифесты для развертывания приложения в Kubernetes, используя Helm для управления конфигурацией.

Тема 8

Тестирование микросервисов

Цель занятия

писать интеграционные тесты для своего приложения.

Краткое содержание

сравнение интеграционного и юнит-тестирования;
TDD и BDD;
написание тестов на языке Gherkin;
использование godog для запуска тестов;
использование docker-compose.

Тема 9

System Design

Цель занятия

получить представление о разработке высоконагруженных систем.

Краткое содержание

разберем решения домашних заданий;
посмотрим на типичные ошибки;
узнаем некоторые тонкости, о которых редко задумываются.

Модуль 6. Проект

Тема 1

Проектная работа. Вводное

Цель занятия

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.

Краткое содержание

проект и требования к нему;
сроки выполнения и проверка задания;
вопросы.

Домашние задания

Проект

Цель

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

Тема 2

Консультация по проектам

Цель занятия

получить ответы на вопросы, возникшие во время выполнения проектной работы;
получить ответы на вопросы по ДЗ и курсу в целом.

Краткое содержание

возможные вопросы, связанные с проектом;
затруднения, возникшие при выполнении ДЗ;
вопросы по программе.

Тема 3

Итоговое занятие

Цель занятия

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.

Краткое содержание

организационные вопросы;
рынок вакансий по направлению;
статистика курса и вопросы по курсу.