Продвинутый Go-разработчик

01 Кому подойдёт курс

Для Go-разработчиков с опытом 1-2 года:

- Глубже поймёте внутреннее устройство Go
- Разберётесь в лучших практиках
- Научитесь писать производительный, безопасный код
- Повысите свою ценность на рынке

Для Go-разработчиков с опытом более 3 лет:

- Систематизируете знания
- Научитесь проектировать сложные системы и принимать архитектурные решения
- Выйдете на новый уровень экспертизы

Для опытных разработчиков на других языках:

- Быстро войдёте в Go и начнёте применять его в продакшене
- Овладеете ключевыми инструментами
- Расширите карьерные перспективы

02 Чему научитесь

- Писать тесты и проверять функциональность и корректность кода
- Проектировать REST API
- Читать код на Go и понимать решаемую им задачу
- Проводить код-ревью приложений на Go
- Портировать часть продакшен-кода с известного вам языка на Go под руководством более опытного разработчика
- Участвовать в проектировании архитектурных решений для новых сервисов на Go
- Проектировать и писать микросервис самостоятельно
- Перекладывать продуктовые задачи в код на Go;
- Внедрять в сервис на Go информативное и высокопроизводительное логирование;
- Улучшать быстродействие уже написанного кода на Go;
- Реализовывать архитектурные решения и паттерны проектирования на Go;
- Находить и исправлять синтаксические и стилистические ошибки кода;
- Расширять функциональность существующего сложного сервиса;
- Писать продвинутые тесты

03 Как проходит курс

- Сопровождение опытными наставниками
- Теория на платформе Практикума
- Практические тренажёры на платформе
- Воркшопы с экспертами
- Самостоятельные работы с ревью

Продвинутый Go-разработчик

3 проекта в портфолио: сокращатель URL или сервис сбора метрик и алертинга, накопительная система лояльности и менеджер паролей

10 практических работ с проверкой опытным экспертом-ревьюером

Воркшопы с опытными наставниками каждый спринт

Карта курса Продвинутый Go-разработчик

6 месяцев

продолжительность курса

2 ЧАСА

00

Введение и входной тест

2 НЕДЕЛИ | 32 ЧАСА

01

HTTP-приложения, время, контекст

2 НЕДЕЛИ | 27 ЧАСОВ

02

Обмен данными, хранение данных на диске, сжатие данных

2 НЕДЕЛИ | 30 ЧАСОВ

03

Шифрование данных, подключение базы данных, логирование ошибок

2 НЕДЕЛИ | 30 ЧАСОВ

04

Обработка входящих и исходящих запросов в асинхронном режиме

5 НЕДЕЛЬ | 70 ЧАСОВ

05-06

Первый итоговый проект

2 НЕДЕЛИ | 30 ЧАСОВ

07

Паттерны проектирования, антипаттерны программирования, профилирование, стилизация, документация

2 НЕДЕЛИ | 30 ЧАСОВ

08

Статический анализ кода, дженерики и кодогенерация, флаги сборки и компиляции

2 НЕДЕЛИ | 28 ЧАСОВ

09

Экспресс-обзор стандартной библиотеки, генерация случайных чисел, чтение данных и буфер, работа с операционной системой

2 НЕДЕЛИ | 32 ЧАСА

10

Примитивы синхронизации, работа с сетью, protocol buffers и RPC

5 НЕДЕЛЬ | 70 ЧАСОВ

11-12

Второй итоговый проект

Пакеты стандартной библиотеки 01

6 недель
89 часов

В этом модуле расскажем о популярных пакетах Go. Вы научитесь писать и тестировать HTTP-приложения с net/http, управлять передачей данных и временем жизни задач с context, делать запросы в базу данных, познакомитесь с особенностями сериализации и обработки ошибок в Go.

Проекты

Вам предстоит выбрать один из практических треков: «Сервис сокращения URL» или «Сервис сбора метрик и алертинга». Практический трек связывает все задачи курса в единый проект, который вы разрабатываете инкрементально до получения финального результата. Инкремент — это часть кода, которая добавляет новые свойства или функции вашему сервису.

Содержание модуля

01.	**time, context** • время: дата, интервалы, таймеры • контекст: отмена операций и передача данных **net/http** • структура проекта • HTTP-клиент • HTTP-сервер • тестирование HTTP-приложений • HTTP-библиотеки	**Инкремент 1** Ваш код на данном этапе практического трека позволяет раздавать данные клиентам по протоколу HTTP. **Инкремент 2** Код не изменяет функциональности, добавляются тесты для проверки корректности работы системы. **Инкремент 3** Код позволяет более гибко отправлять и принимать данные. **Инкремент 4** Код использует JSON как универсальный формат обмена данными. **Инкремент 5** Код использует переменные окружения ОС для установки состояния сервиса на момент запуска.	3 недели 34 часа

Содержание модуля

02.	**encoding** • структурные теги • стандартные сериализаторы, JSON, XML, Gob • сторонние сериализаторы **os** • переменные окружения • работа с файлами **flag** • аргументы командной строки • compress • сжатие данных	**Инкремент 6** Код использует файлы для персистентного хранения данных на диске. **Инкремент 7** Код использует аргументы командной строки вместо переменных окружения ОС. **Инкремент 8** Код умеет сжимать данные ответа с помощью gzip.	2 недели 27 часов
03.	**hash и crypto** • хеширование и шифрование **database/sql** • пакет gomock, имитация данных для тестирования • обобщённый подход и драйверы • запросы к базе данных • запись в базу данных, SQL-инструкции **errors, log** • интроспекция и логирование ошибок	**Инкремент 9** Код использует JSON как универсальный формат обмена данными. **Инкремент 10** Код использует переменные окружения ОС для установки состояния сервиса на момент запуска. **Инкремент 11** Код использует файлы для персистентного хранения данных на диске. **Инкремент 12** Код использует аргументы командной строки вместо переменных окружения ОС. **Инкремент 13** Код умеет сжимать данные ответа с помощью gzip.	2 недели 30 часов
04.	**Многопоточность** • основы многопоточности • горутины • каналы • синхронизация состояния • паттерны многопоточности	**Инкремент 14** Код использует JSON как универсальный формат обмена данными.	2 недели 30 часов

Конкурентность 02

6 недель
100 часов

Расскажем, что такое многопоточность на уровне операционной системы. Вы узнаете, в чём разница между потоком и процессом, какие существуют виды многопоточности, что происходит при переключении потоков, как бороться с deadlock и datarace. Отвечая на эти и другие вопросы, плавно перейдём к устройству планировщика Go. Потом познакомим с легковесными потоками в языке (goroutines) и другими средствами для написания многопоточных приложений.

Содержание модуля

05-06.	**Первый итоговый проект**	Вы сможете выполнить проект по предложенному нами ТЗ индивидуально или в команде с другими участниками курса. Также вы сможете сделать свой проект, предварительно согласовав его с ментором. Проверять итоговые проекты будут менторы. На выполнение проекта у вас будет 5 недель.	5 недель 70 часов

Паттерны проектирования 03

1 неделя
15 часов

В этом модуле рассмотрим паттерны проектирования и антипаттерны программирования на Go. Вы узнаете, как реализовать наиболее распространённые паттерны проектирования приложений и как избежать неявных ошибок при программировании на Go.

Содержание модуля

07.	**Паттерны проектирования на Go** • порождающие паттерны • структурные паттерны • поведенческие паттерны **Антипаттерны программирования на Go** • постулаты Go • лучшие практики и антипаттерны	1 неделя 15 часов

Тулинг

04

3 недели
45 часов

Сильная сторона Go — встроенные инструменты разработчика. В этом модуле вы научитесь использовать тулинг для стилизации и статического анализа кода, создания документации, кодогенерации и профилирования.

Содержание модуля

07.

Профилирование

- бенчмарки
- инструмент pprof

Стилизация

- форматирование кода: gofmt и goimports

Документация

- генерирование документации командой godoc, спецификация Swagger
- шаблон example_test.go

Инкремент 15
Код исправлен согласно статистике, собранной с помощью pprof.

Инкремент 16
Код отформатирован с помощью gofmt или goimports.

Инкремент 17
Экспортированные функции кода описаны комментариями в формате godoc.

1 неделя
15 часов

08.

Статический анализ кода

- команда go vet
- пакет go/ast
- пакет x/analysis
- пакет staticcheck

Дженерики и кодогенерация

- кодогенерация проектов
- дженерики

Флаги сборки и компиляции

- флаги сборки и компиляции, build constraints

Инкремент 18
Код проверен статическим мультираннером и не содержит ошибок.

Инкремент 19
Код добавляет версию и другие метаданные при компиляции.

2 недели
30 часов

Расширенная стандартная библиотека 05

9 недель
130 часов

В Go многие вещи доступны «из коробки» и приветствуется написание собственных решений на основе стандартной библиотеки. В этом модуле вы завершите знакомство со встроенными пакетами Go и познакомитесь с «расширенной» стандартной библиотекой языка, в которой содержится большое количество полезных пакетов.

Содержание модуля

09.	**Экспресс-обзор стандартной библиотеки** • пакеты стандартной библиотеки (stdlib) • расширенная стандартная библиотека (golang.org/x) **Генерация случайных чисел** • пакеты math/rand и crypto/rand • Чтение данных и буфер • пакет bytes • пакет bufio • работа с операционной системой • пакет os, работа с директориями и процессами • вызов внешних приложений, сигналы	**Инкремент 20** Код умеет использовать асимметричное шифрование. **Инкремент 21** Код умеет работать с файлами конфигов. **Инкремент 22** Код умеет реагировать на сигналы ОС.	2 недели 28 часов
10.	**Примитивы синхронизации** • пакеты sync и x/sync **Работа с сетью** • пакет net, работа с TCP и UDP • IP-адреса **Protocol buffers и gRPC** • protocol buffers и gRPC • разработка gRPC-сервера и клиента	**Инкремент 23** Код умеет обрабатывать запросы с учётом IP-адреса клиента. **Для сервиса сокращения URL:** ограничивает доступ к некоторым хендлерам только для запросов из внутренней сети. **Для сервиса сбора метрик:** сервер принимает метрики только от доверенных IP-адресов, если выставлен специальный флаг. **Инкремент 24** Код умеет обрабатывать запросы по протоколу gRPC.	2 недели 32 часа

Расширенная стандартная библиотека

05

11-12.

Второй итоговый проект
Вы сможете выполнить проект по предложенному нами ТЗ индивидуально или в команде с другими участниками курса. Также вы сможете сделать свой проект, предварительно согласовав его с ментором. Проверять итоговые проекты будут менторы. На выполнение проекта у вас будет 5 недель.

Вебинары
В дополнение к теории и практике менторы будут проводить для вас вебинары один раз в спринт, в конце первой недели. Вебинары нужны для того, чтобы подробнее остановиться на сложных темах, разобрать самые частые ошибки и ответить на вопросы по курсу, Go или программированию в целом.

Сессии 1:1
У каждого участника курса будет возможность один раз в спринт созвониться со своим ментором один на один. В отведённые полчаса вы можете не только задать вопросы по проекту, но и обсудить карьерные ожидания, поделиться опытом разработки или затронуть любую другую интересующую вас тему.

5 недель

70 часов