

Продвинутый Go-разработчик

6 месяцев

продолжительность курса

[Купить курс](#)

Нажмите, чтобы перейти на страницу покупки

2 ЧАСА

00

**Введение
и входной тест**

Шифрование данных,
обработка входящих
и исходящих запросов
в асинхронном режиме

2 НЕДЕЛИ | 30 ЧАСОВ

08

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

3 НЕДЕЛИ | 34 ЧАСА

01

Структура проекта, работа
с HTTP, аргументами
командной строки,
переменными окружения

5 НЕДЕЛЬ | 70 ЧАСОВ

05

**Первый итоговый
проект курса
«Продвинутый
Go-разработчик»**

2 НЕДЕЛИ | 32 ЧАСА

09

Примитивы синхронизации,
работа с сетью, protocol
buffers и gRPC

2 НЕДЕЛИ | 33 ЧАСА

02

Логирование, обмен
данными, сжатие данных,
хранение данных на диске

2 НЕДЕЛИ | 30 ЧАСОВ

06

Паттерны проектирования,
антипаттерны
программирования,
профилирование, стилизация,
документация

5 НЕДЕЛЬ | 70 ЧАСОВ

10

**Второй итоговый
проект курса
«Продвинутый
Go-разработчик»**

2 НЕДЕЛИ | 30 ЧАСОВ

03

Отмена операций,
управление временем
выполнения, подключение
базы данных, интроспекция
ошибок

2 НЕДЕЛИ | 30 ЧАСОВ

07

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

Что вас ждёт на курсе

Вебинары

Один раз в спринт менторы будут проводить вебинары. Они будут дополнять теоретический материал и помогут разобраться с кейсами. Кроме того, это отличная возможность провести работу над ошибками и узнать ответы на волнующие вас вопросы по курсу, Go и программированию в целом.

Чему вы научитесь на курсе Продвинутый Go-разработчик

- 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
 - Переключать 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
-

В этом модуле расскажем о популярных пакетах Go. Вы научитесь писать и тестировать HTTP-приложения, управлять временем выполнения операций, делать запросы к базе данных и обрабатывать ошибки. Также вы познакомитесь с особенностями логирования, чтения и записи в файл, сериализации и десериализации данных.

Проекты

Вам предстоит выбрать один из практических треков: «Сервис сокращения URL» или «Сервис сбора метрик и алертинга». Практический трек связывает все задания 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

№	Продолжительность	Теория	Практика
Спринт 1	3 недели 34 часа	<p>Пакет net/http. Работа с HTTP</p> <ul style="list-style-type: none">Структура проектаСоздание HTTP-сервераТестирование HTTP-приложенияИспользование HTTP-клиентаВыбор HTTP-библиотеки <p>Пакет flag. Чтение аргументов командной строки</p> <ul style="list-style-type: none">Аргументы командной строки <p>Пакет os. Получение переменных окружения</p> <ul style="list-style-type: none">Переменные окружения	<p>Инкремент 1</p> <p>Ваш код на данном этапе практического трека позволяет раздавать данные клиентам по протоколу HTTP.</p> <p>Инкремент 2</p> <p>Код не изменяет функциональности, добавляются тесты для проверки корректности работы системы.</p> <p>Инкремент 3</p> <p>Код позволяет более гибко отправлять и принимать данные.</p> <p>Инкремент 4</p> <p>Код использует аргументы командной строки.</p> <p>Инкремент 5</p> <p>Код использует переменные окружения ОС для установки состояния сервиса на момент запуска.</p>

Содержание модуля

№	Продолжительность	Теория	Практика
Спринт 2	2 недели 33 часа	<p>Пакет log. Логирование в приложении</p> <ul style="list-style-type: none">Стандартные и сторонние пакеты для логированияЛогирование через middleware <p>Пакет encoding. Сериализация и десериализация данных</p> <ul style="list-style-type: none">Основы REST APIСтруктурные тегиСтандартные сериализаторыСторонние сериализаторы <p>Пакет compress. Сжатие данных Оптимизация передачи данных</p> <p>Пакет os. Операции с файлами</p> <ul style="list-style-type: none">Чтение и запись в файл	<p>Инкремент 6 Код умеет логировать события в системе.</p> <p>Инкремент 7 Код использует JSON как универсальный формат обмена данными.</p> <p>Инкремент 8 Код умеет сжимать данные ответа с помощью gzip.</p> <p>Инкремент 9 Код использует файлы для персистентного хранения данных на диске.</p>
Спринт 3	2 недели 30 часов	<p>Пакеты time, context. Отмена операций и управление временем выполнения</p> <ul style="list-style-type: none">Время: даты, интервалы, таймерыКонтекст: отмена операций <p>Пакет database/sql. Взаимодействие с базами данных SQL</p> <ul style="list-style-type: none">Пакет gorm: имитация баз данныхАбстрактный интерфейс и SQL-драйверыЗапросы к базе данныхЗапись в базу данных <p>Пакет errors. Обработка ошибок</p> <ul style="list-style-type: none">Интроспекция ошибок	<p>Инкремент 10 Участник курса подключает сервис к СУБД PostgreSQL и проектирует схему базы данных.</p> <p>Инкремент 11 Код умеет делать запросы к БД и использует её в качестве персистентного хранилища.</p> <p>Инкремент 12 Код умеет эффективно совершать сложные запросы к БД.</p> <p>Инкремент 13 Код умеет определять типы ошибок.</p>
Спринт 4	2 недели 23 часа	<p>Пакеты hash, crypto. Безопасность информации</p> <ul style="list-style-type: none">Хеширование и шифрованиеАвторизация: JSON Web Token	<p>Инкремент 14 Код умеет использовать пакет crypto для шифрования данных.</p>

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 data race. Отвечая на эти и другие вопросы, плавно перейдём к устройству планировщика Go. Потом познакомим вас с легковесными потоками в языке (goroutines) и другими средствами для написания многопоточных приложений.

Содержание модуля

№	Продолжительность	Теория	Практика
Спринт 4	2 недели 23 часа	Многопоточность <ul style="list-style-type: none">• Основы многопоточности• Многопоточность в Go• Каналы• Паттерны многопоточности	Инкремент 15 Код умеет обрабатывать входящие и исходящие данные в асинхронном режиме.
Спринт 5	5 недель 70 часов	Первый итоговый проект Вы напишете итоговый проект по предложенному нами ТЗ или реализуете собственную идею, предварительно согласовав её с ментором. Работать можно индивидуально или в команде с другими участниками курса. Проверять итоговые проекты будут менторы. На выполнение проекта отводится 4 недели, и ещё 1 неделя на проверку.	

Паттерны проектирования

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

Спринт 6	1 неделя 15 часов	Паттерны проектирования на Go <ul style="list-style-type: none">• Порождающие паттерны• Структурные паттерны• Поведенческие паттерны Антипаттерны программирования на Go <ul style="list-style-type: none">• Постулаты Go• Лучшие практики, антипаттерны	
----------	---------------------	---	--

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

№	Продолжительность	Теория	Практика
Спринт 6	1 неделя 15 часов	<p>Профилирование</p> <ul style="list-style-type: none">• Бенчмарки• Инструмент pprof <p>Стилизация</p> <ul style="list-style-type: none">• Форматирование кода: gofmt и goimports <p>Документация</p> <ul style="list-style-type: none">• Генерирование документации командой godoc.• Спецификация Swagger• Шаблон example_test.go	<p>Инкремент 16</p> <p>Код исправлен согласно статистике, собранной с помощью pprof</p> <p>Инкремент 17</p> <p>Код отформатирован с помощью gofmt или goimports.</p> <p>Инкремент 18</p> <p>Экспортируемые функции кода описаны комментариями в формате godoc.</p>
Спринт 7	2 недели 30 часов	<p>Статический анализ кода</p> <ul style="list-style-type: none">• Команда go vet• Пакет go/ast• Пакет x/analysis• Пакет staticcheck <p>Дженерики и кодогенерация</p> <ul style="list-style-type: none">• Кодогенерация• Дженерики <p>Флаги сборки и компиляции</p> <ul style="list-style-type: none">• Флаги сборки и компиляции: build constraints	<p>Инкремент 19</p> <p>Код проверен статическим мультираннером и не содержит ошибок.</p> <p>Инкремент 20</p> <p>Код добавляет версию и другие метаданные при компиляции.</p>

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

№	Продолжительность	Теория	Практика
Спринт 8	2 недели 30 часов	<p>Экспресс-обзор стандартной библиотеки</p> <ul style="list-style-type: none">• Пакеты стандартной библиотеки (stdlib)• Расширенная стандартная библиотека (golang.org/x) <p>Генерация случайных чисел</p> <ul style="list-style-type: none">• Пакеты math/rand и crypto/rand <p>Чтение данных и буфер</p> <ul style="list-style-type: none">• Пакет bytes• Пакет bufio <p>Работа с операционной системой</p> <ul style="list-style-type: none">• Пакет os. Работа с директориями и процессами• Вызов внешних приложений, сигналы	<p>Инкремент 21 Код умеет использовать асимметричное шифрование.</p> <p>Инкремент 22 Код умеет работать с файлами конфигов.</p> <p>Инкремент 23 Код умеет реагировать на сигналы ОС.</p>
Спринт 9	2 недели 30 часов	<p>Примитивы синхронизации</p> <ul style="list-style-type: none">• Пакеты sync и x/sync <p>Работа с сетью</p> <ul style="list-style-type: none">• Пакет net. Работа с TCP и UDP• IP-адреса <p>Protocol buffers и gRPC</p> <ul style="list-style-type: none">• Protocol buffers и gRPC• Разработка gRPC-сервера и клиента	<p>Инкремент 24 Код умеет обрабатывать запросы с учётом IP-адреса клиента.</p> <p>Для сервиса сокращения URL: ограничивает доступ к некоторым хендлерам только для запросов из внутренней сети.</p> <p>Для сервиса сбора метрик: сервер принимает метрики только от доверенных IP-адресов, если выставлен специальный флаг.</p> <p>Инкремент 25 Код умеет обрабатывать запросы по протоколу gRPC.</p>

Второй итоговый проект

№	Продолжительность	Содержание
Спринт 10	5 недель 70 часов	Второй итоговый проект Вы напишете итоговый проект по предложенному нами ТЗ или реализуете собственную идею, предварительно согласовав её с ментором. Работать можно индивидуально или в команде с другими участниками курса. Проверять выполненные участниками курса проекты будут менторы. На выполнение проекта отводится 4 недели, и ещё 1 неделя на проверку.

В этом модуле вы узнаете, как работать с самыми популярными структурами данных с помощью языка Go: хеш-таблицами, связными списками и графами.

Содержание модуля

Продолжительность	Теория
Введение в алгоритмы 27 часов	<ul style="list-style-type: none">• Понятие алгоритма• Понятие сложности• Временная и пространственная сложность• Алгоритмические собеседования
Основные структуры данных 20 часов	<ul style="list-style-type: none">• Массив, связный список, стек, очередь• Сложность операций вставки, поиска и удаления• Представление данных в памяти• Пространственная сложность алгоритма
Рекурсия и сортировки 20 часов	<p>Рекурсия</p> <ul style="list-style-type: none">• Понятие рекурсии• Принцип «разделяй и властвуй»• Бинарный поиск <p>Сортировки</p> <ul style="list-style-type: none">• Квадратичные сортировки• Сортировка слиянием• Быстрая сортировка• Линейная сортировка подсчётом
Хеш-функции и хеш-таблицы 20 часов	<ul style="list-style-type: none">• Абстракция отображения• Понятие и свойства хеш-функции, примеры• Структура данных хеш-таблица• Коллизии и способы их разрешения