

# Карта курса

## Продвинутый

## Go-разработчик

[Купить курс](#)

1 часа	89 часов	105 часов	15 часов	45 часов	60 часов	75 часов
<b>Входная диагностика</b>	<b>Модуль 1</b> Пакеты стандартной библиотеки	<b>Модуль 2</b> Конкурентность	<b>Модуль 3</b> Паттерны проектирования	<b>Модуль 4</b> Тулинг	<b>Модуль 5</b> Расширенная стандартная библиотека	<b>Итоговая аттестация</b>

### Чему вы научитесь на курсе:

- писать тесты и проверять функциональность и корректность кода;
- находить и исправлять синтаксические и стилистические ошибки кода;
- проектировать REST API;
- читать код на Go и понимать решаемую им задачу;
- проводить код-ревью приложений на Go;
- реализовывать архитектурные решения и паттерны проектирования 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;
- внедрять в сервис на Go информативное и высокопроизводительное логирование;
- улучшать быстродействие уже написанного кода на Go;
- проектировать и писать микросервис самостоятельно;
- расширять функциональность существующего сложного сервиса;
- перекладывать продуктовые задачи в код на Go;
- писать продвинутые тесты.



# Пакеты стандартной библиотеки

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 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.

### Тема 1. `time, context`

- время: дата, интервалы, таймеры,
- контекст: отмена операций и передача данных

### Тема 2. `net/http`

- структура проекта,
- HTTP-клиент,
- HTTP-сервер,
- тестирование HTTP-приложений,
- HTTP-библиотеки,

### Тема 3. `encoding`

- структурные теги,
- стандартные сериализаторы, JSON, XML, Gob,
- сторонние сериализаторы

### Тема 4. `os`

- переменные окружения,
- работа с файлами

### Тема 5. `flag`

- аргументы командной строки

### Тема 6. `compress`

- стандартные методы сжатия,
- сторонние библиотеки и форматы

### Тема 7. `hash, crypto`

- хеширование и шифрование,

### Тема 8. `database/sql`

- пакет `gomock`, имитация данных для тестирования,
- обобщённый подход и драйверы,
- запросы к базе данных,
- запись в базу данных, стейтменты и транзакции

## Модуль 2

105 часов

# Конкурентность

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.

### Тема 1. Многопоточность

- основы многопоточности,
- горутины,
- каналы,
- синхронизация состояния,
- паттерны многопоточности,

## Первый выпускной проект

Вы сможете выполнить проектную работу по предложенному нами ТЗ индивидуально или в команде с другими студентами. Также вы сможете сделать свой проект, предварительно согласовав его с ментором. Проверять выпускные проекты будут менторы. На выполнение проекта у вас будет 4 недели.

## Модуль 3

15 часов

# 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.

### Тема 1. Паттерны проектирования на Go

- порождающие паттерны,
- структурные паттерны,
- поведенческие паттерны

### Тема 2. Антипаттерны программирования на Go

- постулаты Go,
- лучшие практики и антипаттерны,
- проникновение в код антипаттернов



## Модуль 4

45 часов

# Тулинг

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

### Тема 1. Профилирование

- бенчмарки,
- использование инструмента pprof для сбора статистики.

### Тема 2. Стилизация

- форматирование кода: gofmt и goimports

### Тема 3. Документация

- генерирование документации командой godoc, спецификация Swagger,
- шаблон example\_test.go

### Тема 4. Статический анализ кода

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

### Тема 5. Дженерики и кодогенерация

- дженерики,
- кодогенерация проектов,
- кодогенерация как дополнение к дженерикам

### Тема 6. Флаги сборки и компиляции

- флаги сборки и компиляции, build constraints

## Модуль 5

60 часов

# Расширенная стандартная библиотека

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.

### Тема 1. Экспресс-обзор стандартной библиотеки

- пакеты стандартной библиотеки (stdlib),
- расширенная стандартная библиотека (golang.org/x)

### Тема 2. Генерация случайных чисел

- использование пакетов math/rand и crypto/rand

### Тема 3. Чтение данных и буфер

- применение пакетов bytes и bufio,
- принципы работы пакетов

### Тема 4. Работа с операционной системой

- пакет os, работа с директориями и процессами,
- вызов внешних приложений, сигналы

### Тема 5. Примитивы синхронизации

- применение примитивов - пакетов sync и x/sync

### Тема 6. Работа с сетью

- применение пакетов - пакет net, работа с TCP и UDP,
- IP-адреса

### Тема 7. Protocol buffers и gRPC

- protocol buffers и gRPC,
- разработка gRPC-сервера и клиента



Итоговая  
аттестация

75 часов

## Второй выпускной проект

Вы сможете выполнить проект по предложенному нами ТЗ индивидуально или в команде с другими студентами. Также вы сможете сделать свой проект, предварительно согласовав его с ментором. Проверять выпускные проекты будут менторы. На выполнение проекта у вас будет 4 недели.

## 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 студента 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.