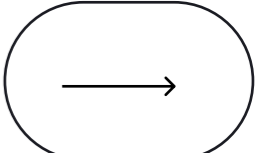


Карты курсов Go-разработчик с нуля и Продвинутый Go-разработчик

Go-разработчик с нуля			Продвинутый Go-разработчик		
Кому подойдёт					
Курс рассчитан на новичков, которые не программировали ранее на других языках или чувствуют себя недостаточно уверенно в бэкенд-разработке.			Курс рассчитан на специалистов с опытом 1-2 года в коммерческой разработке на одном из серверных языков программирования. Для комфортного обучения нужно знать синтаксис Go и понимать основы бэкенд-разработки.		
Продолжительность					
8 месяцев обучения по 10-12 часов в неделю.			6 месяцев обучения по 10-15 часов в неделю.		
Уровень навыков					
По окончании курса студенты будут обладать знаниями и навыками, необходимыми для выполнения задач Go-разработчика уровня junior.			По окончании курса студенты будут обладать знаниями и навыками, необходимыми для выполнения задач Go-разработчика уровня middle.		
<p><i>Умение выполнять задачи указанного уровня не гарантирует, что выпускники смогут легко трудоустроиться на аналогичную позицию. Успех трудоустройства и грейд во многом зависят от прошлого опыта студента. Со знаниями уровня middle, но без опыта коммерческой разработки устроиться на позицию middle-разработчика будет труднее.</i></p>					
Сопровождение на курсе					
<p>Студентов сопровождают наставники, ревьюеры и куратор. Наставники отвечают на вопросы в учебном мессенджере и проводят вебинары. Ревьюеры проверяют практические работы студентов. Куратор помогает с организационными вопросами.</p>			<p>Студентов сопровождают ментор и куратор. Ментор отвечает на вопросы в учебном мессенджере, проводит вебинары и проверяет практические работы студентов. Он ведёт группу до 15 человек на протяжении всего обучения. У каждого студента есть возможность несколько раз созвониться с ментором один на один. Куратор помогает с организационными вопросами.</p>		
Инструменты и технологии, рассматриваемые на курсе					
Go	REST API	YAML	Go	Стилизация	
Git	Swagger	JWT	SQL	Документация	
GitHub	Linux	ORM	REST API	Дженерики	
SQL	HTML	CI/CD	HTTP	Кодогенерация	
Базы данных	JSON	Docker	Базы данных	TCP/UDP	
HTTP	XML	Docker Compose	Многопоточность	gRPC	
			Профилирование	Паттерны	
<p><i>Некоторые технологии, например REST API и базы данных, рассматриваются на обоих курсах, но с разной степенью погружения. На курсе «Go-разработчик с нуля» глубина изучения технологий ограничивается задачами, которые решают разработчики уровня junior, а на курсе «Продвинутый Go-разработчик» рассматриваются задачи, которые решают разработчики уровня middle.</i></p>					



Go-разработчик с нуля

Продвинутый Go-разработчик

Проекты

Студенты выполняют несколько небольших проектов в конце спринтов и один выпускной проект в конце курса.

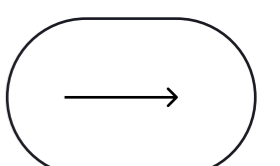
Студенты пишут один сквозной проект на протяжении всего обучения, инкрементально дорабатывая и улучшая функциональность сервиса. Помимо сквозного проекта, на курсе есть два больших выпускных проекта — в середине и в конце обучения.

Программы курсов

Ниже вы найдёте программы курсов «Go-разработчик с нуля» и «Продвинутый Go-разработчик».

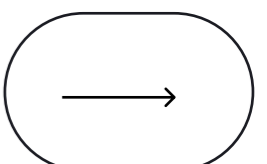
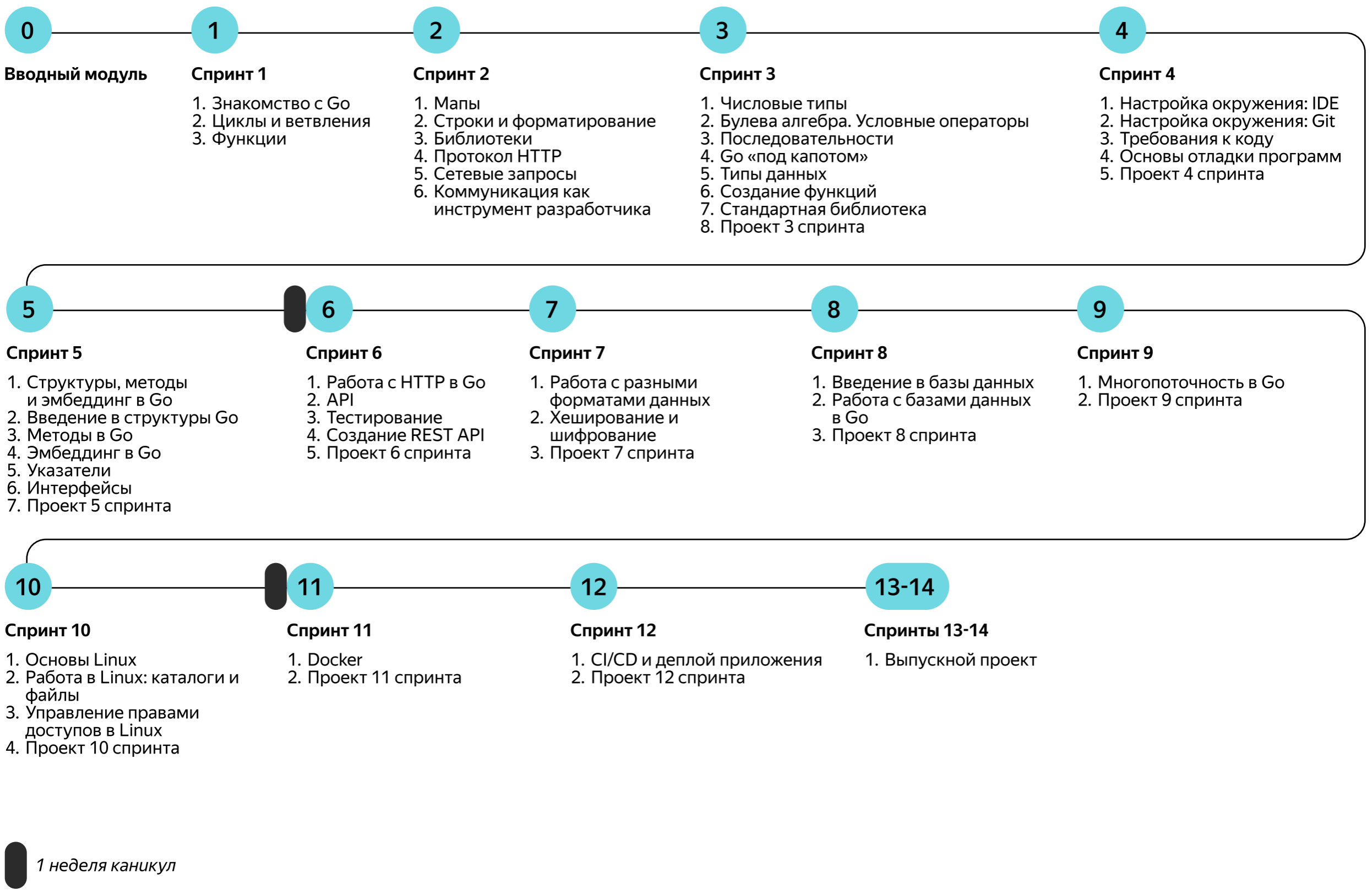
[Go-разработчик с нуля *стр. 4*](#)

[Продвинутый Go-разработчик *стр. 10*](#)



Карта курса **Go-разработчик с нуля**

[Купить курс](#)



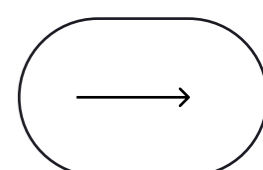
Вебинары

Один раз в спринт наставники проводят вебинары. Они дополняют теоретический материал и помогают разобраться с заданиями. 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

Go-разработчик с нуля:

- изучите синтаксис и базовые возможности языка
- научитесь работать с IDE и GitHub'ом
- сможете создавать бэкенд для веб-приложений
- освоите работу с протоколом HTTP
- изучите архитектуру REST API и научитесь создавать простые API
- научитесь конфигурировать базы данных и подключать их к приложениям
- освоите SQL-запросы для работы с базами данных
- изучите работу с многопоточностью и горутинами — отличительной особенностью языка Go
- сможете разворачивать приложения с использованием Docker и настраивать простой CI/CD
- изучите базовую автоматизацию работы с помощью командной строки Linux



Модуль 1

ОСНОВЫ Go

Модуль посвящён изучению основных положений и концепций программирования на примере языка Go. Рассматриваются как общие для языков бэкенда постулаты и принципы создания кода, так и специфические особенности Go. На этом этапе студенты приобретают навыки создания минимально функциональных программ.

Спринт 1

2 недели
17 часов

Теория

1. Знакомство с Go
2. Циклы и ветвления
3. Функции

Практика

- Первые шаги в создании простейших команд на Go
- Знакомство с особенностями языка через анализ простых кодовых конструкций
- Создание простых циклов и функций

Проект

Студенты создают основу голосового помощника «Алиса» и настраивают базовые функции взаимодействия с пользователем.

Спринт 2

2 недели
23 часа

Теория

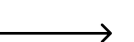
1. Мапы
2. Строки и форматирование
3. Библиотеки
4. Протокол HTTP
5. Сетевые запросы
6. Коммуникация как инструмент разработчика

Практика

- Создание мап, управление ими
- Перебор элементов мапы, использование мапы в качестве прототипа базы данных
- Основы работы со строками
- Импорт библиотек, использование пакетов для решения простых задач
- Изучение структуры сайтов, простые манипуляции с URL
- Передача параметров в URL с помощью Go
- Обработка ошибок

Проект

Студенты продолжают добавлять в голосовой помощник функции: получение информации о погоде и времени на основе списка контактов из адресной книги.



Спринт 3

2 недели
30 часов

Теория

1. Числовые типы
2. Булева алгебра. Условные операторы
3. Последовательности
4. Go «под капотом»
5. Типы данных
6. Создание функций
7. Стандартная библиотека
8. Проект 3 спринта

Практика

- Базовые арифметические операции
- Настройка логических выражений с помощью условных операторов
- Создание циклов
- Продолжение работы со строками
- Углублённая работа с функциями

Проект

Студенты пишут модуль для фитнес-трекера, который будет обрабатывать данные о пользователе.

Спринт 4

2 недели
26 часов

Теория

1. Настройка окружения: IDE
2. Настройка окружения: Git
3. Требования к коду
4. Основы отладки программ
5. Проект 4 спринта
6. Софтскилы для разработчика

Практика

- Основы работы с командной строкой
- Начало работы с IDE
- Настройка Git, базовые операции
- Оптимизация кода небольших программ

Проект

Студенты оптимизируют код программного модуля для фитнес-трекера, используя изученные принципы отладки программ и форматирования кода на Go.

Спринт 5

2 недели
23 часа

Теория

1. Структуры, методы и эмбединг в Go
2. Введение в структуры Go
3. Методы в Go
4. Эмбединг в Go
5. Указатели
6. Интерфейсы
7. Проект 5 спринта

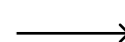
Практика

- Реализация структур в Go на основе бэкенда компьютерной игры
- Создание методов и реализация эмбединга
- Настройка указателей
- Создание интерфейсов

Проект

Студенты внедряют в проект фитнес-трекера новые кодовые конструкции (структуры, методы), настраивают эмбединг, работу указателей и интерфейсов.

1 неделя каникул



Модуль 2

HTTP в Go/REST API

В модуле сделан акцент на веб-разработке как на одном из ключевых компонентов создания современных веб-приложений. Студенты изучают основные принципы сетевого взаимодействия программ и учатся их настраивать.

Спринт 6

2 недели
24 часа

Теория

1. Работа с HTTP в Go
2. API
3. Тестирование

Практика

- Создание сервера
- Создание клиента
- Разработка API и подключение к сервису
- Тестирование программы

Проект

Студенты создают клиент и сервер, которые обмениваются запросами и ответами, а также подключают сторонний API и пишут юнит-тесты для сервиса.

Спринт 7

3 недели
25 часов

Теория

1. Создание REST API
2. Работа с разными форматами данных
3. Хеширование и шифрование
4. Проверочный тест модуля

Практика

- Создание API в соответствии с концепцией REST
- Базовые операции со Swagger
- Преобразование формата данных
- Настройка аутентификации пользователя

Проект

Студенты создают API для готового Telegram-бота с возможностью аутентификации пользователя.

Модуль 3

SQL и базы данных

Модуль погружает студентов в важный раздел бэкенд-разработки, связанный с логикой создания хранилищ данных и организацией их содержания. Также студенты учатся подключать базы данных к уже готовым сервисам.

Спринт 8

2 недели
20 часов

Теория

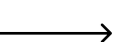
1. Введение в базы данных

Практика

- Обращение к базе данных, основные команды
- Изменение структуры данных в БД
- Настройка подключения сервиса к БД

Проект

Студенты подключают готовую базу данных к заранее описанному сервису и выполняют манипуляции по настройке взаимодействия между сервисом и БД.



Спринт 9

2 недели
26 часов

Теория

1. Работа с базами данных в Go
2. Проверочный тест модуля

Практика

- Работа с пакетом database/sql
- Продвинутый уровень работы с БД

Проект

Студенты конфигурируют базу данных из предыдущего спринта новыми инструментами, оптимизируют взаимодействие сервиса и БД, настраивают миграцию данных в БД.

Модуль 4

Многопоточность в Go

Модуль посвящён многопоточности в Go как одной из уникальных особенностей языка. Студенты учатся оптимизировать нагрузку на приложение при выполнении большого количества операций.

Спринт 10

2 недели
31 час

Теория

1. Многопоточность
2. Проверочный тест модуля

Практика

- Создание горутин
- Использование каналов
- Создание многопоточного кода

Проект

Студенты пишут многопоточный код для параллельного вычисления суммы элементов массива, интегрируют код в веб-сервис, разработанный в предыдущих темах курса.

1 неделя каникул

Модуль 5

Базовый Linux

В модуле рассматриваются основы работы с командной строкой и взаимодействие с программами на уровне операционной системы. Студенты учатся оптимизировать скорость самого процесса разработки и отладки приложений.

Спринт 11

2 недели
23 часа

Теория

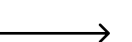
1. Основы Linux
2. Работа на Linux: каталоги и файлы
3. Права доступа в Linux
4. Проверочный тест модуля

Практика

- Изменение иерархии файлов
- Оптимизация работы программы с помощью манипуляций в shell
- Изменения типа прав доступа к файлам

Проект

Студенты используют основные команды Linux: навигация по файловой системе, создание новых каталогов и файлов, изменение прав доступа. Итоговый результат: отладка работы веб-сервиса, разработанного на курсе ранее.



Модуль 6

Основы CI/CD и работы с Docker

В этом модуле студенты отрабатывают навыки упаковки приложения, чтобы подготовить его к передаче пользователям.

Спринт 12

2 недели
30 часов

Теория

1. Docker
2. CI/CD и деплой приложения
3. Проверочный тест модуля

Практика

- Сборка контейнеров для подготовки деплоя веб-сервиса
- Настройка пайплайна с помощью GitHub Actions для дальнейшего деплоя готового приложения

Проект

Студенты выполняют контейнеризацию сервиса, создают образ программы и готовят её к деплою с помощью Docker. Также они настраивают автоматический деплой готового сервиса через GitHub Actions.

Спринт 13-14

4 недели
50 часов

Выпускной проект

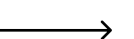
Бэкенд для планировщика задач

В выпускном проекте нужно разработать бэкенд для планировщика задач. Планировщик хранит задачи, каждая из них содержит дедлайн, заголовок и комментарий. Задачи могут повторяться по заданному правилу. Если отметить задачу как выполненную, она переносится на следующую дату по правилу. Нужно спроектировать базу данных и написать веб-сервер, который реализует аналог TODO-листа.

Проект

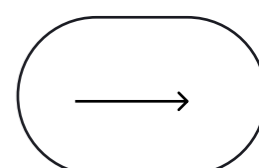
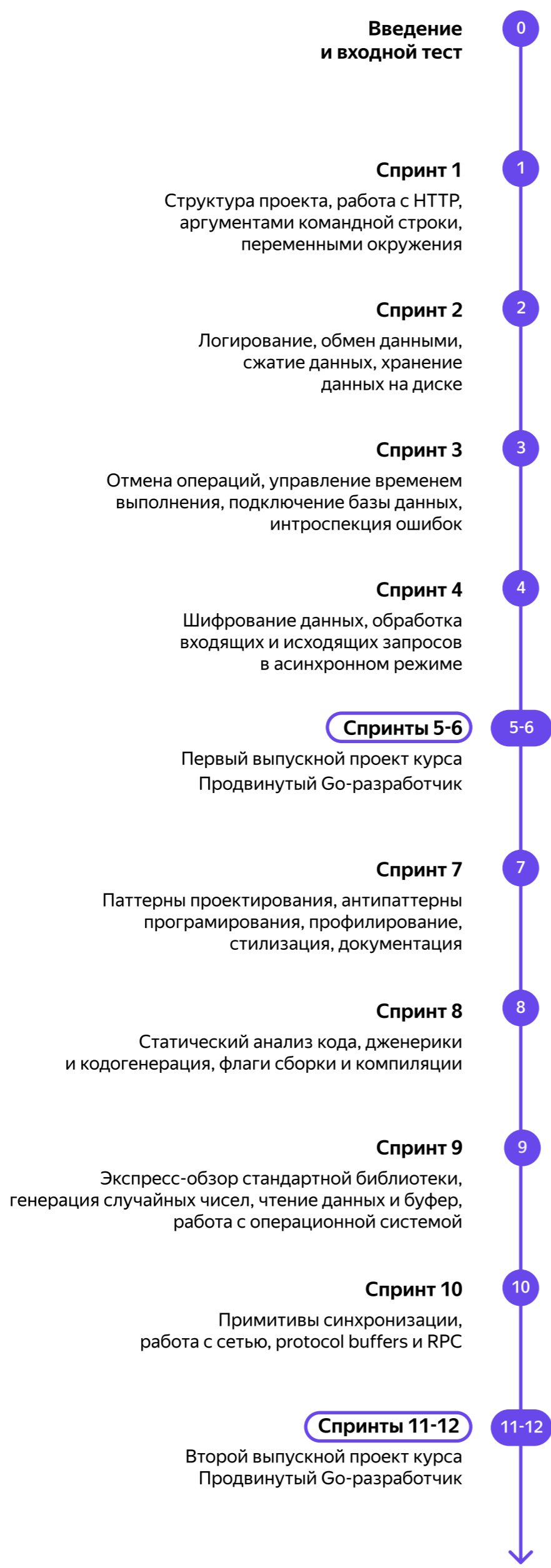
Запуск первой версии веб-сервера, который возвращает готовые файлы фронтенда;

- Создание базы данных;
- Реализация алгоритма для вычисления следующей даты по правилам;
- Реализация API-методов бэкенда;
- Задания со звёздочкой: аутентификация и создание docker-файла.



Карта курса Продвинутый Go-разработчик

[Купить курс](#)



Вебинары

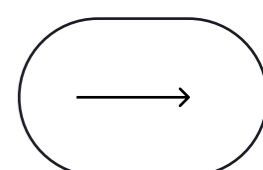
Один раз в спринт менторы будут проводить вебинары. Они будут дополнять теоретический материал и помогут разобраться с заданиями. Кроме того, это отличная возможность провести работу над ошибками и узнать ответы на волнующие вас вопросы по курсу, Go и программированию в целом.

Сессии 1:1

У вас будет возможность один раз в спринт созвониться с ментором один на один. В отведённые полчаса вы сможете задать вопросы по вашему проекту, обсудить карьерные ожидания, поделиться опытом разработки или затронуть любую другую интересующую вас тему.

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;
- 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.



Пакеты стандартной библиотеки

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.

Спринт 1

3 недели
34 часа

Теория

Пакет net/http. Работа с HTTP

- Структура проекта
- Создание HTTP-сервера
- Тестирование HTTP-приложения
- Использование HTTP-клиента
- Выбор HTTP-библиотеки

Пакет flag. Чтение аргументов командной строки

- Аргументы командной строки

Пакет os. Получение переменных окружения

- Переменные окружения

Практика

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 аргументы командной строки.

Инкремент 5

Код использует переменные окружения ОС для установки состояния сервиса на момент запуска.

Спринт 2

2 недели
33 часа

Теория

Пакет log. Логирование в приложении

- Стандартные и сторонние пакеты для логирования
- Логирование через middleware

Пакет encoding. Сериализация и десериализация данных

- Основы REST API
- Структурные теги
- Стандартные сериализаторы
- Сторонние сериализаторы

Пакет compress. Сжатие данных

- Оптимизация передачи данных

Пакет os. Операции с файлами

- Чтение и запись в файл

Практика

Инкремент 6

Код умеет логировать события в системе.

Инкремент 7

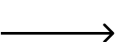
Код использует JSON как универсальный формат обмена данными.

Инкремент 8

Код умеет сжимать данные ответа с помощью gzip.

Инкремент 9

Код использует файлы для персистентного хранения данных на диске.



Спринт 3

2 недели
30 часов

Теория

Пакеты `time`, `context`.
Отмена операций и управление временем выполнения

- Время: даты, интервалы, таймеры
- Контекст: отмена операций

Пакет `database/sql`. Взаимодействие с базами данных SQL

- Пакет `gorm`: имитация баз данных
- Абстрактный интерфейс и SQL-драйверы
- Запросы к базе данных
- Запись в базу данных

Пакет `errors`. Обработка ошибок

- Интроспекция ошибок

Практика

Инкремент 10

Студент подключает сервис к СУБД PostgreSQL и проектирует схему базы данных.

Инкремент 11

Код умеет делать запросы к БД и использует её в качестве персистентного хранилища.

Инкремент 12

Код умеет эффективно совершать сложные запросы к БД.

Инкремент 13

Код умеет определять типы ошибок.

Спринт 4

2 недели
23 часа

Теория

Пакеты `hash`, `crypto`. Безопасность информации

- Хеширование и шифрование
- Авторизация: JSON Web Token

Практика

Инкремент 14

Код умеет использовать пакет `crypto` для шифрования данных.

Модуль 2

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 `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.

Теория

Многопоточность

- Основы многопоточности
- Многопоточность в Go
- Каналы
- Паттерны многопоточности

Практика

Инкремент 15

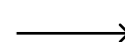
Код умеет обрабатывать входящие и исходящие данные в асинхронном режиме.

Спринты 5-6

5 недель
70 часов

Первый выпускной проект

Вы напишете выпускной проект по предложенному нами ТЗ или реализуете собственную идею, предварительно согласовав её с ментором. Работать можно индивидуально или в команде с другими студентами. Проверять выпускные проекты будут менторы. На выполнение проекта отводится 4 недели, и ещё 1 неделя — на проверку.



Модуль 3

Спринт 7

1 неделя
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.

Теория

Паттерны проектирования на Go

- Порождающие паттерны
- Структурные паттерны
- Поведенческие паттерны

Антипаттерны программирования на Go

- Постулаты Go
- Лучшие практики, антипаттерны

Модуль 4

Спринт 7

1 неделя
15 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.

Теория

Профилирование

- Бенчмарки
- Инструмент pprof

Стилизация

- Форматирование кода: gofmt и goimports

Документация

- Генерирование документации командой godoc. Спецификация Swagger
- Шаблон example_test.go

Практика

Инкремент 16

Код исправлен согласно статистике, собранной с помощью pprof.

Инкремент 17

Код отформатирован с помощью gofmt или goimports.

Инкремент 18

Экспортируемые функции кода описаны комментариями в формате godoc.

Теория

Статический анализ кода

- Команда go vet
- Пакет go/ast
- Пакет x/analysis
- Пакет staticcheck

Дженерики и кодогенерация

- Кодогенерация
- Дженерики

Флаги сборки и компиляции

- Флаги сборки и компиляции: build constraints

Практика

Инкремент 19

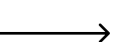
Код проверен статическим мультираннером и не содержит ошибок.

Инкремент 20

Код добавляет версию и другие метаданные при компиляции.

Спринт 8

2 недели
30 часов



Модуль 5

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.

Спринт 9

2 недели
30 часов

Теория

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

Практика

Инкремент 21

Код умеет использовать асимметричное шифрование.

Инкремент 22

Код умеет работать с файлами конфигов.

Инкремент 23

Код умеет реагировать на сигналы ОС.

Спринт 10

2 недели
32 часа

Теория

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

Практика

Инкремент 24

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 25

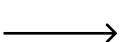
Код умеет обрабатывать запросы по протоколу gRPC.

Спринты 11-12

5 недель
70 часов

Второй выпускной проект

Вы напишете ещё один выпускной проект по предложенному нами ТЗ или реализуете собственную идею, предварительно согласовав её с ментором. Работать можно индивидуально или в команде с другими студентами. Проверять выпускные проекты будут менторы. На выполнение проекта отводится 4 недели, и ещё 1 неделя — на проверку.



Дополнительный модуль

Модуль 6

Алгоритмы и структуры данных

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

Спринт 13

27 часов

Введение в алгоритмы

Теория

Понятие алгоритма
Понятие сложности
Временная и пространственная сложность
Алгоритмические собеседования

Спринт 14

20 часов

Основные структуры данных

Теория

Массив, связный список, стек, очередь
Сложность операций вставки, поиска и удаления
Представление данных в памяти
Пространственная сложность алгоритма

Спринт 15

20 часов

Рекурсия и сортировки

Теория

Рекурсия

- Понятие рекурсии
- Принцип «разделяй и властвуй»
- Бинарный поиск

Сортировки

- Квадратичные сортировки
- Сортировка слиянием
- Быстрая сортировка
- Линейная сортировка подсчётом

Спринт 16

20 часов

Хеш-функции и хеш-таблицы

Теория

Абстракция отображения
Понятие и свойства хеш-функции, примеры
Структура данных хеш-таблица
Коллизии и способы их разрешения