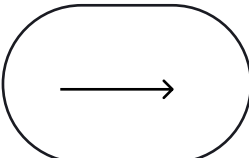
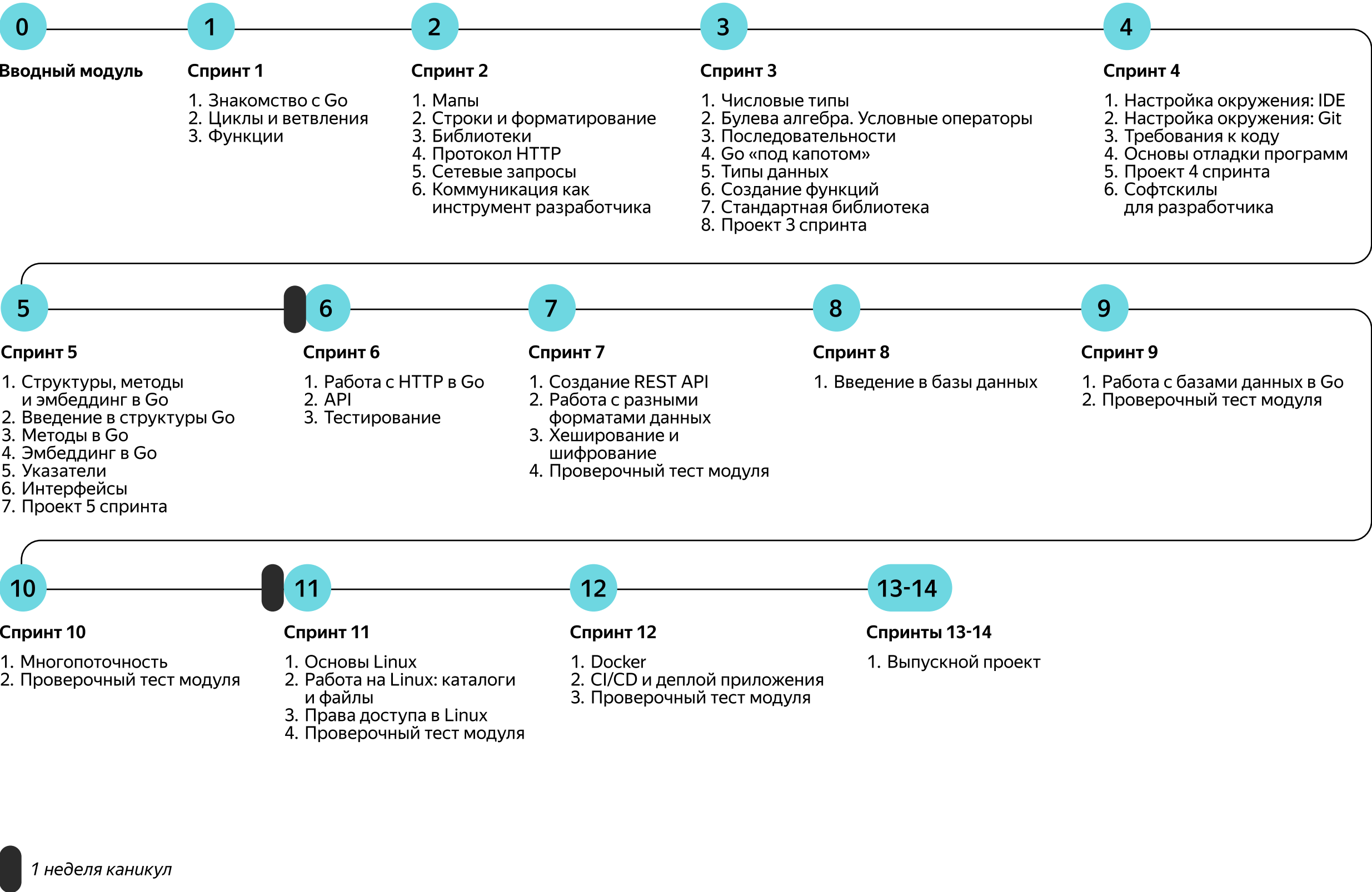


# Карта курса Go-разработчик с нуля

Купить курс



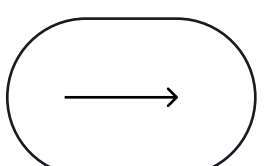
# Вебинары

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:

- писать тесты, проверять функциональность и корректности кода;
- искать и исправлять синтаксические и стилистические ошибки в коде;
- проектировать REST API;
- делать код-ревью приложений на Go;
- применять паттерны проектирования на Go;
- проектировать архитектурные решения для новых сервисов на Go;
- внедрять в сервис на Go информативное и высокопроизводительное логирование.



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

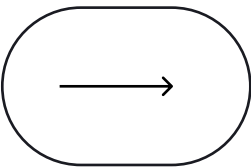
- 1. Мапы
- 2. Строки и форматирование
- 3. Библиотеки
- 4. Протокол HTTP
- 5. Сетевые запросы
- 6. Коммуникация как инструмент разработчика

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

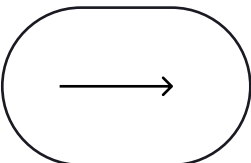
Практика

- 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

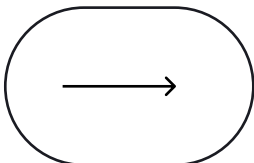
- 1. Введение в базы данных

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

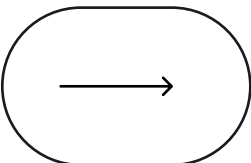
1. Основы Linux
2. Работа на Linux: каталоги и файлы
3. Права доступа в Linux
4. Проверочный тест модуля

### 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 онлайн-маркетплейса

В выпускном проекте студенты разрабатывают бэкенд для магазина билетов по типу «Афиши». Задача — применить полученные на курсе знания и навыки, в частности продемонстрировать умение проектировать и реализовывать сервисы на Go.

Проект

- Реализация клиент-серверной модели приложения
- Реализация RESTful API для аутентификации и авторизации пользователей, листинга и поиска услуг внутри магазина
- Проектирование схемы базы данных
- Использование middleware для логирования событий, поиска ошибок и ограничения количества запросов
- Написание тестов для сервиса
- Деплой бэкенда на облачном сервисе

