

Go-разработчик с нуля

Пояснения к словам иностранного происхождения

Здесь собраны пояснения к словам иностранного происхождения, которые используются в программе курса, но пока не включены в словарную норму русского языка.

01 Кому подойдёт курс

Начинающим без опыта

Если вы только начинаете путь в разработке и у вас нет технического опыта. Мы объясняем всё с нуля — без воды, с фокусом на практике и реальных навыках. Достаточно уметь пользоваться компьютером на базовом уровне — остальному мы научим.

Специалистам из смежных IT-сфер

Если вы уже работаете в смежных отраслях или даже когда-то пробовали писать код, курс поможет вам структурировать знания и понять, как в действительности применять полученные знания на практике. Вы сможете собрать портфолио, а также подготовиться к переходу на новую должность или смене работы.

02 Какие знания и навыки освоите

- **Основы Go:** синтаксис, типы данных, функции, пакеты — сможете писать простые программы.
- **Работа с IDE, Git и GitHub:** коммиты, ветки, слияние, команды — сможете сохранять код, вести проекты и работать вместе с другими разработчиками.
- **Создание REST API:** маршрутизация, обработка запросов, структура ответов — изучите архитектуру, через которую фронтенд или мобильное приложение смогут получать данные, а также научитесь создавать простые API.
- **Создание бэкенда для веб-приложений:** освоите работу с протоколом HTTP.
- **Основы реляционных баз данных и SQL:** научитесь обслуживать базы данных и подключать их к приложениям, а также освоите SQL-запросы для работы с базами данных.
- **Горутины и каналы:** научитесь работать с главным преимуществом Go для быстрой и эффективной работы — параллельностью и многопоточностью.
- **Запуск приложений в Docker и настройка простого CI/CD:** упакуете код так, чтобы он работал везде: у вас, на сервере, у коллег. Поймёте, как код попадает в продукт — без ручных ошибок с помощью CI/CD.
- **Командная строка Linux:** изучите базовую автоматизацию работы с помощью командной строки Linux.
- **Алгоритмы и структуры данных.**

Go-разработчик с нуля

02 Какие знания и навыки освоите

Расширенный тариф:

- **Кэширование и асинхронность:** кэширование в Redis, брокеры сообщений (RabbitMQ), фоновые задачи — научитесь радикально повышать производительность и отзывчивость приложений, разгружая основные системы с помощью кэша и организуя фоновую обработку задач.
- **Событийно-ориентированная архитектура и микросервисы:** Apache Kafka, декомпозиция, трейсинг — освоите переход от монолита к отказоустойчивым микросервисам, которые взаимодействуют через потоки событий, и научитесь настраивать мониторинг их работы.
- **Паттерны надёжности, Resilience Patterns:** Circuit Breaker, gRPC, вклад в опенсорс — изучите продвинутые паттерны для создания устойчивых к сбоям систем, высокопроизводительное межсервисное взаимодействие.
- **Проектирование и интеграция сложных систем:** микросервисная архитектура, Kafka, PostgreSQL — сможете проектировать, разрабатывать и интегрировать компоненты сложной системы для аналитической обработки данных в реальном времени, обеспечивая надёжность и масштабируемость.
- **Навыки работы с опенсорс-проектами:** получите опыт реального контрибьютинга в открытые проекты.
- **Навыки кодинга с ИИ:** узнаете, как грамотно использовать искусственный интеллект для работы программиста, повысить качество кода, найти баги и оптимизировать производительность.
- **Проектный месяц:** работа над групповым синтезированным проектом. Вы примените все полученные навыки на практике в условиях, которые будут имитировать реальную рабочую задачу — от проектирования архитектуры до тестирования и оптимизации. Итог — готовый проект для портфолио.

Go-разработчик с нуля

03	Как проходит обучение	• Теория и практика на платформе Практикума • Команда сопровождения будет помогать вам на протяжении всего обучения • Онлайн-семинары для закрепления материала или разбора сложных моментов • Расширенный тариф: опыт групповой работы
----	-----------------------	---

Что вас ждёт на курсе

Актуальные инструменты и подходы к защите инфраструктуры компании	Практика, основанная на решении реальных рабочих задач	Обучение от экспертов из Яндекса и других крупных компаний
---	--	--

Кто такие Go-разработчики

Пояснения к словам иностранного происхождения

Здесь собраны пояснения к словам иностранного происхождения, которые используются в программе курса, но пока не включены в словарную норму русского языка.

01 В чём суть профессии

Разработчик на Go создаёт надёжный и быстрый бэкенд — «внутреннюю начинку» для веб- и мобильных приложений, которая обрабатывает данные и обеспечивает работу сервисов. Компаниям нужны такие специалисты, чтобы их платформы выдерживали большую нагрузку, быстро работали и не падали при росте числа пользователей.

Go особенно ценят за производительность и простоту в масштабировании — его используют в крупных компаниях вроде Google, Kaspersky и Яндекс.

02 В каких сферах работают

Go-разработчики востребованы в компаниях, где важны высокая производительность, масштабируемость и стабильность систем.

Они часто работают в финтехе (банки, платёжные сервисы), облачных платформах и крупных интернет-сервисах.

03 Какие задачи решают

Создают серверную часть приложений — например, пишут код, который обрабатывает заказы в сервисе доставки: принимает данные, проверяет оплату, отправляет уведомление курьеру.

Строят микросервисы для больших систем — разбивают сложный сервис (например, маркетплейс) на отдельные блоки: один отвечает за каталог, другой — за корзину, третий — за пользователей. Go помогает сделать их быстрыми и устойчивыми.

Разрабатывают инструменты для других разработчиков — например, создают утилиты для автоматического разворачивания приложений или сбора логов, которые упрощают и ускоряют работу всей команды.

Кто такие Go-разработчики

04 Карьерные перспективы

Go — один из ключевых языков в бэкенд-разработке крупных компаний и высоконагруженных сервисов, и спрос на него стабилен в финтехе, IT-продуктовых компаниях, госпроектах и DevOps-направлениях. Даже на джуниор-уровне вакансии часто предлагают стартовый доход выше среднего по сравнению с другими языками — 120 000₽.

Конкуренция среди новичков действительно высока, но она сглаживается глубиной навыков и качеством портфолио. Те, кто умеет не просто писать код, но ещё и разворачивать сервисы, работать с базами, Docker и API, имеют преимущества на рынке.

От джуниора можно вырасти до мидла и синьор-специалиста или уйти в архитектуру решений, DevOps, SRE или облачные платформы — Go часто становится «мостиком» в эти востребованные и высокооплачиваемые направления.

Джуниор может начать свою карьеру с фриланса и опенсорс-проектов, получив навыки поиска на курсе.

Зарплатная вилка

120 000 ₽

Младший Go-разработчик

250 000 ₽

Специалист по разработке на Go

364 000 ₽

Старший Go-разработчик

На курсе — всё, что нужно, чтобы начать карьеру аналитика

Диплом о переподготовке

Непрерывная практика

Портфолио из 10 проектов

Помощь в трудоустройстве

Go-разработчик с нуля

Сравнение тарифов

	Go-разработчик	Go-разработчик расширенный
Длительность	8 месяцев	11 месяцев
Каникулы	6 недель	7 недель
Нагрузка	от 12 часов в неделю	от 12 часов в неделю
Практика	Проекты в каждом спринте	Проекты в каждом спринте
Помощь в трудоустройстве	✓	✓
Вакансии и стажировки от партнёров	✓	✓
Сопровождение: наставник, преподаватель, ревьюеры, куратор, техподдержка	✓	✓
Вебинары	✓	✓ + воркшопы по работе с ИИ и карьерным вопросам
Проекты в портфолио	10 проектов: 9 практических + 1 итоговый	15 проектов: 13 практических + 1 групповой + 1 итоговый
Итоговый проект	Выполняется самостоятельно	Выполняется самостоятельно + предварительный групповой проект

Преимущества расширенного тарифа

01	Больше практики	В базовом тарифе вы сможете выполнить 10 проектов для портфолио. Программа расширенного тарифа предлагает их же + 4 практических проекта для отработки навыков джуниор плюс (Redis, RabbitMQ, Apache Kafka, MSA) + 1 групповой проект, приближенный к реальной работе с заказчиком + итоговый проект. Почему усиливаем тариф именно этими навыками? • Redis: начинающий разработчик, умеющий грамотно кэшировать, сразу становится в разы ценнее, так как решает реальные проблемы производительности. • Понимание асинхронности и основ RabbitMQ: умение писать асинхронный код и работать с очередями задач расширяет круг решаемых задач. • Apache Kafka, MSA: позволит работать с отказоустойчивыми микросервисами, а это повышает ценность специалиста. • Circuit Breaker, gRPC: актуальный навык создания отказоустойчивых систем, который пользуется спросом на рынке. Таким образом, сильный джуниор — это тот, кто эффективно кэширует данные с помощью Redis и может организовать фоновые задачи через брокер (чаще всего RabbitMQ), знаком с микросервисами и умеет создавать отказоустойчивые системы. Кроме того, мы дополнили программу навыком поиска опенсорс-проектов как ещё одной точки входа в профессию для джуниора и добавили групповой проект для получения опыта командной работы.
02	Сопровождение итогового проекта приближено к реальной работе	Как и в любом профессиональном обучении, и базовый, и расширенный тарифы включают итоговый проект. Но в расширенном тарифе ваш групповой проект будет приближен к реальной работе. С вами будет работать наставник, который выступит в роли руководителя: он будет ставить задачи, проверять решения и регулярно давать обратную связь. Такая модель помогает не просто сдать проект, а прочувствовать, как устроен процесс в настоящей команде, и быстрее адаптироваться на первой работе.
03	Экспертиза Яндекса	Материалы в тарифе созданы вместе с командой опытных программистов, которые давно сотрудничают с Яндекс Практикумом на основе их реального опыта. Эксперты вложили в материалы свои знания, подходы и лучшие практики из работы.

Go-разработчик с нуля

8 или 11 месяцев
продолжительность курса

15 проектов
в портфолио

1 НЕДЕЛЯ | 12 ЧАСОВ

00
Подготовительный
блок. Введение

2 НЕДЕЛИ | 24 ЧАСА

01
Computer
Science: введение.
Дополнительный
курс

2 НЕДЕЛИ | 24 ЧАСА

02
Модуль 1.
Основы GO.
1 спринт

2 НЕДЕЛИ | 24 ЧАСА

03
Модуль 1.
Основы GO.
2 спринт

2 НЕДЕЛИ | 24 ЧАСА

04
Модуль 1.
Основы GO.
3 спринт

2 НЕДЕЛИ | 24 ЧАСА

05
Модуль 1.
Основы GO.
4 спринт

ВХОДИТ В СПРИНТ

06
Проект: основа
для голосового
помощника

2 НЕДЕЛИ | 24 ЧАСА

07
Модуль 1.
Основы GO.
5 спринт

ВХОДИТ В СПРИНТ

08
Проект: модуль
для фитнес-трекера

1 НЕДЕЛЯ | 12 ЧАСОВ

09
Computer Science:
база

2 НЕДЕЛИ | 24 ЧАСА

10
Дополнительный
спринт. Основы
работы с ИИ

2 НЕДЕЛИ | 24 ЧАСА

11
Модуль 2. HTTP в Go/
REST API. 6 спринт

ВХОДИТ В СПРИНТ

12
Проект: тестирование
API при помощи
библиотеки Testify

2 НЕДЕЛИ | 24 ЧАСА

13
Модуль 2. HTTP в Go/
REST API. 7 спринт

ВХОДИТ В СПРИНТ

14
Проект: API
для сервера +
его тестирование
при помощи Postman

2 НЕДЕЛИ | 24 ЧАСА

15
Модуль 3. SQL и базы
данных. 8 спринт

ВХОДИТ В СПРИНТ

16
Проект: сервис
для отслеживания
посылок

2 НЕДЕЛИ | 24 ЧАСА

17
Дополнительный
спринт. ИИ для
профессиональных
задач

2 НЕДЕЛИ | 24 ЧАСА

18
Модуль 4.
Многопоточность
в Go. 9 спринт

ВХОДИТ В СПРИНТ

19
Проект:
многопоточная
программа на Go

Go-разработчик с нуля

2 НЕДЕЛИ | 24 ЧАСА

20

Модуль 5. Linux.
10 спринт

ВХОДИТ В СПРИНТ

21

Проект: скрипт
на Bash

1 НЕДЕЛЯ

22

Каникулы

2 НЕДЕЛИ | 24 ЧАСА

23

Модуль 6. Основы CI/
CD и работы с Docker.
11 спринт

ВХОДИТ В СПРИНТ

24

Проект:
настройка Dockerfile
для готового проекта

2 НЕДЕЛИ | 24 ЧАСА

25

Модуль 6. Основы CI/
CD и работы с Docker.
12 спринт

ВХОДИТ В СПРИНТ

26

Проект: создание
и настройка workflow
в Docker Hub

4 НЕДЕЛИ | 48 ЧАСОВ

27

Выпускной проект
базового тарифа

2 НЕДЕЛИ | 24 ЧАСА

28

Отказоустойчивость
и асинхронность.
13 спринт

ВХОДИТ В СПРИНТ

29

Проект расширенного
тарифа 13 спринта:
сервис генерации и
кэширования отчетов

2 НЕДЕЛИ | 24 ЧАСА

30

Отказоустойчивость
и асинхронность.
14 спринт

ВХОДИТ В СПРИНТ

31

Проект расширенного
тарифа 14 спринта:
система уведомлений
на основе событий

Расширенный тариф

Расширенный тариф

Расширенный тариф

Расширенный тариф

2 НЕДЕЛИ | 24 ЧАСА

32

Отказоустойчивость
и асинхронность.
15 спринт

ВХОДИТ В СПРИНТ

33

Проект расширенного
тарифа 15 спринта:
развёртывание
отказоустойчивого
сервиса

2 НЕДЕЛИ | 24 ЧАСА

34

Отказоустойчивость
и асинхронность.
16 спринт

ВХОДИТ В СПРИНТ

35

Проект расширенного
тарифа 16 спринта:
интеграция и
аналитика

Расширенный тариф

Расширенный тариф

Расширенный тариф

Расширенный тариф

1 НЕДЕЛЯ

36

Каникулы

2 НЕДЕЛИ | 24 ЧАСА

37

Computer Science:
мышление (для обоих
тарифов)

4 НЕДЕЛИ | 48 ЧАСОВ

38

Проектный месяц
расширенного
тарифа

4 НЕДЕЛИ | 48 ЧАСОВ

39

Выпускной проект
расширенного
тарифа

1 неделя | 12 часов

Фритрек. В этом блоке мы рассказываем про учёбу и даем базовые понятия языка Go. Это необязательный для прохождения блок, но изучив его, вы будете увереннее чувствовать себя в начале обучения.

Содержание

01. Введение	1. Онбординг 2. Основы Go	1 неделя
--------------	------------------------------	----------

2 недели | 24 часа

Факультативный курс. В этом блоке мы рассказываем о базовых принципах работы компьютера. Это тоже необязательный для прохождения блок, но изучив его, вы будете увереннее чувствовать себя в начале обучения.

Содержание

01. Computer Science: введение	- Как устроен компьютер - Компьютер и операционная система - Интернет: как компьютеры находят друг друга - Алгоритмы: как компьютер принимает решения	2 недели
--------------------------------	---	----------

10 недель | 240 часов

В этом модуле вы разберётесь в базовых понятиях Go, познакомитесь с основами тестирования и отладки кода, а также настроите окружение.

Содержание

01. Спринт 1	- Введение в курс - Переменные - Типы данных - Вывод на экран - Итоговое задание. 1 спринт	Проект	2 недели
02. Спринт 2	- Условный оператор - Циклы - Массивы и слайсы - Коммуникация как инструмент разработчика - Мапы - Итоговое задание. 2 спринт	Проект	2 недели
03. Спринт 3	- Функции - Обработка ошибок - Настройка окружения: IDE - Настройка окружения: GIT - Итоговое задание. 3 спринт	Проект	2 недели
04. Спринт 4	- Работа со временем - Пакеты и модули - Требования к коду - Тестирование - Основы отладки программ - Итоговое задание. 4 спринт	Проект	2 недели
05. Спринт 5	- Структуры, методы и эмбединг: первый взгляд - Введение в структуры Go - Методы в Go - Эмбединг в Go - Указатели - Интерфейсы - Итоговое задание. 5 спринт	Проект	2 недели

2 недели | 24 часа

Факультативный курс. В этом блоке мы рассказываем о базовых принципах работы компьютера, но подробнее, чем во введении. Это необязательный для прохождения блок, но изучив его, вы будете чувствовать себя увереннее.

Содержание

01. Computer Science: введение	- Компьютер изнутри: архитектура и данные - Операционные системы: как программы и железо понимают друг друга - Как работает компьютерная сеть - Алгоритмы: как компьютер решает задачи - Путь данных: от алгоритма до физического сигнала	2 недели
--------------------------------	---	----------

2 недели | 24 часа

Дополнительный блок. Здесь мы рассказываем про основы работы с ИИ и как можно использовать его для своей деятельности.

Содержание

01. Дополнительный спринт. Основы работы с ИИ	- Введение - Введение в ИИ - ИИ для обучения и исследований - Заключение	2 недели
---	--	----------

4 недели | 48 часов

Модуль, где вы познакомитесь с HTTP, работой с файлами и REST API, а также узнаете о хэшировании и шифровании.

Содержание

01. Спринт 6	- Работа с HTTP в Go - Работа с разными форматами данных - Работа с файлами - Создание REST API - Итоговое задание. 6 спринт	Проект	2 недели
02. Спринт 7	- Хэширование и шифрование - Тестирование - Итоговое задание. 7 спринт	Проект	2 недели

2 недели | 24 часа

В этом модуле вы познакомитесь с тем, как устроены базы данных, какие они бывают и как использовать их в Go.

Содержание

01. Спринт 8	- Введение в базы данных - Работа с базами данных в Go - Итоговое задание. 8 спринт	Проект	2 недели
--------------	---	--------	----------

Дополнительный спринт.
ИИ для профессиональных задач

07

2 недели | 24 часа

Дополнительный блок. Здесь вы углубите свои знания о том, как использовать ИИ, и научитесь применять его в разработке.

Содержание

01.	Дополнительный спринт. ИИ для профессиональных задач	- Введение - Базовые кейсы в разработке - Продвинутые кейсы в разработке - Заключение	2 недели
-----	--	---	----------

Модуль 4. Многопоточность в Go

08

2 недели | 24 часа

Модуль, где вы познакомитесь с понятием многопоточности и его использованием в Go.

Содержание

01.	Спринт 9	- Многопоточность - Итоговое задание. 9 спринт	Проект	2 недели
-----	----------	--	--------	----------

Модуль 5. Linux

09

2 недели | 24 часа

В модуле рассматриваются основы работы с командной строкой и взаимодействие с программами на уровне операционной системы.

Содержание

01.	Спринт 10	- Основы Linux - Работа в Linux: каталоги и файлы - Управление правами доступа в Linux - Итоговое задание. 10 спринт	Проект	2 недели
-----	-----------	--	--------	----------

Модуль 6. Основы CI/CD и работы с Docker

10

2 недели | 24 часа

В этом модуле вы освоите работу с контейнерами и образами, научитесь использовать Docker и познакомитесь с CI/CD.

Содержание

01.	Спринт 11	• Docker	Проект	2 недели
02.	Спринт 12	• CI/CD и deploy приложения	Проект	2 недели

Отказоустойчивость и асинхронность

11

8 недель | 96 часов

Этот модуль посвящен механизмам асинхронной обработки (RabbitMQ), кэширования (Redis), брокерам сообщений (Kafka) и основам построения отказоустойчивого сервиса (Circuit Breaker).

Содержание

01.	Спринт 13	• Сервис генерации и кэширования отчётов	Проект	2 недели
02.	Спринт 14	• Система уведомлений на основе событий	Проект	2 недели
03.	Спринт 15	• Развёртывание отказоустойчивого сервиса	Проект	2 недели
04.	Спринт 16	• Интеграция с PostgreSQL	Проект	2 недели

4 недели | 48 часов

Групповой проект, в котором у учащихся будет возможность создать микросервисное приложение своими руками.

Содержание

01.	Спринт 17	• Микросервисы	Проект	2 недели
02.	Спринт 18	• Аналитика и оптимизация	Проект	2 недели

4 недели | 48 часов

Финальное выпускное задание. Здесь для создания приложения придется применить все навыки, полученные за время учёбы.

Содержание

01.	Спринт 19	• Выпускной проект 1.1	Проект	2 недели
02.	Спринт 20	• Выпускной проект 1.2	Проект	2 недели

2 недели | 24 часа

Факультативный курс. В этом блоке мы рассказываем о том, как устроено техническое мышление, об универсальных подходах к решениям задач и грамотных процессах. Блок необязателен для прохождения, однако изучив его, вы будете увереннее чувствовать себя на собеседованиях.

Содержание

01.	Computer Science: мышление	• Как научиться думать как инженер • Система • Визуализация систем и процессов • Документация и коммуникация • Работа с задачами и декомпозицией • Проектирование и системный анализ • Agile и интеграция практик	Самостоятельный проект	2 недели
-----	----------------------------	---	------------------------	----------