

# Rust для действующих разработчиков

## 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

Разработчики на других языках программирования

Погрузятся в Rust и перейдут на новый стек: поймут различия между языками, освоят владение и заимствование, начнут писать безопасный и быстрый код.

Действующие Rust-разработчики

Систематизируют знания, освоят многопоточность, асинхронность и unsafe-код. Научатся проектировать архитектуру, работать с производительностью и писать фронтенд-часть на Rust.

Разработчики, которым интересен блокчейн

Погрузятся в Rust как в основной язык Web3-разработки: рассмотрят протоколы консенсуса и криптографические алгоритмы, научатся писать смарт-контракты под Solana.

02

Какие знания и навыки освоите

- Писать безопасный и производительный код на Rust
- Реализовывать многопоточные и асинхронные приложения
- Создавать HTTP/gRPC API и веб-сервисы
- Работать с CLI-инструментами и WebAssembly
- Проектировать архитектуру приложений
- Использовать FFI, подключать и создавать C-style-библиотеки
- Оптимизировать и отлаживать код
- Разворачивать Rust-приложения и писать интеграционные тесты
- Применять паттерны проектирования и писать идиоматичный код

03

Как проходит курс

- Сопровождение опытными наставниками
- Теория на платформе Практикума
- Практика на реальных кейсах
- Воркшопы с экспертами

## Что вас ждёт на обучении

Диплом

о профессиональной переподготовке или сертификат

6 практических работ

для закрепления навыков

Обучение от экспертов

из Яндекса и других крупных компаний

# Rust для действующих разработчиков

## Сравнение версий

|                                                                                                                     | Rust для действующих разработчиков           | Rust для действующих разработчиков: переход с Python, C++ или C | Rust для действующих разработчиков: погружение в блокчейн |
|---------------------------------------------------------------------------------------------------------------------|----------------------------------------------|-----------------------------------------------------------------|-----------------------------------------------------------|
| Длительность                                                                                                        | 4 месяца обучения                            | 4,5 месяца обучения                                             | 5 месяцев                                                 |
| В каких случаях подойдёт                                                                                            | Если вы хотите углубить свои знания Rust     | Если вы хотите плавно перейти на Rust с других языков           | Если вы хотите писать блокчейн-проекты на Rust            |
| Количество проектных работ с проверкой ревьюером                                                                    | 6 практических заданий с проверкой ревьюером | 6 практических заданий с проверкой ревьюером                    | 7 практических заданий с проверкой ревьюером              |
| Воркшопы с экспертами                                                                                               | 4 воркшопа с экспертами                      | 4 воркшопа с экспертами                                         | 5 воркшопов с экспертами                                  |
| Модуль по основам Rust: Cargo, типы данных, управление памятью, базовый синтаксис и элементы стандартной библиотеки | ✗                                            | ✓                                                               | ✓                                                         |
| Модуль по блокчейн-разработке                                                                                       | ✗                                            | ✗                                                               | ✓                                                         |

# Rust для действующих разработчиков

4 месяца

продолжительность курса

6 проектов

в портфолио

4 воркшопа

с наставником

1 ЧАС



Бесплатный модуль

2 НЕДЕЛИ | 20 ЧАСОВ

00

Основы Rust

Модуль есть в тарифе для переходящих студентов и в тарифе с блокчейном

3 НЕДЕЛИ | 45 ЧАСОВ

01

Основные фичи Rust

3 НЕДЕЛИ | 45 ЧАСОВ

02

Сеть и многопоточность

4 НЕДЕЛИ | 60 ЧАСОВ

03

Асинхронные веб-приложения и CLI-инструменты на Rust

2 НЕДЕЛИ | 40 ЧАСОВ

04

Механизмы FFI и использование unsafe-кода в Rust

2 НЕДЕЛИ | 35 ЧАСОВ

05

Использование инструментов для отладки и оптимизации кода на Rust

2 НЕДЕЛИ | 35 ЧАСОВ

06

Проектирование и рефакторинг

4 НЕДЕЛИ | 40 ЧАСОВ

07

Blockchain-разработка

Модуль есть только в тарифе с блокчейном

# Бесплатный модуль

|       |                                                                                                                                                                                                                                                     |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 час | Познакомьтесь с программой, форматом обучения и подходом. Узнаете, как устроено обучение на Rust, какие навыки вы получите, какие проекты будете делать и чем отличается каждая из трёх версий курса. Всё это поможет понять, подходит ли вам курс. |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

## Содержание

- Привет от команды Яндекс Практикума: знакомство с курсом
- Как мы учим Rust: демо-уроки курса

# Основы Rust00

|                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2 недели   20 часов | Научитесь создавать простые проекты на Rust, описывать собственные типы данных и реализовывать базовую логику. Познакомьтесь с элементами стандартной библиотеки для хранения данных, работы со строками и обработки ошибок. Также освоите утилиты языка для проверки и улучшения кода.<br><br><b>Практика</b><br>Выполните задания локально: напишете код, проверите его с помощью встроенных инструментов Rust и сравните результат с ожидаемым. |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

## Содержание

|                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|----------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 01. Проекты и синтаксис Rust                 | <ul style="list-style-type: none"><li>Введение в Rust</li><li>Установка окружения</li><li>Настройка Rust-проекта с Cargo</li><li>Работа с модулями и зависимостями</li><li>Типы данных, переменные и функции</li><li>Управление потоком: условия, циклы и паттерн-матчинг</li><li>Создание данных, владение ими и перемещение</li><li>Ссылки и заимствование</li><li>Дженерики и трейты</li><li>Форматирование строк и трейты Display, Debug</li></ul> |
| 02. Безопасность и управление памятью        | <ul style="list-style-type: none"><li>Fn-трейты и замыкания</li><li>Обработка ошибок: Result/Option и трейт Error</li><li>Обработка ошибок: best practice</li><li>Умные указатели и внутренняя мутабельность</li></ul>                                                                                                                                                                                                                                 |
| 03. Структуры данных и стандартные коллекции | <ul style="list-style-type: none"><li>Коллекции и работа с ними (Vec, HashMap, String)</li><li>Итераторы и трейты для коллекций</li></ul>                                                                                                                                                                                                                                                                                                              |

3 недели | 45 часов

Научитесь организовывать код в модули, писать тесты, использовать стандартную библиотеку и обрабатывать ошибки. Разберётесь с алгебраическими типами, лайфтаймами, дженериками и трейтами.

Проект

Создадите банковскую систему с переводами и внесением/выводом средств. А также тестовую среду, имитирующую поведение пользователей.

Воркшоп

Разберётесь, как работает borrow checker, и научитесь писать собственные структуры данных на Rust. Увидите, как выполнение требований borrow checker делает ваш код более надёжным и читаемым, не мешая решать задачи.

Содержание

|     |                                                                 |                                                                                                                                                                                                       |
|-----|-----------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 01. | Модульность, инфраструктура и работа со стандартной библиотекой | <ul style="list-style-type: none"><li>Типы данных для хранения банковских аккаунтов</li><li>Абстрактный и стандартный ввод/вывод. Модули и тесты</li></ul>                                            |
| 02. | Алгебраические типы и лайфтаймы                                 | <ul style="list-style-type: none"><li>Алгебраические типы данных и лайфтаймы</li><li>Обработка ошибок: Result, ?, unwrap, надёжная логика</li></ul>                                                   |
| 03. | Дженерики, трейты и абстракции                                  | <ul style="list-style-type: none"><li>Трейты и статический полиморфизм на дженериках</li><li>Макросы. Часть 1: Основы и декларативные макросы</li><li>Макросы. Часть 2: Процедурные макросы</li></ul> |

3 недели | 45 часов

Изучите многозадачность, потокобезопасность и сетевое взаимодействие в Rust. Научитесь реализовывать многопоточные приложения с безопасным доступом к данным, а также разрабатывать сетевые клиенты и серверы с использованием TCP и UDP.

Проект

Разработаете два приложения для стриминга котировок: одно генерирует искусственные данные и передаёт их по UDP, а другое запрашивает и принимает данные, отправляя Ping-сообщения для поддержания соединения. Каждый поток будет работать в отдельном потоке для обработки данных и обеспечения надёжности передачи.

Воркшоп

Разберётесь, как из простой вычислительной задачи вырастает распределённая система с сетевым взаимодействием. Создадите на Rust распределённое приложение по схеме «мастер—воркеры»: настроите сетевой обмен (TCP/UDP), распараллелите вычисления на потоках и реализуете отправку задач и сбор результатов.

Содержание

|                                                 |                                                                                                                                                                                                                                                                                                    |
|-------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 01. Многопоточность и потокобезопасность в Rust | <ul style="list-style-type: none"><li>• Проблемы многопоточности</li><li>• Сигнатура функции с thread::spawn. Трейты Send и Sync. Лайфтаймы и scoped потоки</li><li>• Разделение и синхронизация доступа к данным</li></ul>                                                                        |
| 02. Сетевое взаимодействие: TCP и UDP           | <ul style="list-style-type: none"><li>• TCP vs UDP</li><li>• TCP клиент-сервер на примере управления банковскими ячейками</li><li>• Трансляция данных по UDP на примере мониторинга состояния помещения</li><li>• Опциональные зависимости</li><li>• Трейт-объекты и object safety rules</li></ul> |



4 недели | 60 часов

Изучите создание асинхронных веб-сервисов и CLI-инструментов на Rust. Научитесь работать с многозадачностью, использовать tokio для асинхронного кода, создавать HTTP и gRPC API, а также взаимодействовать с базами данных.

Проект

Разработаете четыре компонента: веб-сервер блога с HTTP и gRPC API, Rust-клиент для выполнения CRUD операций, CLI-клиент для взаимодействия через командную строку и WebAssembly фронтенд для работы с блогом в браузере.

Воркшоп

Разберётесь, как собрать простое веб-приложение "Top ten crates" на Rust с использованием Dioxus и WebAssembly.

Содержание

|                    |                                                                                                                                                                                                                                                               |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 01. Асинхронность  | <ul style="list-style-type: none"><li>• Кооперативная и вытесняющая многозадачность. Синтаксис async/await</li><li>• Асинхронность под капотом</li><li>• Инструменты из tokio</li></ul>                                                                       |
| 02. Rust backend   | <ul style="list-style-type: none"><li>• Сериализация и десериализация данных с serde</li><li>• HTTP-сервер с axum</li><li>• Логирование, сервисы, middleware и типы ошибок в axum</li><li>• Работа с БД и сторонние сервисы</li><li>• Работа с gRPC</li></ul> |
| 03. CLI и Frontend | <ul style="list-style-type: none"><li>• Создание CLI-приложений: структура, ввод, обработка ошибок</li><li>• Основы WebAssembly: кросс-компиляция, wasm-pack, wasm-bindgen</li><li>• Реализация веб-интерфейса на Rust</li></ul>                              |

# Механизмы FFI и использование unsafe-кода в Rust

04

2 недели | 40 часов

Изучите виды линковки (статическая/динамическая), роль ABI и элементы C ABI от calling conventions до data layout, чтобы уверенно стыковать Rust с C и другими языками. Освоите сборку статических и динамических библиотек на Rust, линковку C-библиотек (на этапе компиляции и в рантайме), экспорт функций под C ABI, а также написание unsafe-кода с безопасными обёртками и практиками предотвращения UB.

**Проект**  
Сделаете CLI-приложение для обработки PNG, подключающее плагины как динамические библиотеки. Реализуете два плагина: зеркальный разворот и размытие с параметрами.

## Содержание

|     |                        |                                                                                                                                          |
|-----|------------------------|------------------------------------------------------------------------------------------------------------------------------------------|
| 01. | Виды линковки и Си ABI | <ul style="list-style-type: none"><li>Системные библиотеки и виды линковки</li><li>Си ABI</li></ul>                                      |
| 02. | Библиотеки в Rust      | <ul style="list-style-type: none"><li>Сборка и линковка библиотек с Си ABI в Rust</li><li>Unsafe Rust</li><li>Система плагинов</li></ul> |



# Использование инструментов для отладки и оптимизации кода на Rust

05

2 недели | 35 часов

Научитесь искать ошибки и узкие места в коде, использовать инструменты для отладки, динамического анализа и профилирования. Будете исправлять баги, убирать небезопасные конструкции и оптимизировать производительность, подтверждая улучшения с помощью бенчмарков.

## Проект

Найдёте и исправите ошибки в коде, а также проведёте оптимизацию: напишете бенчмарки, улучшите проблемные места и подтвердите рост производительности.

## Содержание

### 01. Оптимизация

- Основы инструментирования. Как работают отладчики и профилировщики
- Этапы оптимизации производительности. Профилирование и бенчмарки

### 02. Отладка

- Отладка: GDB, LLDB, VSCode
- Динамический анализ с Miri и Valgrind

|                     |                                                                                                                                                                                                                                                                                                                      |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2 недели   35 часов | Улучшите архитектуру, читабельность и реюзабельность кода.                                                                                                                                                                                                                                                           |
|                     | <b>Проект</b><br>Проанализируете предоставленный «плохой» Rust-код, найдёте антипаттерны и выполните рефакторинг, заменив их на более безопасные, идиоматичные и эффективные решения.                                                                                                                                |
|                     | <b>Live coding</b><br>Разберёте паттерны, которые реально используются в популярных Rust-крейтах, и научитесь применять их на практике: Visitor на примере <code>serde</code> , Composite через <code>Service</code> в <code>actix-web</code> и композицию функций на примере парсер-комбинаторов <code>nom</code> . |

## Содержание

|     |                                              |                                                                                                                                                                    |
|-----|----------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 01. | Паттерны. Элементы ООП и ФП                  | <ul style="list-style-type: none"><li>• Best practice и антипаттерны</li><li>• Аналогии с ООП и ФП</li><li>• Rust-паттерны: builder, new-type, state</li></ul>     |
| 02. | Описание контрактов и сохранение инвариантов | <ul style="list-style-type: none"><li>• Контракты и инварианты</li><li>• Инвариант безопасности</li><li>• Инструменты для описания и проверки контрактов</li></ul> |

|                     |                                                                                                                                                                                                                                                                                                 |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 4 недели   40 часов | Вы изучите разработку на блокчейне Solana с использованием Rust — от базовых принципов блокчейна, криптографии и смарт-контрактов до создания, тестирования и аудита безопасности собственного смарт-контракта с интеграцией оракула и SPL-токенов.                                             |
|                     | <b>Проект</b><br>Разработаете и задеплоите смарт-контракт в Solana                                                                                                                                                                                                                              |
|                     | <b>Воркшоп</b><br>Выполните базовые операции в Solana CLI (кошелёк, airdrop, сеть, токен, транзакции) и проверку транзакций в Explorer. Соберёте счётчик на Anchor — запустите тесты, задеплоите программу и подключите её к фронтенду для отправки инкрементов и просмотра состояния on-chain. |

# Содержание

|                                                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-----------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 01. Введение в блокчейн на RUST                           | <ul style="list-style-type: none"><li>• Интро в блокчейн</li><li>• Хеш-функции</li><li>• Алгоритмы цифровой подписи</li><li>• Алгоритмы консенсуса</li></ul>                                                                                                                                                                                                                                                                                                                                                                          |
| 02. Solana. Ключевые концепции и знакомство с разработкой | <ul style="list-style-type: none"><li>• Знакомство с Solana. Ключевые концепции</li><li>• Solana Localnet: CLI/RPC-практика и ключевые концепции</li><li>• Транзакции и инструкции</li><li>• Обзор рантайма SVM</li><li>• Первая on-chain программа: деплой, вызов, логи</li><li>• Аккаунты и хранение данных</li><li>• PDA: программные адреса (без приватного ключа) и подпись программы</li><li>• CPI на практике: invoke, invoke_signed</li><li>• Программа токенов. Token extensions</li><li>• Мини-проект: VaultBoard</li></ul> |
| 03. Погружение в разработку (Solana)                      | <ul style="list-style-type: none"><li>• Anchor Foundations: среда, структура программы и тестовый контур</li><li>• Security by Design: векторы атак Solana и защитные паттерны Anchor</li><li>• Anchor на практике: Оракул</li><li>• SPL Tokens на практике: Token Factory с оплатой в SOL по USD-цене из Оракула</li><li>• Производительность Solana-программ: Compute Units, аккаунты и CPI-стоимость</li><li>• Формальная верификация смарт-контрактов</li><li>• Аудит смарт-контрактов</li></ul>                                  |

## Завершающий модуль

1 час

Подведёте итоги курса

