



# Rust Developer

Rust Developer

Длительность курса: 150 академических часов

# 1 Введение в Rust

## 1 Основные концепции и особенности Rust

### Цели занятия:

перечислить отличия Rust от других языков;  
определить задачи, для которых подходит Rust;  
подготовить окружение для разработки;  
написать, собрать и запустить простейшее приложение на Rust.

### Краткое содержание:

управление ресурсами;  
установка компилятора и вспомогательных утилит;  
создание, сборка и запуск проекта.

---

## 2 Система сборки. Вспомогательные утилиты. Выбор среды разработки

### Цели занятия:

объяснить зачем нужна система сборки;  
собирать приложения и библиотеки;  
использовать утилиты в составе Rust;  
выбрать среду разработки.

### Краткое содержание:

система сборки Cargo;  
линтеры clippy и check;  
форматирование с fmt;  
IDE для Rust.

### Домашние задания

#### 1 Сломанный FizzBuzz

Цель: Применить на практике утилиты fmt и clippy.

Научиться использовать систему сборки cargo.  
Установить и настроить среду разработки

Результатом является:  
Отформатированный и исправленный код  
программы FizzBuzz.  
Установленная среда разработки.

Имеется плохая реализация программы  
FizzBuzz:

<https://pastebin.com/FSrzmqMD>

В ней содержится ошибка, а также несколько  
менее серьёзных проблем.

Исправьте все недостатки в этом коде с  
помощью утилит fmt и clippy, а после этого  
соберите и запустите программу.

О clippy: <https://bit.ly/2Zc9X5E>

О fmt: <https://bit.ly/39r0BVs>

О cargo: <https://doc.rust-lang.ru/book/ch01-03-hello-cargo.html>

---

**Цели занятия:**

создавать простейшие синтаксические конструкции.

**Краткое содержание:**

функции, переменные, встроенные типы данных и замыкания;

концепция владения;

ссылки, указатели и unsafe блоки;

концепция заимствования;

ветвление, циклы, структуры, перечисления и

сопоставление с шаблоном;

массивы и типажи.

---

## **Цели занятия:**

создавать простейшие синтаксические конструкции.

## **Краткое содержание:**

концепция владения;  
ссылки, слайсы, указатели и unsafe блоки;  
концепция заимствования.

## **Домашние задания**

### **1** Умный дом

Цель: Создать прототип двух устройств ""умного дома"": умной розетки и термометра.

Под прототипом подразумеваются наличие функций и типов, описывающих функционал устройств. Реализация не требуется.

Умная розетка умеет:

- Предоставлять текстовое описание.
- Включаться и выключаться.
- Предоставлять данные о текущей потребляемой мощности.

Термометр умеет:

- Выдавать данные о текущей температуре.

Утилита `srpu` не выдаёт предупреждений.  
Код корректно отформатирован с помощью утилиты `fmt`.

Помощь:

- Макрос `todo!()` позволяет не реализовывать функцию сразу, уведомляя компилятор, что разработчик знает, что делает, оставляя функцию пустой.
  - Символ `"" _ ""` перед именем сущности сообщает компилятору, что разработчик в курсе, что эта сущность не используется. Это отключает предупреждение о том, что сущность не используется."
-

**5 Обобщённое  
программирование**

**Цели занятия:**

использовать продвинутое синтаксические конструкции языка.

**Краткое содержание:**

обобщения, трейты и время жизни.

---

**6 Макросы**

**Цели занятия:**

использовать продвинутое синтаксические конструкции языка.

**Краткое содержание:**

декларативные и процедурные макросы.

---

## 7 Дополнительные вопросы синтаксиса

### Цели занятия:

пользоваться особо трудными для понимания элементами синтаксиса Rust.

### Краткое содержание:

список тем формируется исходя из того, с чем студенты испытывали трудности на предыдущих занятиях.

### Домашние задания

#### 1 Умный дом №2

Цель: Разработать прототип библиотеки "умный дом".

Библиотека позволяет сформировать структуру дома и разместить в нём устройства.

- Дом имеет название и содержит несколько помещений.
- Библиотека позволяет запросить список помещений, добавлять и удалять помещения в доме.
- Помещение имеет уникальное название и содержит несколько устройств.
- Устройство имеет уникальное в рамках помещения название.
- Библиотека позволяет добавлять и удалять устройство в доме. А также получать список устройств в помещении.
- Библиотека имеет функцию, принимающую обобщённый тип, предоставляющий текстовую информацию о текущем состоянии устройства по имени комнаты и устройства. Функция должна возвращать текстовый отчёт о полном состоянии дома.

Утилита `clrru` не выдаёт предупреждений. Код корректно отформатирован с помощью утилиты `fmt`.

### 1 Разбиение проекта на файлы и логические блоки. Тесты и примеры

#### Цели занятия:

разделять код на модули;  
использовать сущности из других модулей;  
писать модульные, интеграционные и функциональные тесты.

#### Краткое содержание:

разбиение кода на модули и файлы;  
импорт сущностей из модулей;  
Examples для функциональных тестов;  
тестовые модули для модульных тестов.

#### Домашние задания

##### 1 Делим и тестируем прототип "умного дома"

Цель: Научиться разделять код на модули, использовать сущности из других модулей, писать модульные и функциональные тесты.

Результатом является:  
Многофайловый прототип библиотеки "умный дом" с тестами.

Разделить логически целостные элементы библиотеки "'умный дом'" на отдельные файлы.  
Покрыть тестами требования к библиотеке.  
Требования:

- Дом имеет название и содержит несколько помещений.
- Библиотека позволяет запросить список помещений, добавлять и удалять помещения в доме.
- Помещение имеет уникальное название и содержит несколько устройств.
- Устройство имеет уникальное в рамках помещения название, тип и описание.
- Библиотека позволяет добавлять, получать и удалять любое устройство в доме. Получать список устройств в помещении.
- Типы устройств: термометр, умная розетка.

- Термометр позволяет узнать температуру.
  - Умная розетка позволяет включать и выключать себя. Предоставляет информацию о текущем состоянии и потребляемой мощности.
  - Библиотека позволяет строить отчёт о состоянии всех устройств в доме.
- 

2 **Пакеты и крейты.  
Использование сторонних крейтов.  
Версии и features**

**Цели занятия:**

создавать приложения и библиотеки;  
создавать пакет, содержащий несколько крейтов.

**Краткое содержание:**

Cargo для создания проектов;  
Rust и git.

---

3 **Консультация по вопросам ДЗ (Q&A)**

**Цели занятия:**

получить ответы на вопросы по ДЗ;  
получить ответы на вопросы по приложениям.

**Краткое содержание:**

типичные ошибки при выполнении ДЗ;  
наставники и преподаватели ответят на ваши вопросы.

## 1 Основные перечисления и типы

### Цели занятия:

использовать основные перечисления из стандартной библиотеки;  
реализовывать и использовать типы из стандартной библиотеки.

### Краткое содержание:

Result, Option и другие перечисления;  
Default, Display и другие типы;  
Derive макрос.

### Домашние задания

#### 1 Обработка ошибок в прототипе "умного дома"

Цель: Научиться использовать основные типы и перечисления стандартной библиотеки Rust.

Результатом является:  
Прототип библиотеки "умный дом" с обработкой ошибок.

Функции библиотеки "умный дом" должны возвращать Option в случае, если есть вероятность не получить результат.  
Функции библиотеки "умный дом" должны возвращать Result в случае, отсутствие результата может быть вызвано различными причинами.

---

## 2 Умные указатели и коллекции

### Цели занятия:

использовать умные указатели и коллекции из стандартной библиотеки.

### Краткое содержание:

Box, Rc, Cell, RefCell и их многопоточные аналоги;  
Vec, String, HashMap и другие коллекции.

---

### Цели занятия:

применять сущности из стандартной библиотеки для вводи и вывода данных.

### Краткое содержание:

сокеты, файлы и абстракции над ними.

### Домашние задания

#### 1 Инфраструктура "умного дома"

Цель: Научиться применять сущности из стандартной библиотеки.

Результатом является:  
Инфраструктура библиотеки "умный дом".

Добавить реализацию для следующих требований к библиотеке ""умный дом"":

- Дом имеет название и содержит несколько помещений.
  - Библиотека позволяет запросить список помещений, добавлять и удалять помещения в доме.
  - Помещение имеет уникальное название и содержит несколько устройств.
  - Устройство имеет уникальное в рамках помещения название, тип и описание.
  - Библиотека позволяет добавлять, получать и удалять любое устройство в доме. Получать список устройств в помещении.
  - Типы устройств: термометр, умная розетка.
  - Библиотека позволяет строить отчёт о состоянии всех устройств в доме.
-

## 4 Пишем клиент-серверное приложение

### Цели занятия:

создать простой TCP сервер и клиент.

### Краткое содержание:

приём и передача данных по TCP. Работа с потоками данных.

### Домашние задания

#### 1 Умная розетка

Цель: Научиться использовать сетевые инструменты стандартной библиотеки.

Результатом является:  
Модуль работы с умной розеткой и её имитатор.

"Умная розетка позволяет включать и выключать себя с помощью TCP запроса.  
Предоставляет информацию о текущем состоянии и потребляемой мощности в виде ответа на TCP запрос.  
Также, для проверки нового функционала, реализовать приложение, имитирующее работу умной розетки.

## 1 Проблемы многопоточного кода. Запуск параллельного потока

### Цели занятия:

описать проблемы создания многопоточных приложений;  
запустить выполнения кода в параллельном потоке.

### Краткое содержание:

гонка данных;  
Dead lock;  
создание потока и передача в него кода.

---

## 2 Контроль корректности многопоточного приложения на стадии компиляции

### Цели занятия:

передавать и разделять данные в многопоточном приложении.

### Краткое содержание:

каналы, мютексы, атомарные переменные.

### Домашние задания

#### 1 Термометр

Цель: Научиться использовать отдельный поток для I/O задач.

Результатом является:  
Модуль работы с термометром и его имитатор.

Термометр позволяет узнать температуру, периодически отправляя UDP пакет с данными на заданный адрес.

Для прослушивания UDP предлагается запустить цикл получения датаграмм в отдельном потоке. Также, для проверки нового функционала, реализовать приложение, имитирующее работу термометра.

---

### 3 Пишем свой ThreadPool

#### Цели занятия:

писать многопоточные приложения и библиотеки.

#### Краткое содержание:

пул потоков;  
потоки и задачи.

---

### 4 Типы многозадачности. Асинхронность

#### Цели занятия:

описать различные типы многозадачности;  
писать асинхронный код.

#### Краткое содержание:

вытесняющая и кооперативная многозадачность;  
ключевые слова `async/await`;  
фреймворк `tokio`.

---

### 5 Асинхронность под капотом

#### Цели занятия:

объяснить, как работает асинхронный код.

#### Краткое содержание:

Futures и стейт-машины.

---

6 **Делаем клиент-серверное приложение асинхронным**

**Цели занятия:**

разрабатывать асинхронные приложения.

**Краткое содержание:**

асинхронные сокеты;  
асинхронный клиент и сервер.

**Домашние задания**

1 Асинхронное сетевое взаимодействие

Цель: Научиться разрабатывать асинхронные приложения.

Результатом является:  
Асинхронные модули работы с термометром и умной розеткой.

Модули взаимодействия с умной розеткой и термометром должны предоставлять асинхронный интерфейс.  
Системные потоки не должны создаваться вручную.  
Рекомендуется использовать библиотеку tokio.

---

7 **Межпроцессное взаимодействие и синхронизация**

**Цели занятия:**

передавать данные между процессами.

**Краткое содержание:**

межпроцессное взаимодействие через сокеты, файл, разделяемую память, и другие механизмы.

## 1 Rust и ООП

### Цели занятия:

использовать объектно ориентированные подходы для разработки на Rust.

### Краткое содержание:

инкапсуляция, наследование, полиморфизм и абстракция;  
динамическая диспетчеризация.

### Домашние задания

#### 1 Паттерны проектирования

Цель: Научиться пользоваться инструментами ООП в Rust.

Результатом является:  
Статическая и динамическая реализация паттерна проектирования.

Разработать статическую и динамическую реализацию одного из следующих паттернов проектирования:

- Декоратор
- Компоновщик
- Посетитель

Продемонстрировать использование обеих реализаций на примерах.

---

## 2 Rust и ФП

### Цели занятия:

использовать элементы функционального программирования в Rust.

### Краткое содержание:

декларативное итерирование;  
дополнительный функционал через обобщения.

---

### 3 **Обработка ошибок**

#### **Цели занятия:**

выбрать способ обработки ошибок для своего проекта.

#### **Краткое содержание:**

неустранимые и исправимые ошибки;  
перечисление Result;  
библиотеки для упрощения обработки ошибок.

#### **Домашние задания**

##### 1 Крейт thiserror

Цель: Научиться пользоваться библиотекой thiserror.

Результатом является:  
Рефакторинг библиотеки "умный дом".

Заменить ручную реализацию обработки ошибок в библиотеке "умный дом" на использование крейта thiserror.

---

### 4 **Инварианты**

#### **Цели занятия:**

разрабатывать надёжные и удобные в использовании сущности.

#### **Краткое содержание:**

создание и сохранение инварианта.

---

## 5 Паттерны

### Цели занятия:

удобные приёмы и подходы в разработке на Rust.

### Краткое содержание:

Builder, strategy, new type и другие паттерны.

### Домашние задания

#### 1 Rust паттерны

Цель: Научиться применять паттерны языка Rust.

Результатом является:  
Два Rust паттерна.

Реализовать и протестировать два Rust паттерна.

Список вариантов:

<https://rust-unofficial.github.io/patterns/>

## 1 Популярные утилитарные библиотеки

### Цели занятия:

сериализовать и десериализовать данные;  
создавать случайные числа и последовательность;  
писать логи.

### Краткое содержание:

Serde, rand, log и варианты реализации;  
Другие утилитарные библиотеки.

---

## 2 Логирование, отладка и профилирование

### Цели занятия:

использовать инструменты для логирования, отладки  
и профилирования ПО на Rust

### Краткое содержание:

реализации фасада log;  
отладчики и профилировщики.

---

### Цели занятия:

назвать основные фреймворки и библиотеки для backend разработки;  
создать свой веб-сервер.

### Краткое содержание:

основные фреймворки и библиотеки для backend разработки;  
веб-сервер с использованием actix-web.

### Домашние задания

#### 1 Веб-сервер "умного дома"

Цель: Научиться пользоваться веб-фреймворками Rust.

Результатом является:  
Веб-сервер "умного дома".

Реализовать с использованием веб-фреймворка сервер, реализующий функционал "умного дома":

- Дом имеет название и содержит несколько помещений.
  - Библиотека позволяет запросить список помещений, добавлять и удалять помещения в доме.
  - Помещение имеет уникальное название и содержит несколько устройств.
  - Устройство имеет уникальное в рамках помещения название, тип и описание.
  - Библиотека позволяет добавлять, получать и удалять любое устройство в доме. Получать список устройств в помещении.
  - Типы устройств: термометр, умная розетка.
  - Термометр позволяет узнать температуру.
  - Умная розетка позволяет включать и выключать себя. Предоставляет информацию о текущей потребляемой мощности.
  - Библиотека позволяет строить отчёт о состоянии всех устройств в доме.
-

**Цели занятия:**

описать различные подходы к созданию GUI фреймворков;  
создать GUI приложение с использованием iced.

**Краткое содержание:**

подходы к реализации GUI фреймвоков;  
фреймворк iced.

**Домашние задания****1** Графический интерфейс "умного дома"

Цель: Научиться пользоваться GUI фреймворками Rust.

Результатом является:  
Приложение "умный дом" с GUI.

Реализовать с использованием GUI фреймворка приложение, реализующий функционал ""умного дома"":

- Дом имеет название и содержит несколько помещений.
  - Библиотека позволяет запросить список помещений, добавлять и удалять помещения в доме.
  - Помещение имеет уникальное название и содержит несколько устройств.
  - Устройство имеет уникальное в рамках помещения название, тип и описание.
  - Библиотека позволяет добавлять, получать и удалять любое устройство в доме. Получать список устройств в помещении.
  - Типы устройств: термометр, умная розетка.
  - Термометр позволяет узнать температуру.
  - Умная розетка позволяет включать и выключать себя. Предоставляет информацию о текущей потребляемой мощности.
  - Библиотека позволяет строить отчёт о состоянии всех устройств в доме.
-

## 5 Rust и GameDev

### Цели занятия:

использовать основные игровые и графические движки.

### Краткое содержание:

игровые и графические движки.

---

## 6 Rust и Frontend

### Цели занятия:

описать способы использования ПО на Rust во frontend разработке;  
компилировать ПО на Rust в WebAssembly.

### Краткое содержание:

технология WebAssembly;  
компиляция Rust кода для WebAssembly.

---

## 7 Взаимодействие с другими языками

### Цели занятия:

выбрать представление данных в памяти;  
создать и использовать DLL с C интерфейсом.

### Краткое содержание:

repr(C) и другие представления данных в памяти;  
создание и использование DLL библиотек.

### Домашние задания

#### 1 Динамическая библиотека "умный дом"

Цель: Научиться создавать динамические библиотеки с "C" интерфейсом.

Результатом является:

Динамическая библиотека "умный дом".

Создать динамическую библиотеку ""умный дом"" и тестовое приложение, загружающее её.

- Дом имеет название и содержит несколько помещений.
- Библиотека позволяет запросить список помещений, добавлять и удалять помещения в доме.
- Помещение имеет уникальное название и содержит несколько устройств.
- Устройство имеет уникальное в рамках помещения название, тип и описание.
- Библиотека позволяет добавлять, получать и удалять любое устройство в доме. Получать список устройств в помещении.
- Типы устройств: термометр, умная розетка.
- Термометр позволяет узнать температуру.
- Умная розетка позволяет включать и выключать себя. Предоставляет информацию о текущей потребляемой мощности.
- Библиотека позволяет строить отчёт о состоянии всех устройств в доме.

**1**    **Выбор темы и  
организация  
проектной  
работы**

**Цели занятия:**

выбрать и обсудить тему проектной работы;  
спланировать работу над проектом;  
ознакомиться с регламентом работы над проектом

**Краткое содержание:**

правила работы над проектом и специфика проведения  
итоговой защиты;  
требования к результату проекта и итоговой  
документации.

**Домашние задания**

**1**    Проектная работа

Цель: В этом дз необходимо выбрать и утвердить в чате по ДЗ темы проекта, разработать и презентовать проект.

1. Выбрать тему
  2. Утвердить темы в чате по ДЗ
  3. Презентовать проект
- 

**2**    **Консультация  
по проектам и  
домашним  
заданиям**

**Цели занятия:**

получить ответы на вопросы по проекту, ДЗ и по курсу.

**Краткое содержание:**

вопросы по улучшению и оптимизации работы над проектом;  
затруднения при выполнении ДЗ;  
вопросы по программе.

---

**3 Защита  
проектных  
работ**

**Цели занятия:**

защитить проект и получить рекомендации экспертов.

**Краткое содержание:**

презентация проектов перед комиссией;  
вопросы и комментарии по проектам.

---

**4 Подведение  
итогов курса**

**Цели занятия:**

узнать, как получить сертификат об окончании курса,  
как взаимодействовать после окончания курса с OTUS и  
преподавателями, какие вакансии и позиции есть для  
выпускников (опционально - в России и за рубежом) и  
на какие компании стоит обратить внимание.

**Краткое содержание:**

организационные вопросы;  
рынок вакансий по направлению;  
статистика курса и вопросы по курсу.