

# Оглавление

---

Предисловие .....	12
Благодарности.....	13
О книге.....	14
В чем особенность книги .....	14
Целевая аудитория.....	15
Структура издания.....	15
Как читать книгу.....	16
О коде .....	16
Об авторе.....	18
Иллюстрация на обложке .....	19
От издательства .....	20
О научном редакторе русскоязычного издания .....	20

## Часть I Структурные элементы Rust

<b>Глава 1.</b> Введение в паттерны Rust.....	22
1.1. О чем эта книга .....	23
1.2. Что такое паттерны проектирования.....	24
1.3. Чем эта книга отличается от других.....	28
1.4. Инструменты, необходимые для работы с Rust.....	28
Резюме .....	29
<b>Глава 2.</b> Базовые структурные компоненты Rust.....	30
2.1. Дженерики.....	31
2.1.1. Система типов, отвечающая принципу полноты по Тьюрингу.....	31
2.1.2. Почему дженерики.....	32
2.1.3. Основы дженериков .....	33
2.1.4. Знакомимся с Option в Rust .....	35
2.1.5. Структуры маркеров и фантомные типы.....	36
2.1.6. Ограничения трейтов обобщенных параметров.....	39

2.2. Трейты .....	40
2.2.1. Почему трейты не относятся к объектно-ориентированному программированию.....	40
2.2.2. Что содержится в трейте.....	41
2.2.3. Разбираемся в трейтах на примере объектно-ориентированного кода .....	42
2.2.4. Совмещение дженериков и трейтов.....	45
2.2.5. Автоматический вывод трейтов .....	49
2.2.6. Трейт-объекты.....	50
Резюме .....	54
<b>Глава 3. Поток кода.....</b>	<b>55</b>
3.1. Что такое сопоставление с шаблоном.....	55
3.1.1. Основы сопоставления с шаблоном .....	56
3.1.2. Чистые сопоставления с помощью оператора ? .....	61
3.2. Функциональный Rust .....	64
3.2.1. Основы функционального программирования в Rust.....	65
3.2.2. Захват переменных замыкания .....	66
3.2.3. Знакомство с итераторами .....	68
3.2.4. Получение итератора с помощью with iter(), into_iter() и iter_mut().....	72
3.2.5. Возможности итераторов .....	80
Резюме .....	84
<b>Часть II</b>	
<b>Основные паттерны</b>	
<b>Глава 4. Паттерны в Rust.....</b>	<b>86</b>
4.1. Получение ресурса есть инициализация .....	87
4.1.1. RAII в C и C++ .....	87
4.1.2. RAII в Rust.....	91
4.2. Передача аргументов по значению и по ссылке.....	96
4.2.1. Передача по значению .....	96
4.2.2. Передача по ссылке.....	97
4.2.3. Когда передавать по значению, а когда — по ссылке .....	99
4.3. Конструкторы .....	100
4.4. Видимость членов объекта и доступ к ним.....	102
4.5. Обработка ошибок.....	104
4.6. Глобальное состояние .....	107
4.6.1. Крейт lazy-static.rs.....	110
4.6.2. Крейт once_cell.....	111

## 8 Оглавление

4.6.3. Крейт <code>static_init</code> .....	112
4.6.4. <code>std::cell::OnceCell</code> .....	112
Резюме.....	113
<b>Глава 5. Паттерны проектирования: расширяем функционал</b> .....	114
5.1. Метaprogramмирование с помощью макросов.....	115
5.1.1. Базовый декларативный макрос в Rust.....	116
5.1.2. Когда использовать макросы.....	118
5.1.3. Использование макросов для написания мини-DSL.....	122
5.1.4. Использование макросов для соблюдения принципа «Не повторяйся».....	124
5.2. Необязательные аргументы функций.....	127
5.2.1. Необязательные аргументы в Python.....	127
5.2.2. Необязательные аргументы в C++.....	128
5.2.3. Необязательные аргументы в Rust, вернее, их отсутствие.....	128
5.2.4. Эмуляция необязательных аргументов с помощью трейтов.....	128
5.3. Паттерн «Строитель».....	131
5.3.1. Реализация паттерна «Строитель».....	132
5.3.2. Расширение строителя трейтами.....	134
5.3.3. Расширение строителя макросами.....	135
5.4. Паттерн «Гибкий интерфейс».....	138
5.4.1. Гибкий строитель.....	138
5.4.2. Тестирование гибкого строителя.....	141
5.5. Паттерн «Наблюдатель».....	141
5.5.1. А почему не обратные вызовы?.....	141
5.5.2. Реализация наблюдателя.....	142
5.6. Паттерн «Команда».....	145
5.6.1. Определение паттерна «Команда».....	145
5.6.2. Реализация паттерна «Команда».....	146
5.7. Паттерн «Новый тип».....	149
Резюме.....	152
<b>Глава 6. Проектирование библиотеки</b> .....	153
6.1. Обдумайте дизайн библиотеки.....	154
6.2. Делайте что-то одно и делайте это качественно и правильно.....	155
6.3. Избегайте излишнего абстрагирования.....	155
6.4. Старайтесь использовать простые типы.....	156
6.5. Пользуйтесь инструментами.....	157
6.6. Хорошие художники копируют; великие — воруют (из стандартной библиотеки).....	157
6.7. Документируйте каждый нюанс и приводите примеры.....	158

6.8. Не ломайте код пользователя.....	158
6.9. Помните о состоянии.....	159
6.10. Помните об эстетичности .....	160
6.11. Оценка эргономичности библиотеки Rust .....	160
6.11.1. Вернемся к связанным спискам.....	161
6.11.2. Улучшение дизайна API с помощью <code>rustdoc</code> .....	162
6.11.3. Улучшение связанного списка с помощью дополнительных тестов.....	169
6.11.4. Упрощение отладки библиотеки .....	171
Резюме .....	174

### Часть III

#### Более сложные паттерны

<b>Глава 7.</b> Использование трейтов, дженериков и структур для особых задач .....	176
7.1. Постоянные дженерики.....	177
7.2. Реализация трейтов для типов из других крейтов.....	179
7.2.1. Структуры-обертки .....	179
7.2.2. Использование трейта <code>Deref</code> для развертывания обернутой структуры.....	180
7.3. Трейты-расширения .....	181
7.4. <code>Blanket</code> -трейты.....	183
7.5. Трейты-маркеры.....	185
7.6. Использование структур в качестве тегов .....	187
7.7. Объекты-ссылки .....	189
Резюме .....	194
<b>Глава 8.</b> Конечные автоматы, корутины, макросы и прелюдии .....	195
8.1. Конечный автомат на основе трейтов.....	196
8.2. Корутины.....	200
8.3. Процедурные макросы.....	205
8.4. Прелюдии .....	209
Резюме .....	212

### Часть IV

#### Предотвращение проблем и создание надежного ПО

<b>Глава 9.</b> Неизменяемость .....	214
9.1. Преимущества неизменяемости .....	215
9.2. Почему неизменяемость не панацея .....	217
9.3. Как воспринимать неизменяемые данные .....	218
9.4. Принцип неизменяемости в Rust.....	219

## 10 Оглавление

9.5. Анализ основ неизменяемости в Rust .....	220
9.6. Обеспечение неизменяемости с помощью трейтов .....	223
9.7. Использование типа Cow для обеспечения неизменяемости .....	224
9.8. Получение неизменяемых структур данных с помощью крейтов.....	227
9.8.1. Использование крейта im .....	228
9.8.2. Использование крейта gprds.....	229
Резюме .....	230
<b>Глава 10. Антипаттерны .....</b>	<b>231</b>
10.1. Что такое антипаттерн.....	232
10.2. Использование unsafe .....	233
10.2.1. Что делает unsafe .....	234
10.2.2. Где можно использовать unsafe .....	235
10.2.3. Когда следует использовать unsafe .....	238
10.2.4. Нужно ли беспокоиться насчет unsafe.....	238
10.3. Использование метода unwrap().....	239
10.4. Пренебрежение Vec .....	239
10.5. Излишнее клонирование .....	243
10.6. Использование трейта Deref для эмуляции полиморфизма.....	244
10.7. Глобальные данные и синглтоны.....	248
10.8. Слишком много умных указателей .....	249
10.9. Что дальше?.....	250
Резюме .....	251
<b>Приложение. Установка Rust .....</b>	<b>253</b>
П.1. Установка инструментов для работы с книгой.....	253
П.1.1. Установка для macOS с помощью Homebrew.....	253
П.1.2. Установка для систем Linux.....	253
П.1.3. Установка для Windows .....	254
П.2. Управление rustc и прочими компонентами Rust через rustup .....	254
П.2.1. Установка rustc и других компонентов .....	254
П.2.2. Переключение штатного набора инструментов с помощью rustup.....	255
П.2.3. Обновление компонентов Rust .....	255