

УДК 004.45:004.438Rust
ББК 32.972.11
Б68

Блэнди Дж., Орендорф Дж.

Б68 Программирование на языке Rust / пер. с англ. А. А. Слинкина. – М.: ДМК Пресс, 2018. – 550 с.: ил.

ISBN 978-5-97060-236-2

Rust – новый язык системного программирования, сочетающий высокую производительность и низкоуровневый контроль, характерные для C и C++, с безопасной работой с памятью и потоками. В начале книги рассмотрены типы данных и основные конструкции языка – выражения, модули, структуры, перечисления и образцы. Далее описываются характеристики и универсальные типы. В следующих главах приводятся сведения о замыканиях и итераторах. Наконец в последних главах книги вы найдете исчерпывающую информацию о коллекциях, обработке текста, вводе-выводе, конкурентности, макросах и небезопасном коде.

Издание предназначено для системных программистов, созревших для поиска альтернативы C++.

УДК 004.45:004.438Rust
ББК 32.972.11

Authorized Russian translation of the English edition of Programming Rust, ISBN 9781491927281
© 2018 Jim Blandy, Jason Orendorff.

This translation is published and sold by permission of O'Reilly Media, Inc., which owns or controls all rights to publish and sell the same.

Все права защищены. Любая часть этой книги не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами без письменного разрешения владельцев авторских прав.

ISBN 978-1-491-92728-1 (англ.)
ISBN 978-5-97060-236-2 (рус.)

Copyright © 2016 2018 Jim Blandy, Jason Orendorff
© Оформление, издание, перевод, ДМК Пресс, 2018

Содержание

Предисловие	14
Глава 1. Почему появился Rust?	18
Типобезопасность	19
Глава 2. Краткий обзор Rust	23
Скачивание и установка Rust	23
Простая функция	25
Написание и выполнение автономных тестов	27
Обработка аргументов командной строки	28
Простой веб-сервер	32
Конкурентность	37
Что такое множество Мандельброта	38
Разбор пары аргументов командной строки	42
Отображение пикселей на комплексные числа	44
Рисование множества	46
Запись файла изображения	47
Конкурентная программа рисования множества Мандельброта	48
Выполнение программы рисования множества Мандельброта	52
Невидимая безопасность	54
Глава 3. Базовые типы	55
Машинные типы	58
Целые типы	58
Типы с плавающей точкой	60
Тип bool	62
Символы	62
Кортежи	64
Указательные типы	65
Ссылки	66
Боксы	66
Простые указатели	66
Массивы, векторы и срезы	67
Массивы	67
Вектор	68
Поэлементное построение векторов	71
Срезы	71
Строковые типы	73
Строковые литералы	73
Байтовые строки	74
Строки в памяти	74
Тип String	75
Использование строк	76
Другие типы, похожие на строки	77
Более сложные типы	77

Глава 4. Владение	78
Владение	79
Передача владения	84
Другие операции с передачей	88
Передача владения и поток управления	89
Передача владения и индексированное содержимое	90
Копируемые типы: исключения из правила передачи владения	92
Rc и Arc: совместное владение	95
Глава 5. Ссылки	98
Ссылки как значения	101
Сравнение ссылок в Rust и в C++	101
Присваивание ссылкам	102
Ссылки на ссылки	103
Сравнение ссылок	103
Ссылки не бывают нулевыми	104
Заемствование ссылок на произвольные выражения	104
Ссылки на срезы и объекты характеристик	105
Безопасность ссылок	105
Заемствование локальной переменной	105
Получение ссылок в качестве параметров	108
Передача ссылок в качестве аргументов	110
Возврат ссылок	111
Структуры, содержащие ссылки	112
Различные параметрические времена жизни	114
Опускание параметрического времени жизни	115
Разделяемость и изменяемость	117
Оружие против моря объектов	123
Глава 6. Выражения	126
Язык выражений	126
Блоки и точки с запятой	127
Объявления	128
if и match	130
Циклы	132
Выражение return	134
Зачем в Rust цикл loop	135
Вызовы функций и методов	136
Поля и элементы	137
Операторы ссылки	139
Арифметические, поразрядные, логические операторы и операторы сравнения	139
Присваивание	140
Приведение типов	140
Замыкания	141
Приоритеты и ассоциативность	142
Что дальше	144
Глава 7. Обработка ошибок	145
Паника	145
Раскрутка стека	146

Снятие процесса	147
Тип Result	147
Обнаружение ошибок	148
Псевдонимы типа Result	149
Печать информации об ошибках	150
Распространение ошибок	151
Работа с ошибками нескольких типов	152
Ошибки, которых «не может быть»	153
Игнорирование ошибок	155
Обработка ошибок в main()	155
Объявление пользовательского типа ошибки	156
Почему именно тип Result?	157
Глава 8. Крейты и модули	158
Крейты	158
Сборочные профили	161
Модули	161
Модули в отдельных файлах	162
Пути и импорт	164
Стандартная прелюдия	166
Артикулы – строительные блоки в Rust	166
Превращение программы в библиотеку	168
Каталог src/bin	170
Атрибуты	171
Тесты и документация	173
Интеграционные тесты	175
Документация	176
Дос-тесты	178
Задание зависимостей	180
Версии	181
Cargo.lock	182
Публикация крейтов на сайте crates.io	183
Рабочие пространства	185
Прочие вкусности	185
Глава 9. Структуры	187
Структуры с именованными полями	187
Кортежеподобные структуры	190
Безэлементные структуры	191
Размещение структуры в памяти	191
Определение методов с помощью ключевого слова impl	192
Универсальные структуры	195
Структуры с параметрическим временем жизни	196
Выведение стандартных характеристик для структурных типов	197
Внутренняя изменяемость	198
Глава 10. Перечисления и образцы	202
Перечисления	203
Перечисления, содержащие данные	205
Перечисления в памяти	206
Обогащенные структуры данных на основе перечислений	206

Универсальные перечисления.....	208
Образцы.....	211
Литералы, переменные и метасимволы в образцах.....	213
Кортежные и структурные образцы.....	215
Ссылочные образцы.....	216
Сопоставление с несколькими возможностями.....	218
Охранные выражения.....	219
@-образцы.....	219
Где еще используются образцы.....	220
Построение двоичного дерева.....	221
Общая картина.....	222
Глава 11. Характеристики и универсальные типы.....	224
Использование характеристик.....	226
Объекты характеристик.....	227
Размещение объекта характеристики в памяти.....	228
Универсальные функции.....	229
Что использовать.....	232
Определение и реализация характеристик.....	233
Методы по умолчанию.....	234
Характеристики и сторонние типы.....	235
Употребление Self в характеристиках.....	237
Подхарактеристики.....	238
Статические методы.....	239
Полностью квалифицированные вызовы методов.....	240
Характеристики, определяющие связи между типами.....	241
Ассоциированные типы, или Как работают итераторы.....	242
Универсальные характеристики, или Как работает перегрузка операторов.....	245
Парные характеристики, или Как работает rand::random().....	245
Обратное конструирование ограничений.....	247
Заключение.....	250
Глава 12. Перегрузка операторов.....	251
Арифметические и поразрядные операторы.....	252
Унарные операторы.....	254
Бинарные операторы.....	255
Составные операторы присваивания.....	255
Сравнение на равенство.....	257
Сравнение на больше-меньше.....	260
Index и IndexMut.....	261
Прочие операторы.....	264
Глава 13. Вспомогательные характеристики.....	265
Характеристика Drop.....	266
Характеристика Sized.....	268
Характеристика Clone.....	271
Характеристика Copy.....	272
Характеристики Deref и DerefMut.....	273
Характеристика Default.....	276
Характеристики AsRef и AsMut.....	277
Характеристики Borrow и BorrowMut.....	279

Характеристики From и Into	280
Характеристика ToOwned	282
Borrow и ToOwned за работой: скромное копирование при записи	283
Глава 14. Замыкания	285
Захват переменных	286
Замыкания с заимствованием	287
Замыкания с кражей	287
Типы функций и замыканий	289
Производительность замыканий	291
Замыкания и безопасность	292
Замыкания, которые убивают	293
FnOnce	293
FnMut	295
Обратные вызовы	297
Эффективное использование замыканий	299
Глава 15. Итераторы	302
Характеристики Iterator и IntoIterator	303
Создание итераторов	305
Методы iter и iter_mut	305
Реализации характеристики IntoIterator	305
Метод drain	307
Другие источники итераторов	308
Адаптеры итераторов	309
map и filter	309
filter_map и flat_map	311
scan	313
take и take_while	314
skip и skip_while	315
peekable	316
fuse	317
Обратимые итераторы и rev	317
inspect	318
chain	319
enumerate	319
zip	320
by_ref	321
cloned	322
cycle	322
Потребление итераторов	323
Простое аккумулятивное: count, sum, product	323
max, min	324
max_by, min_by	324
max_by_key, min_by_key	324
Сравнение последовательностей	325
any и all	326
position, rposition и ExactSizeIterator	326
fold	327
nth	327
last	328

find	328
Построение коллекций: collect и FromIterator	328
Характеристика Extend	330
partition	331
Реализация собственных итераторов	332
Глава 16. Коллекции	336
Обзор	337
Тип Vec<T>	338
Доступ к элементам	338
Итерирование	340
Увеличение и уменьшение вектора	340
Соединение	343
Расщепление	343
Перестановка элементов	345
Сортировка и поиск	345
Сравнение срезов	347
Случайные элементы	347
В Rust отсутствуют ошибки недействительности	347
Тип VecDeque<T>	348
Тип LinkedList<T>	350
Тип BinaryHeap<T>	350
Типы HashMap<K, V> и BTreeMap<K, V>	351
Записи	354
Обход отображения	356
Типы HashSet<T> и BTreeSet<T>	356
Обход множества	357
Когда равные значения различны	357
Операции над множествами как единым целым	358
Хеширование	359
Применение пользовательского алгоритма хеширования	360
За пределами стандартных коллекций	361
Глава 17. Строки и текст	362
Общие сведения о Юникоде	362
ASCII, Latin-1 и Юникод	362
UTF-8	363
Направление текста	365
Символы (char)	365
Классификация символов	365
Работа с цифрами	366
Преобразование регистра символов	366
Преобразование в целое число и обратно	367
Типы String и str	367
Создание значений типа String	368
Простая инспекция	369
Дописывание и вставка текста	369
Удаление текста	371
Соглашения о поиске и итерировании	371
Образцы для поиска текста	372
Поиск и замена	372

Обход текста.....	373
Усечение	375
Преобразование регистра	376
Создание значений других типов из строк	376
Преобразование других типов в строки	376
Заимствование в виде других текстообразных типов	377
Доступ к байтам текста в кодировке UTF-8	378
Порождение текста из данных в кодировке UTF-8	378
Откладывание выделения памяти	379
Строки как универсальные коллекции	381
Форматирование значений	381
Форматирование текстовых значений	383
Форматирование чисел.....	384
Форматирование прочих типов	385
Форматирование значений для отладки	386
Форматирование указателей для отладки.....	387
Ссылка на аргументы по индексу или по имени	387
Динамическая ширина и точность.....	388
Форматирование пользовательских типов	389
Применение языка форматирования в своем коде	391
Регулярные выражения	392
Основы работы с Regex	392
Ленивое построение значений типа Regex.....	393
Нормализация.....	394
Формы нормализации	395
Крейт unicode-normalization.....	396
Глава 18. Ввод и вывод	398
Читатели и писатели	399
Читатели.....	400
Буферизованные читатели	401
Чтение строк	402
Собирание строк.....	405
Писатели	405
Файлы	406
Поиск.....	407
Другие типы читателей и писателей.....	407
Двоичные данные, сжатие и сериализация.....	409
Файлы и каталоги	410
Типы OsStr и Path	410
Методы типов Path и PathBuf	412
Функции доступа к файловой системе	413
Чтение каталогов.....	414
Платформенно-зависимые средства	416
Средства сетевого программирования	417
Глава 19. Конкурентность	420
Вилочный параллелизм	421
Функции spawn и join.....	423
Обработка ошибок в потоках	425
Разделение неизменяемых данных между потоками	426

Rayon	428
И снова о множестве Мандельброта	430
Каналы.....	431
Отправка значений	433
Получение значений	436
Выполнение конвейера.....	437
Возможности и производительность каналов	438
Потокобезопасность: Send и Sync	440
Отправка объектов почти любого итератора по каналу.....	442
За пределами конвейеров.....	443
Разделяемое изменяемое состояние.....	444
Что такое мьютекс?	444
Мьютексы в Rust.....	446
mut и Mutex.....	448
Почему мьютексы – не всегда хорошая идея	448
Взаимоблокировка	449
Отравленные мьютексы.....	450
Каналы с несколькими производителями и мьютексом	450
Блокировки чтения-записи (RwLock)	451
Условные переменные (Condvar).....	452
Атомарные типы	453
Глобальные переменные.....	455
Как выглядит написание конкурентного кода на Rust	457
Глава 20. Макросы.....	458
Основы макросов.....	459
Основы макрорасширения	460
Непредвиденные последствия	461
Повторение	463
Встроенные макросы.....	465
Отладка макросов	466
Макрос json!	467
Типы фрагментов.....	468
Рекурсия в макросах.....	471
Использование характеристик совместно с макросами	471
Области видимости и гигиена.....	473
Импорт и экспорт макросов	476
Предотвращение синтаксических ошибок при сопоставлении.....	477
За пределами macro_rules!	478
Глава 21. Небезопасный код.....	480
Небезопасность от чего?	481
Unsafe-блоки	482
Пример: эффективный тип ASCII-строки	483
Unsafe-функции	485
Unsafe-блок или unsafe-функция?.....	487
Неопределенное поведение.....	488
Небезопасные характеристики.....	490
Простые указатели	492
Безопасное разыменование простых указателей	494
Пример: RefWithFlag	495

Нулевые указатели	498
Размеры и выравнивание типов	498
Арифметика указателей	499
Передача в память и из памяти.....	500
Пример: GarBuffer	503
Безопасность паники в небезопасном коде	510
Иноязычные функции: вызов функций на С и С++ из Rust.....	511
Поиск общего представления данных	511
Объявление иноязычных функций и переменных.....	514
Использование библиотечных функций	515
Низкоуровневый интерфейс с libgit2.....	519
Безопасный интерфейс к libgit2.....	524
Заключение	535
Предметный указатель	536
Об авторах	548
Колофон	549