

Оглавление

Предисловие	22
Благодарности	24
Введение	28
Для кого предназначена эта книга	28
Как пользоваться книгой	28
Как изучать Scala.....	29
Условные обозначения	29
Обзор содержимого	30
Ресурсы	33
Исходный код	33
Глава 1. Масштабируемый язык	34
1.1. Язык, наращиваемый под ваши потребности	35
Выведение новых типов	36
Выведение новых управляющих конструкций.....	37
1.2. Что делает Scala масштабируемым языком	39
Scala — объектно-ориентированный язык	40
Scala — функциональный язык.....	41
1.3. Почему именно Scala.....	43
Scala обладает совместимостью	43
Scala лаконичен	44
Scala принадлежит к языкам высокого уровня.....	45
Scala является статически типизированным языком	47
1.4. Истоки происхождения Scala.....	50
Резюме.....	51
Глава 2. Первые шаги в Scala	53
Шаг 1. Освоение интерпретатора Scala.....	53
Шаг 2. Определение переменных	55
Шаг 3. Определение функций.....	57
Шаг 4. Создание сценариев на языке Scala	58

Шаг 5. Организация цикла с помощью while и принятие решения с помощью if	59
Шаг 6. Последовательный перебор элементов с помощью foreach и for	61
Резюме.....	63
Глава 3. Последующие шаги в Scala	64
Шаг 7. Параметризация массивов типами	64
Шаг 8. Использование списков	68
Шаг 9. Применение кортежей.....	71
Шаг 10. Использование наборов и отображений.....	72
Шаг 11. Обучение распознаванию функционального стиля	76
Шаг 12. Считывание строк из файла.....	79
Резюме.....	82
Глава 4. Классы и объекты	83
4.1. Классы, поля и методы	83
4.2. Подразумеваемость использования точки с запятой	87
4.3. Синглтон-объекты.....	88
4.4. Приложение на языке Scala	91
4.5. Трейт App	93
Резюме.....	93
Глава 5. Основные типы и операции	94
5.1. Некоторые основные типы	94
5.2. Литералы.....	95
Целочисленные литералы	95
Литералы чисел с плавающей точкой	97
Символьные литералы	97
Строковые литералы	98
Литералы обозначений	99
Булевы литералы	100
5.3. Строковая интерполяция	100
5.4. Все операторы являются методами	102
5.5. Арифметические операции	104
5.6. Методы отношений и логические операции	106
5.7. Поразрядные операции	107
5.8. Одинаковое содержимое объектов.....	109
5.9. Приоритетность и ассоциативность операторов.....	110

5.10. Обогащающие оболочки	113
Резюме	113
Глава 6. Функциональные объекты	114
6.1. Спецификация класса Rational	114
6.2. Конструкция класса Rational.....	115
6.3. Переопределение метода toString	116
6.4. Проверка соблюдения предварительных условий	117
6.5. Добавление полей	118
6.6. Рекурсивные ссылки	120
6.7. Дополнительные конструкторы	120
6.8. Закрытые поля и методы	121
6.9. Определение операторов.....	123
6.10. Идентификаторы в Scala	124
6.11. Перегрузка методов	127
6.12. Подразумеваемое преобразование.....	129
6.13. Предостережение	130
Резюме.....	130
Глава 7. Встроенные структуры управления	131
7.1. Выражения if.....	132
7.2. Циклы while	133
7.3. Выражения for	135
Обход элементов коллекций	135
Фильтрация.....	137
Вложенные итерации	137
Привязки промежуточных переменных	138
Создание новой коллекции	139
7.4. Обработка исключений с помощью выражений try	140
Выдача исключений	140
Перехват исключений	141
Условие finally	142
Выдача значения	142
7.5. Выражения сопоставлений.....	143
7.6. Жизнь без break и continue	145
7.7. Область видимости переменных	147
7.8. Реорганизация кода, написанного в императивном стиле.....	150
Резюме.....	152

Глава 8. Функции и замыкания	153
8.1. Методы	153
8.2. Локальные функции.....	154
8.3. Функции первого класса	156
8.4. Краткие формы функциональных литералов.....	158
8.5. Синтаксис заместителя	159
8.6. Частично применяемые функции	160
8.7. Замыкания	163
8.8. Специальные формы вызова функций.....	166
Повторяющиеся параметры.....	166
Именованные аргументы.....	167
Значения параметров, используемые по умолчанию	168
8.9. Концевая рекурсия	168
Трассировка функций с концевой рекурсией	169
Ограничения концевой рекурсии.....	171
Резюме.....	172
Глава 9. Управляющие абстракции	173
9.1. Сокращение повторяемости кода	173
9.2. Упрощение клиентского кода.....	177
9.3. Карринг	178
9.4. Создание новых управляющих структур.....	180
9.5. Параметры до востребования	182
Резюме.....	184
Глава 10. Композиция и наследование	185
10.1. Библиотека двумерной разметки.....	185
10.2. Абстрактные классы.....	186
10.3. Определение методов без параметров	187
10.4. Расширение классов.....	189
10.5. Переопределение методов и полей	191
10.6. Определение параметрических полей.....	192
10.7. Вызов конструкторов родительского класса	193
10.8. Использование модификаторов override.....	194
10.9. Полиморфизм и динамическое связывание	196
10.10. Объявление терминальных элементов.....	198
10.11. Использование композиции и наследования.....	199
10.12. Реализация above, beside и toString	200

10.13. Определение фабричного объекта	202
10.14. Методы heighten и widen	205
10.15. А теперь соберем все вместе.....	206
Резюме.....	207
Глава 11. Иерархия в Scala	208
11.1. Иерархия классов в Scala	208
11.2. Реализация элементарных типов.....	212
11.3. Низшие типы.....	214
11.4. Определение своих собственных классов значений.....	214
Резюме.....	217
Глава 12. Трейты	218
12.1. Как работают трейты	218
12.2. Сравнение скудных интерфейсов с насыщенными.....	221
12.3. Пример: прямоугольные объекты.....	222
12.4. Трейт Ordered	224
12.5. Трейты в качестве наращиваемых изменений	226
12.6. Почему не используется множественное наследование.....	230
12.7. Так все же с трейтами или без?.....	233
Резюме.....	234
Глава 13. Пакеты и импортируемый код	235
13.1. Помещение кода в пакеты.....	235
13.2. Краткая форма доступа к родственному коду.....	237
13.3. Импортирование кода	239
13.4. Подразумеваемое импортирование	243
13.5. Модификаторы доступа.....	243
Закрытые элементы	243
Защищенные элементы	244
Открытые элементы	245
Область защиты	245
Видимость и объекты-спутники	247
13.6. Объекты пакетов	248
Резюме.....	249
Глава 14. Утверждения и тесты	250
14.1. Утверждения.....	250
14.2. Тестирование в Scala	251

14.3. Информативные отчеты об ошибках	253
14.4. Использование тестов в качестве спецификаций	254
14.5. Тестирование на основе свойств	257
14.6. Подготовка и проведение тестов.....	258
Резюме.....	260
Глава 15. Case-классы и поиск по шаблону	261
15.1. Простой пример	261
Case-классы	262
Поиск по шаблону	263
Сравнение match со switch	265
15.2. Разновидности шаблонов	265
Универсальные шаблоны сопоставления	266
Шаблоны-константы.....	266
Шаблоны-переменные.....	267
Шаблоны-конструкторы	269
Шаблоны-последовательности	269
Шаблоны-кортежи	270
Типизированные шаблоны	270
Затирание типов	272
Привязка переменной	273
15.3. Ограничители шаблонов	273
15.4. Наложение шаблонов.....	274
15.5. Запечатанные классы.....	275
15.6. Тип Option.....	277
15.7. Повсеместное использование шаблонов	279
Шаблоны в определениях переменных.....	279
Последовательности вариантов в качестве частично применяемых функций	279
Шаблоны в выражениях for	282
15.8. Большой пример	282
Резюме.....	289
Глава 16. Работа со списками	290
16.1. Литералы списков.....	290
16.2. Тип списка	290
16.3. Создание списков.....	291
16.4. Основные операции, проводимые над списками.....	292
16.5. Шаблоны-списки	293

16.6. Методы первого порядка, определенные в классе List.....	294
Объединение двух списков.....	294
Принцип «разделяй и властвуй»	295
Получение длины списка: length	296
Обращение к концу списка: init и last	297
Реверсирование списков: reverse	297
Префиксы и суффиксы: drop, take и splitAt.....	298
Выбор элемента: apply и indices	299
Линеаризация списка списков: flatten	300
Объединение в пары и обратное разбиение: zip и unzip	300
Отображение списков: toString и mkString.....	301
Преобразование списков: iterator, toArray, copyToArray	302
Пример: сортировка методом слияния	302
16.7. Методы высшего порядка, определенные в классе List.....	304
Отображение, применяемое в отношении списка: map, flatMap и foreach.....	305
Фильтрация списков: filter, partition, find, takeWhile, dropWhile и span.....	306
Применение предикатов к спискам: forall и exists	307
Свертка списков: /: и :\.....	308
Пример: реверсирование списков с помощью свертки.....	310
Сортировка списков: sortWith.....	311
16.8. Методы объекта List.....	312
Создание списков из их элементов: List.apply.....	311
Создание диапазона чисел: List.range	312
Создание однообразных списков: List.fill	312
Табулирование функции: List.tabulate	313
Объединение нескольких списков: List.concat.....	313
16.9. Совместная обработка нескольких списков	313
16.10. Осмысление имеющегося в Scala алгоритма вывода типа.....	314
Резюме.....	317
Глава 17. Работа с другими коллекциями	318
17.1. Последовательности.....	318
Списки	318
Массивы.....	319
Списочный буфер.....	320
Буфер массива	321
Строки, реализуемые посредством StringOps.....	321

17.2. Наборы и отображения	322
Использование наборов	323
Применение отображений	324
Наборы и отображения, используемые по умолчанию	326
Отсортированные наборы и отображения	327
17.3. Какие коллекции выбрать: изменяемые или неизменяемые	328
17.4. Инициализация коллекций	331
Преобразование в массив или список	332
Преобразования между изменяемыми и неизменяемыми наборами и отображениями	333
17.5. Кorteжи	334
Резюме	335
Глава 18. Изменяемые объекты	336
18.1. Чем обуславливается изменяемость объекта	336
18.2. Переменные и свойства с возможностью присваивания нового значения	338
18.3. Практический пример: моделирование дискретных событий	341
18.4. Язык для цифровых схем	342
18.5. API моделирования	345
18.6. Моделирование электронной логической схемы	349
Класс Wire	350
Метод inverter	351
Методы andGate и orGate	352
Вывод, получаемый в результате моделирования	353
Запуск симулятора	353
Резюме	355
Глава 19. Параметризация типов	356
19.1. Функциональные очереди	356
19.2. Скрытие информации	359
Закрытые конструкторы и фабричные методы	359
Альтернативный вариант: закрытые классы	361
19.3. Аннотация вариантности	362
19.4. Проверка аннотаций вариантности	366
19.5. Нижние ограничители	369
19.6. Контрвариантность	370
19.7. Закрытые данные объекта	373

19.8. Верхние ограничители	375
Резюме	377
Глава 20. Абстрактные элементы	378
20.1. Краткий обзор абстрактных элементов	378
20.2. Элементы типа	379
20.3. Абстрактные val-переменные	379
20.4. Абстрактные var-переменные	380
20.5. Инициализация абстрактных val-переменных	381
Предварительно проинициализированные поля	383
Ленивые val-переменные	384
20.6. Абстрактные типы	387
20.7. Типы, зависящие от пути	390
20.8. Уточняющие типы	392
20.9. Перечисления	393
20.10. Практический пример: работа с валютой	394
Резюме	402
Глава 21. Подразумеваемые преобразования и параметры	403
21.1. Подразумеваемые преобразования	403
21.2. Правила для подразумеваемых преобразований	405
Названия подразумеваемых преобразований	408
Где применяются подразумеваемые элементы	408
21.3. Подразумеваемое преобразование в ожидаемый тип	409
21.4. Преобразование получателя	410
Взаимодействие с новыми типами	410
Имитация нового синтаксиса	412
Подразумеваемые классы	413
21.5. Подразумеваемые параметры	414
21.6. Контекстные ограничители	419
21.7. Когда применяются множественные преобразования	421
21.8. Отладка подразумеваемых элементов	423
Резюме	425
Глава 22. Реализация списков	426
22.1. Принципиальный взгляд на класс List	426
Объект Nil	427
Класс ::	428

Еще несколько методов.....	428
Создание списка	429
22.2. Класс ListBuffer	431
22.3. Практическое использование класса List.....	432
22.4. Внешняя функциональность.....	435
Резюме.....	436
Глава 23. Возвращение к выражениям for	437
23.1. Выражения for	438
23.2. Задача N королев.....	439
23.3. Выполнение запросов с помощью выражений for	442
23.4. Трансляция выражений for.....	443
Трансляция выражений for с одним генератором	444
Трансляция выражений for, начинающихся с генератора и фильтра.....	444
Трансляция выражений for, начинающихся с двух генераторов	444
Трансляция шаблонов в генераторах	445
Трансляция определений	446
Трансляция, применяемая для циклов	447
23.5. Движение в обратном направлении	447
23.6. Обобщение сведений о выражениях for	448
Резюме.....	450
Глава 24. Углубленное изучение коллекций	451
24.1. Изменяемые и неизменяемые коллекции.....	452
24.2. Согласованность коллекций	454
24.3. Трейт Traversable	455
24.4. Трейт Iterable.....	459
Зачем нужны и Traversable, и Iterable	461
Подкатегории Iterable	462
24.5. Трейты последовательностей Seq, IndexedSeq и LinearSeq	463
24.6. Наборы	467
24.7. Отображения.....	470
24.8. Конкретные классы неизменяемых коллекций.....	474
Списки	474
Потоки	474
Векторы	475
Неизменяемые стеки.....	477
Неизменяемые очереди.....	477

Диапазоны	478
Хеш-извлечения	478
Красно-черные деревья	479
Неизменяемые наборы битов	479
Списочные отображения	480
24.9. Конкретные классы изменяемых коллекций	480
Буферы массивов	481
Списочные буферы	481
Построители строк	482
Связанные списки	482
Двунаправленные связанные списки	482
Изменяемые списки	482
Очереди	483
Последовательности, применяющие массив	483
Стеки	484
Стеки, использующие массивы	484
Хеш-таблицы	484
Слабые хеш-отображения	485
Совместно используемые отображения	486
Изменяемые наборы битов	486
24.10. Массивы	486
24.11. Строки	491
24.12. Характеристики производительности	492
24.13. Равенство	494
24.14. Отображения	494
24.15. Итераторы	499
24.16. Создание коллекции с нуля	506
24.17. Преобразования между коллекциями Java и Scala	508
Резюме	509
Глава 25. Архитектура коллекций Scala	510
25.1. Построители	510
25.2. Вынесение за скобки общих операций	511
25.3. Внедрение новых коллекций	516
Внедрение последовательностей	516
Внедрение новых наборов и отображений	525
Краткие выводы	530
Резюме	530

Глава 26. Экстракторы	531
26.1. Пример извлечения адресов электронной почты.....	531
26.2. Экстракторы	532
26.3. Шаблоны без переменных или с одной переменной.....	535
26.4. Экстракторы переменного количества аргументов	536
26.5. Экстракторы и шаблоны последовательностей.....	539
26.6. Сравнение экстракторов и case-классов.....	539
26.7. Регулярные выражения	541
Составление регулярных выражений	541
Поиск регулярных выражений.....	542
Извлечение с использованием регулярных выражений	543
Резюме.....	544
Глава 27. Аннотации	545
27.1. Зачем нужны аннотации.....	545
27.2. Синтаксис аннотаций	546
27.3. Стандартные аннотации	548
Нежелательность	548
Изменяемые поля.....	549
Двоичная сериализация	549
Автоматически создаваемые методы get и set.....	550
Tailrec.....	551
Unchecked	551
Методы прямого доступа.....	551
Резюме.....	551
Глава 28. Работа с XML	552
28.1. Слабоструктурированные данные	552
28.2. Краткий обзор XML.....	553
28.3. Литералы XML.....	554
28.4. Сериализация	556
28.5. Разборка данных XML-формата на части	557
28.6. Десериализация.....	559
28.7. Загрузка и сохранение	559
28.8. Поиск по шаблону в XML	560
Резюме.....	563
Глава 29. Модульное программирование с использованием объектов	564
29.1. Суть проблемы.....	565
29.2. Приложение для работы с рецептами	566

29.3. Абстракция	569
29.4. Разбиение модулей на трейты.....	572
29.5. Компоновка во время выполнения	573
29.6. Отслеживание экземпляров модулей	575
Резюме	576
Глава 30. Равенство объектов	577
30.1. Понятие равенства в Scala	577
30.2. Написание метода равенства	578
Просчет 1: определение equals с неверной сигнатурой	578
Просчет 2: изменение equals без соответствующего изменения hashCode	580
Просчет 3: определение equals с использованием изменяемых полей	582
Просчет 4: невозможность определения equals в качестве отношения эквивалентности	583
30.3. Определение равенства для параметризованных типов	589
30.4. Рецепты для equals и hashCode	593
Рецепт для equals	594
Рецепт для hashCode.....	595
Резюме	596
Глава 31. Сочетание кодов Scala и Java	597
31.1. Использование Scala из Java	597
Основные правила	597
Типы значений.....	598
Синглтон-объекты.....	598
31.2. Аннотации.....	600
Аннотации Java	601
Написание собственных аннотаций	602
31.3. Подстановочные типы.....	604
31.4. Совместная компиляция Scala и Java	606
31.5. Интегрирование Java 8 в Scala 2.12.....	607
Лямбда-выражения и SAM-типы	607
Использование Stream-объектов Java 8 из Scala 2.12	608
Резюме	610
Глава 32. Фьючерсы и многопоточные вычисления	611
32.1. Неприятности в раю.....	611
32.2. Асинхронное выполнение и Try	613

32.3. Работа с фьючерсами.....	615
Преобразование фьючерсов с помощью функции map	615
Преобразование фьючерсов с помощью выражений for	616
Создание Future: Future.failed, Future.successful, Future.fromTry и Promise	617
Фильтрация: filter и collect.....	618
Обработка сбоев: failed, fallBackTo, recover и recoverWith	619
Отображение обеих возможностей: transform	622
Объединение фьючерсов: zip, Future.fold, Future.reduce, Future.sequence и Future.traverse.....	623
Реализация побочных эффектов: foreach, onComplete и andThen	625
32.4. Тестирование с применением Future-объектов.....	627
Резюме.....	629
Глава 33. Синтаксический анализ с применением комбинаторов	630
33.1. Пример: арифметические выражения	631
33.2. Запуск парсера	633
33.3. Основные парсеры — регулярные выражения.....	634
33.4. Еще один пример: JSON	634
33.5. Вывод парсера	636
33.6. Реализация парсер-комбинаторов.....	641
Входные данные парсера	642
Результаты парсера	643
Класс Parser	643
Использование псевдонимов, создаваемых с помощью ключевого слова this.....	644
Парсеры для анализа отдельно взятых токенов	645
Последовательная композиция.....	645
Альтернативная композиция	646
Работа с рекурсией	647
Преобразование результата	647
Парсеры, не считывающие никаких входных данных.....	647
Возвращение Option-значения и повторение	648
33.7. Строковые литералы и регулярные выражения	648
33.8. Лексинг и парсинг.....	649
33.9. Сообщения об ошибках.....	650
33.10. Сравнение отката с LL(1).....	652
Резюме.....	653

Глава 34. Программирование GUI	655
34.1. Первое Swing-приложение	655
34.2. Панели и разметки.....	657
34.3. Обработка событий.....	659
34.4. Пример: программа перевода градусов между шкалами Цельсия и Фаренгейта.....	662
Резюме.....	664
Глава 35. Электронная таблица SCells	665
35.1. Визуальная среда.....	665
35.2. Разделение введенных данных и отображения	668
35.3. Формулы	670
35.4. Синтаксический анализ формул	672
35.5. Вычисление	676
35.6. Библиотеки операций	678
35.7. Распространение изменений.....	681
Резюме.....	684
Приложение. Сценарии Scala на Unix и Windows	686