
Содержание

Предисловие	17
К читателю	17
Краткий обзор книги	18
Условные обозначения	20
Примеры исходного кода	21
Благодарности	21
От издательства	23
Глава 1. Введение в язык Java	25
1.1. Программная платформа Java	25
1.2. Характерные особенности Java	26
1.2.1. Простота	27
1.2.2. Объектно-ориентированный характер	27
1.2.3. Поддержка распределенных вычислений в сети	28
1.2.4. Надежность	28
1.2.5. Безопасность	28
1.2.6. Независимость от архитектуры компьютера	29
1.2.7. Переносимость	29
1.2.8. Интерпретируемость	30
1.2.9. Производительность	30
1.2.10. Многопоточность	31
1.2.11. Динамичность	31
1.3. Апплеты и Интернет	31
1.4. Краткая история развития Java	33
1.5. Распространенные заблуждения относительно Java	36
Глава 2. Среда программирования на Java	39
2.1. Установка Java Development Kit	40
2.1.1. Загрузка JDK	40
2.1.2. Установка JDK	41
2.1.3. Установка библиотек и документации	44
2.2. Использование инструментов командной строки	45
2.3. Применение ИСП	47
2.4. Запуск графического приложения	50
2.5. Построение и запуск апплетов	52

Глава 3. Основные языковые конструкции Java	59
3.1. Простая программа на Java	60
3.2. Комментарии	63
3.3. Типы данных	64
3.3.1. Целочисленные типы данных	64
3.3.2. Числовые типы данных с плавающей точкой	65
3.3.3. Тип данных char	66
3.3.4. Юникод и тип char	67
3.3.5. Тип данных boolean	68
3.4. Переменные	69
3.4.1. Инициализация переменных	70
3.4.2. Константы	70
3.5. Операции	71
3.5.1. Математические функции и константы	72
3.5.2. Преобразование числовых типов	74
3.5.3. Приведение типов	75
3.5.4. Сочетание арифметических операций с присваиванием	75
3.5.5. Операции инкрементирования и декрементирования	76
3.5.6. Операции отношения и логические операции	76
3.5.7. Поразрядные логические операции	77
3.5.8. Круглые скобки и иерархия операций	78
3.5.9. Перечислимые типы	78
3.6. Символьные строки	79
3.6.1. Подстроки	79
3.6.2. Сцепление строк	79
3.6.3. Принцип постоянства символьных строк	80
3.6.4. Проверка символьных строк на равенство	81
3.6.5. Пустые и нулевые строки	82
3.6.6. Кодовые точки и кодовые единицы	82
3.6.7. Прикладной программный интерфейс API класса String	84
3.6.8. Оперативно доступная документация на API	86
3.6.9. Построение символьных строк	89
3.7. Ввод и вывод	90
3.7.1. Чтение вводимых данных	90
3.7.2. Форматирование выводимых данных	93
3.7.3. Файловый ввод и вывод	97
3.8. Управляющая логика	99
3.8.1. Область действия блоков	99
3.8.2. Условные операторы	100
3.8.3. Неопределенные циклы	104
3.8.4. Определенные циклы	107
3.8.5. Оператор switch для многовариантного выбора	110
3.8.6. Операторы прерывания логики управления программой	112
3.9. Большие числа	115

3.10. Массивы	117
3.10.1. Цикл в стиле <code>for each</code>	118
3.10.2. Инициализация массивов и анонимные массивы	119
3.10.3. Копирование массивов	120
3.10.4. Параметры командной строки	121
3.10.5. Сортировка массивов	122
3.10.6. Многомерные массивы	125
3.10.7. Неровные массивы	127
Глава 4. Объекты и классы	131
4.1. Введение в объектно-ориентированное программирование	132
4.1.1. Классы	132
4.1.2. Объекты	134
4.1.3. Идентификация классов	134
4.1.4. Отношения между классами	135
4.2. Применение predefined классов	136
4.2.1. Объекты и объектные переменные	137
4.2.2. Класс <code>LocalDate</code> из библиотеки <code>Java</code>	140
4.2.3. Модифицирующие методы и методы доступа	141
4.3. Определение собственных классов	145
4.3.1. Класс <code>Employee</code>	145
4.3.2. Использование нескольких исходных файлов	147
4.3.3. Анализ класса <code>Employee</code>	148
4.3.4. Первые действия с конструкторами	149
4.3.5. Явные и неявные параметры	150
4.3.6. Преимущества инкапсуляции	151
4.3.7. Привилегии доступа к данным в классе	153
4.3.8. Закрытые методы	153
4.3.9. Неизменяемые поля экземпляра	154
4.4. Статические поля и методы	154
4.4.1. Статические поля	154
4.4.2. Статические константы	155
4.4.3. Статические методы	156
4.4.4. Фабричные методы	157
4.4.5. Метод <code>main()</code>	157
4.5. Параметры методов	160
4.6. Конструирование объектов	166
4.6.1. Перегрузка	166
4.6.2. Инициализация полей по умолчанию	166
4.6.3. Конструктор без аргументов	167
4.6.4. Явная инициализация полей	168
4.6.5. Имена параметров	168
4.6.6. Вызов одного конструктора из другого	169
4.6.7. Блоки инициализации	170
4.6.8. Уничтожение объектов и метод <code>finalize()</code>	173

4.7. Пакеты	174
4.7.1. Импорт классов	175
4.7.2. Статический импорт	176
4.7.3. Ввод классов в пакеты	177
4.7.4. Область действия пакетов	179
4.8. Путь к классам	181
4.8.1. Указание пути к классам	183
4.9. Документирующие комментарии	184
4.9.1. Вставка комментариев	184
4.9.2. Комментарии к классам	185
4.9.3. Комментарии к методам	185
4.9.4. Комментарии к полям	186
4.9.5. Комментарии общего характера	186
4.9.6. Комментарии к пакетам и обзорные	187
4.9.7. Извлечение комментариев в каталог	188
4.10. Рекомендации по разработке классов	189
Глава 5. Наследование	193
5.1. Классы, суперклассы и подклассы	194
5.1.1. Определение подклассов	194
5.1.2. Переопределение методов	195
5.1.3. Конструкторы подклассов	197
5.1.4. Иерархии наследования	200
5.1.5. Полиморфизм	201
5.1.6. Представление о вызовах методов	202
5.1.7. Предотвращение наследования: конечные классы и методы	205
5.1.8. Приведение типов	206
5.1.9. Абстрактные классы	208
5.1.10. Защищенный доступ	213
5.2. Глобальный суперкласс Object	214
5.2.1. Метод equals ()	215
5.2.2. Проверка объектов на равенство и наследование	216
5.2.3. Метод hashCode ()	220
5.2.4. Метод toString ()	222
5.3. Обобщенные списочные массивы	228
5.3.1. Доступ к элементам списочных массивов	230
5.3.2. Совместимость типизированных и базовых списочных массивов	234
5.4. Объектные оболочки и автоупаковка	235
5.5. Методы с переменным числом параметров	238
5.6. Классы перечислений	239
5.7. Рефлексия	241
5.7.1. Класс Class	242
5.7.2. Основы обработки исключений	244
5.7.3. Анализ функциональных возможностей классов с помощью рефлексии	246
5.7.4. Анализ объектов во время выполнения с помощью рефлексии	251

5.7.5. Написание кода обобщенного массива с помощью рефлексии	255
5.7.6. Вызов произвольных методов	258
Рекомендации по применению наследования	262
Глава 6. Интерфейсы, лямбда-выражения и внутренние классы	265
6.1. Интерфейсы	266
6.1.1. Общее представление об интерфейсах	266
6.1.2. Свойства интерфейсов	272
6.1.3. Интерфейсы и абстрактные классы	273
6.1.4. Статические методы	274
6.1.5. Методы по умолчанию	275
6.1.6. Разрешение конфликтов с методами по умолчанию	276
6.2. Примеры интерфейсов	278
6.2.1. Интерфейсы и обратные вызовы	278
6.2.2. Интерфейс <code>Comparator</code>	281
6.2.3. Клонирование объектов	282
6.3. Лямбда-выражения	288
6.3.1. Причины для употребления лямбда-выражений	288
6.3.2. Синтаксис лямбда-выражений	289
6.3.3. Функциональные интерфейсы	292
6.3.4. Ссылки на методы	293
6.3.5. Ссылки на конструкторы	295
6.3.6. Область действия переменных	295
6.3.7. Обработка лямбда-выражений	298
6.3.8. Еще о компараторах	300
6.4. Внутренние классы	301
6.4.1. Доступ к состоянию объекта с помощью внутреннего класса	303
6.4.2. Специальные синтаксические правила для внутренних классов	306
6.4.3. О пользе, необходимости и безопасности внутренних классов	307
6.4.4. Локальные внутренние классы	310
6.4.5. Доступ к конечным переменным из внешних методов	310
6.4.6. Анонимные внутренние классы	313
6.4.7. Статические внутренние классы	316
6.5. Прокси-классы	319
6.5.1. Когда используются прокси-классы	318
6.5.2. Создание прокси-объектов	320
6.5.3. Свойства прокси-классов	323
Глава 7. Исключения, утверждения и протоколирование	325
7.1. Обработка ошибок	326
7.1.1. Классификация исключений	327
7.1.2. Объявление проверяемых исключений	329
7.1.3. Порядок генерирования исключений	331
7.1.4. Создание классов исключений	333
7.2. Перехват исключений	334
7.2.1. Перехват одного исключения	334
7.2.2. Перехват нескольких исключений	336

7.2.3. Повторное генерирование и связывание исключений в цепочку	337
7.2.4. Блок <code>finally</code>	338
7.2.5. Оператор <code>try</code> с ресурсами	341
7.2.6. Анализ элементов трассировки стека	343
7.3. Рекомендации по обработке исключений	346
7.4. Применение утверждений	349
7.4.1. Понятие утверждения	349
7.4.2. Разрешение и запрет утверждений	350
7.4.3. Проверка параметров с помощью утверждений	351
7.4.4. Документирование предположений с помощью утверждений	352
7.5. Протоколирование	353
7.5.1. Элементарное протоколирование	354
7.5.2. Усовершенствованное протоколирование	354
7.5.3. Смена диспетчера протоколирования	356
7.5.4. Интернационализация	357
7.5.5. Обработчики протоколов	358
7.5.6. Фильтры	362
7.5.7. Средства форматирования	362
7.5.8. “Рецепт” протоколирования	363
7.6. Рекомендации по отладке программ	371
Глава 8. Обобщенное программирование	377
8.1. Назначение обобщенного программирования	378
8.1.1. Преимущества параметров типа	378
8.1.2. На кого рассчитано обобщенное программирование	379
8.2. Определение простого обобщенного класса	380
8.3. Обобщенные методы	382
8.4. Ограничения на переменные типа	383
8.5. Обобщенный код и виртуальная машина	385
8.5.1. Стирание типов	385
8.5.2. Преобразование обобщенных выражений	387
8.5.3. Преобразование обобщенных методов	387
8.5.4. Вызов унаследованного кода	389
8.6. Ограничения и пределы обобщений	390
8.6.1. Параметрам типа нельзя приписывать простые типы	390
8.6.2. Во время выполнения можно запрашивать только базовые типы	391
8.6.3. Массивы параметризованных типов недопустимы	391
8.6.4. Предупреждения о переменном числе аргументов	392
8.6.5. Нельзя создавать экземпляры переменных типа	393
8.6.6. Нельзя строить обобщенные массивы	393
8.6.7. Переменные типа в статическом контексте обобщенных классов недействительны	395
8.6.8. Нельзя генерировать или перехватывать экземпляры обобщенного класса в виде исключений	395
8.6.9. Преодоление ограничения на обработку проверяемых исключений	396
8.6.10. Остерегайтесь конфликтов после стирания типов	398

8.7. Правила наследования обобщенных типов	398
8.8. Подстановочные типы	401
8.8.1. Понятие подстановочного типа	401
8.8.2. Ограничения супертипа на подстановки	402
8.8.3. Неограниченные подстановки	405
8.8.4. Захват подстановок	405
8.9. Рефлексия и обобщения	407
8.9.1. Обобщенный класс <code>Class</code>	408
8.9.2. Сопоставление типов с помощью параметров <code>Class<T></code>	409
8.9.3. Сведения об обобщенных типах в виртуальной машине	409
Глава 9. Коллекции	415
9.1. Каркас коллекций в Java	415
9.1.1. Разделение интерфейсов и реализаций коллекций	416
9.1.2. Интерфейс <code>Collection</code>	418
9.1.3. Итераторы	419
9.1.4. Обобщенные служебные методы	421
9.1.5. Интерфейсы в каркасе коллекций Java	424
9.2. Конкретные коллекции	426
9.2.1. Связные списки	427
9.2.2. Списочные массивы	436
9.2.3. Хеш-множества	437
9.2.4. Древовидные множества	440
9.2.5. Односторонние и двухсторонние очереди	444
9.2.6. Очереди по приоритету	446
9.3. Отображения	447
9.3.1. Основные операции над отображениями	447
9.3.2. Обновление записей в отображении	450
9.3.3. Представления отображений	452
9.3.4. Слабые хеш-отображения	453
9.3.5. Связные хеш-множества и отображения	454
9.3.6. Перечислимые множества и отображения	455
9.3.7. Хеш-отображения идентичности	456
9.4. Представления и оболочки	458
9.4.1. Легковесные оболочки коллекций	458
9.4.2. Поддиапазоны	459
9.4.3. Немодифицируемые представления	460
9.4.4. Синхронизированные представления	461
9.4.5. Проверяемые представления	461
9.4.6. О необязательных операциях	462
9.5. Алгоритмы	465
9.5.1. Сортировка и перетасовка	466
9.5.2. Двоичный поиск	468
9.5.3. Простые алгоритмы	470
9.5.4. Групповые операции	471

9.5.5. Взаимное преобразование коллекций и массивов	472
9.5.6. Написание собственных алгоритмов	473
9.6. Унаследованные коллекции	474
9.6.1. Класс <code>Hashtable</code>	474
9.6.2. Перечисления	475
9.6.3. Таблицы свойств	476
9.6.4. Стеки	477
9.6.5. Битовые множества	477
Глава 10. Программирование графики	481
10.1. Общие сведения о библиотеке <code>Swing</code>	482
10.2. Создание фрейма	487
10.3. Расположение фрейма	489
10.3.1. Свойства фрейма	491
10.3.2. Определение подходящих размеров фрейма	492
10.4. Отображение данных в компоненте	495
10.5. Двухмерные формы	501
10.6. Окрашивание цветом	509
10.7. Специальное шрифтовое оформление текста	512
10.8. Воспроизведение изображений	519
Глава 11. Обработка событий	523
11.1. Общее представление об обработке событий	523
11.1.1. Пример обработки событий от щелчков на экранных кнопках	525
11.1.2. Краткое обозначение приемников событий	530
11.1.3. Пример изменения визуального стиля	532
11.1.4. Классы адаптеров	536
11.2. Действия	540
11.3. События от мыши	547
11.4. Иерархия событий в библиотеке <code>AWT</code>	554
11.4.1. Семантические и низкоуровневые события	556
Глава 12. Компоненты пользовательского интерфейса в <code>Swing</code>	559
12.1. Библиотека <code>Swing</code> и проектный шаблон “модель–представление–контроллер”	560
12.1.1. Проектные шаблоны	560
12.1.2. Проектный шаблон “модель–представление–контроллер”	561
12.1.3. Анализ экранных кнопок в <code>Swing</code> по шаблону “модель–представление–контроллер”	565
12.2. Введение в компоновку пользовательского интерфейса	566
12.2.1. Граничная компоновка	569
12.2.2. Сеточная компоновка	571
12.3. Ввод текста	575
12.3.1. Текстовые поля	576
12.3.2. Метки и пометка компонентов	577
12.3.3. Поля для ввода пароля	579

12.3.4. Текстовые области	579
12.3.5. Панели прокрутки	580
12.4. Компоненты для выбора разных вариантов	583
12.4.1. Флажки	583
12.4.2. Кнопки-переключатели	585
12.4.3. Границы	589
12.4.4. Комбинированные списки	593
12.4.5. Регулируемые ползунки	596
12.5. Меню	603
12.5.1. Создание меню	603
12.5.2. Пиктограммы в пунктах меню	605
12.5.3. Пункты меню с флажками и кнопками-переключателями	607
12.5.4. Всплывающие меню	608
12.5.5. Клавиши быстрого доступа и оперативные клавиши	609
12.5.6. Разрешение и запрет доступа к пунктам меню	611
12.5.7. Панели инструментов	616
12.5.8. Всплывающие подсказки	618
12.6. Расширенные средства компоновки	621
12.6.1. Диспетчер сеточно-контейнерной компоновки	622
12.6.2. Диспетчер групповой компоновки	633
12.6.3. Компоновка без диспетчера	643
12.6.4. Специальные диспетчеры компоновки	643
12.6.5. Порядок обхода компонентов	647
12.7. Диалоговые окна	648
12.7.1. Диалоговые окна для выбора разных вариантов	649
12.7.2. Создание диалоговых окон	659
12.7.3. Обмен данными	663
12.7.4. Диалоговые окна для выбора файлов	669
12.7.5. Диалоговые окна для выбора цвета	679
12.8. Отладка программ с ГПИ	685
12.8.1. Рекомендации по отладке программ с ГПИ	685
12.8.2. Применение робота AWT	688
Глава 13. Развертывание приложений Java	693
13.1. Архивные JAR-файлы	694
13.1.1. Создание JAR-файлов	694
13.1.2. Файл манифеста	695
13.1.3. Исполняемые JAR-файлы	696
13.1.4. Ресурсы	697
13.1.5. Герметизация пакетов	700
13.2. Сохранение глобальных параметров настройки приложений	700
13.2.1. Таблица свойств	701
13.2.2. Прикладной программный интерфейс API для сохранения глобальных параметров настройки	706
13.3. Загрузчики служб	712
13.4. Апплеты	713

13.4.1. Простой апплет	714
13.4.2. HTML-дескриптор <code><applet></code> и его атрибуты	718
13.4.3. Передача данных апплетам через параметры	720
13.4.4. Доступ к файлам изображений и звуковым файлам	725
13.4.5. Контекст апплета	726
13.4.6. Взаимодействие апплетов	727
13.4.7. Отображение элементов в браузере	727
13.4.8. “Песочница”	729
13.4.9. Подписанный код	730
13.5. Технология Java Web Start	732
13.5.1. Доставка приложений Java Web Start	732
13.5.2. Прикладной программный интерфейс JNLP API	736

Глава 14. Параллельное программирование **745**

14.1. Назначение потоков исполнения	746
14.1.1. Предоставление возможности для исполнения других задач с помощью потоков	751
14.2. Прерывание потоков исполнения	755
14.3. Состояния потоков исполнения	758
14.3.1. Новые потоки исполнения	758
14.3.2. Исполняемые потоки	759
14.3.3. Блокированные и ожидающие потоки исполнения	759
14.3.4. Завершенные потоки исполнения	761
14.4. Свойства потоков	761
14.4.1. Приоритеты потоков исполнения	761
14.4.2. Потокосые демоны	762
14.4.3. Обработчики необрабатываемых исключений	763
14.5. Синхронизация	765
14.5.1. Пример состояния гонок	765
14.5.2. Объяснение причин, приводящих к состоянию гонок	768
14.5.3. Объекты блокировки	770
14.5.4. Объекты условий	773
14.5.5. Ключевое слово <code>synchronized</code>	778
14.5.6. Синхронизированные блоки	782
14.5.7. Принцип монитора	783
14.5.8. Поля и переменные типа <code>volatile</code>	784
14.5.9. Поля и переменные типа <code>final</code>	785
14.5.10. Атомарность операций	786
14.5.11. Взаимные блокировки	788
14.5.12. Локальные переменные в потоках исполнения	791
14.5.13. Проверка блокировок и время ожидания	792
14.5.14. Блокировки чтения/записи	794
14.5.15. Причины, по которым методы <code>stop()</code> и <code>suspend()</code> не рекомендованы к применению	795
14.6. Блокирующие очереди	796
14.7. Потокобезопасные коллекции	803

14.7.1. Эффективные отображения, множества и очереди	803
14.7.2. Атомарное обновление записей в отображениях	805
14.7.3. Групповые операции над параллельными хеш-отображениями	807
14.7.4. Параллельные представления множеств	809
14.7.5. Массивы, копируемые при записи	810
14.7.6. Алгоритмы обработки параллельных массивов	810
14.7.7. Устаревшие потокобезопасные коллекции	811
14.8. Интерфейсы Callable и Future	812
14.9. Исполнители	817
14.9.1. Пулы потоков исполнения	817
14.9.2. Плановое исполнение потоков	822
14.9.3. Управление группами задач	822
14.9.4. Архитектура вилочного соединения	824
14.9.5. Завершаемые будущие действия	827
14.10. Синхронизаторы	829
14.10.1. Семафоры	830
14.10.2. Защелки с обратным отсчетом	831
14.10.3. Барьеры	831
14.10.4. Обменники	832
14.10.5. Синхронные очереди	832
14.11. Потоки исполнения и библиотека Swing	832
14.11.1. Выполнение продолжительных задач	834
14.11.2. Применение класса SwingWorker	838
14.11.3. Правило единственного потока исполнения	844
Приложение А. Ключевые слова Java	847
Предметный указатель	849
