

Предисловие

Java SE5 и SE6

Java SE6

Четвертое издание

Изменения

Замечания о дизайне обложки

Благодарности

Введение

Предпосылки

Изучение Java

Цели

Обучение по материалам книги

HTML-документация JDK

Упражнения

Сопроводительные материалы

Исходные тексты программ

Стандарты оформления кода

Ошибки

Глава 1. Введение в объекты

Развитие абстракции

Объект обладает интерфейсом

Объект предоставляет услуги

Скрытая реализация

Повторное использование реализации

Наследование

Отношение «является» в сравнении с «похоже»

Взаимозаменяемые объекты и полиморфизм

Однокорневая иерархия

Контейнеры

Параметризованные типы

Создание и время жизни объектов

Обработка исключений: борьба с ошибками

Параллельное выполнение

Java и Интернет

Что такое Web?

Вычисления «клиент—сервер»

Web как гигантский сервер

Программирование на стороне клиента

Модули расширения

Языки сценариев

Java

Альтернативы

.NET и C#

Интернет и интрасети

Программирование на стороне сервера

Резюме

Глава 2. Все является объектом

Для работы с объектами используются ссылки

Все объекты должны создаваться явно

Где хранятся данные

Особый случай: примитивные типы

Числа повышенной точности

Массивы в Java

Объекты никогда не приходится удалять

Ограничение области действия

Область действия объектов

Создание новых типов данных

Поля и методы

Значения по умолчанию для полей примитивных типов

Методы, аргументы и возвращаемые значения

Список аргументов

Создание программы на Java

Видимость имен

Использование внешних компонентов

Ключевое слово `static`

Ваша первая программа на Java

Компиляция и выполнение

Комментарии и встроенная документация

Документация в комментариях

Синтаксис

Встроенный HTML

Примеры тегов

@see: ссылка на другие классы

{ @link пакет.класс#член_класса метка }

j@docRoot }

{ @inheritDoc }

@version

@author

@since

@param

@return

@throws

@deprecated

Пример документации

Стиль оформления программ

Резюме

Упражнения

Глава 3. Операторы

Простые команды печати

Операторы Java

Приоритет

Присваивание

Совмещение имен во время вызова методов

Математические операторы

Унарные операторы плюс и минус

Автоувеличение и автоуменьшение

Операторы сравнения

Проверка объектов на равенство

Логические операторы

Ускоренное вычисление

Литералы

Экспоненциальная запись

Поразрядные операторы

Операторы сдвига

Тернарный оператор
Операторы + и += для String
Типичные ошибки при использовании операторов
Операторы приведения
Округление и усечение
Повышение
В Java отсутствует sizeof
Сводка операторов
Резюме

Глава 4. Управляющие конструкции

true и false
if-else
Циклы
do-while
for
Оператор-запятая
Синтаксис foreach.
return
break и continue
Нехорошая команда goto
switch
Резюме

Глава 5. Инициализация и завершение

Конструктор гарантирует инициализацию
Перегрузка методов
Различение перегруженных методов
Перегрузка с примитивами
Перегрузка по возвращаемым значениям
Конструкторы по умолчанию
Ключевое слово this
Вызов конструкторов из конструкторов
Значение ключевого слова static
Очистка: финализация и уборка мусора
Для чего нужен метод finalize?
Очистка — ваш долг
Условие «готовности»
Как работает уборщик мусора
Инициализация членов класса
Явная инициализация
Инициализация конструктором
Порядок инициализации
Инициализация статических данных
Явная инициализация статических членов
Инициализация нестатических данных экземпляра
Инициализация массивов
Списки аргументов переменной длины
Перечисления
Резюме

Глава 6, Управление доступом

Пакет как библиотечный модуль
Структура кода
Создание уникальных имен пакетов

Конфликты имен
Пользовательские библиотеки
Использование импортирования для изменения поведения
Предостережение при работе с пакетами
Спецификаторы доступа Java
Доступ в пределах пакета
public
Пакет по умолчанию
private
protected
Интерфейс и реализация
Доступ к классам
Резюме

Глава 7. Повторное использование классов

Синтаксис композиции
Синтаксис наследования
Инициализация базового класса
Конструкторы с аргументами
Делегирование
Сочетание композиции и наследования
Обеспечение правильного завершения
Скрытие имен
Композиция в сравнении с наследованием protected
Восходящее преобразование типов
Почему «восходящее преобразование»?
Снова о композиции с наследованием
Ключевое слово final
Неизменные данные
Пустые константы
Неизменные аргументы
Неизменные методы
Спецификаторы final и private
Неизменные классы
Предостережение
Инициализация и загрузка классов
Инициализация с наследованием
Резюме

Глава 8. Полиморфизм

Снова о восходящем преобразовании
Потеря типа объекта
Особенности
Связывание «метод-вызов»
Получение нужного результата
Расширяемость
Проблема: «переопределение» закрытых методов...
Проблема: поля и статические методы
Конструкторы и полиморфизм
Порядок вызова конструкторов
Наследование и завершающие действия
Поведение полиморфных методов при вызове из конструкторов
Ковариантность возвращаемых типов
Наследование при проектировании

Нисходящее преобразование и динамическое определение типов

Резюме

Глава 9. Интерфейсы

Абстрактные классы и методы

Интерфейсы

Отделение интерфейса от реализации

«Множественное наследование» в Java

Расширение интерфейса через наследование

Конфликты имен при совмещении интерфейсов

Интерфейсы как средство адаптации

Поля в интерфейсах

Инициализация полей интерфейсов

Вложенные интерфейсы

Интерфейсы и фабрики

Резюме

Глава 10. Внутренние классы

Создание внутренних классов

Ссылка на внешний класс

.this и .new

Внутренние классы и восходящее преобразование

Внутренние классы в методах и областях действия

Анонимные внутренние классы

Снова о паттерне «Фабричный метод»

Вложенные классы

Классы внутри интерфейсов

Доступ вовне из многократно вложенных классов

Внутренние классы: зачем?

Замыкания и обратные вызовы

Внутренние классы и система управления

Наследование от внутренних классов

Можно ли переопределить внутренний класс?

Локальные внутренние классы

Идентификаторы внутренних классов

Резюме

Глава 11. Коллекции объектов

Обобщенные типы и классы, безопасные по отношению к типам

Основные концепции

Добавление групп элементов

Вывод контейнеров

List

Итераторы

ListIterator

Linked List

Стек

Множество

Map

Очередь

PriorityQueue

Collection и Iterator

ForEach и итераторы

Идиома «Метод-Адаптер»

Резюме

Глава 12. Обработка ошибок и исключения

Основные концепции

Основные исключения

Аргументы исключения

Перехват исключений

Блок try

Обработчики исключений

Прерывание и возобновление

Создание собственных исключений

Вывод информации об исключениях

Спецификация исключений

Перехват любого типа исключения

Трассировка стека

Повторное возбуждение исключения

Цепочки исключений

Стандартные исключения Java

Особый случай: RuntimeException

Завершение с помощью finally

Для чего нужен блок finally?

Использование finally при return

Ловушка: потерянное исключение

Ограничения исключений

Конструкторы

Отождествление исключений

Альтернативные решения

Предыстория

Перспективы

Передача исключений на консоль

Преобразование контролируемых исключений в неконтролируемые

Рекомендации по использованию исключений

Резюме

Глава 13. Строки

Постоянство строк

Перегрузка + и StringBuilder

Непреднамеренная рекурсия

Операции со строками

Форматирование вывода

printf()

System.out.format()

Класс Formatter

Форматные спецификаторы

Преобразования Formatter

String.format()

Вывод файла в шестнадцатеричном виде

Регулярные выражения

Основы

Создание регулярных выражений

Квантификаторы

CharSequence

Pattern и Matcher

find()

Группы

start() и end()
Флаги шаблонов
split()
Операции замены
reset()
Регулярные выражения и ввод-вывод в Java
Сканирование ввода
Ограничители Scanner
Сканирование с использованием регулярных выражений
StringTokenizer
Резюме

Глава 14. Информация о типах

Необходимость в динамическом определении типов (RTTI)
Объект Class
Литералы class
Ссылки на обобщенные классы
Новый синтаксис приведения типа
Проверка перед приведением типов
Использование литералов class
Динамическая проверка типа
Рекурсивный подсчет
Зарегистрированные фабрики
instanceof и сравнение объектов Class
Отражение: динамическая информация о классе
Извлечение информации о методах класса
Динамические заместители
Null-объекты
Фиктивные объекты и заглушки
Интерфейсы и информация типов
Резюме

Глава 15. Обобщенные типы

Сравнение с C++
Простые обобщения
Библиотека кортежей
Класс стека
Random List
Обобщенные интерфейсы
Обобщенные методы
Использование автоматического определения аргументов-типов...
Явное указание типа
Списки аргументов переменной длины и обобщенные методы
Обобщенный метод для использования с генераторами
Генератор общего назначения
Упрощение использования кортежей
Операции с множествами
Анонимные внутренние классы
Построение сложных моделей
Загадка стирания
Подход C++
Миграционная совместимость
Проблема стирания
Граничные ситуации

Компенсация стирания
Создание экземпляров типов
Массивы обобщений
Ограничения
Маски
Насколько умен компилятор?
Контравариантность
Неограниченные маски
Фиксация
Проблемы
Примитивы не могут использоваться как параметры-типы
Реализация параметризованных интерфейсов
Приведения типа и предупреждения
Перегрузка
Перехват интерфейса базовым классом
Самоограничиваемые типы
Необычные рекурсивные обобщения
Самоограничение
Ковариантность аргументов
Динамическая безопасность типов
Исключения
Примеси
Примеси в C++
Примеси с использованием интерфейсов
Использование паттерна «Декоратор»
Примеси и динамические заместители
Латентная типизация
Компенсация отсутствия латентной типизации
Отражение
Применение метода к последовательности
Если нужный интерфейс отсутствует
Моделирование латентной типизации с использованием адаптеров
Использование объектов функций как стратегий
Резюме
Дополнительная литература

Глава 16. Массивы

Особое отношение к массивам
Массивы как полноценные объекты
Возврат массива
Многомерные массивы
Массивы и обобщения
Создание тестовых данных
Arrays.fill()
Генераторы данных
Применение генераторов для создания массивов
Класс Arrays
Копирование массива
Сравнение массивов
Сравнения элементов массивов
Сортировка массива
Поиск в отсортированном массиве
Резюме

Глава 17. Подробнее о контейнерах

Полная таксономия контейнеров

Заполнение контейнеров

Решение с Generator

Генераторы Map

Использование классов Abstract

Функциональность Collection

Необязательные операции

Неподдерживаемые операции

Функциональность List

Set и порядок хранения

Sorted Set

Очереди

Приоритетные очереди

Деки

Карты (Map)

Производительность

Sorted Map

LinkedHashMap

Хеширование и хеш-коды

Понимание hashCode()

Хеширование ради скорости

Переопределение hashCode()

Выбор реализации

Среда тестирования

Выбор List

Опасности микротестов

Выбор между множествами

Выбор между картами

Факторы, влияющие на производительность HashMap

Вспомогательные средства работы с коллекциями

Сортировка и поиск в списках

Получение неизменяемых коллекций и карт

Синхронизация коллекции или карты

Срочный отказ

Удержание ссылок

WeakHashMap

Контейнеры Java версий 1.0/1.1

Vector и Enumeration

Hashtable

Stack

BitSet

Резюме

Глава 18. Система ввода-вывода Java

Класс File

Получение содержимого каталогов

Анонимные внутренние классы

Вспомогательные средства для работы с каталогами

Проверка и создание каталогов

Ввод и вывод

Типы InputStream

Добавление атрибутов и интерфейсов

Чтение из `InputStream` с использованием `FilterInputStream`
Запись в `OutputStream` с использованием `FilterOutputStream`
Классы `Reader` и `Writer`
Источники и приемники данных
Изменение поведения потока
Классы, оставленные без изменений
`RandomAccessFile`: сам по себе
Типичное использование потоков ввода-вывода
Буферизованное чтение из файла
Чтение из памяти
Форматированное чтение из памяти
Вывод в файл
Сокращенная запись для вывода в текстовые файлы
Сохранение и восстановление данных
Чтение/запись файлов с произвольным доступом
Каналы
Средства чтения и записи файлов
Чтение двоичных файлов
Стандартный ввод-вывод
Чтение из стандартного потока ввода
Замена `System.out` на `Print Writer`
Перенаправление стандартного ввода-вывода
Управление процессами
Новый ввод-вывод (`nio`)
Преобразование данных
Извлечение примитивов
Представления буферов
Данные о двух концах
Буферы и манипуляция данными
Подробнее о буферах
Отображаемые в память файлы
Производительность
Блокировка файлов
Блокирование части отображаемого файла
Сжатие данных
Простое сжатие в формате `GZIP`
Многофайловые архивы `ZIP`
Архивы `Java ARchives` (файлы `JAR`)
Сериализация объектов
Поиск класса
Управление сериализацией
Ключевое слово `transient`
Альтернатива для `Externalizable`
Версии
Долговременное хранение
`XML`
Предпочтения
Резюме
Глава 19. Перечислимые типы
Основные возможности перечислений
Статическое импортирование и перечисления
Добавление методов к перечислению

Переопределение методов перечисления
Перечисления в командах switch
Странности values()
Реализация, а не наследование
Случайный выбор
Использование интерфейсов для организации кода
Использование EnumSet вместо флагов
Использование Enum Map
Методы констант
Цепочка обязанностей
Конечные автоматы
Множественная диспетчеризация
Диспетчеризация с использованием перечислений
Использование методов констант
Использование двумерного массива
Резюме
Глава 20. Аннотации
Базовый синтаксис
Определение аннотаций
Мета-аннотации
Написание обработчиков аннотаций
Элементы аннотаций
Ограничения значений по умолчанию
Генерирование внешних файлов
Альтернативные решения
Аннотации не поддерживают наследование
Реализация обработчика
Использование apt для обработки аннотаций
Использование паттерна «Посетитель» с apt
Использование аннотаций при модульном тестировании
Использование @Unit с обобщениями
«Семейства» не нужны
Реализация @Unit
Удаление тестового кода
Резюме
Новые библиотечные компоненты
CountDownLatch
Потоковая безопасность библиотеки
Cyclic Barrier
DelayQueue
Priors! у BlockingQueue
Управление оранжереями на базе Scheduled Executor
Semaphore
Exchanger
Моделирование
Модель кассира
Моделирование ресторана
Распределение работы
Оптимизация
Сравнение технологий мьютексов
Контейнеры без блокировок
Вопросы производительности

Сравнение реализаций Map
Оптимистическая блокировка
ReadWriteLock
Активные объекты
Резюме
Дополнительная литература
Глава 22. Графический интерфейс
Основы Swing
Вспомогательный класс
Создание кнопки
Перехват событий
Текстовые области
Управление расположением компонентов
BorderLayout
Flow Layout
Grid Layout
GridBagLayout
Абсолютное позиционирование
BoxLayout
Лучший вариант?
Модель событий библиотеки Swing
Типы событий и слушателей.
Адаптеры слушателей упрощают задачу
Отслеживание нескольких событий
Компоненты Swing
Кнопки
Группы кнопок
Значки
Подсказки
Текстовые поля
Рамки
Мини-редактор
Флажки
Переключатели
Раскрывающиеся списки
Списки
Панель вкладок
Окна сообщений
Меню
Всплывающие меню
Рисование
Диалоговые окна
Диалоговые окна выбора файлов
HTML для компонентов Swing
Регуляторы и индикаторы выполнения
Выбор внешнего вида и поведения программы
Деревья, таблицы и буфер обмена
JNLP и Java Web Start
Параллельное выполнение и Swing
Продолжительные задачи
Визуальные потоки
Визуальное программирование и компоненты JavaBean

Что такое компонент JavaBean?
Получение информации о компоненте Bean: инструмент Introspector.
Более сложный компонент Bean
Компоненты JavaBean и синхронизация
Упаковка компонента Bean
Поддержка более сложных компонентов Bean
Больше о компонентах Bean
Альтернативы для Swing
Построение веб-клиентов Flash с использованием Flex
Первое приложение Flex
Компилирование MXML
MXML и ActionScript
Контейнеры и элементы управления
Эффекты и стили
События
Связывание с Java
Модели данных и связывание данных
Построение и развертывание
Создание приложений SWT
Установка SWT
Первое приложение
Устранение избыточного кода
Меню
Вкладки, кнопки и события
Графика
Параллельное выполнение в SWT
SWT или Swing?
Резюме
Ресурсы
Приложение А
Приложения, доступные для загрузки
Thinking in C: Foundations for Java
Семинар «Разработка Объектов и Систем»
Приложение Б. Ресурсы
Программные средства
Редакторы и среды разработки
Книги
Анализ и планирование
Python
Список моих книг