

---

# Содержание

ПРЕДИСЛОВИЕ	18
БЛАГОДАРНОСТИ	20
ОБ АВТОРЕ	21
От издательства	22
<b>1</b> ОСНОВОПОЛАГАЮЩИЕ СТРУКТУРЫ ПРОГРАММИРОВАНИЯ	23
1.1. Первая программа на Java	24
1.1.1. Анализ программы " <b>Hello, World!</b> "	25
1.1.2. Компилирование и выполнение первой программы на Java	26
1.1.3. Вызовы методов	28
1.1.4. Консольное приложение JShell	30
1.2. Примитивные типы	34
1.2.1. Целочисленные типы со знаком	34
1.2.2. Числовые типы с плавающей точкой	36
1.2.3. Тип <b>char</b>	37
1.2.4. Логический тип	37
1.3. Переменные	38
1.3.1. Объявление переменных	38
1.3.2. Именованые переменных	38
1.3.3. Инициализация переменных	39
1.3.4. Константы	39
1.4. Арифметические операции	41
1.4.1. Присваивание	41
1.4.2. Основные арифметические операции	42
1.4.3. Математические методы	44
1.4.4. Преобразования числовых типов	44
1.4.5. Операции отношения и логические операции	46
1.4.6. Большие числа	48
1.5. Символьные строки	49
1.5.1. Сцепление символьных строк	49
1.5.2. Подстроки	50
1.5.3. Сравнение символьных строк	51
1.5.4. Взаимное преобразование чисел и символьных строк	52
1.5.5. Прикладной программный интерфейс API для обработки символьных строк	53
1.5.6. Кодовые точки и единицы	55
1.6. Ввод-вывод	58

1.6.1. Чтение вводимых данных	58
1.6.2. Форматированный вывод данных	60
1.7. Управляющая логика	62
1.7.1. Условные переходы	62
1.7.2. Циклы	64
1.7.3. Прерывание и продолжение цикла	66
1.7.4. Область видимости локальных переменных	68
1.8. Обычные и списочные массивы	69
1.8.1. Обращение с массивами	70
1.8.2. Построение массива	70
1.8.3. Списочные массивы	72
1.8.4. Классы-оболочки для примитивных типов данных	73
1.8.5. Расширенный цикл <b>for</b>	74
1.8.6. Копирование обычных и списочных массивов	74
8.7. Алгоритмы обработки массивов	76
1.8.8. Аргументы командной строки	77
1.8.9. Многомерные массивы	78
1.9. Функциональная декомпозиция	80
1.9.1. Объявление и вызов статических методов	80
1.9.2. Массивы параметров и возвращаемые значения	81
1.9.3. Переменное число аргументов	81
Упражнения	82
<b>2</b> ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ	85
2.1. Обращение с объектами	86
2.1.1. Методы доступа и модифицирующие методы	89
2.1.2. Ссылки на объекты	90
2.2. Реализация классов	92
2.2.1. Переменные экземпляра	92
2.2.2. Заголовки методов	93
2.2.3. Тела методов	93
2.2.4. Вызов методов экземпляра	94
2.2.5. Ссылка <b>this</b>	94
2.2.6. Вызов по значению	95
2.3. Создание объектов	97
2.3.1. Реализация конструкторов	97
2.3.2. Перегрузка	98
2.3.3. Вызов одного конструктора из другого	98
2.3.4. Инициализация по умолчанию	99
2.3.5. Инициализация переменных экземпляра	100
2.3.6. Конечные переменные экземпляра	101
2.3.7. Конструкторы без аргументов	101
2.4. Статические переменные и методы	102

2.4.1. Статические переменные	102
2.4.2. Статические константы	103
2.4.3. Статические блоки инициализации	104
2.4.4. Статические методы	104
2.4.5. Фабричные методы	106
2.5. Пакеты	106
2.5.1. Объявление пакетов	107
2.5.2. Утилита <b>jar</b>	108
2.5.3. Путь к классу	109
2.5.4. Доступ к пакетам	111
2.5.5. Импорт классов	112
2.5.6. Статический импорт	113
2.6. Вложенные классы	114
2.6.1. Статические вложенные классы	114
2.6.2. Внутренние классы	115
2.6.3. Правила специального синтаксиса для внутренних классов	118
2.7. Документирующие комментарии	119
2.7.1. Ввод комментариев	120
2.7.2. Комментарии к классам	121
2.7.3. Комментарии к методам	121
2.7.4. Комментарии к переменным	122
2.7.5. Общие комментарии	122
2.7.6. Ссылки	122
2.7.7. Комментарии к пакетам и общие комментарии	123
2.7.8. Извлечение комментариев	124
Упражнения	125
<b>3</b> ИНТЕРФЕЙСЫ И ЛЯМБДА-ВЫРАЖЕНИЯ	129
3.1. Интерфейсы	130
3.1.1. Объявление интерфейса	131
3.1.2. Реализация интерфейса	132
3.1.3. Преобразование в интерфейсный тип	133
3.1.4. Приведение типов и операция <b>instanceof</b>	134
3.1.5. Расширение интерфейсов	135
3.1.6. Реализация нескольких интерфейсов	135
3.1.7. Константы	136
3.2. Статические, закрытые и методы по умолчанию	136
3.2.1. Статические методы	136
3.2.2. Методы по умолчанию	137
3.2.3. Разрешение конфликтов с методами по умолчанию	138
3.2.4. Закрытые методы	140
3.3. Примеры интерфейсов	140
3.3.1. Интерфейс <b>Comparable</b>	140

3.3.2. Интерфейс <b>Comparator</b>	142
3.3.3. Интерфейс <b>Runnable</b>	143
3.3.4. Обратные вызовы в пользовательском интерфейсе	144
3.4. Лямбда-выражения	145
3.4.1. Синтаксис лямбда-выражений	145
3.4.2. Функциональные интерфейсы	147
3.5. Ссылки на методы и конструкторы	148
3.5.1. Ссылки на методы	148
3.5.2. Ссылки на конструкторы	150
3.6. Обработка лямбда-выражений	151
3.6.1. Реализация отложенного выполнения	151
3.6.2. Вызов функционального интерфейса	152
3.6.3. Реализация собственных функциональных интерфейсов	154
3.7. Область видимости лямбда-выражений и переменных	155
3.7.1. Область видимости лямбда-выражения	155
3.7.2. Доступ к переменным из объемлющей области видимости	156
3.8. Функции высшего порядка	159
3.8.1. Методы, возвращающие функции	159
3.8.2. Методы, изменяющие функции	160
3.8.3. Методы из интерфейса <b>Comparator</b>	160
3.9. Локальные и анонимные классы	161
3.9.1. Локальные классы	162
3.9.2. Анонимные классы	162
Упражнения	163
<b>4 НАСЛЕДОВАНИЕ И РЕФЛЕКСИЯ</b>	<b>167</b>
4.1. Расширение классов	169
4.1.1. Суперклассы и подклассы	169
4.1.2. Определение и наследование методов из суперкласса	170
4.1.3. Переопределение методов	170
4.1.4. Создание подкласса	172
4.1.5. Присваивания в суперклассе	172
4.1.6. Приведение типов	173
4.1.7. Конечные методы и классы	174
4.1.8. Абстрактные методы и классы	175
4.1.9. Защищенный доступ	176
4.1.10. Анонимные подклассы	177
4.1.11. Наследование и методы по умолчанию	177
4.1.12. Ссылки на методы типа <b>super</b>	178
4.2. Всеобъемлющий суперкласс <b>Object</b>	179
4.2.1. Метод <b>toString()</b>	180
4.2.2. Метод <b>equals()</b>	181
4.2.3. Метод <b>hashCode()</b>	184

4.2.4. Клонирование объектов	186
4.3. Перечисления	189
4.3.1. Методы перечислений	189
4.3.2. Конструкторы, методы и поля	191
4.3.3. Тела экземпляров	191
4.3.4. Статические члены	192
4.3.5. Переход по перечислению	193
4.4. Динамическая идентификация типов: сведения и ресурсы	194
4.4.1. Класс <b>Class</b>	194
4.4.2. Загрузка ресурсов	197
4.4.3. Загрузчики классов	198
4.4.4. Загрузчик контекста классов	199
4.4.5. Загрузчики служб	201
4.5. Рефлексия	203
4.5.1. Перечисление членов класса	203
4.5.2. Исследование объектов	204
4.5.3. Вызов методов	205
4.5.4. Создание объектов	206
4.5.5. Компоненты JavaBeans	207
4.5.6. Обращение с массивами	208
4.5.7. Заместители	210
Упражнения	212
<b>5</b> ИСКЛЮЧЕНИЯ, УТВЕРЖДЕНИЯ И ПРОТОКОЛИРОВАНИЕ	215
5.1. Обработка исключений	217
5.1.1. Генерирование исключений	217
5.1.2. Иерархия исключений	218
5.1.3. Объявление проверяемых исключений	220
5.1.4. Перехват исключений	222
5.1.5. Оператор <b>try</b> с ресурсами	223
5.1.6. Оператор <b>finally</b>	225
5.1.7. Повторное генерирование и связывание исключений в цепочку	226
5.1.8. Необработываемые исключения и трассировка стека	227
5.1.9. Метод <b>Objects.requireNonNull()</b>	228
5.2. Утверждения	229
5.2.1. Применение утверждений	230
5.2.2. Разрешение и запрет утверждений	230
5.3. Протоколирование	231
5.3.1. Применение регистраторов	232
5.3.2. Регистраторы	232
5.3.3. Уровни протоколирования	233
5.3.4. Другие методы протоколирования	234
5.3.5. Конфигурация протоколирования	235

5.3.6. Обработчики протоколов	236
5.3.7. Фильтры и средства форматирования	239
Упражнения	240
<b>6</b> ОБОБЩЕННОЕ ПРОГРАММИРОВАНИЕ	243
6.1. Обобщенные классы	245
6.2. Обобщенные методы	246
6.3. Ограничения типов	247
6.4. Вариантность типов и метасимволы подстановки	248
6.4.1. Метасимволы подстановки подтипов	249
6.4.2. Метасимволы подстановки супертипов	250
6.4.3. Применение метасимволов подстановки в переменных типа	251
6.4.4. Неограниченные метасимволы подстановки	253
6.4.5. Захват подстановки	253
6.5. Обобщения в виртуальной машине	254
6.5.1. Стирание типов	255
6.5.2. Вставка приведения типов	255
6.5.3. Мостовые методы	256
6.6. Ограничения, накладываемые на обобщения	258
6.6.1. Запрет на аргументы примитивных типов	258
6.6.2. Во время компиляции все типы оказываются базовыми	258
6.6.3. Нельзя получить экземпляры обобщенных типов	259
6.6.4. Нельзя создать массивы параметризованных типов	261
6.6.5. Нельзя употреблять параметры типа класса в статическом контексте	263
6.6.6. Методы не должны конфликтовать после стирания	263
6.6.7. Исключения и обобщения	264
6.7. Обобщения и рефлексия	265
6.7.1. Класс <code>Class&lt;T&gt;</code>	265
6.7.2. Сведения об обобщенном типе в виртуальной машине	267
Упражнения	269
<b>7</b> КОЛЛЕКЦИИ	273
7.1. Краткий обзор каркаса коллекций	274
7.2. Итераторы	279
7.3. Множества	281
7.4. Отображения	283
7.5. Другие коллекции	286
7.5.1. Свойства	286
7.5.2. Множества битов	288
7.5.3. Перечислимые множества и отображения	289
7.5.4. Стеки и разнотипные очереди	290
7.5.5. Слабые хеш-отображения	292

7.6. Представления	292
7.6.1. Мелкие коллекции	292
7.6.2. Диапазоны	293
7.6.3. Неизменяемые представления	294
Упражнения	295
<b>8 ПОТОКИ ДАННЫХ</b>	<b>299</b>
8.1. От итерации к потоковым операциям	300
8.2. Создание потока данных	302
8.3. Методы <b>filter()</b> , <b>map()</b> и <b>flatMap()</b>	304
8.4. Извлечение и соединение потоков данных	306
8.5. Другие потоковые преобразования	307
8.6. Простые методы сведения	308
8.7. Тип <b>Optional</b>	309
8.7.1. Как обращаться с необязательными значениями	309
8.7.2. Как не следует обращаться с необязательными значениями	311
8.7.3. Формирование необязательных значений	311
8.7.4. Сочетание функций необязательных значений с методом <code>flatMap()</code>	312
8.7.5. Преобразование необязательного значения в поток данных	313
8.8. Накопление результатов	314
8.9. Накопление результатов в отображениях	315
8.10. Группирование и разделение	317
8.11. Нисходящие коллекторы	318
8.12. Операции сведения	321
8.13. Потоки данных примитивных типов	323
8.14. Параллельные потоки данных	324
Упражнения	327
<b>9 ОРГАНИЗАЦИЯ ВВОДА-ВЫВОДА</b>	<b>331</b>
9.1. Потоки ввода, вывода, чтения и записи	332
9.1.1. Получение потоков ввода-вывода	333
9.1.2. Ввод байтов	333
9.1.3. Вывод байтов	334
9.1.4. Кодировки символов	335
9.1.5. Ввод текста	338
9.1.6. Вывод текста	340
9.1.7. Ввод-вывод двоичных данных	341
9.1.8. Произвольный доступ к файлам	342
9.1.9. Файлы, отображаемые в памяти	343
9.1.10. Блокировка файлов	343
9.2. Каталоги, файлы и пути к ним	344
9.2.1. Пути к файлам	344

9.2.2. Создание файлов и каталогов	346
9.2.3. Копирование, перемещение и удаление файлов	347
9.2.4. Обход элементов каталога	349
9.2.5. Системы файлов формата ZIP	352
9.3. Подключения по заданному URL	353
9.3.1. Классы <b>URLConnection</b> и <b>URLConnection</b>	353
9.3.2. Прикладной интерфейс API для HTTP-клиентов	354
9.4. Регулярные выражения	357
9.4.1. Синтаксис регулярных выражений	357
9.4.2. Обнаружение одного совпадения	361
9.4.3. Обнаружение всех совпадений	362
9.4.4. Группы	363
9.4.5. Разбиение по разделителям	364
9.4.6. Замена совпадений	365
9.4.7. Признаки	366
9.5. Сериализация	367
9.5.1. Интерфейс <b>Serializable</b>	367
9.5.2. Переходные переменные экземпляра	369
9.5.3. Методы <b>readObject()</b> и <b>writeObject()</b>	369
9.5.4. Методы <b>readResolve()</b> и <b>writeReplace()</b>	371
9.5.5. Контроль версий	372
Упражнения	373

<b>10</b> ПАРАЛЛЕЛЬНОЕ ПРОГРАММИРОВАНИЕ	377
10.1. Параллельные задачи	379
10.1.1. Выполнение задач	379
10.1.2. Будущие действия	382
10.2. Асинхронные вычисления	384
10.2.1. Завершаемые будущие действия	384
10.2.2. Составление завершаемых будущих действий	386
10.2.3. Длительные задачи в обратных вызовах пользовательского интерфейса	390
10.3. Безопасность потоков исполнения	392
10.3.1. Доступность	392
10.3.2. Состояние гонок	394
10.3.3. Методики надежного параллельного программирования	397
10.3.4. Неизменяемые классы	398
10.4. Параллельные алгоритмы	399
10.4.1. Параллельные потоки данных	399
10.4.2. Параллельные операции над массивами	400
10.5. Потокбезопасные структуры данных	401
10.5.1. Параллельные хеш-отображения	402
10.5.2. Блокирующие очереди	404

10.5.3. Другие потокобезопасные структуры данных	405
10.6. Атомарные счетчики и накапливающие сумматоры	406
10.7. Блокировки и условия	409
10.7.1. Блокировки	410
10.7.2. Ключевое слово <b>synchronized</b>	411
10.7.3. Ожидание по условию	413
10.8. Потоки исполнения	415
10.8.1. Запуск потока исполнения	416
10.8.2. Прерывание потока исполнения	416
10.8.3. Локальные переменные в потоках исполнения	418
10.8.4. Различные свойства потоков исполнения	419
10.9. Процессы	420
10.9.1. Построение процесса	420
10.9.2. Выполнение процесса	422
10.9.3. Дескрипторы процессов	423
Упражнения	425
<b>11</b> АННОТАЦИИ	431
11.1. Применение аннотаций	433
11.1.1. Элементы аннотаций	433
11.1.2. Многие и повторяющиеся аннотации	434
11.1.3. Объявление аннотаций	435
11.1.4. Аннотации в местах употребления типов	436
11.1.5. Явное указание получателей аннотаций	438
11.2. Определение аннотаций	439
11.3. Стандартные аннотации	441
11.3.1. Аннотации для компиляции	442
11.3.2. Аннотации для управления ресурсами	443
11.3.3. Мета-аннотации	444
11.4. Обработка аннотаций во время выполнения	446
11.5. Обработка аннотаций на уровне исходного кода	449
11.5.1. Процессоры аннотаций	449
11.5.2. Прикладной программный интерфейс API модели языка	450
11.5.3. Генерирование исходного кода с помощью аннотаций	451
Упражнения	454
<b>12</b> ПРИКЛАДНОЙ ИНТЕРФЕЙС API ДАТЫ И ВРЕМЕНИ	457
12.1. Временная шкала	458
12.2. Местные даты	461
12.3. Корректоры дат	464
12.4. Местное время	466
12.5. Поясное время	467
12.6. Форматирование и синтаксический анализ даты и времени	471

12.7. Взаимодействие с унаследованным кодом	474
Упражнения	475
<b>13</b> ИНТЕРНАЦИОНАЛИЗАЦИЯ	477
13.1. Региональные настройки	478
13.1.1. Указание региональных настроек	479
13.1.2. Региональные настройки по умолчанию	482
13.1.3. Отображаемые имена	483
13.2. Форматы чисел	485
13.3. Денежные единицы	486
13.4. Форматирование даты и времени	487
13.5. Сортировка и нормализация	489
13.6. Форматирование сообщений	491
13.7. Комплекты ресурсов	493
13.7.1. Организация комплектов ресурсов	494
13.7.2. Классы комплектов ресурсов	496
13.8. Кодировки символов	497
13.9. Глобальные параметры настройки	498
Упражнения	500
<b>14</b> КОМПИЛЯЦИЯ И НАПИСАНИЕ СЦЕНАРИЕВ	503
14.1. Прикладной программный интерфейс API компилятора	504
14.1.1. Вызов компилятора	504
14.1.2. Запуск задания на компиляцию	505
14.1.3. Чтение исходных файлов из оперативной памяти	506
14.1.4. Запись байт-кодов в оперативную память	507
14.1.5. Фиксация диагностической информации	508
14.2. Прикладной программный интерфейс API для сценариев	509
14.2.1. Получение механизма сценариев	509
14.2.2. Привязки	511
14.2.3. Переадресация ввода-вывода	511
14.2.4. Вызов функций и методов из сценариев	512
14.2.5. Компилирование сценария	513
14.3. Интерпретатор Nashorn	514
14.3.1. Выполнение Nashorn из командной строки	514
14.3.2. Вызов методов получения, установки и перегружаемых методов	515
14.3.3. Построение объектов Java	516
14.3.4. Символьные строки в JavaScript и Java	518
14.3.5. Числа	519
14.3.6. Обращение с массивами	519
14.3.7. Списки и отображения	521
14.3.8. Лямбда-выражения	521

14.3.9. Расширение классов и реализация интерфейсов в Java	522
14.3.10. Исключения	524
14.4. Написание сценариев командного процессора средствами Nashorn	524
14.4.1. Выполнение команд из командного процессора	525
14.4.2. Интерполяция символьных строк	526
14.4.3. Ввод данных в сценарий	527
Упражнения	528
<b>15</b> МОДУЛЬНАЯ СИСТЕМА НА ПЛАТФОРМЕ JAVA	531
15.1. Понятие модуля	533
15.2. Именованые модулей	535
15.3. Пример модульной программы "Hello, World!"	535
15.4. Требования модулей	537
15.5. Экспорт пакетов	540
15.6. Модули и рефлексивный доступ	543
15.7. Модульные архивные JAR-файлы	546
15.8. Автоматические и безымянные модули	548
15.9. Параметры командной строки для переноса прикладного кода	550
15.10. Переходные и статические требования	552
15.11. Уточненный экспорт и открытие модулей	554
15.12. Загрузка служб	555
15.13. Инструментальные средства для работы с модулями	557
Упражнения	560
ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ	563