

Оглавление

| | |
|------------------------------------|----|
| Благодарности..... | 16 |
| Представляем Kotlin..... | 18 |
| Почему Kotlin? | 18 |
| Для кого написана эта книга | 19 |
| Как пользоваться этой книгой | 19 |
| Для любознательных..... | 20 |
| Задания..... | 20 |
| Шрифтовые обозначения | 20 |
| Заглядывая вперед..... | 21 |
| От издательства | 21 |

ЧАСТЬ I. ПЕРВЫЕ ШАГИ

| | |
|--|----|
| Глава 1. Ваше первое приложение на Kotlin | 24 |
| Установка IntelliJ IDEA..... | 24 |
| Ваш первый проект на Kotlin..... | 25 |
| Ваш первый файл на Kotlin | 31 |
| Запуск вашего файла на языке Kotlin..... | 33 |
| Компиляция и выполнение кода Kotlin/JVM..... | 34 |
| Kotlin REPL..... | 35 |
| Для любознательных: зачем использовать IntelliJ..... | 37 |
| Для любознательных: программирование для JVM | 37 |
| Задание: арифметические вычисления в REPL..... | 38 |
| Глава 2. Переменные, константы и типы | 40 |
| Типы | 40 |
| Объявление переменной | 40 |
| Встроенные типы языка Kotlin..... | 44 |
| Переменные, доступные только для чтения..... | 44 |
| Автоматическое определение типов | 48 |

| | |
|--|----|
| Константы времени компиляции | 49 |
| Изучаем байт-код Kotlin | 52 |
| Для любознательных: простые типы Java в Kotlin | 55 |
| Задание: hasSteed | 56 |
| Задание: «Рог единорога» | 57 |
| Задание: волшебное зеркало..... | 57 |

ЧАСТЬ II. БАЗОВЫЙ СИНТАКСИС

| | |
|---|-----------|
| Глава 3. Условные конструкции..... | 60 |
| Операторы if/else..... | 60 |
| Добавление условий..... | 64 |
| Вложенные команды if/else..... | 66 |
| Более элегантные условные выражения | 67 |
| Логические операторы | 68 |
| Условные выражения | 71 |
| Убираем скобки в выражениях if/else | 74 |
| Интервалы..... | 75 |
| Условное выражение when | 77 |
| Выражения when с объявлением переменных | 79 |
| Выражения when без аргументов | 80 |
| Задание: эксперименты с интервалами..... | 81 |
| Глава 4. Функции | 82 |
| Выделение кода в функции | 82 |
| Анатомия функции | 85 |
| Заголовок функции | 86 |
| Модификатор видимости | 86 |
| Объявление имени функции..... | 87 |
| Параметры функции..... | 87 |
| Тип возвращаемого значения..... | 88 |
| Тело функции..... | 89 |
| Область видимости функции | 89 |
| Вызов функции..... | 91 |
| Пишем свои функции..... | 92 |
| Аргументы по умолчанию | 94 |
| Функции с единственным выражением | 96 |
| Функции с возвращаемым типом Unit | 98 |

| | |
|--|------------|
| Именованные аргументы функций | 99 |
| Для любознательных: тип Nothing | 102 |
| Для любознательных: функции уровня файла в Java | 103 |
| Для любознательных: перегрузка функций | 104 |
| Для любознательных: имена функций в обратных кавычках | 106 |
| Глава 5. Числа | 108 |
| Числовые типы | 108 |
| Целые числа | 110 |
| Дробные числа | 111 |
| Форматирование значений типа Double | 113 |
| Преобразования числовых типов | 115 |
| Для любознательных: числа без знака | 116 |
| Для любознательных: манипуляции с битами | 119 |
| Глава 6. Строки | 121 |
| Интерполяция строк | 121 |
| Необработанные строки | 124 |
| Чтение ввода с консоли | 126 |
| Преобразование строк в числа | 127 |
| Регулярные выражения | 130 |
| Операции со строками | 131 |
| Строки неизменяемы | 133 |
| Сравнение строк | 133 |
| Для любознательных: Юникод | 134 |
| Глава 7. Null-безопасность и исключения | 136 |
| Допустимость null | 136 |
| Явный тип null в Kotlin | 139 |
| Время компиляции и время выполнения | 141 |
| Null-безопасность | 141 |
| Первый вариант: проверка null в операторе if | 143 |
| Второй вариант: оператор безопасного вызова | 144 |
| Использование безопасного вызова с let | 145 |
| Оператор объединения с null | 147 |
| Третий вариант: оператор проверки | 149 |
| Исключения | 151 |

| | |
|---|-----|
| Выдача исключений | 151 |
| Обработка исключений | 153 |
| Выражения try/catch..... | 154 |
| Проверка предусловий | 155 |
| Для любознательных: пользовательские исключения..... | 158 |
| Для любознательных: проверяемые и непроверяемые исключения..... | 160 |

ЧАСТЬ III. ФУНКЦИОНАЛЬНОЕ ПРОГРАММИРОВАНИЕ И КОЛЛЕКЦИИ

| | |
|--|-----|
| Глава 8. Лямбда-выражения и тип функции | 162 |
| Представляем NyetHack..... | 162 |
| Анонимные функции..... | 163 |
| Лямбда-выражения | 164 |
| Тип функции | 166 |
| Неявный возврат | 167 |
| Аргументы функции | 167 |
| Идентификатор it | 169 |
| Получение нескольких аргументов | 169 |
| Поддержка автоматического определения типов..... | 170 |
| Более эффективные лямбда-выражения..... | 171 |
| Определение функции, которая получает функцию..... | 174 |
| Сокращенный лямбда-синтаксис..... | 176 |
| Встраиваемые функции | 177 |
| Лямбда-выражения и стандартная библиотека Kotlin..... | 179 |
| Для любознательных: ссылки на функции | 182 |
| Для любознательных: захват лямбда-выражений..... | 182 |
| Задание: новые титулы и настроения | 184 |
| Глава 9. Списки и множества | 185 |
| Списки..... | 187 |
| Обращение к элементам списка | 188 |
| Границы индексов и безопасные обращения по индексу | 189 |
| Проверка содержимого списка | 190 |
| Изменение содержимого списка | 192 |
| Итерация..... | 196 |
| Чтение файла в список..... | 200 |

| | |
|---|------------|
| Деструктуризация..... | 202 |
| Множества | 204 |
| Создание множества | 204 |
| Добавление элементов в множество..... | 206 |
| Цикл while | 208 |
| Преобразование коллекций | 210 |
| Для любознательных: типы массивов | 211 |
| Для любознательных: «только для чтения» vs «неизменяемый» | 212 |
| Для любознательных: выражение break | 213 |
| Для любознательных: метки return | 214 |
| Задание: форматированный вывод меню таверны | 216 |
| Задание: улучшенное форматирование меню таверны | 216 |
| Глава 10. Ассоциативные массивы | 217 |
| Создание ассоциативного массива..... | 217 |
| Доступ к значениям в ассоциативном массиве..... | 219 |
| Добавление записей в ассоциативный массив | 221 |
| Изменение значений в ассоциативном массиве | 223 |
| Преобразования между списками и ассоциативными массивами..... | 225 |
| Перебор элементов ассоциативного массива | 228 |
| Задание: составные заказы | 231 |
| Глава 11. Основы функционального программирования..... | 232 |
| Преобразование данных..... | 233 |
| map..... | 233 |
| associate..... | 235 |
| Деструктуризация средствами функционального программирования..... | 237 |
| flatMap | 238 |
| map vs flatMap | 241 |
| Фильтрация данных..... | 241 |
| filter..... | 241 |
| Комбинирование данных | 243 |
| zip | 243 |
| Почему именно функциональное программирование? | 245 |
| Последовательности..... | 246 |
| Для любознательных: профилирование..... | 248 |
| Для любознательных: агрегирование данных | 249 |

| | |
|---|------------|
| reduce | 249 |
| fold..... | 250 |
| sumBy | 251 |
| Для любознательных: ключевое слово vararg | 251 |
| Для любознательных: Arrow.kt | 252 |
| Задание: перестановка ключей и значений в ассоциативном массиве | 253 |
| Задание: поиск самого популярного пункта меню..... | 254 |
| Глава 12. Функции области видимости..... | 255 |
| apply..... | 255 |
| let..... | 256 |
| run | 258 |
| with | 259 |
| also..... | 259 |
| takeIf..... | 260 |
| Использование функций области видимости..... | 261 |

ЧАСТЬ IV. ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ

| | |
|--|------------|
| Глава 13. Классы | 266 |
| Объявление класса..... | 266 |
| Создание экземпляров..... | 266 |
| Функции класса | 267 |
| Видимость и инкапсуляция | 268 |
| Свойства класса | 270 |
| Get-методы и set-методы свойств | 272 |
| Видимость свойств | 275 |
| Вычисляемые свойства | 276 |
| Использование пакетов | 278 |
| Для любознательных: более пристальный взгляд на свойства var и val | 280 |
| Для любознательных: защита от изменяемости | 283 |
| Для любознательных: ограничение видимости рамками пакета..... | 285 |
| Глава 14. Инициализация..... | 286 |
| Конструкторы | 286 |
| Главный конструктор..... | 287 |
| Объявление свойств в главном конструкторе | 289 |

| | |
|---|------------|
| Дополнительные конструкторы | 290 |
| Аргументы по умолчанию | 292 |
| Именованные аргументы | 293 |
| Блок инициализации | 294 |
| Порядок инициализации..... | 296 |
| Задержка инициализации | 297 |
| Поздняя инициализация | 298 |
| Отложенная инициализация..... | 302 |
| Для любознательных: подводные камни инициализации | 305 |
| Для любознательных: делегаты свойств..... | 307 |
| Задание: загадка Эскалибура..... | 308 |
| Глава 15. Наследование..... | 310 |
| Объявление класса Room | 310 |
| Создание подкласса | 312 |
| Проверка типов..... | 318 |
| Иерархия типов в языке Kotlin | 320 |
| Приведение типа | 320 |
| Умное приведение типа..... | 322 |
| Рефакторинг кода таверны..... | 323 |
| Для любознательных: Any | 331 |
| Для любознательных: оператор безопасного приведения типа..... | 332 |
| Глава 16. Объекты, классы данных и перечисления | 333 |
| Ключевое слово object..... | 333 |
| Объявления объекта | 334 |
| Объекты-выражения | 339 |
| Объекты-компаньоны | 340 |
| Вложенные классы | 342 |
| Классы данных | 344 |
| toString..... | 345 |
| equals и hashCode..... | 345 |
| сору | 346 |
| Деструктуризация объявлений | 347 |
| Классы-перечисления..... | 348 |
| Перегрузка операторов | 350 |
| Исследуем мир NyetHack..... | 352 |

| | |
|--|------------|
| Для любознательных: объявление структурного сравнения..... | 356 |
| Для любознательных: алгебраические типы данных..... | 358 |
| Для любознательных: классы-значения..... | 360 |
| Задание: другие команды | 361 |
| Задание: реализация карты мира | 362 |
| Задание: колокольный звон..... | 362 |
| Глава 17. Интерфейсы и абстрактные классы..... | 363 |
| Определение интерфейса | 363 |
| Реализация интерфейса | 364 |
| Реализация по умолчанию..... | 367 |
| Абстрактные классы | 368 |
| Сражение в NyetHack..... | 371 |
| Задание: дополнительные монстры | 376 |
| ЧАСТЬ V. KOTLIN ДЛЯ ОПЫТНЫХ ПРОГРАММИСТОВ | |
| Глава 18. Обобщения..... | 378 |
| Определение обобщенных типов | 378 |
| Обобщенные функции | 380 |
| Ограничения обобщений | 381 |
| in и out | 384 |
| Добавление наград в NyetHack | 388 |
| Для любознательных: ключевое слово reified | 394 |
| Глава 19. Расширения..... | 397 |
| Определение функции-расширения..... | 397 |
| Объявление расширения для суперкласса | 399 |
| Обобщенные функции-расширения | 400 |
| Операторные функции-расширения..... | 402 |
| Свойства-расширения..... | 404 |
| Расширения для типов, допускающих null..... | 406 |
| Расширения: как это устроено..... | 407 |
| Видимость расширений..... | 408 |
| Расширения в стандартной библиотеке Kotlin | 409 |
| Для любознательных: литералы функций с получателями..... | 410 |
| Задание: расширение рамок..... | 412 |

| | |
|---|-----|
| Глава 20. Сопрограммы..... | 413 |
| Блокирующие вызовы..... | 414 |
| Включение сопрограмм | 416 |
| Строители сопрограмм | 417 |
| Области видимости сопрограмм..... | 418 |
| Структурированный параллелизм..... | 419 |
| Использование клиента HTTP | 422 |
| asunc и await | 428 |
| Для любознательных: состояние гонки..... | 430 |
| Для любознательных: Kotlin на стороне сервера..... | 433 |
| Задание: никаких отмен | 434 |
| Глава 21. Поток данных | 435 |
| Создание потоков данных..... | 436 |
| MutableStateFlow..... | 443 |
| Завершение потоков данных | 447 |
| Преобразования потоков данных..... | 450 |
| Обработка ошибок в потоках данных..... | 452 |
| Для любознательных: SharedFlow..... | 454 |
| Глава 22. Каналы | 457 |
| Распределение работы с использованием каналов | 457 |
| Отправка данных в канал | 459 |
| Получение данных из канала..... | 460 |
| Закрытие канала | 463 |
| Объединение заданий | 464 |
| Для любознательных: другие особенности поведения каналов..... | 469 |
| Встречные каналы (rendezvous)..... | 469 |
| Буферизованные каналы (buffered) | 469 |
| Неограниченные каналы (unlimited) | 470 |
| Каналы с заменой (conflated)..... | 470 |

ЧАСТЬ VI. СОВМЕСТИМОСТЬ И МУЛЬТИПЛАТФОРМЕННЫЕ ПРИЛОЖЕНИЯ

| | |
|---|-----|
| Глава 23. Совместимость с Java | 474 |
| Взаимодействие с классом Java | 474 |
| Совместимость и null | 476 |

| | |
|--|------------|
| Соответствие типов | 479 |
| Get-методы, set-методы и совместимость | 481 |
| За пределами класса | 483 |
| Исключения и совместимость | 492 |
| Функциональные типы в Java | 495 |
| Глава 24. Знакомство с Kotlin Multiplatform | 497 |
| Что такое Kotlin Multiplatform? | 498 |
| Планирование мультиплатформенного проекта | 499 |
| Первый мультиплатформенный проект | 500 |
| Определение целевой платформы Kotlin/JVM | 502 |
| Определение общего кода | 504 |
| expect и actual | 505 |
| Глава 25. Kotlin/Native | 511 |
| Объявление целевой платформы macOS | 513 |
| Написание нативного кода на Kotlin | 515 |
| Запуск приложения Kotlin/Native | 517 |
| Вывод Kotlin/Native | 519 |
| Для любознательных: Kotlin Multiplatform Mobile | 520 |
| Для любознательных: другие нативные платформы | 522 |
| Глава 26. Kotlin/JS | 524 |
| Объявление поддержки Kotlin/JS | 524 |
| Взаимодействие с DOM | 528 |
| Ключевое слово external | 533 |
| Выполнение произвольного кода JavaScript | 535 |
| Динамические типы | 537 |
| Для любознательных: фреймворки клиентской части | 540 |
| Задание: комиссионные при обмене валюты | 541 |
| Послесловие | 542 |
| Что дальше? | 542 |
| Бесстыдная самореклама | 542 |
| Спасибо вам | 543 |
| Глоссарий | 544 |