

Сергей Пименов

# ЯЗЫК ПРОГРАММИРОВАНИЯ KOTLIN

Киев



2017

УДК 004.4\*Kotlin  
ПЗ2

**Пименов, Сергей**

ПЗ2 Язык программирования Kotlin / Сергей Пименов — К. : «Агентство «ИPIO», 2017. — 304 с.

ISBN 978-617-7453-28-3

Книга представляет собой полное справочное пособие по языку программирования Kotlin. В книге подробно рассмотрены такие вопросы как: типы данных, базовые синтаксические конструкции языка, вопросы объектно-ориентированного программирования, классы и интерфейсы, исключения. Книга изобилует примерами кода, который можно загрузить из репозитория автора. Книга рассчитана на разработчиков разной квалификации и будет полезна как новичкам в программировании, так и опытным программистам, решившим освоить новый отличный язык программирования Kotlin.

УДК 004.4\*Kotlin

*Все права защищены.  
Никакая часть данной книги не может  
быть воспроизведена в какой бы то ни  
было форме без письменного разрешения  
владельцев авторских прав.*

ISBN 978-617-7453-28-3

© Сергей Пименов

© Издательство «Агентство «ИPIO»

# ПРЕДИСЛОВИЕ

Kotlin относительно молодой язык программирования, на момент написания книги ему исполнилось 7 лет. Несмотря на свою молодость, Kotlin признан компанией Google в качестве официального языка программирования для Android с first-class поддержкой.

Основные причины успеха Kotlin — его простота, краткость и выразительность, безопасность и полная совместимость с Java.

## Книга для всех

Эта книга предназначена для всех, кто занимается программированием, независимо от опыта и стажа. Цель этой книги — познакомить читателя с отличным универсальным языком программирования Kotlin.

Автор надеется, что каждый найдет в этой книге что-либо полезное для себя.

## Структура книги

Эта книга служит справочным пособием по языку программирования Kotlin, в котором описываются его синтаксис, ключевые слова и основополагающие принципы программирования на языке Kotlin.

## Исходный код примеров

Исходный код всех примеров, приведенных в этой книге, доступен на GitHub по адресу: <https://github.com/olton/kotlin-examples>

Также большое количество примеров вы найдете в репозитории Egorand/kotlin-playground по адресу <https://github.com/Egorand/kotlin-playground> и на официальном сайте Kotlin по адресу <https://kotlinlang.org/docs/tutorials/>

# АВТОР ВЫРАЖАЕТ БЛАГОДАРНОСТЬ

Любимой жене Татьяне за любовь и поддержку.

Другу и шефу Александру Ольшанскому (<https://www.facebook.com/olshanskiy>) за помощь в издании книги и за ту энергию, которую он проецирует на других людей.

Компании JetBrains за отличную IDE IntelliJ IDEA и другие продукты.

Разработчикам Kotlin за отличный язык программирования.

Всем, кто принимал участие в рождении и развитии языка Kotlin за их участие.

Сайту [kotlinlang.ru](https://kotlinlang.ru) и Олегу Дуброву (<https://github.com/phplego>)

Отдельное спасибо Денису Седченко (<https://www.facebook.com/denissedchenko>) за помощь в тестировании текста книги.

# КНИГА ИЗДАНА И НАПЕЧАТАНА ПРИ ПОДДЕРЖКЕ КОМПАНИЙ ХОЛДИНГА INTERNET INVEST:

Imena.UA

Mirohost

Olshansky & Partners

# СОДЕРЖАНИЕ

<b>Глава 1</b>	<b>История и развитие языка</b> . . . . .	8
	Jetbrains . . . . .	8
	Kotlin . . . . .	9
	Применение Kotlin . . . . .	12
<b>Глава 2</b>	<b>Краткий обзор Kotlin</b> . . . . .	15
	ООП . . . . .	15
	Программа «Привет, мир!» . . . . .	19
	Установка компилятора . . . . .	21
	Компиляция программы . . . . .	22
	Лексика . . . . .	24
	Базовый синтаксис . . . . .	26
	Ключевые слова . . . . .	35
	Пакеты . . . . .	36
<b>Глава 3</b>	<b>Типы данных и переменные</b> . . . . .	41
	Типы данных . . . . .	41
	Числа . . . . .	42
	Символы . . . . .	48
	Строки . . . . .	49
	Массивы . . . . .	51
	Логический тип . . . . .	54
	Приведение типов . . . . .	55
	Псевдонимы типов . . . . .	58
<b>Глава 4</b>	<b>Операции</b> . . . . .	60
	Операции в Kotlin . . . . .	60
	Арифметические операции . . . . .	60
	Операции отношения . . . . .	66
	Равенство . . . . .	69
	Логические операции . . . . .	71
	Поразрядные операции . . . . .	73
	Операция присваивания . . . . .	76
	Тернарная операция . . . . .	76
	Приоритет операций . . . . .	77
	Перегрузка операторов . . . . .	78

<b>Глава 5</b>	<b>Управляющие операторы</b>	83
	Управляющие операторы	83
	Операторы выбора	83
	Выражение if	84
	Оператор ?:	86
	Выражение when	87
	Операторы цикла	90
	Цикл for	90
	Цикл while и do-while	92
	Вложенные циклы	94
	Операторы перехода	95
<b>Глава 6</b>	<b>Функции и лямбды</b>	102
	Функции в Kotlin	102
	Применение функций	104
	Инфиксное обозначение	106
	Параметры функции	106
	Имена в названиях параметров	108
	Функции, возвращающие Unit	109
	Функции с одним выражением	110
	Явные типы возвращаемых значений	110
	Переменное число аргументов	110
	Область действия функций	111
	Функции с хвостовой рекурсией	114
	Лямбда-выражения и анонимные функции	115
	Высокоуровневые функции	120
	Встроенные (inline) функции	123
<b>Глава 7</b>	<b>Классы и объекты</b>	129
	Введение в классы	129
	Общая форма класса	129
	Объявление класса	130
	Конструкторы	135
	Свойства и поля	137
	Методы и перегрузка методов	142
	Класс Stack	145
	Модификаторы доступа	147
	Интерфейсы	151
	Наследование	160
	Абстрактные классы	166
	Классы данных	167
	Изолированные классы	172
	Перечисления	174

	Вложенные классы . . . . .	179
	Объекты . . . . .	180
	Делегирование . . . . .	186
	Обобщения . . . . .	195
	Расширения . . . . .	209
	Улучшаем класс Stack . . . . .	214
<b>Глава 8</b>	<b>Обработка исключений . . . . .</b>	<b>216</b>
	Исключения в Kotlin . . . . .	216
	Классы исключений . . . . .	217
	Необработываемые исключения . . . . .	218
	Обработка исключений . . . . .	219
	try — это выражение . . . . .	221
	Несколько операторов catch . . . . .	221
	Вложенные операторы try . . . . .	223
	Оператор throw . . . . .	224
	Оператор finally . . . . .	226
	Встроенные исключения . . . . .	227
	Создание собственных исключений . . . . .	228
	Цепочки исключений . . . . .	229
<b>Глава 9</b>	<b>Рефлексия и аннотации . . . . .</b>	<b>231</b>
	Рефлексия . . . . .	231
	Аннотации . . . . .	237
<b>Глава 10</b>	<b>Сопрограммы . . . . .</b>	<b>246</b>
	Введение в сопрограммы . . . . .	246
<b>Глава 11</b>	<b>Коллекции . . . . .</b>	<b>256</b>
	Введение . . . . .	256
	Collection и MutableCollection . . . . .	257
	List и MutableList . . . . .	267
	Set и MutableSet . . . . .	268
	Использование коллекций . . . . .	268
	Map и MutableMap . . . . .	271
<b>Глава 12</b>	<b>Другие особенности языка . . . . .</b>	<b>277</b>
	Ключевое слово this . . . . .	277
	Интервалы . . . . .	278
	NULL-безопасность . . . . .	283
<b>Глава 13</b>	<b>Грамматика языка . . . . .</b>	<b>287</b>
	Грамматика . . . . .	287

# ГЛАВА 1

## ИСТОРИЯ И РАЗВИТИЕ ЯЗЫКА

### JETBRAINS

Компания JetBrains была основана в 2000 году тремя программистами: Сергеем Дмитриевым, Евгением Беляевым и Валентином Кипятковым с основной целью — создать мощную, полноценную IDE (интегрированная среда разработки) для Java. Штаб-квартира компании находится в Чехии, но JetBrains имеет множество представительств и в других странах. В штате компании более 500 разработчиков, которые работают в Санкт-Петербурге, Мюнхене, Праге, Бостоне и Москве и создают интеллектуальные инструменты, понимающие семантику кода и повышающие продуктивность работы программистов. На данный момент JetBrains сотрудничает с более чем 3000 компаний по всему миру и в различных сферах деятельности: банки, финансирование, IT-индустрия, биотехнологии, промышленность, программные продукты и многое другое.

Первым продуктом компании был Renamer — небольшая программа, которая позволяла делать простой рефакторинг-переименование для программ на языке Java. Программа давала возможность безопасно переименовывать класс, пакет, метод или переменную в проекте. Вторым продуктом стал CodeSearch — плагин для популярной в то время IDE от Borland JBuilder, который позволял быстро и точно находить все использования символа, метода или класса во всей программе.



Следующим продуктом стала IDE — IntelliJ IDEA, которая до сих пор остается флагманом компании. Первая версия появилась в январе 2001 года и быстро приобрела популярность как первая среда для Java с широким набором интегрированных инструментов для рефакторинга, которые позволяли программистам быстро реорганизовывать исходные тексты программ. Дизайн среды ориентирован на продуктивность работы программистов позволяя сконцентрироваться на функциональных задачах, в то время как IntelliJ IDEA берет на себя выполнение рутинных операций.

В последующие годы компания выпустила на базе IDEA IDE для таких языков программирования, как C#, Ruby, Python, PHP, C/C++, Swift и Objective-C, JavaScript, Go.

Помимо сред разработки компания также создает и другие полезные инструменты, призванные упростить весь цикл — от идеи до реализации и внедрения программных продуктов. Среди таких инструментов стоит отметить: ReSharper, DataGrip, dotPeek, dotTrace, dotMemory, dotCover, Youtrack, TeamCity, Upsource, Hub и MPS.

Весь перечень продуктов компании доступен по адресу: <https://www.jetbrains.com/products.html>

## KOTLIN

В 2010 году компания JetBrains приступила к созданию нового языка программирования. Необходимость такого решения была обусловлена несколькими факторами, в том числе и тем, что стало понятно, что язык Java в некоторых вопросах уже не устраивает компанию, а существующие альтернативные языки не соответствуют тем требованиям, которые выдвигались к языку.

В компании собралось очень много людей с большим экспертным опытом в области языков программирования. Это также способствовало принятию решения о необходимости создать свой новый язык программирования, который бы удовлетворял требованиям текущего момента и мог бы решать задачи будущего.

Основными требованиями к новому языку были:

- Простота и эффективность
- Максимальная безопасность
- Полная совместимость с Java
- Статическая типизация
- Качественная инструментальная поддержка

Язык Kotlin можно охарактеризовать двумя словами: прагматичность и демократичность.

Языки бывают разные: некоторые выходят из академической среды, другие созданы для конкретных платформ. В компании JetBrains были нацелены на практичный язык для максимально широкой аудитории. Он должен был быть демократичным, то есть без заумных вещей. Бывают ситуации, когда программист знает все тонкости языка и благодаря этому пишет хитрый код, — и в этот код никто из джуниор-программистов не может лезть. Компании был нужен язык, который одинаково хорош как для начинающих программистов, так и для продвинутых.

### **Три причины начать дружить с Kotlin.**

В первую очередь разработчики любят Kotlin за его краткость и выразительность. Раньше людей это не очень смущало, потом размер программ стал больше — люди поняли, что пишут очень много совершенно бессмысленных кусков кода только потому, что от них это требует синтаксис языка программирования.

Вторая причина в том, что он полностью совместим с Java и позволяет постепенно мигрировать с Java на Kotlin.

Третья причина — Kotlin безопасен: в семантику языка заложены принципы, предотвращающие целый ряд очень распространенных ошибок, которые обычно случаются в момент исполнения программы. Это позволяет писать более безопасный код, что в конечном итоге помогает сэкономить деньги и снизить затраты на тестирование.

**Сегодня с помощью Kotlin вы можете разрабатывать приложения для:**

- JVM
- Android
- Browser и NodeJS

**Четыре кита, на которых стоит Kotlin:**

- Краткость — меньшее количество шаблонов кода.
- Безопасность — позволяет избежать множества ошибок на этапе разработки, таких как исключения нулевого указателя.
- Interoperable — позволяет использовать существующие библиотеки Java, Android, JavaScript.
- Tool-friendly — используйте качественную инструментальную поддержку языка на уровне IDE и других инструментов.

## Итак, что же такое Kotlin?

Kotlin — современный статически типизированный объектно-ориентированный язык программирования, компилируемый для платформ Java и JavaScript. При полной совместимости с Java Kotlin предоставляет дополнительные возможности, упрощающие повседневную работу программиста и повышающие продуктивность. Он сочетает в себе лаконичность, выразительность, производительность и простоту в изучении. Kotlin компилируется в байткод, работающий поверх JVM. Также он умеет компилироваться в JavaScript и на другие платформы через инфраструктуру LLVM. Язык назван в честь острова Котлин в Финском заливе, на котором расположен город Кронштадт.

### Ключевые возможности Kotlin

- Полная совместимость с Java в обе стороны (код на Java и Kotlin можно безболезненно смешивать в одном проекте)
- Автоматический вывод типов переменных и функций
- Анонимные функции (лямбда-выражения) позволяют писать более компактный код
- Возможности создания проблемно-ориентированных языков (DSL)
- Внешние функции позволяют расширять интерфейс существующих классов, не меняя их
- Выразительная система типов позволяет обнаруживать многие ошибки на этапе компиляции
- Конструкции, сокращающие лишние повторения кода: свойства, значения параметров по умолчанию, мультиприсваивания, классы данных, автоматическое приведение типов и пр.

Авторы ставили целью создать язык более лаконичный и типобезопасный, чем Java, и более простой, чем Scala. Следствием упрощения по сравнению со Scala стали также более быстрая компиляция и лучшая поддержка языка в IDE.

Впервые публично Kotlin был представлен широкой публике в июле 2011 года. Исходный код реализации языка был открыт в феврале 2012. В феврале был выпущен milestone 1, включающий плагин для IDEA. В июне — milestone 2 с поддержкой Android. В декабре 2012 года вышел milestone 4, включающий, в частности, поддержку Java 7.

В феврале 2016 года вышел официальный релиз языка Kotlin. На момент написания книги Kotlin достиг версии 1.1.2, анонс которой состоялся 25 апреля 2017 года.

В мае 2017 года компания Google сообщила, что инструменты языка Kotlin, основанные на JetBrains IDE, будут по стандарту включены в Android Studio 3.0 — официальный инструмент разработки для ОС Android.

Kotlin позиционируется разработчиками как объектно-ориентированный язык промышленного уровня, а также как язык, который сможет заменить Java. При этом он полностью совместим с Java, что позволяет разработчикам постепенно перейти с Java на Kotlin. В частности, в Android язык интегрируется с помощью Gradle, что позволяет для существующего Android-приложения внедрять новые функции на Kotlin без переписывания приложения целиком.

Среди компаний, которые так или иначе применяют язык в своих разработках, можно назвать Google (часть компилятора Android DataBindings), Expedia (мобильное приложение), Square (SQLDelight compiler), Prezi (использование на сервере).

В 2016 году около 40 тыс. программистов использовали Kotlin, а количество кода на нем в открытых репозиториях GitHub удваивается каждые несколько месяцев и уже перевалило за 2 млн строк. Ну и в самой JetBrains, конечно, интенсивно используют Kotlin как для новых продуктов, так и для развития старых, например, IntelliJ IDEA.

Kotlin — это очень простой язык, призванный решать серьезные задачи.

## ПРИМЕНЕНИЕ KOTLIN

### Kotlin для Server-side

Kotlin отлично подходит для разработки приложений на стороне сервера, позволяя писать краткий и выразительный код, сохраняя при этом полную совместимость с существующими стеками Java-технологий и плавной кривой обучения.

- **Выразительность:** инновационные языковые функции Kotlin, такие как поддержка type-safe builders и делегированных свойств, помогают создавать мощные и простые в использовании абстракции.
- **Масштабируемость:** поддержка Kotlin для coroutines дает возможность создавать серверные приложения, которые масштабируются до огромного количества клиентов со скромными требованиями к оборудованию.

- **Взаимодействие:** Kotlin полностью совместим со всеми основанными на Java фреймворками, что позволяет вам оставаться в привычном технологическом стеке, наслаждаясь преимуществами более современного языка.
- **Миграция:** Kotlin поддерживает постепенную, пошаговую миграцию больших кодовых баз с Java на Kotlin. Вы можете начать писать новый код в Kotlin, сохраняя старые части вашей системы на Java.
- **Инструментарий:** в дополнение к большой поддержке IDE в целом Kotlin предлагает инструментарий, специфичный для конкретной платформы (например, для Spring) в плагине для IntelliJ IDEA Ultimate.

Приложения Kotlin могут быть развернуты на любом хосте, который поддерживает Java Web-приложения, включая Amazon Web Services, Google Cloud Platform и другие.

### Kotlin для Android

Kotlin отлично подходит для разработки приложений для Android, принося все преимущества современного языка в платформу Android без введения каких-либо новых ограничений.

- **Совместимость:** Kotlin полностью совместим с JDK 6. Это гарантирует, что приложения Kotlin могут работать на старых устройствах Android без проблем. Инструментарий Kotlin полностью поддерживается в Android Studio и совместим с системой сборки Android.
- **Производительность:** приложение Kotlin работает так же быстро, как эквивалент Java, благодаря очень похожей структуре байт-кода. Благодаря поддержке встроенных функций Kotlin код с использованием лямбд часто работает даже быстрее, чем тот же код, написанный на Java.
- **Совместимость:** Kotlin на 100% совместим с Java, что позволяет использовать все существующие библиотеки Android в приложениях Kotlin. Это включает обработку аннотаций, поэтому привязка данных и Dagger тоже работают.
- **Footprint:** у Kotlin очень компактная библиотека времени исполнения, которая может быть дополнительно уменьшена за счет использования ProGuard. В реальном приложении среда

исполнения Kotlin добавляет всего несколько сотен методов и менее 100 Кбайт к размеру файла .apk.

- **Время компиляции:** Kotlin поддерживает эффективную инкрементную компиляцию. Поэтому, хотя для чистых сборок есть дополнительные накладные расходы, инкрементные сборки обычно бывают быстрыми и быстрее, чем с Java.

## Kotlin для JavaScript

Kotlin поддерживает трансляцию кода в JavaScript. Текущая реализация нацелена на ECMAScript 5.1, но есть планы в конечном итоге также нацелиться на ECMAScript 2015.

Когда вы выбираете целью компиляции JavaScript, любой код Kotlin, который является частью проекта, а также стандартная библиотека, поставляемая вместе с Kotlin, компилируется в JavaScript. Однако это исключает JDK и любую используемую JVM или Java-инфраструктуру или библиотеку. Любой файл, который не является Kotlin, будет игнорироваться во время компиляции.

Компилятор Kotlin старается выполнить следующие задачи:

- Обеспечить оптимальный размер получаемого кода JavaScript
- Обеспечить генерацию читабельного кода JavaScript
- Обеспечить взаимодействие с существующими модульными системами
- Обеспечить такую же функциональность в стандартной библиотеке, будь то таргетинг JavaScript или JVM (в максимально возможной степени)

Kotlin может использоваться совместно с существующими сторонними библиотеками и фреймворками, такими как JQuery или ReactJS. Чтобы получить доступ к сторонним инфраструктурам с помощью строго типизированного API, вы можете конвертировать определения TypeScript из репозитория определений типизированного типа в Kotlin с помощью инструмента ts2kt. Кроме того, вы можете использовать динамический тип для доступа к любой инфраструктуре без строгой типизации.

Kotlin также совместим с CommonJS, AMD и UMD, что делает взаимодействие с различными модульными системами простым.