

Scala — это язык программирования, основанный на Java. Его можно использовать для разных целей, он лаконичнее и упорядоченнее, чем Java, но местами сложнее в освоении. Язык опирается на объектно-ориентированный подход: все сущности в коде представлены в виде объектов.

Scala создавался как преемник Java, но не получил такого же широкого распространения по нескольким причинам: необходимость иначе подходить к архитектуре, переучиваться с более привычной Java и многое менять. Тем не менее разработчики на Scala нужны — и получают довольно высокую зарплату.

Язык полностью совместим с Java, работает под той же виртуальной машиной, поэтому нередко его изучают Java-разработчики, которые хотят освоить больше связанных технологий. Еще есть реализация Scala на JavaScript, но она пока менее распространена.

Название читается как «скала» с ударением на первый слог.

Для чего нужен язык Scala

Scala — язык общего назначения. Это значит, что у него нет жестко определенной области применения: его можно использовать практически для любых задач. Чаще всего Scala применяют в отдельных областях веба: высоконагруженные новостные сайты, социальные сети, образовательные сервисы и другие «тяжелые» проекты. Для веба у Scala есть отдельные фреймворки — Lift и Play.

Тем не менее Scala можно использовать и для других задач. В частности, тех, для которых по умолчанию выбирают Java, то есть таких, где важна полная кроссплатформенность. Тут языку сильно помогает наличие JVM, о которой мы поговорим позже. Технически его можно использовать и для «умной» техники, и для программ под любую операционную систему.

Особенности Scala

Scala довольно похож на Java, но если Java — многословный язык, то Scala более краток и лаконичен. Однако код на нем довольно непросто читается: нужно иметь привычку, чтобы понимать, что написано. У Scala есть несколько характерных особенностей.

Строгая типизация. Типизация в Scala — статическая и строгая. Это означает, что тип переменной жестко задается, присваивается раз и навсегда, а еще язык проверяет, чтобы, например, в функцию передавались только переменные определенного типа. В нем нельзя «вольнo» обращаться с типами: например, передать число туда, где нужна строка. Есть автовыведение типов, то есть тип результата какого-то выражения компилятор языка подсчитает сам.

Объектно-ориентированный подход. Scala имеет сильный уклон в объектно-ориентированный подход, даже сама программа в нем считается объектом. Несмотря на то что язык создавался как мультипарадигменный, с ним преимущественно работают именно через ООП. Объектно-ориентированное программирование имеет свои плюсы и минусы: из плюсов — структурирование сущностей в коде, из минусов — громоздкость больших проектов.

Работа под JVM. Основная реализация Scala написана под JVM, поэтому язык полностью кроссплатформенный. JVM — это Java Virtual Machine, особая технология виртуализации, созданная изначально для языка Java. JVM позволяет запустить код на любом устройстве: виртуальная машина сама отвечает за все «технические» детали, связанные с системными и «железными» особенностями. Программисту остается писать код и не думать о совместимости. Код запустится внутри виртуальной системы, а она работает на чем угодно, вплоть до кофеварок (кстати, по одной из версий, именно из-за этого появилось название языка Java)

Преимущества Scala

ООП. Объектно-ориентированный подход позволяет организовать код, сделать сущности в нем более четкими и структурированными. Это важно, чтобы не запутаться. А возможности Scala позволяют реализовать ООП очень широко и точно.

Лаконичность. Более краткий синтаксис — это удобно. Так код занимает меньше места, его быстрее писать и читать.

Кроссплатформенность. Код на Scala можно запустить где угодно, это очевидный плюс. Не нужно беспокоиться о совместимости и о «железе».

Полная совместимость с Java. Scala полностью «понимает» Java-код, может работать совместно с ним и поддерживает все его функции.

Программы на этих двух языках могут спокойно общаться между собой без ошибок, даже если код сложный.

Масштабируемость. Scala устроен так, что проект на нем можно легко масштабировать в большую или меньшую сторону. Это удобно, если вы проектируете систему, которая в будущем может серьезно расширяться.

Недостатки Scala

Сложность. Scala сложнее, чем Java: чтобы его выучить, желательно уже знать Java, а синтаксис и особенности построения программ в нем совершенно другие и довольно непростые. Поэтому не каждый программист в принципе пойдет на то, чтобы его изучать, когда в мире уже есть Java.

Медленная компиляция. Работать программа может быстро, а вот компилироваться, то есть «собираться» из исходного кода в исполняемый, — медленно. Если проект большой и в нем много частей, это критично.

Низкая распространенность. Scala менее популярен, чем Java, из-за высокой сложности и необходимости совершенно иначе подходить к построению программ. В частности, в России сейчас вакансий относительно немного. Компаниям проще воспользоваться более

привычной Java, чем с нуля строить совершенно иную архитектуру или вообще переводить на нее уже существующий большой проект. Но Scala постепенно набирает популярность: например, на него перешла социальная сеть Twitter, на нем работает Coursera. Так что, возможно, все еще впереди.