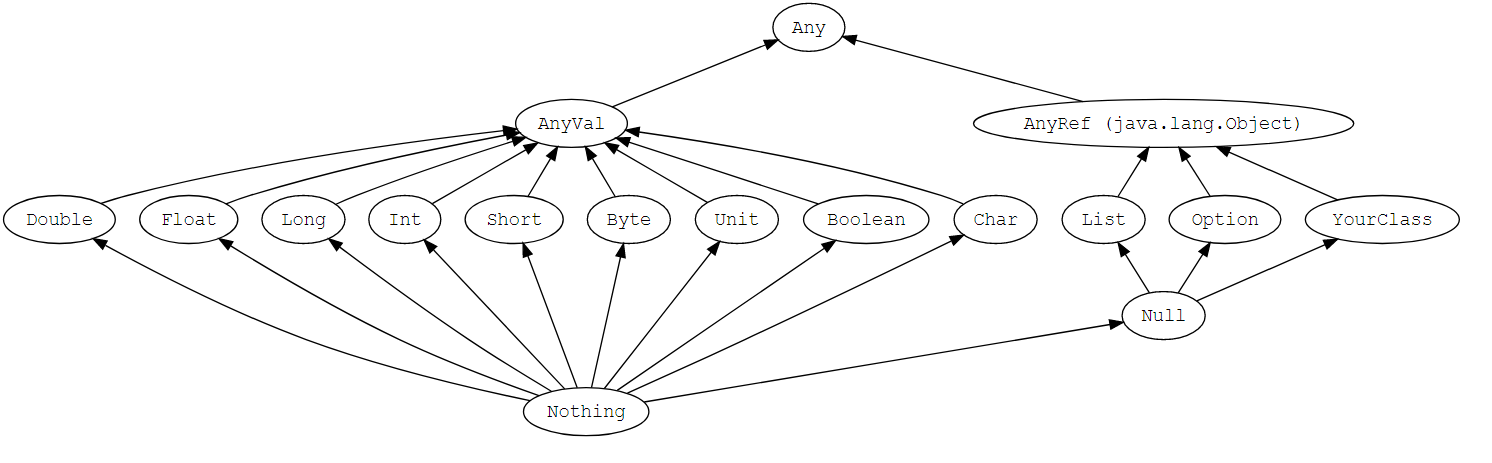
Стандартная библиотека Scala

Примитивы, коллекции, алгебраические типы данных.

Типы данных

В Scala все значения имеют тип, включая числовые значения и функции. Диаграмма ниже иллюстрирует подмножество иерархии типов.



Иерархия типов Scala

[Any](https://www.scala-lang.org/api/2.12.1/scala/Any.html) это супертип всех типов, также называемый верхним типом. Он определяет несколько универсальных методов, таких как equals, hashCode и toString. У Any есть два прямых подкласса: AnyVal и AnyRef.

AnyVal представляет числовые типы. Существует девять предварительно определенных числовых типов и они никогда не могут быть равны ‘null’: Double, Float, Long, Int, Short, Byte, Char, Unit, и Boolean. Unit - это числовой тип, который не содержит значимой информации (также обозначает пустое множество). Есть только один представитель типа Unit, который можно объявить вот так: (). Все функции должны возвращать что-то, поэтому иногда Unit полезный для возврата тип.

AnyRef представляет ссылочные типы. Все типы, не относящиеся к “числовым типам”, называются ссылочными типами. Каждый объявляемый пользователем тип в Scala является подтипом AnyRef. Если в Scala исходить из контекста среды исполнения Java, AnyRef соответствует java.lang.Object.

Вот пример, демонстрирующий, что строки, целые числа, символы, логические значения и функции являются объектами, как и любой другой объект:

val list: List[Any] = List(

"a string",

732, // целое число

'c', // символ

true, // логическое значение

() => "анонимная функция возвращающая строку"

)

list.foreach(element => println(element))

Объявляем переменную list типа List[Any]. Список инициализируется элементами различных типов, но все они являются экземпляром scala.Any, так что вы можете добавить их в список.

Ниже приведен вывод программы:

a string

732

c

true

<function>

Приведение типа

Числовые типы могут быть приведены следующим образом:

Например:

val x: Long = 987654321

val y: Float = x // 9.8765434E8 (заметьте, что некоторая точность теряется в этом случае.)

val face: Char = '☺'

val number: Int = face // 9786

Приведение типа - однонаправленно. Следующий пример не скомпилируется:

val x: Long = 987654321

val y: Float = x // 9.8765434E8

val z: Long = y // обратно не подходит

Вы также можете приводить к своему подтипу. Об этом мы поговорим позже в ходе нашего обзора.

Nothing и Null

Nothing является подтипом всех типов, также называемым нижним типом. Нет значения, которое имеет тип Nothing. Обычно он используется чтоб дать сигнал о не вычислимости, например брошено исключение, выход из программы, бесконечное зацикливание (т.е. это тип выражения, которое не вычисляется).

Null подтип всех ссылочных типов (т.е. любой подтип AnyRef). Он имеет одно значение, определяемое ключевым словом литерала null. Nullпредоставляется в основном для функциональной совместимости с другими языками JVM и почти никогда не должен использоваться в коде Scala. Об альтернативах null мы поговорим позднее.