Scala basic tips for Java developers

Topics

- Overview
- Scala X Java
- Variables
- Data types
- Option, Some and None
- Operators / If else / Loop
- Classes & Objects
- Traits
- Access modifiers
- Functions
- Closure
- Pattern matching
- Extractors
- Exception handling
- Regular expression

Overview

- Object oriented
- Functional
- Runs on JVM
- Easy integration with Java
- Scala REPL

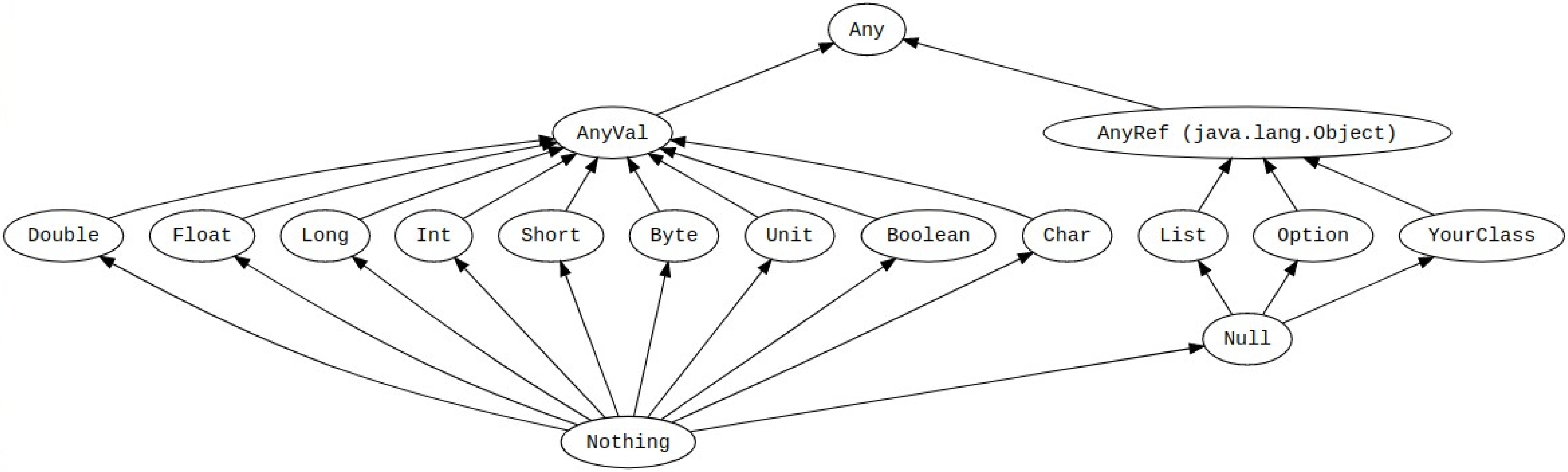
Scala vs Java

- All types are objects
- Type inference
- Nested Functions
- Functions are objects
- Domain specific language (DSL) support
- Traits
- Closures
- Concurrency support inspired by Erlang

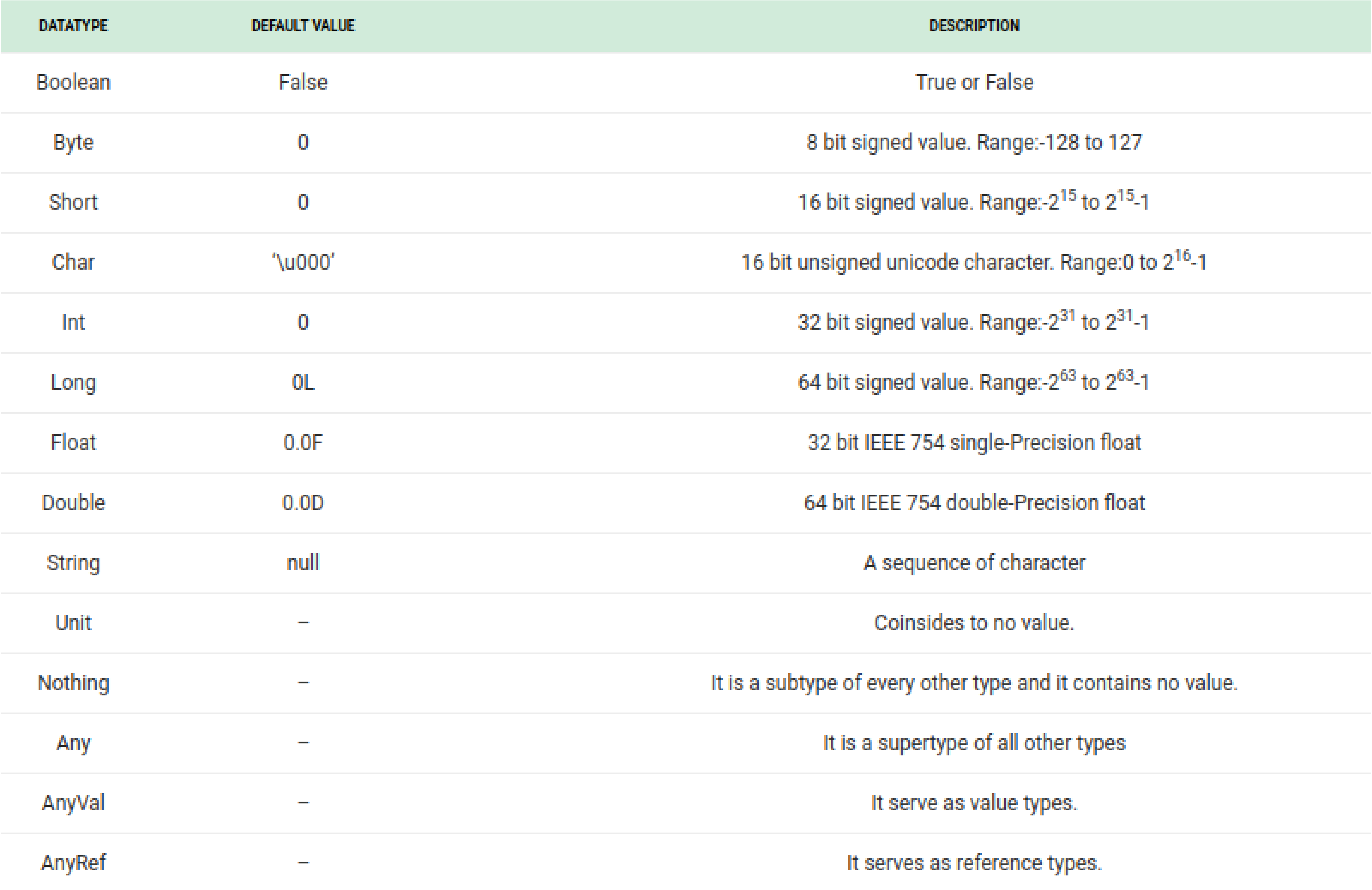
Variable

- Variable declaration
  - var : Variable
  - val : value (like final in java)
- Variable Type Inference
- Multiple assignments
- Variable Scope
  - Fields (var / val)
  - Method (val)
  - Local (var / val)

Data types


Any
AnyVal
AnyRef (java.lang.Object)
Double
Float
Long
Int
Short
Byte
Unit
Boolean
Char
List
Option
YourClass
Null
Nothing

Data types



DATATYPE	DEFAULT VALUE	DESCRIPTION
Boolean	False	True or False
Byte	0	8 bit signed value. Range:-128 to 127
Short	0	16 bit signed value. Range:-2^{15} to 2^{15}-1
Char	‘\u000’	16 bit unsigned unicode character. Range:0 to 2^{16}-1
Int	0	32 bit signed value. Range:-2^{31} to 2^{31}-1
Long	0L	64 bit signed value. Range:-2^{63} to 2^{63}-1
Float	0.0F	32 bit IEEE 754 single-Precision float
Double	0.0D	64 bit IEEE 754 double-Precision float
String	null	A sequence of character
Unit	–	Coinsides to no value.
Nothing	–	It is a subtype of every other type and it contains no value.
Any	–	It is a supertype of all other types
AnyVal	–	It serve as value types.
AnyRef	–	It serves as reference types.

Data types

- Arrays
- Collections
  - Mutable
  - Imutable
  - Types
    - List
    - Map
    - Set
    - Tuple
    - Iterators

Option, Some and None

- Use it to avoid return null values
- Some and None are child of Option
- Either, Left, and Right
  - Either is just like Option
  - Right is just like Some
  - Left is just like None, except you can include content with it to describe the problem

Operators / If else / Loop

- If / else
- While – loop
- Do – while loop
- For loop
- Yield

Classes & Objects

- Class
- Class extend
- Implicit class
- Singleton class

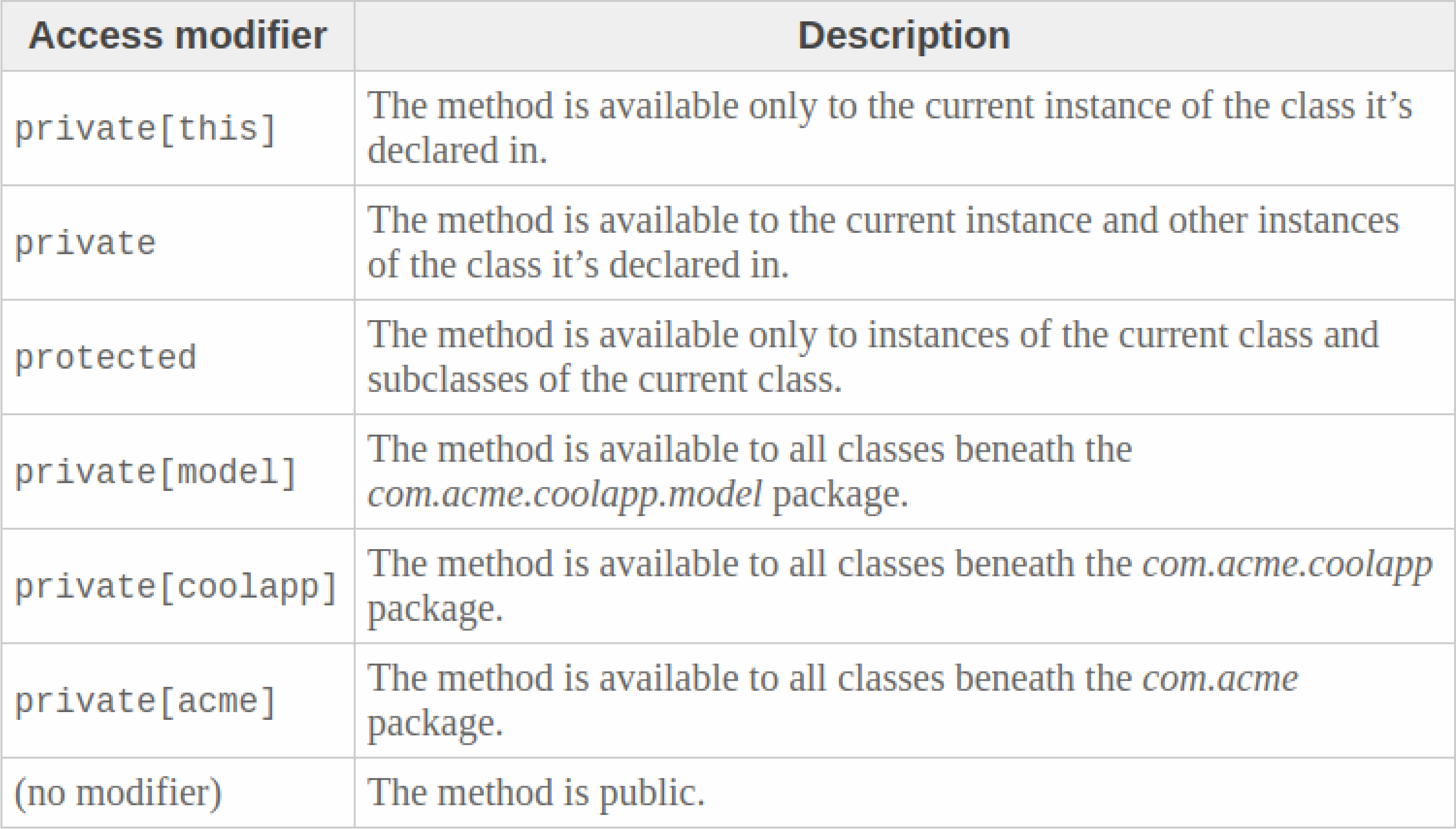
Traits

- Similar to java interface
- Encapsulates method and fields
- Possible to implementeds methods
- A class can have any number of traits

Access modifiers

- Public
- Protected
- Private
- Scope of protection

Access modifiers



Access modifier	Description
`private[this]`	The method is available only to the current instance of the class it’s declared in.
`private`	The method is available to the current instance and other instances of the class it’s declared in.
`protected`	The method is available only to instances of the current class and subclasses of the current class.
`private[model]`	The method is available to all classes beneath the *com.acme.coolapp.model* package.
`private[coolapp]`	The method is available to all classes beneath the *com.acme.coolapp* package.
`private[acme]`	The method is available to all classes beneath the *com.acme* package.
(no modifier)	The method is public.

Functions

- Methods are function inside an object
- Variable arguments
- Default parameter
- Named arguments
- Nested functions
- High-order functions
- Anonymous functions
- Currying Functions

Closure

- A closure is a function , whose return value depends on the value of one or more variables declared outside this function

Pattern matching

- Expressions
- Case

Extractors

- Apply
- Unapply
- Use with pattern matching

Regular Expression

- Rich String – r( )
- Find methods

Exception handling

- Try
- Catch
- Finally

References

- https://www.amazon.com/Programming-Scala-Updated-2-13-ebook/dp/B082T2ZNJG
- https://alvinalexander.com/scala
- https://docs.scala-lang.org/
- https://rocketeer.be/articles/concurrency-in-erlang-scala/
- https://www.tutorialspoint.com/scala/index.htm