How Classes are Organized

Packages

- Classes and objects are organized in packages
- To place a class or object inside a package, use a package clause at the top of your source file
- It is customary to organize your project so that the package corresponds to the path to the file
- You can refer it by its *fully qualified name*

Imports

- To avoid having to write long package names, you have imports
- Imports come in several forms:
    - to import one class: ***import week3.Rational***
    - to import more classes: ***import week3.{Rational, Hello}***
    - to import everything in a package: ***import week3._***
- The first two forms are called *named imports* and the last form is called a *wildcard import*
- You can import from either a package or an object
- Conceptually, a package, in Scala, is just a very large object that has members that are each an individual source file

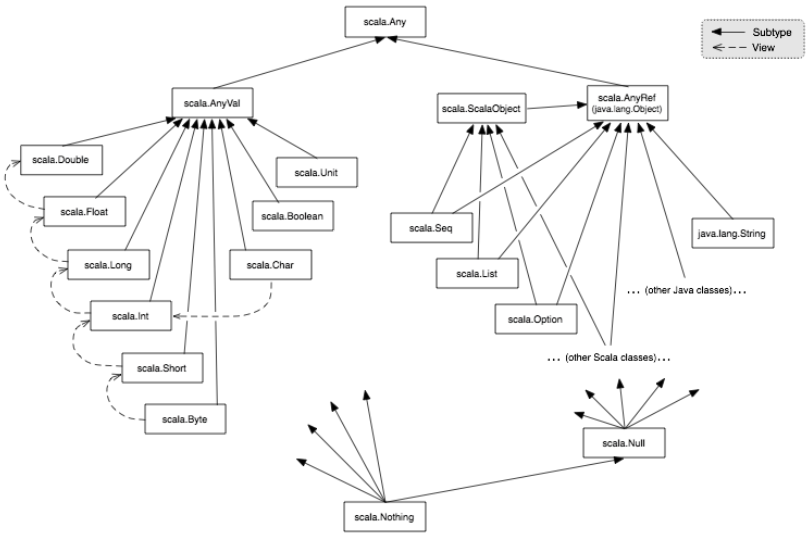
Automatic Imports

- Some entities are automatically imported in any Scala program
- There are:
    - All members of package ***scala***
    - All members of package ***java.lang***
    - All members of the singleton object ***java.Predef***
- Here are the fully qualified names of some types and functions which you have seen so far:
    - Int --> ***scala.Int***
    - Boolean --> ***scala.Boolean***
    - Object --> ***java.lang.Object***
    - require --> ***scala.Predef.require***
    - assert --> ***scala.Predef.assert***

Traits

- In Java, as well as in Scala, a class can only have one superclass
- A trait is declared like an abstract class, just with trait instead of abstract class
- Classes, objects and traits can inherit from at most one class, but arbitrary many traits
- Traits resemble interfaces in Java, but are more powerful because they can have parameters and can contain fields and concrete methods

Scala's Class Hierarchy



Top Types

At the top of the type hierarchy we find:

- ***Any***
  - The base type of all types
  - Methods: ***==, !=, equals, hashCode, toString***
- ***AnyRef***
  - The base type of all reference types
  - Alias of ***java.lang.Object***
- ***AnyVal***
  - The base type of all primitive types

The Nothing Type

- ***Nothing*** is at the bottom of Scala's type hierarchy
- It is a subtype of every other type
- There is no value of type ***Nothing***
- Why is that useful?
  - To signal abnormal termination
  - As an element type of empty collections

Exceptions

- Scala's exception handling is similar to Java's
- The expression ***throw Exception*** aborts evaluation with the exception ***Exception***
- The type of this expression is *Nothing*, because there is no value that is returned from the expression

Exercise: *What is the type of:* ***if true then 1 else false***

--> *AnyVal*