

O'REILLY®

Scala Core Programming

Sealed Traits, Collections, and Functions

Daniel Hinojosa

About Me...

Daniel Hinojosa

Programmer, Developer, Consultant,
Instructor, and Speaker



Notable Content:

Testing in Scala (Book)
Beginning Scala Programming (Video)
Java & TDD (Video Training)
Scala Beyond The Basics
Scala Core Programming Series

Speaker:

OSCON
No Fluff Just Stuff Tour
DevNexus

dhinojosa@evolutionnext.com
@dhinojosa

Structure of the Class

- Project Using Tests
- Performed in IntelliJ and Eclipse
- Labs Included in Project
- Definitely Ask Questions

Checklist before we proceed...

- We provided pre-setup configuration already.
- Be sure `JAVA_HOME` is set and that the following works:
 - `javac -version`
 - `java -version`
- Be sure SBT is setup and ensure that the following works
 - `sbt about`
- Be sure that Scala plugins are installed in IntelliJ and Eclipse

Bringing in our Project

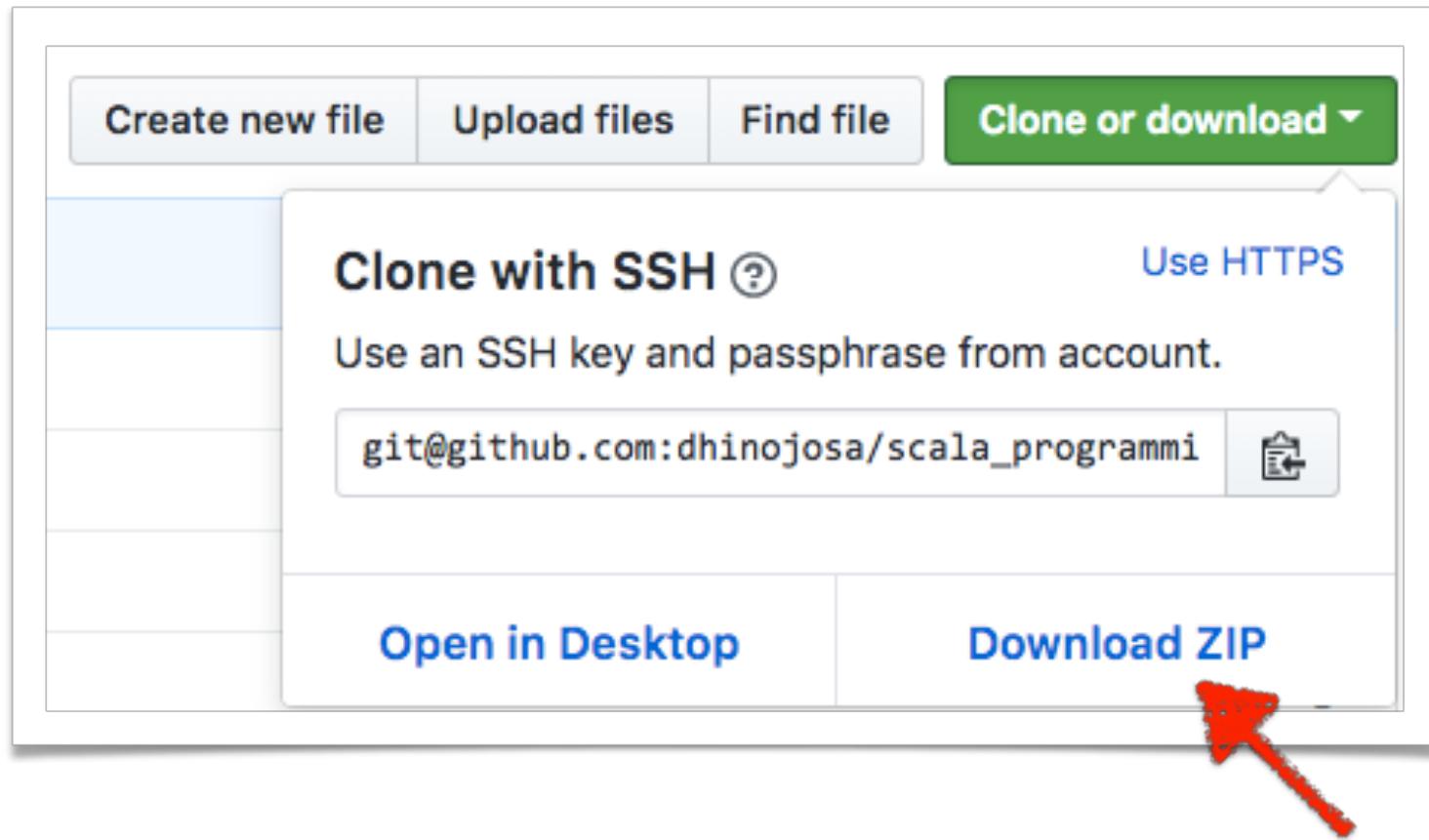
Repository Location, Clone it!

https://github.com/dhinojosa/scala_core_programming_2

or use the abbreviation...

<https://bit.ly/2Igo0yp>

But, if don't know Git too well...



Once downloaded...Run SBT

```
-----  
~/Development/scala_core_programming_2(master*) » sbt
```

You should see something like this...

```
[~/Development/scala_core_programming_2(master*) » sbt
[info] Loading settings from idea.sbt,plugins.sbt ...
[info] Loading global plugins from /Users/danno/.sbt/1.0/plugins
[info] Loading settings from plugins.sbt ...
[info] Loading project definition from /Users/danno/Development/scala_core_programming_2/project
[info] Loading settings from build.sbt ...
[info] Set current project to scala_core_programming_2 (in build file at /Users/danno/Development/scala_core_programming_2/)
[info] sbt server started at local:///Users/danno/.sbt/1.0/server/60474e/sock
sbt:scala_core_programming_2> █
```

You'll need to update your project...

```
[~/Development/scala_core_programming_2(master*) » sbt
[info] Loading settings from idea.sbt,plugins.sbt ...
[info] Loading global plugins from /Users/danno/.sbt/1.0/plugins
[info] Loading settings from plugins.sbt ...
[info] Loading project definition from /Users/danno/Development/scala_core_programming_2/project
[info] Loading settings from build.sbt ...
[info] Set current project to scala_core_programming_2 (in build file
no/Development/scala_core_programming_2/)
[info] sbt server started at local:///Users/danno/.sbt/1.0/server/
60474e/sock
[sbt:scala_core_programming_2> update
[info] Updating ...
[info] Done updating.
[success] Total time: 2 s, completed Nov 5, 2018, 9:10:56 PM
sbt:scala_core_programming_2> █
```

Updating Your Project

- This will download Scala and all your dependencies into your `~/.ivy2`
- This may take a while depending on your connection speed.
- If you have any problems, there might be issues with network connection, or proxy based issues

Proxy Issues

- If you have problems because of any proxy server, add the following to your environment variables and change username and password accordingly

```
export JAVA_OPTS="$JAVA_OPTS \
-Dhttp.proxyHost=yourserver \
-Dhttp.proxyPort=8080 \
-Dhttp.proxyUser=username \
-Dhttp.proxyPassword=password"
```

Lab: Let's get setup!

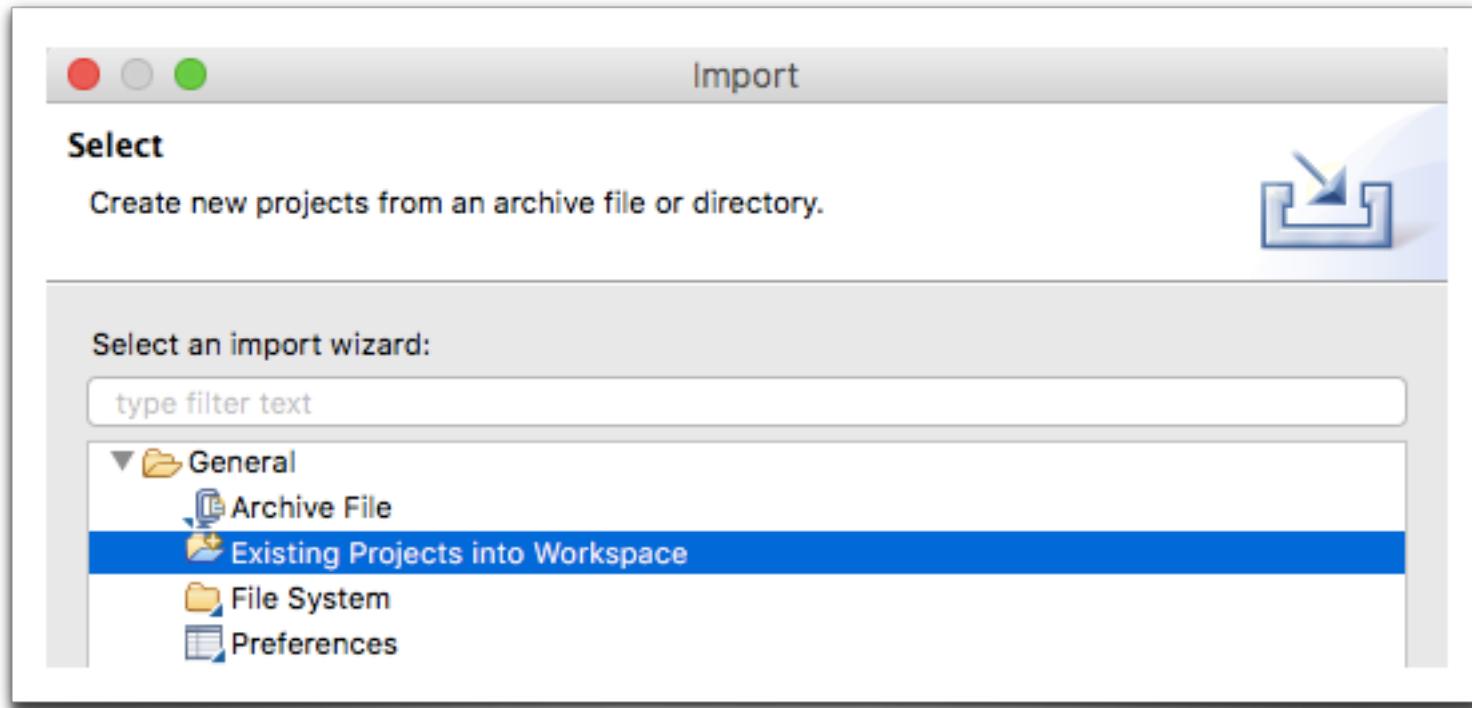
- Clone or download
`scala_programming_fundamentals_2` from
[https://github.com/dhinojosa/
scala_programming_fundamentals_2](https://github.com/dhinojosa/scala_programming_fundamentals_2)
- Run sbt at the command line
- Once in the sbt console, run update
- Wait for the download to complete

Setup for Eclipse

Run the Eclipse Plugin in SBT

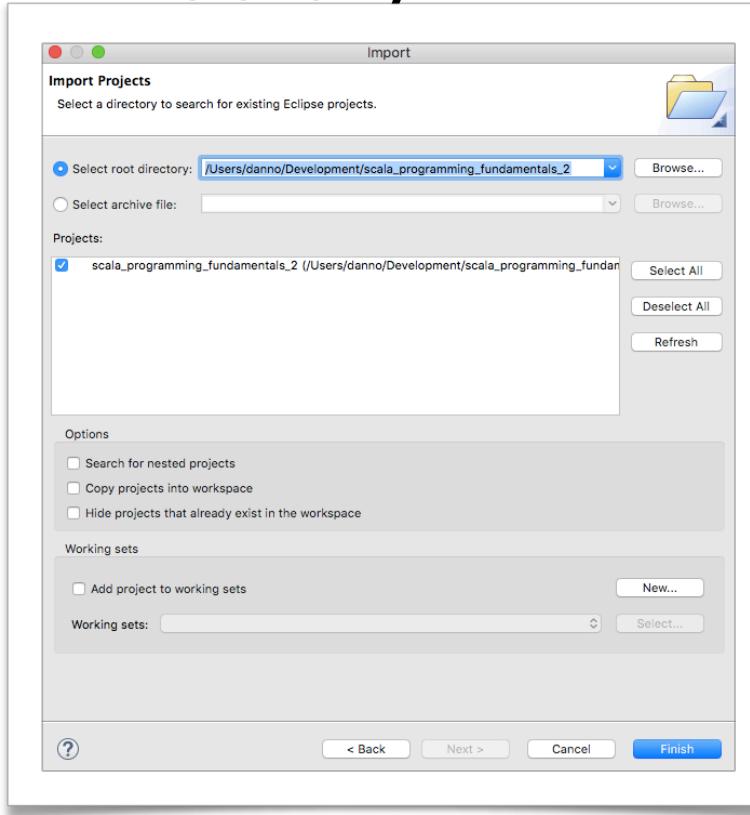
```
[sbt:scala_core_programming_2> eclipse
[info] About to create Eclipse project files for your project(s).
[info] Successfully created Eclipse project files for project(s):
[info] scala_core_programming_2
sbt:scala_core_programming_2> █
```

Import Existing Project



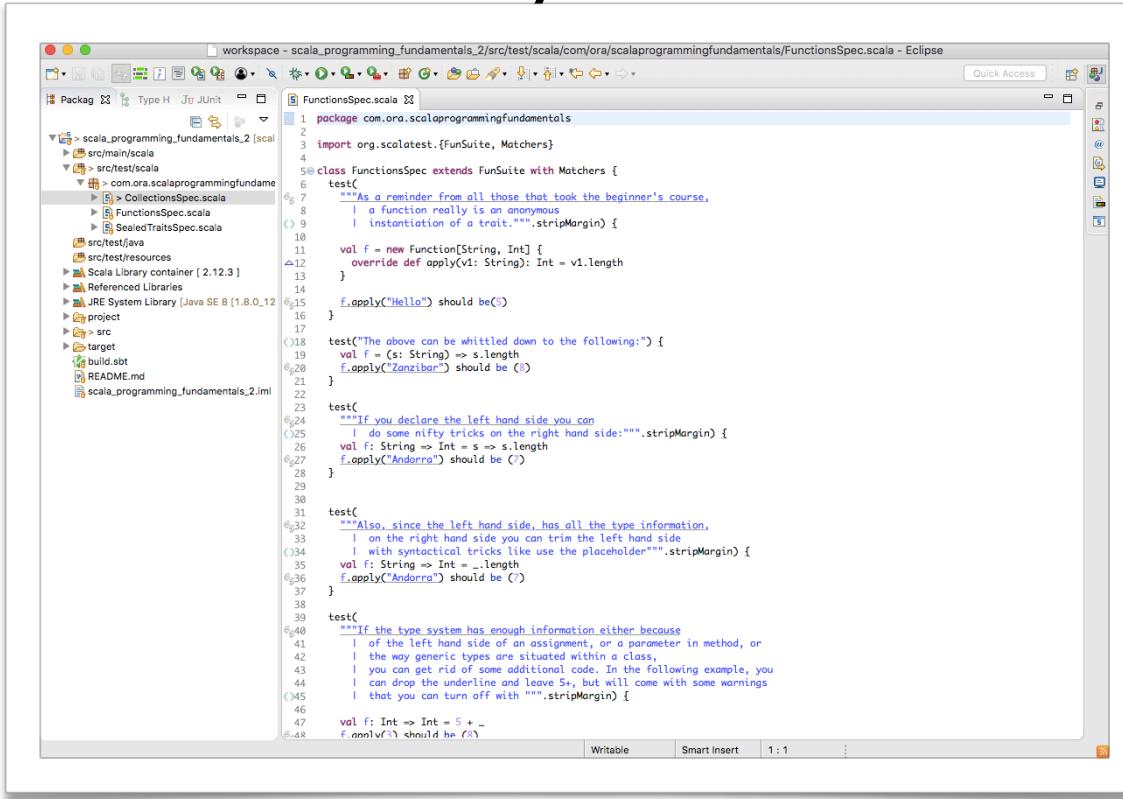
File > Import Project >
Select Existing Projects Into Workspace > Next

Select Root Directory



Select the Root Directory of the Project using Browse,
Ensure that the Project is Found, hit Finish

Project is now ready



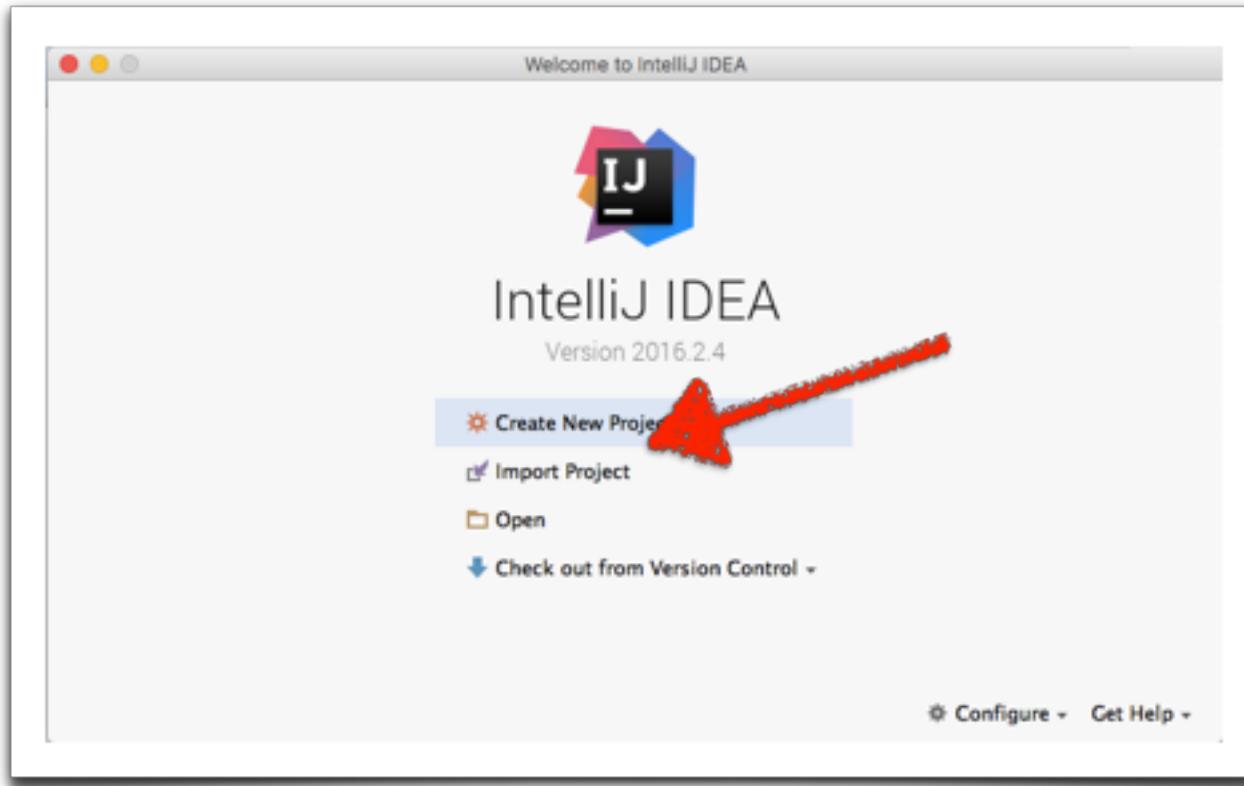
```
workspace - scala_programming_fundamentals_2/src/test/scala/com/ora/scalaprogrammingfundamentals/FunctionsSpec.scala - Eclipse

1 package com.oras.scalaprogrammingfundamentals
2
3 import org.scalatest.{FunSuite, Matchers}
4
5 @class FunctionsSpec extends FunSuite with Matchers {
6   test(
7     """As a reminder from all those that took the beginner's course,
8      | a function really is an anonymous instantiation of a trait.""".stripMargin)
9   {
10     val f = new Function[String, Int] {
11       override def apply(v1: String): Int = v1.length
12     }
13
14     f.apply("Hello") should be(5)
15   }
16
17   test("The above can be whittled down to the following:")
18   {
19     val f = (s: String) => s.length
20     f.apply("Zanzibar") should be(8)
21   }
22
23   test(
24     """If you declare the left hand side you can
25       | do some nifty tricks on the right hand side:""".stripMargin)
26   {
27     val f: String => Int = s => s.length
28     f.apply("Andorra") should be(8)
29   }
30
31   test(
32     """Also, since the left hand side, has all the type information,
33       | on the right hand side you can trim the left hand side
34       | with syntactical tricks like use the placeholder:""".stripMargin)
35   {
36     val f: String => Int = _.length
37     f.apply("Andorra") should be(8)
38   }
39
40   test(
41     """If the type system has enough information either because
42       | of the left hand side of an assignment, or a parameter in method, or
43       | the way generic types are situated within a class,
44       | you can get rid of some additional code. In the following example, you
45       | can drop the underline and leave 5+, but will come with some warnings
46       | that you can turn off with """.stripMargin)
47
48   val f: Int => Int = 5 + _
49   f.apply(1) should be(6)
50 }
```

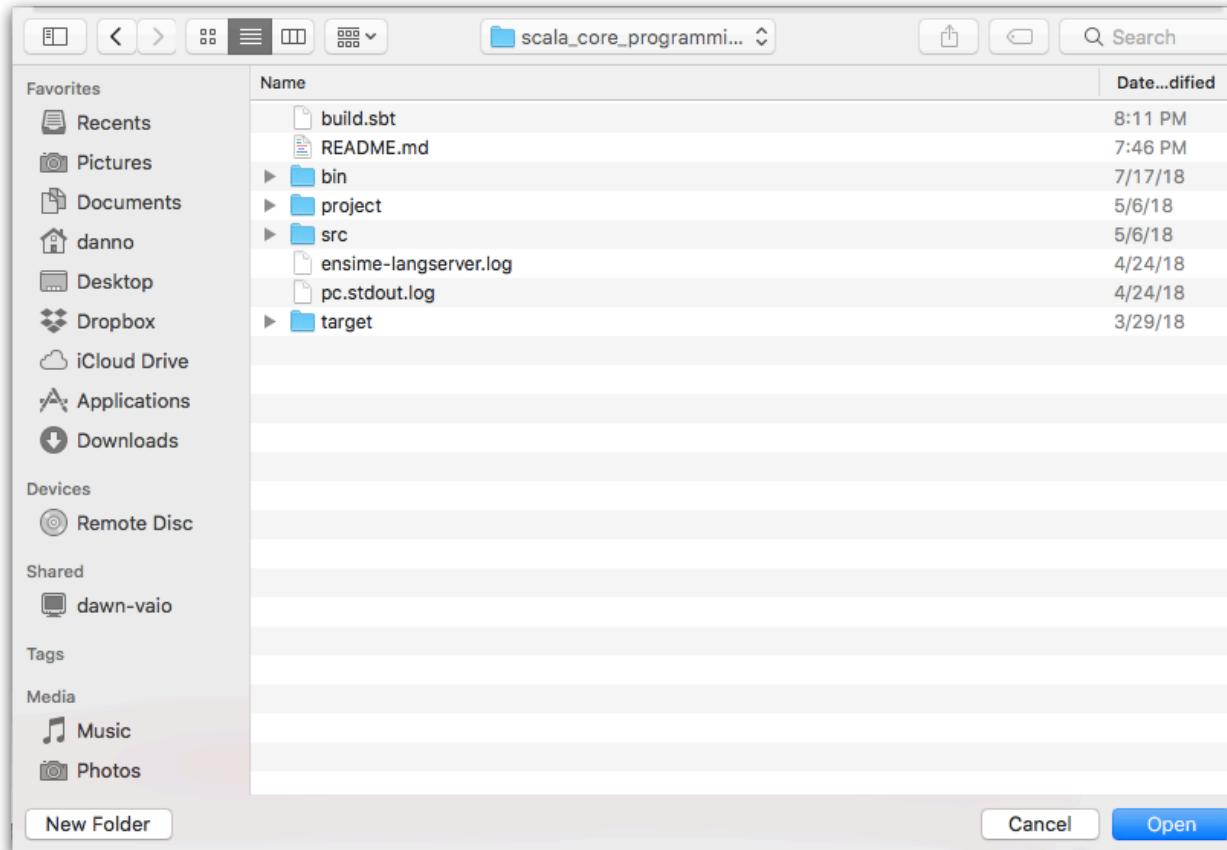
Select the Root Directory of the Project using Browse,
Ensure that the Project is Found, hit Finish

Setup for IntelliJ

From IntelliJ Intro Screen, or from the splash screen, select Import Project



Locate the `scala_core_programming_2` project and click OK, accept defaults



Project is now ready...

The screenshot shows the IntelliJ IDEA interface with the following details:

- Project View:** On the left, the project structure is displayed under "scala_core_programming_2". It includes ".ensime_cache", ".idea", ".settings", "bin", "project [scala_core_programming_2-build]", "src" (containing "main" and "test" subfolders), and "target" (containing ".cache-main", ".cache-tests", ".classpath", ".gitignore", ".project", and "build.sbt").
- Code Editor:** The main window shows the file "SealedTraitsSpec.scala" with the following code:

```
package com.ora.scalacoreprogramming
import ...
class SealedTraitsSpec extends FunSuite with Matchers {
  test("""A trait is analogous to an interface in Java, a
        | no concrete implementation is considered abstract
        | inheritance of a trait is added using the keyword
        | .stripMargin) {
    trait Vehicle {
      def increaseSpeed(ms: Int): Vehicle
      def decreaseSpeed(ms: Int): Vehicle
      def currentSpeedMetersPerHour: Int
    }

    trait Fun
    trait FreshAir

    case class Bicycle(currentSpeedMetersPerHour: Int)
      extends Vehicle with Fun with FreshAir {

      override def increaseSpeed(ms: Int): Vehicle =
        this.copy(currentSpeedMetersPerHour + ms)

      override def decreaseSpeed(ms: Int): Vehicle =
        this.copy(currentSpeedMetersPerHour - ms)
    }
}
```
- Toolbars and Status Bar:** The top bar shows tabs for "FunctionsSpec.scala", "CollectionsSpec.scala", and "SealedTraitsSpec.scala". The status bar at the bottom shows "sbt: dump project structure from sbt", "147:12", "LF", "UTF-8", "Git: master", and a terminal icon.

Sealed Traits and Abstract Classes, Collections, and Functions

Thank You