

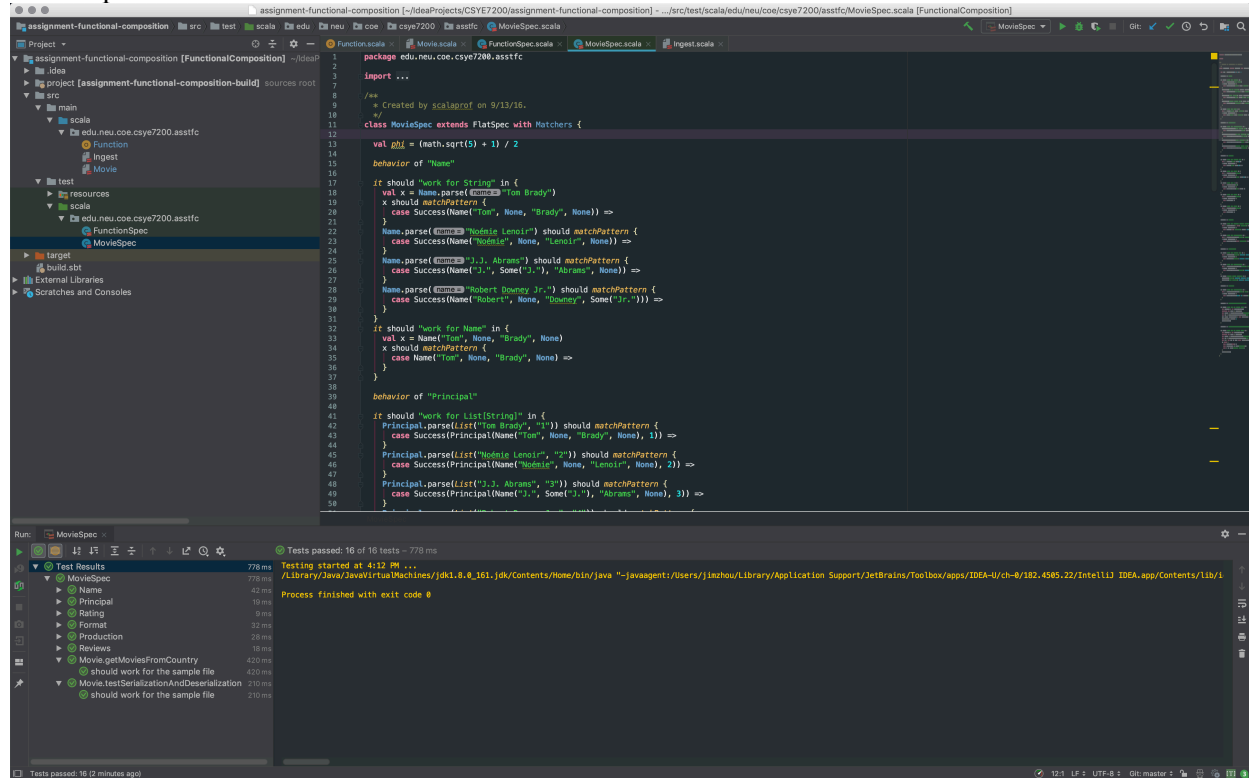
Assignment 5: Function Composition

Name: Qixiang Zhou

NUID: 001822974

Snapshot of the test case

MovieSpec:



The screenshot shows the IntelliJ IDEA interface with the `MovieSpec.scala` file open. The file is part of the `edu.neu.coe.csye7200.assfrc` package and extends `FlatSpec` with `Hatchers`. It defines a `Movie` class and tests its `parse` and `parseList` methods. The test results pane at the bottom shows that all 16 tests passed.

```
package edu.neu.coe.csye7200.assfrc

import ...

// Created by scalapref on 9/13/16.
//

class MovieSpec extends FlatSpec with Hatchers {

  val phi = (math.sqrt(5) + 1) / 2

  behavior of "Name"

  it should "work for String" in {
    val x = Name.parse("Tom Brady")
    x should matchPattern {
      case Success(Name("Tom", None, "Brady", None)) =>
    }
    Name.parse("Noddy Lenoir") should matchPattern {
      case Success(Name("Noddy", None, "Lenoir", None)) =>
    }
    Name.parse("J.J. Abrams") should matchPattern {
      case Success(Name("J.J.", Some("J.J."), "Abrams", None)) =>
    }
    Name.parse("Robert Downey Jr.") should matchPattern {
      case Success(Name("Robert", None, "Downey", Some("Jr."))) =>
    }
  }

  it should "work for Name" in {
    val x = Name("Tom", None, "Brady", None)
    x should matchPattern {
      case Name("Tom", None, "Brady", None) =>
    }
  }

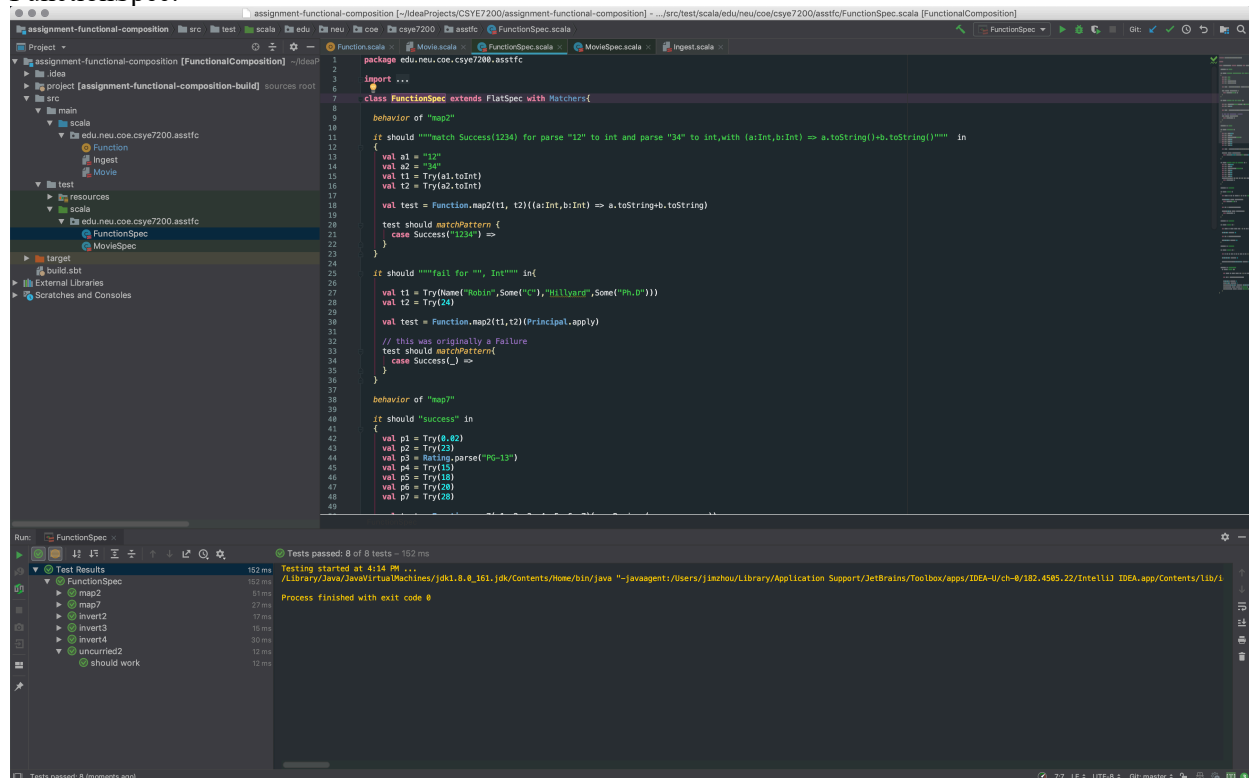
  behavior of "Principal"

  it should "work for List(String)" in {
    Principal.parseList("Tom Brady", "1") should matchPattern {
      case Success(Principal(Name("Tom", None, "Brady", None), 1)) =>
    }
    Principal.parseList("Noddy Lenoir", "2") should matchPattern {
      case Success(Principal(Name("Noddy", None, "Lenoir", None), 2)) =>
    }
    Principal.parseList("J.J. Abrams", "3") should matchPattern {
      case Success(Principal(Name("J.J.", Some("J.J."), "Abrams", None), 3)) =>
    }
  }
}
```

Run: MovieSpec - Tests passed: 16 of 16 tests - 778 ms

Test Results	Time
MovieSpec	778 ms
Name	42 ms
Principal	16 ms
Rating	9 ms
Format	32 ms
Production	28 ms
Reviews	18 ms
Movie.getMoviesFromCountry	420 ms
should work for the sample file	420 ms
Movie.testSerializationAndDeserialization	210 ms
should work for the sample file	210 ms

FunctionSpec:



The screenshot shows the IntelliJ IDEA interface with the `FunctionSpec.scala` file open. The file is part of the `edu.neu.coe.csye7200.assfrc` package and extends `FlatSpec` with `Hatchers`. It tests the `map2` function. The test results pane at the bottom shows that all 8 tests passed.

```
package edu.neu.coe.csye7200.assfrc

import ...

class FunctionSpec extends FlatSpec with Hatchers {

  behavior of "map2"

  it should "match Success(1234) for parse '12' to int and parse '34' to int, with (a:Int,b:Int) => a.toString()+b.toString()" in {
    val a1 = "12"
    val a2 = "34"
    val t1 = Try(a1.toInt)
    val t2 = Try(a2.toInt)
    val test = Function.map2(t1, t2)((a:Int,b:Int) => a.toString()+b.toString)
    test should matchPattern {
      case Success("1234") =>
    }
  }

  it should "fail for ' '", Int"" in {
    val t1 = Try(Name("Robin", Some("C"), "Hillward", Some("Ph.D")))
    val t2 = Try(24)
    val test = Function.map2(t1, t2)(Principal.apply)
    // this was originally a Failure
    test should matchPattern {
      case Success(_) =>
    }
  }

  behavior of "map7"

  it should "success" in {
    val p1 = Try(8.82)
    val p2 = Try(23)
    val p3 = Try(15)
    val p4 = Try(15)
    val p5 = Try(18)
    val p6 = Try(28)
    val p7 = Try(28)
  }
}
```

Run: FunctionSpec - Tests passed: 8 of 8 tests - 162 ms

Test Results	Time
FunctionSpec	162 ms
map2	51 ms
map7	27 ms
invert2	17 ms
invert3	16 ms
invert4	30 ms
uncurried2	12 ms
should work	12 ms