01: Coding Scala Way – Part 1

Posted on October 14, 2016 by Arulkumaran Kumaraswamipillai

Example #1: Read from a list & write to a list

Scala the Java Way

```
1
2  object ListToList extends App {
3
4    val inputList = List("Java", "Scala", "Ruby")
5    val outputList = scala.collection.mutable.List
6
7    for(item <- inputList) {
8      outputList += item + " Programming"
9    }
10
11   println(outputList.toList)
12 }
13
```

Output: List(Java Programming, Scala Programming, Ruby Programming)

Scala the Scala Way: Using the "map" functional combinator

Immutable code as shown below using the "map" function.

```
1
2 object ListToList extends App {
3
4   val inputList = List("Java", "Scala", "Ruby")
5   val outputList = inputList.map(item => item + "
6   println(outputList)  //List(Java Programming, S
7 }
8
```

Scala the Scala Way: Using recursion

```
1
2  object ListToList extends App {
3
4    val inputList = List("Java", "Scala", "Ruby")
5
6    def add(xs: List[String]) : List[String] = {
7      xs match {
8        case List() => List[String]()
9        case y :: ys => (y + " Programming") :: ad
10     }
11   }
12
13   println(add(inputList)) //List(Java Programmin
14 }
15
```

Example #2: Aggregate a list of values

Scala the Java Way by mutating state

```
1
2  object AggregateValue extends App {
3    var total = 0 //mutable
4
5    def sum(input: List[Int]): Int = {
6      for (num <- input) {
7        total += num
8      }
9
```


16 Technical Key Areas

80+ step by step Java Tutorials

```
10      total
11    }
12
13    println(sum(List(1,2,3)))// 6
14  }
15
```

Scala the Scala Way using the "foldLeft" functional combinator

```
1
2  object AggregateValue extends App {
3    var total = 0 //mutable
4
5    def sum(input: List[Int]): Int = {
6      input.foldLeft(0)((accumulator, item) => acc
7    }
8
9    println(sum(List(1,2,3)))// 6
10 }
11
```

Example #3: Conditional Aggregates

Scala the Java Way

```
1
2  object AggregatingConditionally extends App {
3
4    def statsByExperience(input: List[Person]): Sa
5      val stats = new SalaryByExperienceTotals()
6
7      for (person <- input) {
8        if (person.experience < 10) {
9          stats.underTen += person
10       } else if (person.experience >= 10 && pers
11         stats.tenToTwenty += person
12       } else if (person.experience > 20) {
13         stats.overTwenty += person
14       }
15     }
16
17     stats
18   }
19
20   val persons = List(Person(25, 4500000), Person
21   println(statsByExperience(persons))
22
23 }
24
25 case class Person(experience: Int, salaryInCents
26
27 case class SalaryByExperienceTotals(
28   underTen: scala.collection.mutable.ListBuffer[
```


Spring & HIbernate Tu
Tools Tutorials (19)
Other Tutorials (45)

100+ Java pre-interview coding tests

open all | close all
Can you write code? (
♦ Complete the given
Converting from A to I
Designing your classe
Java Data Structures
Passing the unit tests
What is wrong with th
Writing Code Home A
Written Test Core Jav
Written Test JEE (1)

How good are your .....?

open all | close all
Career Making Know-
Job Hunting & Resum

```
29   tenToTwenty: scala.collection.mutable.ListBuff
30   overTwenty: scala.collection.mutable.ListBuffe
31  ) {
32
33   override def toString: String = {
34     underTen + " , " + tenToTwenty  + " , " + ov
35   }
36 }
37
```

Output: ListBuffer(Person(5,2500000)) ,
ListBuffer(Person(20,9500000)) ,
ListBuffer(Person(25,4500000))

Scala the Scala Way using case statements for the conditions & immutable List

```
1
2  object AggregatingConditionally extends App {
3
4    def statsByExperience(input: List[Person]): Sa
5
6      input.foldLeft(new SalaryByExperienceTotals(
7        person match {
8          case _ if(person.experience < 10) => sta
9          case _ if(person.experience >= 10 && per
10         case _ => stats.copy(overTwenty = person
11       }
12     }
13   }
14
15   val persons = List(Person(25, 4500000), Person
16   println(statsByExperience(persons))
17
18 }
19
20 case class Person(experience: Int, salaryInCents
21
22 case class SalaryByExperienceTotals(
23   underTen: List[Person] = List(),
24   tenToTwenty: List[Person] = List(),
25   overTwenty: List[Person] = List()
26  ) {
27
28   override def toString: String = {
29     underTen + " , " + tenToTwenty  + " , " + ov
30   }
31 }
32
```

Output: ListBuffer(Person(5,2500000)) ,
ListBuffer(Person(20,9500000)) ,
ListBuffer(Person(25,4500000))

Popular Member Posts

♦ 11 Spring boot interview questions & answers

850 views

♦ Q11-Q23: Top 50+ Core on Java OOP Interview Questions & Answers

768 views

001A: ♦ 7+ Java integration styles & patterns interview questions & answers

399 views

18 Java scenarios based interview Questions and Answers

387 views

♦ 7 Java debugging interview questions & answers

308 views

01b: ♦ 13 Spring basics Q8 – Q13 interview questions & answers

305 views

01: ♦ 15 Ice breaker questions asked 90% of the time in Java job interviews with hints

297 views

♦ 10 ERD (Entity-Relationship Diagrams) Interview Questions and Answers

293 views

♦ Q24-Q36: Top 50+ Core on Java classes, interfaces and generics interview questions & answers

246 views

001B: ♦ Java architecture & design concepts interview questions & answers

204 views

Bio | Latest Posts

Arulkumaran Kumaraswamipillai

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via Amazon.com in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription

based site.**945+** paid members. join my LinkedIn Group. **Reviews**

About Arulkumaran Kumaraswamipillai

Mechanical Eng to freelance Java developer in 3 yrs. Contracting since 2003, and attended 150+ Java job interviews, and often got 4 - 7 job offers to choose from. It pays to prepare. So, published Java interview Q&A books via Amazon.com in 2005, and sold 35,000+ copies. Books are outdated and replaced with this subscription based site.**945+** paid members. join my LinkedIn Group. **Reviews**

‹ 12: Q78 – Q80 Recursion in Scala Q&As explained with diagrams

Posted in member-paid, Scala way of coding

Empowers you to open more doors, and fast-track

Technical Know Hows

☀ Java generics in no time ☀ Top 6 tips to transforming your thinking from OOP to FP ☀ How does a HashMap internally work? What is a hashing function?
☀ 10+ Java String class interview Q&As ☀ Java auto un/boxing benefits & caveats ☀ Top 11 slacknesses that can come back and bite you as an experienced Java developer or architect

Non-Technical Know Hows

☀ 6 Aspects that can motivate you to fast-track your career & go places ☀ Are you reinventing yourself as a Java developer? ☀ 8 tips to safeguard your Java career against offshoring ☀ My top 5 career mistakes

Prepare to succeed

☀ Turn readers of your Java CV go from “Blah blah” to “Wow”? ☀ How to prepare for Java job interviews? ☀ 16 Technical Key Areas ☀ How to choose from multiple Java job offers?

Select Category

© Disclaimer

The contents in this Java-Success are copy righted. The author has the right to correct or enhance the current content without any prior notice.

These are general advice only, and one needs to take his/her own circumstances into consideration. The author will not be held liable for any damages caused or alleged to be caused either directly or indirectly by these materials and resources. Any trademarked names or labels used in this blog remain the property of their respective trademark owners. No guarantees are made regarding the accuracy or usefulness of content, though I do make an effort to be accurate. Links to external sites do not imply endorsement of the linked-to sites.

↑