04: Q19 – Q26 Scala Data Structures basics interview questions & answers

Posted on August 10, 2016 by Arulkumaran Kumaraswamipillai

Q19. What is the use of Tuples in Scala?

A19. A **Tuple** is an immutable collection that can hold objects of different types. In other words, it groups together logical collections of items without using a class. There are 22 Tuple class in Scala from **Tuple1**, **Tuple2**, **Tuple3**, etc up to **Tuple22**.

For example, Tuple3 takes 3 parameters of different object types. In Scala everything is an Object. No primitives.

```
1
2  package com.mytutorial
3
```

```
4  object Simple extends App {
5
6    //type definition is optional
7    val mixedBag: Tuple3[Int,String, BigDecimal] =
8    println("Id=" + mixedBag._1 + ", name=" + mixe
9  }
10
```

Output:

```
1
2  Id=1234, name=John, salary=46500.0
3
```

Q20. How will you use a Tuple to store 5 different object types?
A20. Use a **Tuple5**.

```
1
2  val mixedBag  = (1234, "John", BigDecimal(46500.0
3
```

OR

```
1
2   val mixedBag: Tuple5[Int, String, BigDecimal, Bi
3
```

Q21. Can you explain the following Scala code, and what would be the output?

```
1
2  object WorkingWithCollection extends App {
3
4    def toMap(input: List[String]): Map[String, In
5      def traverse(list: List[String])(result: Map
6        case hd :: tail => traverse(tail)(result +
7        case Nil => result
8      }
9
10     traverse(input)(Map[String, Int]()) //3
11   }
12
13   val names = List("Java", "JEE", "Scala", "Java
14   val mappedByName = toMap(names)
15
16   println(mappedByName);
17
18 }
19
```


16 Technical Key Areas

80+ step by step Java Tutorials

A21 It counts the number of string values based on its occurrences. It uses **Currying** on //1 and **recursion** //3 & //2.

"hd" is the **head** element of the list and "tail" is a sub list with the head element removed. It keeps recursing until the sub list becomes empty, and the recursion ends by returning the result.

Output:

```
1
2 Map(Java -> 3, JEE -> 1, Scala -> 2)
3
```

Q22. What is "+" in the "result **+** (hd -> (result.getOrElse(hd, 0) + 1))"?
A22. It is NOT a concatenation operator. It is a function named "+". So, you can rewrite it as shown below with a "**.**".

```
1
2 result.+(hd -> (result.getOrElse(hd, 0) + 1))
3
```

As per the trait "Map" API in Scala:

```
1
2 def +(kvs: (A, B)*): Map[A, B]
3
```

[use case] Adds key/value pairs to this map, returning a new map.

Also be aware that a Map inherits from "**Iterable**", which has a "**++**" function, which combines two collections. In other words: Returns a new iterable collection containing the elements from the left hand operand followed by the elements from the right hand operand.

Q23. Isn't it inefficient to create new Map every recursive call on line **//2** with

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

```
1
2 result + (hd -> (result.getOrElse(hd, 0) + 1))
3
```

where a new map is created by adding a new map “hd -> (result.getOrElse(hd, 0) + 1)” to existing map “result”. “hd” (i.e. next head element) is the key and “(result.getOrElse(hd, 0) + 1)” is the count value.

A23. Yes and No. Yes for creating a new instance of a Map every iteration, which is is an expensive operation on the JVM, but the key/value elements within the previous Map are NOT copied. Scala’s **immutable data structures** have a property known as **persistence**, which allows you to just re-point the references when you create a new container from an old container.

Q24. How will you modify the above code so that it can count any types of objects, and not just string objects?
A24. Typing is used [] in Scala, and “T” is a generic type. E.g. String, Pet, etc.

```
1
2  object WorkingWithCollection extends App {
3
4    def toMap[T](input: List[T]) : Map[T, Int] = {
5      def traverse(list: List[T])(result: Map[T, I
6        case hd :: tail => traverse(tail)(result +
7        case Nil => result
8      }
9
10     traverse(input)(Map[T, Int]())
11   }
12
13   val names = List("Java", "JEE", "Scala", "Java
14   val mappedByName = toMap(names)
15
16   println(mappedByName);
17
18 }
19
```

Q25. Can the above code simplified using the Scala’s collection library?
A25. Yes. Scala’s collection library is well crafted. When learning a language, it pays to look at the available collections and their functionality. Look at the API for

"**Traversable**" trait, which has a "**groupBy**" function, which split the elements of a collection into several sub-collections.

```
1
2  object WorkingWithCombinatorsAndContainers exten
3
4    //a list of names container
5    val names = List("Java", "JEE", "Scala", "Java
6
7    val result = names.groupBy(x => x).mapValues {
8
9    println(result);
10 }
11
```

The "**groupBy**" creates sub lists as shown below

```
1
2 Map(JEE -> List(JEE), Scala -> List(Scala, Scala)
3
```

The Map."**mapValues**" is function that gets the size of the sub lists that were created by the "**groupBy**" function.

Q26. Can you use the function "**map**" instead of "mapValues" in the previous code?
A26. Yes.

```
1
2  object WorkingWithCombinatorsAndContainers exten
3
4    //a list of names container
5    val names = List("Java", "JEE", "Scala", "Java
6
7    val result = names.groupBy(x => x).map{x => (x
8
9    println(result);
10 }
11
```

This shows that "**map**" and "**mapValues**" work differently. mapValues, unlike map, returns a view on the original Map. It maintains the references to original Map and its function "_.size".

Popular Member Posts

♦ 11 Spring boot interview questions & answers

850 views

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

769 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**

‹ 05: Create or append a Sequence file to HDFS – Hadoop API tutorial

05: Spark SQL & CSV with DataFrame Tutorial ›

Posted in member-paid, Scala Interview Q&As

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

 ↑