

Industrial strength Java/JEE Career Companion to open more doors

search here ...

Go

Home

Java FAQs

600+ Java Q&amp;As

Career

Tutorials

Member

Why?

Can u Debug?

Java 8 ready?

Top X

Productivity Tools

Judging Experience?

[Home](#) › [member-paid](#) › 10: Q66 – Q70 Pattern matching in Scala Interview Q&As

# 10: Q66 – Q70 Pattern matching in Scala Interview Q&As

Posted on [September 21, 2016](#) by [Arulkumaran Kumaraswamipillai](#)

0

Like

Share

0

G+1

**Q66.** What is the purpose of pattern matching in Scala?

**A66.** It is for the “**flow control**”. Most imperative languages support “if-then-else”, case statements, and loops like while, for, foreach, etc. Scala does support some of these, but the most often used “**conditional expression**” is based on pattern matching. In the example below, an integer input is matched against either 0 or any integer value.

```

1
2 object Example2 extends App {
3
4     def factorial(input: Int) : Int = input match
5         case 0 => 1
6         case n => n * factorial(n-1) //recursion
7     }
8

```

**600+ Full  
Stack  
Java/JEE  
Interview  
Q&As ♥Free  
♦FAQs**

[open all](#) | [close all](#)[Ice Breaker Interview](#)[Core Java Interview C](#)[JEE Interview Q&A \(3](#)[Pressed for time? Jav](#)[SQL, XML, UML, JSC](#)[Hadoop & BigData Int](#)[Java Architecture Inte](#)[Scala Interview Q&As](#)[Scala way of coding](#)[01: Coding Scala](#)[01: ♥ Q1 – Q6 Scal](#)[02: Q6 – Q12 Scala](#)[03: Q13 – Q18 Scala](#)[04: Q19 – Q26 Scala](#)[05: Q27 – Q32 Scala](#)[06: Q33 – Q40 Scala](#)[07: Q41 – Q48 Scala](#)[08: Q49 – Q58 Scala](#)[09: Q59 – Q65 High](#)[10: Q66 – Q70 Pat](#)[11: Q71 – Q77 – S](#)

```

9   println(factorial(3)) // 3 * 2 * 1 = 6
10 }
11

```

Compared to Java's switch/case statements, Scala's pattern matching is an **expression** and not a statement, meaning that it always results in a value. Secondly, Scala's alternative expressions never "fall through" into the next case. Thirdly, if none of the patterns match, an exception named "MatchError" will be thrown. This means you always have to make sure that all

cases are covered or add a default case where there's nothing to do.

**Q67.** Can Scala's pattern matching used for the collections?

**A67.** Yes. Here is an example of pattern matching in action for the flow control.

```

1
2 object Example2 extends App {
3
4   val days = List("Sunday", "Monday", "Tuesday")
5
6   def iter(list: List[String]) : List[String]= {
7     case dayBeingProcessed :: remainingDays => {
8       println("Processing: " + dayBeingProcessed)
9       iter(remainingDays)
10    }
11    case Nil => Nil //when the list is empty
12  }
13
14  iter(days)
15 }
16
17

```

**Output:**

```

1
2 Processing: Sunday
3 Processing: Monday
4 Processing: Tuesday
5

```

"dayBeingProcessed :: remainingDays" is a notation meaning "head :: tail", which means head (i.e. first element) followed by a tail (i.e. all the remaining elements except the head). You can also think of it as "first::rest".

12: Q78 – Q80 Rec  
[Spring, Hibernate, & J](#)  
[Testing & Profiling/Sa](#)  
[Other Interview Q&A 1](#)  
[Free Java Interview](#)

## 16 Technical Key Areas

[open all](#) | [close all](#)

- [Best Practice \(6\)](#)
- [Coding \(26\)](#)
- [Concurrency \(6\)](#)
- [Design Concepts \(7\)](#)
- [Design Patterns \(11\)](#)
- [Exception Handling \(3\)](#)
- [Java Debugging \(21\)](#)
- [Judging Experience In](#)
- [Low Latency \(7\)](#)
- [Memory Management](#)
- [Performance \(13\)](#)
- [QoS \(8\)](#)
- [Scalability \(4\)](#)
- [SDLC \(6\)](#)
- [Security \(13\)](#)
- [Transaction Managen](#)

## 80+ step by step Java Tutorials

[open all](#) | [close all](#)

- [Setting up Tutorial \(6\)](#)
- [Tutorial - Diagnosis \(2\)](#)
- [Akka Tutorial \(9\)](#)
- [Core Java Tutorials \(2\)](#)
- [Hadoop & Spark Tuto](#)
- [JEE Tutorials \(19\)](#)
- [Scala Tutorials \(1\)](#)

**Q68.** The following code will randomly fail by throwing an exception “The number is too big”. How will you fix the code so that when the number is too big, it returns a value of 70.

```

1
2 import scala.util.Try
3
4 object ServiceThatCanFail extends App {
5
6   def getNextNumber:Long = {
7     val number = Math.round(Math.random() * 100)
8     if(number <= 70) {
9       number
10    } else {
11      throw new Exception("The number is too big")
12    }
13  }
14
15  //write code
16  println(getNextNumber)
17 }
18
19

```

**A68.** The “Try” with “Success” and “Failure” algebraic types with pattern matching as shown below.

```

1
2 import scala.util.Try
3 import scala.util.Failure
4 import scala.util.Success
5
6 object ServiceThatCanFail extends App {
7
8   def getNextNumber: Try[Long] = Try {
9     val number = Math.round(Math.random() * 100)
10    if(number <= 70) {
11      number
12    } else {
13      throw new Exception("The number is too big")
14    }
15  }
16
17  getNextNumber match {
18    case Failure(e) => println(70)
19    case Success(result) => println(result)
20  }
21 }
22

```

**Q69.** Can Scala pattern match on objects?

**A69.** Yes. Case classes are Scala’s way to allow pattern matching on objects without requiring much boilerplate code. Here is an example.

[Spring & Hibernate Ti](#)  
[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 object ShapeTest extends App {
3   def whatShape(shape: Shape) = shape match {
4     case Square(_) => println("Found a square")
5     case Rectangle(_, _) => println("Found a rectangle")
6     case _ => println("NOT FOUND!")
7   }
8
9   whatShape(new Square(5)) // Found a square
10  whatShape(new Rectangle(6, 5)) // Found a rectangle
11  whatShape(null) // NOT FOUND!
12 }
13
14 trait Shape
15 case class Square(width: Double) extends Shape
16 case class Rectangle(width: Double, length: Double) extends Shape
17

```

**Q70.** Can you replace the following code with pattern matching without the **if/else** and **isInstanceOf** statements?

```

1
2 object BetterMatching extends App {
3
4   def f(x: Any): String = {
5     if(x.isInstanceOf[Int]) {
6       "integer: " + x
7     } else if (x.isInstanceOf[Double]) {
8       "a double"
9     } else if (x.isInstanceOf[String]) {
10      "It is a String value of " + x
11    } else {
12      throw new Exception("Unsupported type")
13    }
14  }
15
16  println(f("Java"))
17  println(f(5))
18 }
19

```

**A70.**

```

1
2 object BetterMatching extends App {
3
4   def f(x: Any): String = x match {
5     case i:Int => "integer: " + i
6     case _:Double => "a double"
7     case s:String => "It is a String value of " + s
8   }
9
10  println(f("Java")) //It is a String value of J
11  println(f(5)) //integer: 5
12 }
13

```

## 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

294 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](https://www.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](#)

◀ 09: Q59 – Q65 Higher Order Functions in Scala Interview Q&As

11: Q71 – Q77 – Scala type parametrization & variance interview

Q&As ▶

**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 unboxing 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.