

Industrial strength Java/JEE Career Companion to open more doors

search here ...

Go

Home

Java FAQs

600+ Java Q&As

Career

Tutorials

Member

Why?

Can u Debug?

Java 8 ready?

Top X

Productivity Tools

Judging Experience?

[Home](#) › [Interview](#) › [Core Java Interview Q&A](#) › [Java 8](#) › Java 8 using the
Predicate functional interface

Java 8 using the Predicate functional interface

Posted on [November 25, 2014](#) by [Arulkumaran Kumaraswamipillai](#) — No

[Comments](#) ↓

0

G+1

The **java.util.function** package in Java 8 has a number of functional interfaces. Let's look at the ***Predicate*** interface that is used for filtering objects from a collection.

Here is the Java 8 example:

Step 1: Create a *Driver* domain class using the builder design pattern.

```
1 public class Driver {
2     enum SEX {
3         MALE, FEMALE
4     }
5
6     private String name;
```

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

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

- ✚ Ice Breaker Interview
- ✚ Core Java Interview C
- ✚ Java Overview (4)
- ✚ Data types (6)
- ✚ constructors-methc
- ✚ Reserved Key Wor
- ✚ Classes (3)
- ✚ Objects (8)
- ✚ OOP (10)
- ✚ GC (2)
- ✚ Generics (5)
- ✚ FP (8)
- ✚ IO (7)
- ✚ Multithreading (12)
- ✚ Algorithms (5)
- ✚ Annotations (2)
- ✚ Collection and Dat
- ✚ Differences Betwee
- ✚ Event Driven Progr
- ✚ Exceptions (2)
- ✚ Java 7 (2)

```

7   private int age;
8   private SEX sex;
9
10      //builder, and getters,
11
12 }
13

```

If not sure how to apply the builder design pattern to the Driver class, check under design patterns key area or search this blog for “builder design”.

Step 2: Define a static method to have methods that return **Predicates** via static methods.

```

1  package com.java8.examples;
2
3  import java.util.function.Predicate;
4
5  public class DriverPredicate {
6
7      public static Predicate<Driver> isEligibleForDr
8      return p -> p.getAge() > 18;
9  }
10
11  public static Predicate<Driver> isEligibleMaleD
12  return p -> p.getAge() > 18 && p.getSex() == D
13  }
14
15 }
16
17
18

```

Step 3: Create sub lists using the predicates. It is much easier to create sub lists in Java 8.

```

1  package com.java8.examples;
2
3  import java.util.Arrays;
4  import java.util.List;
5  import java.util.stream.Collectors;
6
7  public class DriverTest {
8
9      public static void main(String[] args) {
10         List<Driver> drivers = Arrays.asList(
11             new Driver.Builder().name("John").age(34).se
12             new Driver.Builder().name("James").age(24).s
13             new Driver.Builder().name("Samantha").age(16
14         );
15
16
17         //using the predicates

```

Java 8 (24)

- 01: ♦ 19 Java 8 I
- 02: ♦ Java 8 Stre
- 03: ♦ Functional
- 04: ♥♦ Top 6 tips
- 04: Convert Lists
- 04: Understandir
- 05: ♥ 7 Java FP
- 05: ♦ Finding the
- 06: ♥ Java 8 way
- 07: ♦ Java 8 API
- 08: ♦ Write code
- 10: ♦ ExecutorSe
- Fibonacci numb
- Java 8 String str
- Java 8 using the
- Java 8: 7 useful
- Java 8: Different
- Java 8: Does “O
- Java 8: What is c
- Learning to write
- Non-trivial Java 8
- Top 6 Java 8 fea
- Top 8 Java 8 fea
- Understanding J

JVM (6)

Reactive Programn

Swing & AWT (2)

JEE Interview Q&A (3

Pressed for time? Jav

SQL, XML, UML, JSC

Hadoop & BigData Int

Java Architecture Inte

Scala Interview Q&As

Spring, Hibernate, & I

Testing & Profiling/Sa

Other Interview Q&A 1

Free Java Interview

```

18 List<Driver> eligibleDriverList = drivers.stream()
19     .collect(Collectors.toList());
20
21 List<Driver> eligibleMaleDriverList = drivers.stream()
22     .collect(Collectors.toList());
23
24 System.out.println("eligibleDriverList-->" + eligibleDriverList);
25 System.out.println("eligibleMaleDriverList-->" + eligibleMaleDriverList);
26
27 }
28
29 }
30

```

Output:

eligibleDriverList-->[Driver [name=John, age=34, sex=MALE],
Driver [name=James, age=24, sex=MALE]]
eligibleMaleDriverList-->[Driver [name=John, age=34,
sex=MALE], Driver [name=James, age=24, sex=MALE]]

Optional: You could have written the above trivial code without the *DriverPredicate* class as shown below. But, in industrial strength applications, you will end up violating the DRY (Don't Repeat Yourself) principle by repeating the filter logic in lambda expressions in multiple places. If the rule changes, you need to modify it in multiple locations.

```

1 package com.java8.examples;
2
3 import java.util.Arrays;
4 import java.util.List;
5 import java.util.stream.Collectors;
6
7 public class DriverTest {
8
9     public static void main(String[] args) {
10         List<Driver> drivers = Arrays.asList(
11             new Driver.Builder().name("John").age(34).sex(MALE),
12             new Driver.Builder().name("James").age(24).sex(MALE),
13             new Driver.Builder().name("Samantha").age(16).sex(FEMALE)
14         );
15
16         //using the predicates
17         List<Driver> eligibleDriverList = drivers.stream()
18             .collect(Collectors.toList());
19
20         List<Driver> eligibleMaleDriverList = drivers.stream()
21             .collect(Collectors.toList());
22
23         System.out.println("eligibleDriverList-->" + eligibleDriverList);
24         System.out.println("eligibleMaleDriverList-->" + eligibleMaleDriverList);
25
26     }
27 }

```

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 \(1\)](#)
- [Low Latency \(7\)](#)
- [Memory Management \(1\)](#)
- [Performance \(13\)](#)
- [QoS \(8\)](#)
- [Scalability \(4\)](#)
- [SDLC \(6\)](#)
- [Security \(13\)](#)
- [Transaction Management \(1\)](#)

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 Tutorial \(1\)](#)
- [JEE Tutorials \(19\)](#)
- [Scala Tutorials \(1\)](#)
- [Spring & Hibernate Tutorials \(1\)](#)
- [Tools Tutorials \(19\)](#)
- [Other Tutorials \(45\)](#)

100+ Java pre-interview

```

28
29 }
30

```

Popular Posts

♦ 11 Spring boot interview questions & answers

825 views

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

765 views

18 Java scenarios based interview Questions and Answers

400 views

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

388 views

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

295 views

♦ 7 Java debugging interview questions & answers

293 views

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

285 views

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

279 views

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

239 views

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

201 views

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](#)

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](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](#)

◀ Understanding Java 8 Streams and working with collections using Lambda expressions

Top 8 Java 8 feature list ▶

Posted in Java 8, member-paid

Leave a Reply

Logged in as [geethika](#). [Log out?](#)

Comment

[Post Comment](#)

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?](#)

© 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.