♦ Java enum interview questions & answers

Posted on June 29, 2015 by Arulkumaran Kumaraswamipillai

0 Like Share Tweet 0 G+1 Share

Covers the power of **enum**and **3 design patterns** that can be applied with the help of enums.

Q1. Why is it a best practice to use enums over static final int or String constants?

A1.

1) static final allows invalid values. Any integer value can be assigned, although there are only 4 valid choices.

```
1 season = 25;
2
```

9 tips to earn more | What can u do to go places? | **945+** members. LinkedIn Group. **Reviews**

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

open all | close all

- Ice Breaker Interview
  - 01: ♦ 15 Ice breake
  - 02: ♥♦ 8 real life ex
  - 03: ♦10+ Know you
  - 04: Can you think c
  - 05: ♥ What job inte
  - 06: ► Tell me abou
  - 07: ♥ 20+ Pre inter
- Core Java Interview C
  - Java Overview (4)
    - 01: ♦ ♥ 17 Java c

With enums:

```
1 public enum Season {Spring, Summer, AUTUMN, WINTE
2
3 private Season season= Season.WINTER ; //only one
4
```

2) No easy methods to convert values to and from a string form. Enums have methods like season.toString() and valueOf("…") method to convert a string value to enum.

3) Static final is less readable and requires more maintenance due to additional look-up code. If there are additions, deletions, or re-orderings to a constant class, the other dependent classes using the constant class will not automatically be readjusted to reflect these changes.

4) Fragile for loops as iterating over all values is subject to errors which are not diagnosed at compile time if the values are rearranged, deleted, or added to. There is no way to use the enhanced for loop like you can for the enums.

```
1 //using J2SE 5.0 for each loop
2 for (Season season : Season.values( )) {
3     listWeather.add(new Weather(season));
4 }
5
```

5) Static final only has values no behavior. Enums can have behavior as shown below with the **command objects** and **template pattern**.

```
1
2  import java.util.EnumSet;
3  import java.util.HashMap;
4  import java.util.Map;
5
6  public enum Season {
7
8      WINTER("WT") {
9          @Override
10         public void execute( ) {
11             System.out.println("Winter...");
12         }
13     },
14     SPRING("SP") {
15         @Override
```


02: ♥♦ Java Con
03: ♦ 9 Core Jav
04: ♦ Top 10 mos
Data types (6)
01: Java data typ
02: ♥♦ 10 Java S
03: ♦ ♥ Java aut
04: Understandi
05: Java primitiv
Working with Da
constructors-metho
Java initializers,
Reserved Key Wor
♥♦ 6 Java Modifi
Java identifiers
Classes (3)
♦ Java abstract c
♦ Java class loac
♦ Java classes a
Objects (8)
▶ Beginner Java
♥♦ HashMap & H
♦ 5 Java Object
♦ Java enum inte
♦ Java immutabl
♦♥ Object equals
Java serializatio
Mocks, stubs, do
OOP (10)
♥ Design princip
♦ 30+ FAQ Java
♦ Why favor com
08: ♦ Write code
Explain abstracti
How to create a
Top 5 OOPs tips
Top 6 tips to go a
Understanding C
What are good r
GC (2)
♦ Java Garbage

```
16          public void execute( ) {
17              System.out.println("Spring....");
18          }
19      };
20
21      private static final Map<String, Season> loo
22              new HashMap<String, Season>( );
23
24      static {
25          for (Season s : EnumSet.allOf(Season.cla
26              lookup.put(s.getCode( ), s);
27      }
28
29      private String code;
30
31      private Season(String code) {
32          this.code = code;
33      }
34
35      public String getCode( ) {
36          return code;
37      }
38
39      //template method
40      public abstract void execute( );
41
42      //reverse lookup method
43      public static Season getByCode(int code) {
44          return lookup.get(code);
45      }
46  }
47
48
```

Unsightly switch/case or if/else statements can be minimized or avoided with the help of template method pattern shown above. The client code would use something like season.execute( ). The static getByCode(String) method provides the reverse lookup by simply getting the value from the Map. The static block that populates the Map, uses a specialized implementation of Set, java.util.EnumSet, which has better performance than java.util.HashSet. Java 5.0 also provides a more compact and specialized implementation of Map for enums with java.util.EnumMap.

Q2. Can enums be used in switch statements?

A2. Yes. The switch statement was enhanced to allow use of enums. It is important to note that the case values don't have to be qualified with the enum class name, which can be determined from the switch control value.

```
1  Weather w = new Weather(Season.WINTER);
2  switch (w.season) {
```


- 03: Java GC tun
- Generics (5)
  - ♥ Java Generics
  - ♥ Overloaded m
  - ♦ 12 Java Gener
  - ♦ 7 rules to reme
  - 3 scenarios to ge
- FP (8)
  - 01: ♦ 19 Java 8 I
  - 02: ♦ Java 8 Stre
  - 03: ♦ Functional
  - 04: ♥♦ Top 6 tips
  - 05: ♥ 7 Java FP
  - Fibonacci numbe
  - Java 8 String str
  - Java 8: What is
- IO (7)
  - ♥ Reading a text
  - ♦ 15 Java old I/C
  - 06: ♥ Java 8 way
  - Processing large
  - Processing large
  - Read a text file f
  - Reloading config
- Multithreading (12)
  - 01: ♥♦ 15 Beginr
  - 02: ♥♦ 10+ Java
  - 03: ♦ More Java
  - 04: ♦ 6 popular J
  - 05: ♦ How a thre
  - 06: ♦ 10+ Atomic
  - 07: 5 Basic multi
  - 08: ♦ ThreadLoc
  - 09: Java FutureT
  - 10: ♦ ExecutorSe
  - Java ExecutorSe
  - Producer and Co
- Algorithms (5)
  - ♦ Splitting input t
  - ♦ Tree traversal
  - ♥ ♦ Java coding

```
3     //values are not fully qualified with the enu
4     case WINTER:
5          System.out.println("It is Winter");
6          break;
7     case SPRING:
8          System.out.println("It is Spring");
9          break;
10    default:
11          break;
12 }
13
```

Q3. How will you get an integer equivalent of an enum value?
A3. Using the ordinal( ) method.

```
1 w.season.ordinal( );
```

Q4. How will you convert a String value to an enum value?
A4. Using the valueOf( ) method. You can use the methods name( ) or toString( ) to convert an enum value back to a string value.

Q5. Would you use equals( ) or == to compare enum values?
A5. Both equals(..) and == amount to the same thing, and can be used interchangeably as enums are implicitly public static final.

Q6. Do you have to override equals( ) and hashCode( ) methods for enum?
A6. No. Since you can only create one instance (i.e. singleton) of each season, you don't have to override these methods. The class Enum declares equals( ), hashCode( ), clone( ), and compareTo( ) methods as final. You can override the toString( ) method to provide better information for debugging.

Q7. Are enums immutable?
A7. Yes. As with any class, it is easy to provide methods in an enum type which can change the state of an enum constant. Hence, the term "enum constant" is misleading. What is constant is the identity of the enum element, not its state. Perhaps a better term would have been "enum" instead of "enum constant". The enum classes can have behavior,

- Searching algori
- Swapping, partiti
- Annotations (2)
  - 8 Java Annotatio
  - More Java annot
- Collection and Data
  - ♦ Find the first n
  - ♦ Java Collectior
  - ♥ Java Iterable \
  - ♥♦ HashMap & H
  - ♦ Sorting objects
  - 02: ♦ Java 8 Stre
  - 04: Understandir
  - 4 Java Collectior
  - If Java did not ha
  - Java 8: Different
  - Part-3: Java Tre
  - Sorting a Map by
  - When to use whi
- Differences Betwee
  - ♥ Java Iterable \
  - ♦ Multithreading
  - ♦ Why do Proxy,
  - Core Java Modif
  - Differences betw
  - Java Collection i
- Event Driven Progr
  - Event Driven Pro
  - Event Driven Pro
- Exceptions (2)
  - ♦ Java exceptior
  - Top 5 Core Java
- Java 7 (2)
  - Java 7 fork and j
  - Java 7: Top 8 ne
- Java 8 (24)
  - 01: ♦ 19 Java 8 I
  - 02: ♦ Java 8 Stre
  - 03: ♦ Functional
  - 04: ♥♦ Top 6 tips
  - 04: Convert Lists

hence the responsibility is on the developers to make it immutable.

Since **enum is basically a special class type, and can have methods and fields** just like any other class, you can apply the "**template method**" design pattern to create enumerations that are factories or command objects as illustrated above.

Q8. How can you use enums to create a singleton class?
A8. The singleton pattern restricts the instantiation of a class to one object. Here is a simple example.

```
1  public enum UserServiceFactoryEnum {
2
3      INSTANCE;
4
5      //private constructor
6      private UserServiceFactoryEnum() {}
7
8      public void doSomething(String name) {
9          System.out.println("Singleton using Enum
10     }
11 }
```

Now, how to use it?

```
1
```

Popular Posts

♦ 11 Spring boot interview questions & answers

857 views

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

825 views

18 Java scenarios based interview Questions and Answers

447 views

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

401 views

♦ 7 Java debugging interview questions & answers

04: Understandi
05: ♥ 7 Java FP
05: ♦ Finding the
06: ♥ Java 8 way
07: ♦ Java 8 API
08: ♦ Write code
10: ♦ ExecutorSe
Fibonacci numbe
Java 8 String str
Java 8 using the
Java 8: 7 useful
Java 8: Different
Java 8: Does "O
Java 8: What is
Learning to write
Non-trival Java 8
Top 6 Java 8 fea
Top 8 Java 8 fea
Understanding J
JVM (6)
♦ Java Garbage
01: jvisualvm to
02: jvisualvm to
05: Java primitiv
06: ♦ 10+ Atomic
5 JMX and MBea
Reactive Programm
07: Reactive Pro
10: ♦ ExecutorSe
3. Multi-Threadir
Swing & AWT (2)
5 Swing & AWT i
Q6 – Q11 Swing
JEE Interview Q&A (3
JEE Overview (2)
♦ 8 Java EE (aka
Java EE intervie
Web basics (8)
01: ♦ 12 Web ba
02: HTTP basics
03: Servlet interv

311 views

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

302 views

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

292 views

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

286 views

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

263 views

8 Git Source control system interview questions & answers

215 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.

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

- 04: JSP overview
- 05: Web patterns
- 06: ♦ MVC0, MV
- 07: When to use
- 08: Web.xml inte
- WebService (11)
  - 01: ♥♦ 40+ Java
  - 02: ♦ 6 Java RES
  - 03: ♥ JAX-RS ho
  - 04: 5 JAXB inter
  - 05: RESTFul We
  - 06: RESTful Wel
  - 07: HATEOAS R
  - 08: REST constr
  - 09: 11 SOAP We
  - 10: SOAP Web S
  - 11: ♥ JAX-WS ho
- JPA (2)
  - 10: Spring, Java
  - 8 JPA interview q
- JTA (1)
  - JTA interview Q&
- JDBC (4)
  - ♦ 12 FAQ JDBC
  - JDBC Overview
  - NamedParamete
  - Spring, JavaCon
- JMS (5)
  - ♦ 16 FAQ JMS ir
  - Configuring JMS
  - JMS versus AMQ
  - Spring JMS with
  - Spring JMS with
- JMX (3)
  - 5 JMX and MBea
  - Event Driven Pro
  - Yammer metrics
- JNDI and LDAP (1)
  - JNDI and LDAP
- Pressed for time? Jav
- Job Interview Ice B

35,000+ copies. Books are outdated and replaced with this subscription based site.

‹ javap, jps, jmap, and jvisualvm tutorial – analyzing the heap histogram

♥ Hidden Unix, Git & SVN config files ›

Posted in member-paid, Objects

01: ♦ 15 Ice brea
02: ♥♦ 8 real life
03: ♦10+ Know y
FAQ Core Java Jol
♥♦ Q1-Q10: Top
♦ Q11-Q23: Top
♦ Q24-Q36: Top
♦ Q37-Q42: Top
♦ Q43-Q54: Top
01: ♥♦ 15 Beginr
02: ♥♦ 10+ Java
FAQ JEE Job Inter
♦ 12 FAQ JDBC
♦ 16 FAQ JMS ir
♦ 8 Java EE (aka
♦ Q01-Q28: Top
♦ Q29-Q53: Top
01: ♦ 12 Web ba
06: ♦ MVC0, MV
JavaScript mista
JavaScript Vs Ja
JNDI and LDAP
JSF interview Q&
JSON interview
FAQ Java Web Ser
01: ♥♦ 40+ Java
02: ♦ 6 Java RES
05: RESTFul We
06: RESTful Wel
09: 11 SOAP We
Java Application Ar
001A: ♦ 7+ Java
001B: ♦ Java arc
04: ♦ How to go
Hibernate Job Inter
01: ♥♦ 15+ Hiber
01b: ♦ 15+ Hiber
06: Hibernate Fir
8 JPA interview
Spring Job Intervie
♦ 11 Spring boot

- 01: ♥♦ 13 Spring
- 01b: ♦ 13 Spring
- 04 ♦ 17 Spring b
- 05: ♦ 9 Spring B
- Java Key Area Ess
  - ♦ Design pattern
  - ♥ Top 10 causes
  - ♥♦ 01: 30+ Writir
  - ♦ 12 Java desigr
  - ♦ 18 Agile Devel
  - ♦ 5 Ways to debu
  - ♦ 9 Java Transac
  - ♦ Monitoring/Pro
  - 02: ♥♦ 13 Tips to
  - 15 Security key a
  - 4 FAQ Performa
  - 4 JEE Design Pa
  - 5 Java Concurre
  - 6 Scaling your Ja
  - 8 Java memory r
- OOP & FP Essentia
  - ♦ 30+ FAQ Java
  - 01: ♦ 19 Java 8 I
  - 04: ♥♦ Top 6 tips
- Code Quality Job In
  - ♦ Ensuring code
  - ♦ 5 Java unit tes
- SQL, XML, UML, JSC
  - ERD (1)
    - ♦ 10 ERD (Entity
  - NoSQL (2)
    - ♦ 9 Java Transac
    - 3. Understandinç
  - Regex (2)
    - ♥♦ Regular Expr
    - Regular Express
  - SQL (7)
    - ♦ 15 Database d
    - ♦ 14+ SQL interv
    - ♦ 9 SQL scenaric
    - Auditing databas

- Deleting records
- SQL Subquery ir
- Transaction man
- UML (1)
  - ♦ 12 UML intervi
- JSON (2)
  - JSON interview
  - JSON, Jackson,
- XML (2)
  - XML basics inter
  - XML Processing
- XSD (2)
  - 11 FAQ XSD inte
  - XSD reuse interv
- YAML (2)
  - YAML with Java
  - YAML with Sprin
- Hadoop & BigData Int
  - ♥ 01: Q1 – Q6 Had
  - 02: Q7 – Q15 Hadc
  - 03: Q16 – Q25 Hac
  - 04: Q27 – Q36 Apa
  - 05: Q37 – Q50 Apa
  - 05: Q37-Q41 – Dat
  - 06: Q51 – Q61 HBa
  - 07: Q62 – Q70 HDI
- Java Architecture Inte
  - ♥♦ 01: 30+ Writing
  - 001A: ♦ 7+ Java int
  - 001B: ♦ Java archit
  - 01: ♥♦ 40+ Java W
  - 02: ♥♦ 13 Tips to w
  - 03: ♦ What should I
  - 04: ♦ How to go ab
  - 05: ETL architectur
  - 1. Asynchronous pr
  - 2. Asynchronous pr
- Scala Interview Q&As
  - 01: ♥ Q1 – Q6 Scal
  - 02: Q6 – Q12 Scala
  - 03: Q13 – Q18 Sca

- 04: Q19 – Q26 Sca
- 05: Q27 – Q32 Sca
- 06: Q33 – Q40 Sca
- 07: Q41 – Q48 Sca
- 08: Q49 – Q58 Sca
- 09: Q59 – Q65 Hig
- 10: Q66 – Q70 Pat
- 11: Q71 – Q77 – S
- 12: Q78 – Q80 Rec
- Spring, Hibernate, & M
  - Spring (18)
    - Spring boot (4)
      - ♦ 11 Spring bo
      - 01: Simple Sp
      - 02: Simple Sp
      - 03: Spring boo
    - Spring IO (1)
      - Spring IO tuto
    - Spring JavaConf
      - 10: Spring, Ja
      - Spring, JavaC
      - Spring, JavaC
      - Spring, JavaC
    - 01: ♥♦ 13 Spring
    - 01b: ♦ 13 Spring
    - 02: ► Spring DII
    - 03: ♥♦ Spring DI
    - 04 ♦ 17 Spring b
    - 05: ♦ 9 Spring Bo
    - 06: ♥ Debugging
    - 07: Debugging S
    - Spring loading p
  - Hibernate (13)
    - 01: ♥♦ 15+ Hiber
    - 01b: ♦ 15+ Hiber
    - 02: Understandi
    - 03: Identifying ar
    - 04: Identifying ar
    - 05: Debugging H
    - 06: Hibernate Fi
    - 07: Hibernate mi

- 08: Hibernate au
- 09: Hibernate en
- 10: Spring, Java
- 11: Hibernate de
- 12: Hibernate cu
- AngularJS (2)
  - ♥ 8 AngularJS in
  - More Angular JS
- Git & SVN (6)
  - ♥ Git & Maven fo
  - ♥ Merging Vs rel
  - ♥ Understanding
  - 6 more Git interv
  - 8 Git Source cor
  - Setting up Cygw
- JMeter (2)
  - ♥ JMeter for test
  - ♦ JMeter perform
- JSF (2)
  - JSF interview Q&
  - More JSF intervi
- Maven (3)
  - ♥ Git & Maven fo
  - 12 Maven intervi
  - 7 More Maven ir
- Testing & Profiling/Sa
  - Automation Testing
    - ♥ Selenium and
  - Code Coverage (2)
    - Jacoco for unit te
    - Maven and Cobe
  - Code Quality (2)
    - ♥ 30+ Java Code
    - ♦ Ensuring code
  - jvisualvm profiling (
    - 01: jvisualvm to
    - 02: jvisualvm to
    - 03: jvisualvm to
  - Performance Testir
    - ♥ JMeter for test
    - ♦ JMeter perform

- Unit Testing Q&A (2
  - BDD Testing (4)
    - Java BDD (Be
    - jBehave and E
    - jBehave and j
    - jBehave with t
  - Data Access Uni
    - ♥ Unit Testing
    - Part #3: JPA H
    - Unit Test Hibe
    - Unit Test Hibe
  - JUnit Mockito Sp
    - JUnit Mockito
      - Spring Con
      - Unit Testing
    - Part 1: Unit te
    - Part 2: Mockit
    - Part 3: Mockit
    - Part 4: Mockit
    - Part 5: Mockit
  - Testing Spring T
    - Integration Un
    - Unit testing Sp
  - ♦ 5 Java unit tes
  - JUnit with Hamc
  - Spring Boot in ur
- Other Interview Q&A f
  - Finance Domain In
    - 12+ FX or Forex
    - 15 Banking & fin
  - FIX Interview Q&A
    - 20+ FIX basics in
    - Finding your way
  - Groovy Interview Q
    - Groovy Coding C
      - Cash balance
      - Sum grades G
    - ♥ Q1 – Q5 Groov
    - ♦ 20 Groovy clos
    - ♦ 9 Groovy meta
    - Groovy method c

Q6 – Q10 Groov
JavaScript Intervie
JavaScript Top I
♥ Q1 – Q10 J
♦ Q11 – Q20
♦ Q21 – Q30
♦ Q31 – Q37
JavaScript mi
JavaScript Vs J
JavaScript Vs
Unix Interview Q&A
♥ 14 Unix intervi
♥ Hidden Unix, C
sed and awk to
Shell script inter
Unix history com
Unix remoting in
Unix Sed comma
▶Free Java Intervie
▶ Java Integration
▶ Java Beginner I
02: ▶ Spring DIP, I
06: ▶ Tell me abou

As a Java Architect

Java architecture & design concepts interview Q&As with diagrams | What should be a typical Java EE architecture?

Senior Java developers must have a good handle on

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 I
- Low Latency (7)
- Memory Managemen
- 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)
- Spring & HIbernate Tu
- Tools Tutorials (19)
- Other Tutorials (45)

Preparing for Java written & coding tests

open all | close all
- ♦ Complete the given
- Can you write code? (
- 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…to go places?

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

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.

© 2016 Java-Success.com

↑

Responsive Theme powered by WordPress