Javarevisited

Blog about Java, Programming, Spring, Hibernate, Interview Questions, Books and Online Course Recommendations from Udemy, Pluralsight, Coursera, etc

Home core java spring hibernate collections multithreading design patterns interview questions coding data structure OOP java 8 books About Me

Java Certifications JDBC jsp-servlet JSON SQL Linux Courses online resources jvm-internals REST Eclipse jQuery Java IO Java XML

TUESDAY, NOVEMBER 12, 2019

10 JDK 10 Features Java Developers Should Know

Hello Guys, while we all are talking about Java 9 and some people are yet to adopt Java 8 in their projects, here comes the new release of Java, the JDK 10. It's available to download now on Oracle's website. This is the first release in the latest release schedule, where you have a new Java release every six months, I know it's quite early, but that's a reality now. Apart from 6 monthly release, Every 3 years there is an LTS release, which is sort of major Java release. The next LTS release is Java 11 which should be out in September. So you can aim for that. Then you can update three years after that to the next LTS release.

10 Java 10 Features Developer Should Know

Even though JDK 10 looks come really quick, there are some interesting features coming in JDK 10. I have yet to go into the details of that but on prima-facia they look useful and interesting to me.

Here is a quick note on those features:

1. Local Variable Type Inference (JEP 286)

Similar to Javascript, Kotlin, and Scala, now Java will also have a var keyword that allows you to declare a local variable without specifying its type. The type will be inferred from context for example when you say var name = "Java" then compile will already know the type is String.

I don't know how useful this will be as I am quite used to of seeing `int i =0` or `String name = "Java"` and I liked the type information present in the variable declaration line but looks like Java is going the way Scala and Kotlin are and trying to incorporate changes from there.

Also, note that the var keyword can only be used for local variables i.e. variables inside methods or code blocks, you cannot use it for member variable declaration inside the class body.

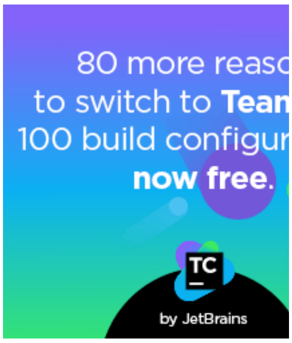
And, finally, it doesn't make Java a dynamically typed language like Python, Java is still a statically typed language and once the type is assigned you cannot change it. For example, `var name = "Java"` is ok but then `name = 3;` is not ok.

As Sander Mak puts in his Pluarlsight course **What's New in Java 10** this is one of the most eye-catching features of Java 10 which reduces the amount of boilerplate code needed to declare local variables in Java.

It's not so obvious from simple example but considers where a method returns a complex list type which requires a lot of angle brackets and generics to declare the type, this really saves time in those case:

```
var list = List.of(1, 2.0, "3")
```

Here list will be inferred as `List<? extends Serializable & Comparable<..>>` which is an intersection type. You can also learn more about Java 10 features on **The Complete Java MasterClass** course on Udemy, which is the most comprehensive resource and also updated for Java 10.



Follow by Email

Email address... Submit

Interview Questions

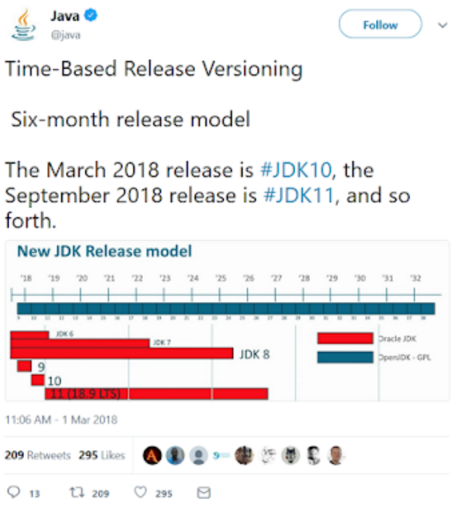
core java interview question (169)
Coding Interview Question (72)
data structure and algorithm (70)
interview questions (48)
object oriented programming (31)
SQL Interview Questions (30)
design patterns (30)
thread interview questions (30)

2. Time-Based Release Versioning (JEP 322)

From JDK 10 release, Java has adopted a new schedule of a new release every six months. There is a lot of debate whether this is a practical approach or not as many are saying that its good that you will get new features every six month and many are complaining that its too little a time to adopt a JDK.

Anyway, this will be the way forward with an LTS release every three years, consider that a major release.

Also, The Update element will increment one month after the Feature element is incremented so the April 2018 Java release will be JDK 10.0.1, July 2018 release will be named JDK 10.0.2 and so on.



Java @java

Follow

Time-Based Release Versioning

Six-month release model

The March 2018 release is #JDK10, the September 2018 release is #JDK11, and so forth.

New JDK Release model

11:06 AM - 1 Mar 2018

209 Retweets 295 Likes

13 209 295

3. Garbage-Collector Interface (JEP 304)

This is one more interesting and useful Java 10 feature which increases the code isolation of different garbage collectors and introduces a clean interface for garbage collectors.

This means it easier to exclude a GC from a JDK build and it also easier to add a new GC without it affecting the codebase. You can further see Java Memory Management learn more about G1 Garbage Collection and the difference between G1 and Concurrent Mark Sweep Garbage collector.

4. Parallel Full GC for G1 (JEP 307)

Another interesting feature which improves G1 worst-case latencies by making the full GC parallel.

collections interview questions (25)
spring interview questions (19)
database interview questions (16)
servlet interview questions (15)
Programming interview question (6)
hibernate interview questions (6)

Best of Javarevisited

How Spring MVC works internally?
How to design a vending machine in Java?
How HashMap works in Java?
Why String is Immutable in Java?
10 Articles Every Programmer Must Read
How to convert lambda expression to method reference in Java 8?
10 Tips to improve Programming Skill
10 OOP design principles programmer should know
How Synchronization works in Java?
10 tips to work fast in Linux
5 Books to improve Coding Skills

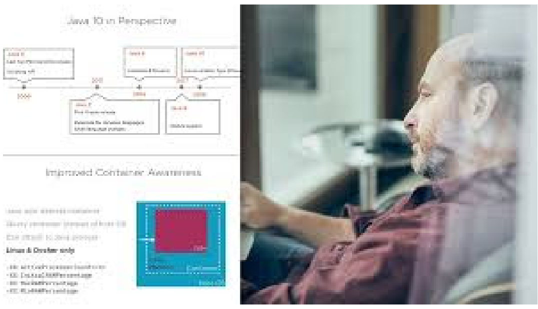
Java Tutorials

date and time tutorial (21)
FIX protocol tutorial (15)
Java Certification OCPJP SCJP (26)
java collection tutorial (73)
java IO tutorial (28)
Java JSON tutorial (12)
Java multithreading Tutorials (56)
Java Programming Tutorials (18)
Java xml tutorial (16)
JDBC (29)
jsp-servlet (37)
online resources (155)

If you remember, In Java 9 release, G1 was made the default GC for JVM, which was designed to avoid full GC, but when the concurrent collections couldn't reclaim memory quick enough it would end up falling back on a full GC, and that creates a problem.

This change will parallelize the full GC algorithm so that in the unlikely event of a G1 Full GC, same number of threads can be used as in the concurrent collections to improve the overall performance.

If you want to learn more about Parallel Full GC for G1 then you should watch **What's New in Java 10** by Sander Mak to learn about the performance improvement done on Java 10 release. He has nicely put together some important JDK feature in a short course on Pluarlsight.



5. Heap Allocation on Alternative Memory Devices (JEP 316)

This sounds a really cool feature that enables the HotSpot VM to allocate the Java object heap on an alternative memory device, specified by the user.

For example, this feature makes it possible to assign lower priority processes to use the NV-DIMM memory, and instead only allocate the higher priority processes to the DRAM in a multi-JVM environment.

Btw, If you don't know much about JVM, I suggest you to first start with **Understanding the Java Virtual Machine: Memory Management** By Kevin Jones, a good introductory course on JVM.

6. Consolidate the JDK Forest into a Single Repository (JEP 296)

This new Java 10 feature is all about housekeeping. It will combine the numerous repositories of the JDK forest into a single repository.

7. Root Certificates (JEP 319)

This is another important change Java 10 is bringing. If you remember, JDK 10 was created with the close collaboration with OpenJDK and this is evident from this feature.

It will provide a default set of root Certification Authority making OpenJDK builds more appealing to developers.

It also aims to reduce the difference between the OpenJDK and Oracle JDK builds. Critical security components such as TLS will now work by default in OpenJDK builds

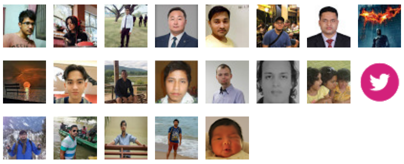
8. Experimental Java-Based JIT Compiler (JEP 317)

This is another interesting feature which enables the Java-based JIT compiler, Graal, to be used as an experimental JIT compiler on the Linux/x64 platform.

If you remember, Graal was already added back in Java 9, but now you can enable it with the

Followers

Followers (4811) Next



Follow

Subscribe to Download the E-book

Download The E-book

Building a REST API with Spring 4?

Email address...

Submit

Categories

- courses (93)
- SQL (55)
- linux (38)
- database (36)
- Eclipse (28)
- Java Certification OCPJP SCJP (26)
- JVM Internals (23)
- JQuery (18)
- REST (17)
- Maven (12)
- Testing (11)
- general (11)

Blog Archive

▼ 2019 (259)
▼ November (38)
Top 10 Gift Ideas for Programmers, Software Engine...
Top 10 Computer Programming, Artificial Intelligen...
Top 10 UNIX and Linux System Programming Books- Be...
Top 5 Courses to learn C# (C-Sharp) in 2020 - Best...
How to send HTTP request using curl and wget comma...
Top 5 Books to Learn Scala and Functional Programm...
Top 5 Courses to Learn The C Programming in 2020 -...

following JVM arguments:

```
-XX:+UnlockExperimentalVMOptions -XX:+UseJVMCICompiler
```

If you don't know Grall is a new Java-based JIT compiler which is the basis of an experimental Ahead-of-Time (AOT) compiler.

However, keep in mind that it is in an experimental stage and you should not use it for production.

9. Thread-Local Handshakes (JEP 312)

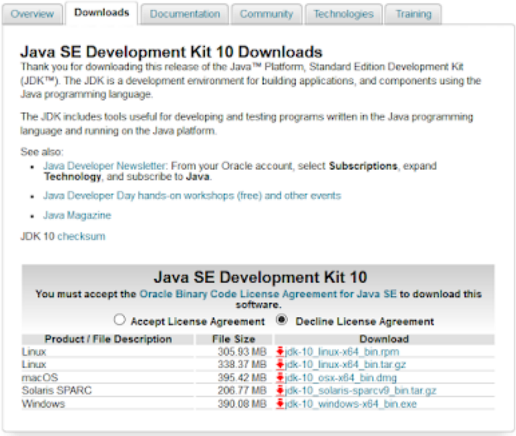
This Java 10 feature lays the groundwork for improved VM performance, by making it possible to execute a callback on application threads without performing a global VM savepoint. This would mean that the JVM could stop individual threads and not just all of them.

There are several small improvements done as part of this feature or JEP 312 to improve VM performance e.g. some memory barriers have been removed the JVM and biased locking is improved by only stopping individual threads for revoking biases.

10. Remove the Native-Header Generation Tool (JEP 313)

This is another Java 10 feature that focuses on housekeeping. It will remove the `javah` tool from the JDK, a separate tool to generate header files when compiling JNI code, as this can be done through javac.

You can download JDK 10 from Oracle's website here to play with the new features:



Java SE Development Kit 10 Downloads

Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications, and components using the Java programming language.

The JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform.

See also:

- Java Developer Newsletter: From your Oracle account, select **Subscriptions**, expand **Technology**, and subscribe to **Java**.
- Java Developer Day hands-on workshops (free) and other events
- Java Magazine

JDK 10 checksum

Java SE Development Kit 10

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

○ Accept License Agreement ● Decline License Agreement

Product / File Description	File Size	Download
Linux	305.93 MB	jdk-10_linux-x64_bin.rpm
Linux	338.37 MB	jdk-10_linux-x64_bin.tar.gz
macOS	395.42 MB	jdk-10_osx-x64_bin.dmg
Solaris SPARC	206.77 MB	jdk-10_solaris-sparcv9_bin.tar.gz
Windows	390.08 MB	jdk-10_windows-x64_bin.exe

That's all about some **interesting features of Java 10 or JDK 10**. There are a lot more low level and API changes which you can find on Oracle's official release notes. I'll also blog about those changes as and when I come to know about, so you can stay tuned for more JDK 10 articles and tutorials here in Javarevisited.

Top 5 Courses to Learn IntelliJIDEA and Android St...
10 Example of find command in UNIX and Linux
Top 5 Hibernate Books for Java Developers - Best, ...
Top 5 Free Courses to Learn Scala and Functiona P...
10 Example of lsof commands in Linux and UNIX
10 reasons to Learn Scala Programming Language
Top 5 TensorFlow and Machine Learning Online Train...
10 JDK 10 Features Java Developers Should Know
Top 5 Android Online Training Courses for Java Dev...
Top 10 Most Popular Programming Languages of the L...
How to Enroll on Spring Professional Certification...
Top 5 courses to learn Apache Maven for Java Devel...
How Spring MVC Framework works? How HTTP Request i...
find + du + sort + head command example in Linux -...
Top 5 Best Java 8 Tutorials, Courses, and Books
10 Tips Improve Your Speed in Bash, and Linux Comm...
Top 5 Java 8 and 9 Courses for Programmers - Best ...
Do you Need to Pass OCAJP before taking OCPJP - Co...
Kotlin vs Java - Which language Android Programmer...
Top 5 Books for Programming/Coding Interviews - Be...
6 ways to declare and initialize Two-dimensional (...
Top 5 Course to Learn Ansible for DevOps in 2019 -...
Top 5 Courses to Learn TypeScript for Web Develope...
How do you find length of a Singly Linked list usi...
Top 5 Websites to Learn Python Coding for FREE - B...
Top 5 courses to learn Tableau in 2019 - Best of L...
Difference between @RestController and @Controller...
Java vs Python - Which Programming Language Should...
5 Websites to Learn SQL Online for FREE
10 Books Every Java Programmer Should Read - Best ...
Top 5 Courses to learn JUnit, Unit Testing and Tes...

► October (62)
► September (44)
► August (34)
► July (26)
► June (14)
► May (18)
► April (11)
► March (5)
► February (1)
► January (6)

► 2018 (40)
► 2017 (774)
► 2016 (109)

References

Other **Useful Resources** for Java Programmers

- The Complete Java MasterClass
- What's New in Java 10 by Sander Mak
- Top 10 Java 8 Tutorials for Programmers
- Top 10 Java 9 Tutorials for Programmers
- 10 Things Java Developer Should learn in 2019
- 10 Frameworks Java and Web Developer Should learn in 2019
- 20 Libraries Java developer should know
- Top 5 Courses to Learn Python in 2019
- Top 5 Spring Boot Features Java developer should learn
- Top 5 Courses to learn Spring Framework in depth

Thanks for reading this article so far. If you like these Java 10 features then please share with your friends and colleagues. If you have any questions or feedback then please drop a note.

All the best with JDK 10

P. S. - If you want to learn Java from scratch and looking for some recommendations, you can also check my post about the top 5 Java courses for beginners. It contains some of the best Java courses from Udemy, Pluralsight, and Coursera to kickstart your Java journey.

By javin paul at November 12, 2019

Labels: best of javarevisited , core java , Java 10

5 comments :

Anonymous said...

Interesting features but I agree 6 month release cycle is very aggressive for anyone to adopt.

March 21, 2018 at 5:56 AM

Javin said...

Another useful feature to add into this list Application Class Data Sharing as rightly pointed out by Nicolai Parlog on twitter. This was a feature which was earlier available to commercial JVM but no its available to open JDK and can significantly improve application launch time. He has also blogged about it on his blog and you can read it for more details https://blog.codefx.org/java/application-class-data-sharing/

March 22, 2018 at 6:46 AM

Unknown said...

Hi Javin...
Earlier there was an option to see the index of all your posts year and month wise.(tree structure) at the right side of the blogs. Now that option is not showing. Is there any way/index to browse across all your blogs?

March 22, 2018 at 11:40 PM

Javin said...

Hello Unknown,

I have removed that option for now but will add it back as some other readers has also requested that.

Thanks
Javin

March 24, 2018 at 8:36 AM

1. Oracle's Java Tech Network
2. jQuery Documentation
3. Microsoft SQL Server Documentation
4. Java SE 8 API Documentation
5. Spring Documentation
6. Oracle's JAva Certification
7. Spring Security 5 Documentation

Pages

Privacy Policy

Javin said...

Hello guys,

Here is another useful link to effectively use the var or local variable type inference in Java 10 code:

Style Guidelines for Local Variable Type Inference in Java
http://openjdk.java.net/projects/amber/LVTIstyle.html

A good read to understand how using var can reduce noise and improve code quality.

March 24, 2018 at 8:37 AM

Post a Comment

Enter your comment...

Comment as: hakobhn@gma ▾ **Sign out**

Publish **Preview** □ Notify me

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)

Search This Blog

Search