9 things Java programmers should know in 2018 if they want a successful career

You want to start the new year in a right direction? Join TipsMake.com to consult 9 things Java programmers should know in 2018 if you want a successful career!

1. Statistics of programming languages ??that are "hated" most
2. Why Web Designer should learn Ruby on Rails?
3. Instructions on how to compile and execute Java using Command Prompt

You want to start the new year in a right direction? This is a time when each of us needs to create a plan, set goals and make a commitment to a successful year.

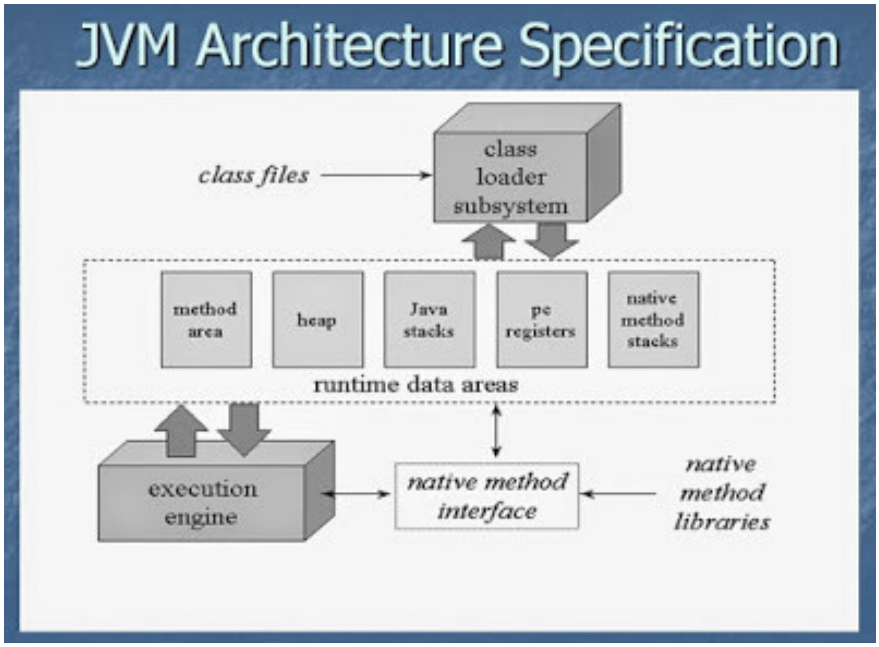
As a Java Developer and the author of a Java blog, the article author regularly gets questions from Java programmers around the world on how to improve performance. .

Last year, I shared 10 tips to become a better Java programmer and the article was right but when I started a new year, I decided to write new goals for Java programmers to improve yourself, expand your knowledge of Java programming and add value to future career development.

So, not to waste time, please join TipsMake.com to review **9 good ideas for Java programmers in 2018** . You can get inspired by these ideas to create your own goals and determination.

1. Learn about adjusting Java performance

In the last few years, I have conducted more than 50 interviews with experienced Java developers and a skill I find lacking in them is knowledge and understanding of internal JVM, the behavior of GC and Java performance tuning (Java performance tuning).



As the Java experience rises and becomes a senior Java developer with more than 5-6 years of experience, you are required to know the whole big picture and small details about Java fundamentals.

If you can't describe an application or can't understand why it's slow or how to make it faster then you should read a good book about internal JVM and adjust performance, for example Scott Oaks ' **Java Performance the Definitive Guide** ' best-selling book.

I have read this book many years ago and still refer back whenever I have time. My goal is to read it again this year.

If you like online courses more than books, I recommend looking through the course sequence Understanding Java Virtual Machine (JVM) - an abstract device of Pluarlsight that explains memory management, class boarding, Security and Reflect.

2. Every day 2 hours code

Another thing I noticed last year was that as the experience increased, you often spent a lot of time doing email replies, catalysts, troubleshooting, mentoring, and as a project manager. good.

What you fall behind is code, which is the most important skill of a programmer.

If you feel your code is not enough, be determined to code every day. At the very least, write something, whether it's your project, an open source framework, a library or a utility.

Getting started is always the hardest thing, I suggest that whenever you feel hesitant, read and refactor your code first. Then you will like and write code.

In addition, you can solve problems based on an interview with Cracking the Coding and 150 programming questions. This not only ensures that you are ready for the interview but also enhances your knowledge of data structures, algorithms and programming logic.

3. Prepare a Java application once a month

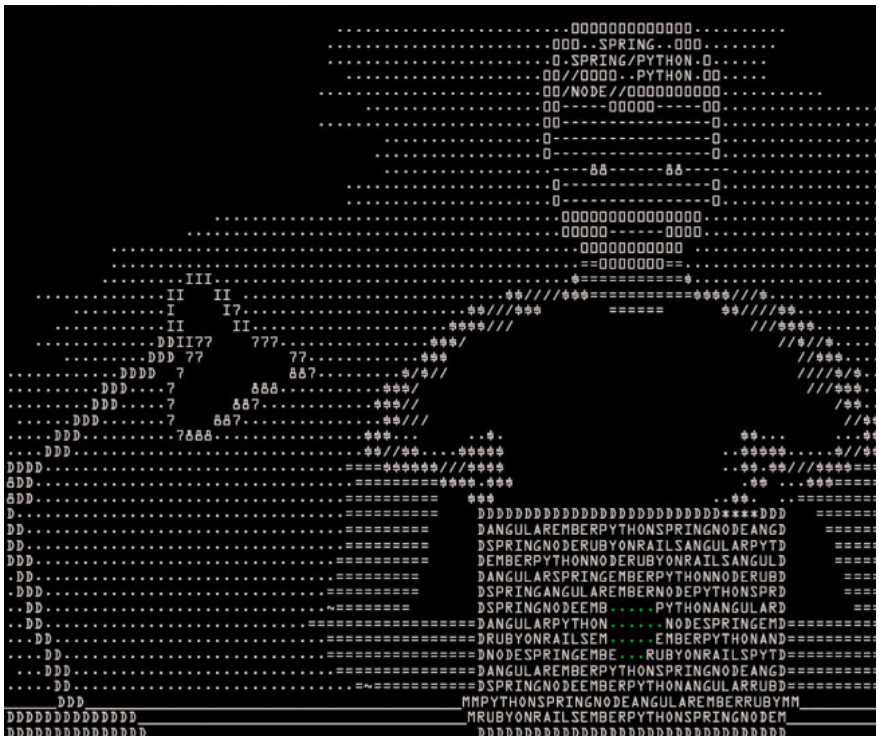
This comes with the first determination in reading good books about internal JVM and performance tuning. If reading is not enough, you must apply that knowledge to your real project.

I recommend that you profile your Java application, at least once a month, and spend a lot of time learning and analyzing the results.

You can also export your Java process or if you have a problem in a recent program, dump it and find out which object takes up most of your memory. Is there a memory leak in your Java application?

What are the causes? What will happen if another 100K new customers access your application? If you can answer all these questions comfortably, you have good preparation. If you need some guidance to solve memory and CPU problems, I recommend that you review Richard Warburton's **Understanding and Solving Java Memory Problems** .

4. Join the code challenge



Again, this goal is partly related to the second determination above - **writing code for 2 hours a day** . Let's face it because sometimes you don't have enough code opportunities in your actual project.

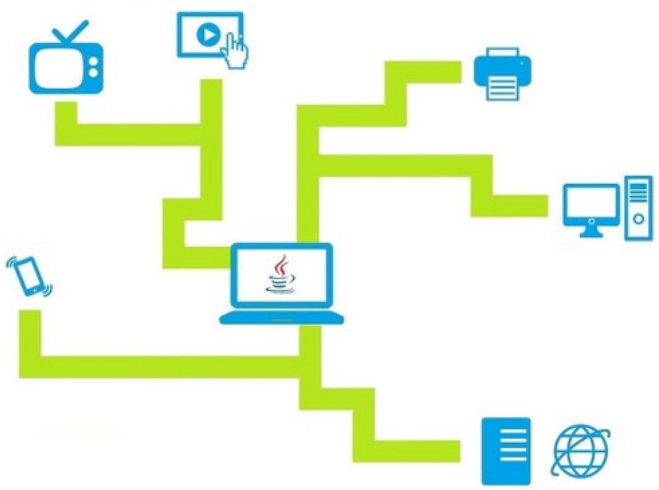
If you are looking for code exercises, there is nothing better than participating in programming and code challenges. Or refer to our article:

1. C ++ exercises have solutions (sample code) for variables, data types, and operators
2. C ++ exercises have solutions (sample code)
3. Basic Java exercises, with sample decoding

There are many websites on the Internet that contain programming challenges and give you the opportunity to test your own skills, but Top 20 websites that learn free programming that need to bookmark immediately can be best.

If you are looking for some tougher programming challenges, you can also check out the list of good websites to practice code.

5. Learn about Network programming in Java



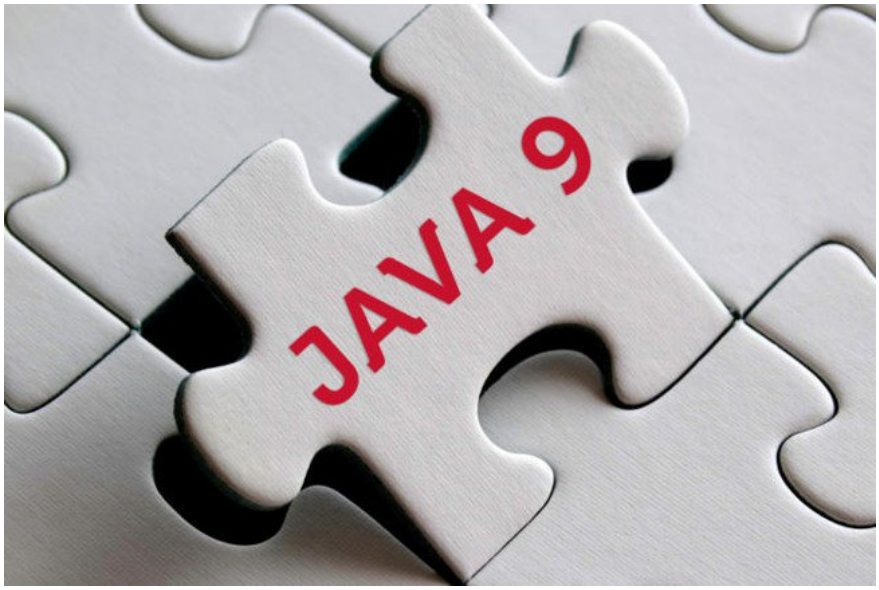
Another thing I learned from the interview is that Java developers need to improve socket network programming and the basic principles of their networks. I have asked many questions from my Java programming programming question list, but developers are almost unable to answer.

Some of them even tried to outline the main difference between TCP and UDP, which I think is too basic to ask any Java developer with 2-5 years experience.

If you feel you don't have the opportunity to learn socket programming in Java or you don't need them because you're a Java web developer working with JSP, Servlet and JSF, then I recommend reading at least one good book on the Internet. Java, NIO and socket network programming.

The book I recommend reading is the TCP / IP Sockets in Java, Second Edition: Practical Guide for Programmers (The Practical Guides). Very easy to read, interesting and a good way to learn the basic principles of socket network programming.

6. Java 9



In 2017 we saw big releases, one of which was JDK 9. I haven't started with JDK 9 yet, but this is the first thing I will look into in 2018. If you want to learn the new features of Java 9 like Jigsaw, Reactive Streams, API enhancements, . then **Java 9 MasterClass** on Udemy is a good course to get started.

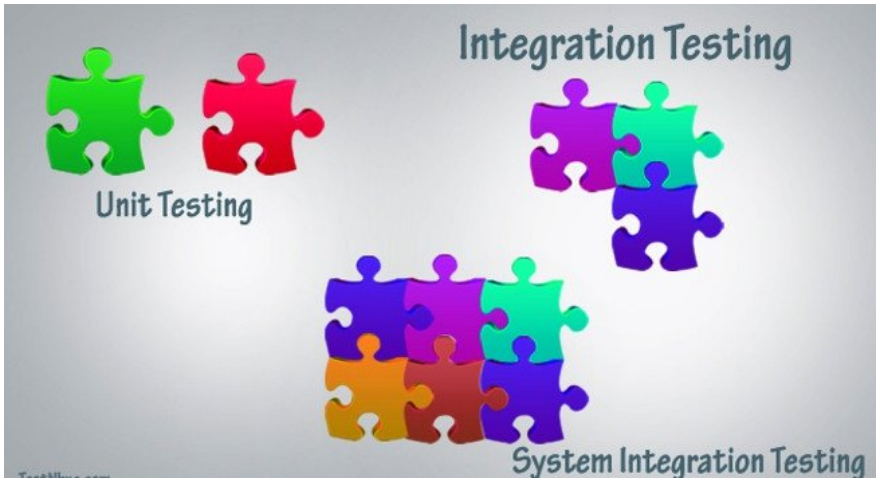
7. Spring 5.0

Another major release for the Java world in 2017 is Spring Framework 5.0. With many interesting features like the reactive programming model and reception of Java 8 and Java 9, Spring 5.0 adoption will accelerate in 2018, which is why every Java developer should learn it. If you need help, **Spring 5.0: Beginner to Guru** is a good course to get started.

8. Spring Security 5.0

Another interesting release of 2017 that I will study this year is **Spring Security 5.0** . Spring Security 5.0 is a major version, they rewrite some modules and fix several hundred errors. The most interesting part is the OAuth 2.0 module. Unfortunately, currently there are not many resources to learn Spring Security 5.0, but Eugen has updated the **Learn Spring Security** software including version 5.0 and added a new module on OAuth 2.0.

9. Unit testing



If you want to be a better developer in 2018, you should work on unit testing skills. Not only with unit testing, but also automated testing in general. This also includes integrated testing. You can learn JUnit 5 and other advanced unit testing libraries like Mockito, PowerMock, Cucumber and Robot to bring your unit testing skills to the next level.

Mockito is really powerful and allows you to write a unit testing for complex classes by mocking dependencies and focusing only on objects that have been tested. If you are a beginner of unit testing and want to learn it in 2018, then **JUnit and Mockito Crash** courses from Udemy are a good starting point.

Summary

Above is my advice! If you are a Java programmer with a few years of experience, you can also get inspired by this list to set personal goals. You can add Android, Docker and Spark learning lists, as this is essential for any

Java programmer.

I tried to make it simple and easy to do because I believe that small success will lead to great success. Set small goals and achieve them better than setting big, impractical and unsuccessful goals before starting.

What are you looking forward to? Write down what you want to do this year and share them with us. At the end of the year, you can come back here and tell us how many goals you have achieved.

Author: Javin Paul

Refer to some more articles:

1. Top 5 popular CSS Framework that you should keep in mind
2. Arrays and objects in JavaScript are like stories and newspapers!
3. If you want to be a data scientist, learn these 3 languages ??right away!

Having fun!