

# 02从零开始学Java02之Java到底是个啥?

## 博主简介

### 一. Java简介

1. 起源
2. 概念(掌握)
3. 技术体系
4. Java特性

### 二. Java发展史

### 三. Java版本(重点)

1. JDK版本
2. 版本选择

### 四. 配套视频

### 五. 今日作业

1. 第一题:
2. 第二题:
3. 第三题:
4. 第四题:

## 博主简介

- 我是**壹壹哥(孙玉昌)**，十年软件开发授课经验，CSDN博客专家、阿里云专家博主、infoQ专家博主；
- 关注**壹壹哥(孙玉昌)**，带你玩转Java，轻松实现从入门到放弃，哦不，到熟悉😁
- 曾任国内物流行业某独角兽企业架构师；
- 曾主导研发油田、广电、教育、物流等行业项目，拥有丰富的行业方案解决经验；
- 现从事软件开发教育工作，培养了大量入职阿里、华为、百度、IBM等名企的高薪学生；
- 坚持“一心做教育，做一流教育”的理念，被学生爱称为“一一哥”、“壹壹哥”。

全文大约 **【5500】** 字，不说废话，只讲可以让你学到技术、明白原理的纯干货！文章带有丰富案例及配图、视频，只为你更好的理解和运用文中的技术概念，给你带来具有足够启迪的教程.....

既然各位小伙伴现在已经决定要学习Java了，那首先得对Java有基本的了解吧，所以接下来壹哥先用一篇文章，来给大家介绍一下Java到底是个啥。

#### 友情提示：

大家要记住，无论我们学习任何技术，都要知道**这个技术是什么、干嘛的、怎么用、为什么这么用，以及使用时出现了问题该怎么解决！**

## 一. Java简介

### 1. 起源

既然各位打算以后要靠Java来养家糊口了，那对Java的了解是不是应该更多一些呢？比如Java这个技术是怎么产生的？接下来壹哥给大家讲个小故事，让咱们一起来看看Java的起源吧。

20世纪90年代，经济高速发展，大家有钱了就想追求更高的生活品质。当时发明了很多稀奇古怪的电子产品，比如有钱人必备的“大哥大”、BB机、电视机顶盒、烤箱等。我们知道，硬件产品必须要有软件系统进行支撑才能运行，就好比人必须有灵魂才能称之为人类。所以当时为了配合硬件的发展，研发出了著名的单片机系统，这就大幅提升了电子产品的智能化程度。

当时有一个叫做“Sun”(不是Son哦)的公司，眼光很敏锐。为了抢占广大的消费电子产品市场，在1991年成立了一个名为“Green”的项目组，由詹姆斯·高斯林(高司令)、帕特里克等几个工程师一起负责研发新技术，专攻家电产品里的嵌入式应用开发。一开始Green项目组打算用C++进行嵌入式开发，但由于C++写出来的程序特别复杂和庞大，不适合在嵌入式的环境中运行。而且不同厂家生产的电子产品，其底层所使用的硬件系统也各有不同，C++写出来的东西无法实现“一次编码，处处运行”的跨平台需求。

所以为了解决以上这些问题，Green项目组根据嵌入式的特点，对C++进行了改造，去除了C++中不太实用及影响安全的部分，并针对实时性要求，开发出一种称为Oak(橡树)的面向对象语言。但此时Oak除了比C++更简单之外，其实也没有太大的区别，所以当时的硬件厂商对Oak没有太大的热情。而Oak也就因为缺乏硬件的支持无法进入市场，从而被搁置了下来。

到了1994年6、7月间，Green团队决定改变接下来的发展目标。因为Oak本身是一种精简的语言，程序非常小，很适合在网络上传输，所以他们决定把Oak用在万维网上。到了1995年，Sun公司正式决定启用Oak，并推出了可以随同网页在网络上传输的Applet技术。

结果就在此时，Sun公司发现Oak这个商标被别的公司给提前抢注了。所以没办法，只能将Oak重新改名！大家都知道，取名字是一个很令人抓狂的事，大家一时都找不到合适的名字来重新命名。在“百思不得其解”之际，当时正在喝82年爪哇岛咖啡的詹姆斯·高斯林灵机一动，干脆就把Oak改名叫“Java”得了。于是，从此Oak更名为Java，所以高斯林也就成了公认的“Java他爹”！幸亏高斯林当时不是在蹲坑，要不然，em.....

到了1996年1月，Sun公司发布了Java的第一个开发工具包(JDK 1.0)，这也是Java发展历程中的重要里程碑，标志着Java真正成为了一种独立的开发工具。

以上就是Java语言的起源由来，现在你对它的历史知道了吗？

## 2. 概念(掌握)

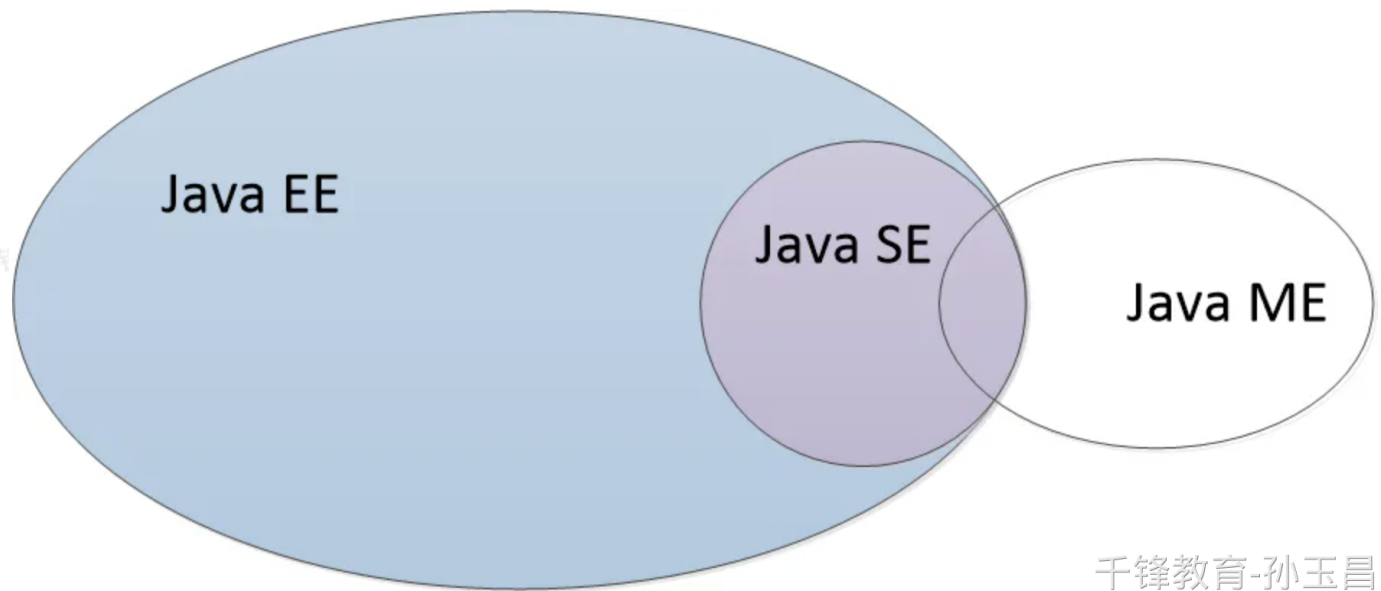
通过上面的介绍，相信各位已经对Java有所了解了，但实际上Java到底是个什么鬼，有些朋友可能还是没有搞清楚。别急，壹哥再给大家把Java的核心概念提炼一下。

- Java是曾经大名鼎鼎的Sun公司，在1995年5月正式推出的一种面向对象的程序设计语言。
- Java一开始是Sun公司的产品，但后来Sun被Oracle(甲骨文)给收购了，所以**现在Java属于Oracle公司**。
- Java语言是由James Gosling(詹姆斯·高斯林，“高司令”)和同事们共同研发的，通常我们认为高斯林是“Java之父”。
- **Java是一种介于编译型和解释型之间的编程语言，但更偏向于解释型。**编译型语言写出的代码会根据CPU类型的不同，编译成不同的机器码执行，工作原理类似于“翻译书籍”，如C、C++等；解释型语言可以由解释器直接加载源码运行，工作原理类似于“同声传译”，但运行效率较低，如JavaScript、Python、Ruby等。
- 为了使得定位更清晰，Sun公司将Java分成了三个技术体系：
  - **JavaSE(J2SE)**：Java 2 Platform Standard Edition，Java平台标准版，包含标准的JVM和标准库；
  - **JavaEE(J2EE)**：Java 2 Platform,Enterprise Edition，Java平台企业版，在JavaSE的基础上增加了大量的API和库，用于企业级项目开发；
  - **JavaME(J2ME)**：Java 2 Platform Micro Edition，Java平台微型版，主要针对移动平台开发。
- 2005年6月，JavaOne大会召开，发布Java SE 6版本，并正式将Java版本名称中的数字“2”取消。从此J2EE更名为Java EE，J2SE更名为Java SE，J2ME更名为 Java ME。所以如果有谁再跟你说什么“J2SE、J2EE、J2ME”，你就送他一个黑人问号脸，“你到底是有多out才会说出这么老古董的话”？

### 3. 技术体系

虽然壹哥在上面的小节中已经给大家说过，Sun把Java分成了JavaSE、JavaEE、JavaME共3个平台体系，但我还是想把这个知识点再跟大家强调一下，因为有太多的小白曾经问过我：“壹哥，我到底是要学JavaSE，还是JavaEE？JavaSE和JavaEE到底是什么关系？……”所以在我们正式开始学习Java之前，壹哥还是希望能够把各位心中的这些疑惑都给你解决掉。

为了搞清楚JavaSE、JavaEE、JavaME之间的关系，壹哥给大家绘制了下面这张图片：



从上图中你应该可以看出来，JavaEE的“范围是最大的”，这表示什么意思呢？这其实主要是说，JavaEE这个分类里包含的API(应用程序接口，我们可以暂时理解为Java提供的各种“工具”)最多，而JavaSE则完全包含在JavaEE里面，JavaME中有一小部分的API和JavaSE是重合的。

由此我们可以得出一个结论，作为初学者，我们需要先学习JavaSE里提供的各种常用API，等把JavaSE学完了就有“资格”去学习JavaEE或JavaME了！也就是说，**JavaSE是Java的必备入门基础，初学者肯定要先学习JavaSE**。这就好比你要学习中医技术，有针灸、推拿、中药等几个方向可以供你选择。但无论你要选择哪个方向，你都得先学习《中医基础理论》这门课，学完了这个才能有资格去学习其他具体的某个方向。现在你知道JavaSE与JavaEE的关系了吗？

但有的小伙伴还会好奇，要不要学习JavaME呢？壹哥告诉你，这个就算了！JavaME本来是Sun用于开发手机等设备的技术平台，但后来Android、iOS的异军突起，现在几乎就没人使用JavaME了，所以大家知道有这么一个东西就可以了。

我们学习完JavaSE之后，还不具备企业级开发的能力，**接下来要学习JavaEE里的各种API**。所以**我们学习Java的正确路线，就是先学JavaSE，再学JavaEE**，而壹哥的《从零开始学Java》系列文章，主要就是讲解JavaSE哦！

其实对于初学者来说，你大可不必过于纠结这些没什么实用价值的概念。等学到后面，你自然而然就会明白它们之间的具体区别，我们只需对其有个大致的了解，把精力放在最核心的问题上即可。

## 4. Java特性

对于Java，我们还需要了解一下Java的一些优势和特性，有时候面试官会贱兮兮的问你：你对Java了解多少？你为什么学习Java而不是其他语言？Java有哪些优势.....针对这样的问题，咱们就简单回答一下Java的核心特性即可。

- **使用简单：**Java的语法与C和C++很接近，可以方便大量C系程序员快速转向Java。但又舍弃了C++中那些特别复杂难用的特性，如操作符重载、多继承、自动强制类型转换等。另外Java不支持指针，而是支持引用，并提供了自动分配和GC垃圾回收机制，程序员不必关心内存管理。
- **面向对象：**Java提供了类、接口和单继承等面向对象的特性，并支持类与接口之间的实现机制，全面支持动态绑定。可以说，Java语言是第一个非常纯粹的面向对象的程序设计语言。
- **健壮安全：**Java具有强类型机制、异常处理、GC垃圾自动收集、安全检查机制、丢弃指针等特性，保证了Java程序的健壮性。并且Java提供了安全机制来防止恶意代码的攻击，可以通过ClassLoader加载器进行字节代码检查，通过SecurityManager安全管理机制设置安全哨兵，保证了Java程序的安全性。
- **跨平台/可移植性：**Java基于JVM虚拟机机制，.java文件会被JVM编译成适合不同平台的.class字节码文件，实现了“一次编写，处处运行”的跨平台特性。正因为这个特性，也使得Java代码可以轻松实现不同平台间的移植。
- **开源/免费：**在2006年的JavaOne大会上，Sun公司宣布要将Java开源，之后陆续将JDK的各个部分在GPL v2(GNU General Public License v2)协议下公开了源码(即OpenJDK)。除了极少量的产权代码外，OpenJDK几乎包括了Sun JDK的全部代码，两者的代码基本上完全一样。目前JDK 8及以前的版本都是完全免费的，JDK 9之后Oracle宣布会对企业进行收费，但是企业可以免费使用Open JDK。
- **高性能：**Java是介于编译型与解释型之间的语言，但更偏向于解释型，执行性能比C语言更高效。而且现在Java的编译器不断更新，性能已经与C++类似了。
- **支持网络和多线程：**Java支持 Internet网络开发，提供了用于网络编程的类库，比如URL、URLConnection、Socket、ServerSocket、RMI(远程方法激活)等，并且支持多线程和并发同步机制。

## 二. Java发展史

对一个即将开始学习Java的小白来说，我们还是有必要适当地了解一下Java的发展史的，看看这些年Java都经历了哪些历史变革吧。

- **1995年5月23日，Java诞生；**
- 1996年1月，JDK 1.0诞生；
- 1997年4月2日，Java One会议召开，参与者超过万人，打破当时全球同类会议规模纪录；
- 1997年9月，Java Developer Connection社区成员超十万人；
- 1998年2月，JDK 1.1被下载超过 2,000,000次；
- 1998年12月8日，JAVA2企业平台J2EE发布；
- 1999年6月，SUN发布Java的三个版本：JavaSE、JavaEE、JavaME；
- 2000年5月8日，JDK 1.3发布；
- 2000年5月29日，JDK 1.4发布；
- 2001年6月5日，NOKIA宣布到2003年将会出售超 1亿部支持Java的手机；
- 2002年2月26日，J2SE 1.4发布，Java计算能力有了大幅提升；
- 2004年9月30日，J2SE 1.5发布，成为Java语言发展史上的又一里程碑。**为了表示该版本的重要性，J2SE 1.5更名为 Java SE 5.0；**
- 2005年6月，Java One大会召开，Sun公司发布Java SE 6。此时Java各种版本已经全部更名，取消了其中的数字 "2"：J2EE更名为Java EE，J2SE更名为Java SE，J2ME更名为Java ME；
- 2006年12月，Sun公司发布JRE 6.0；
- **2009 年04月20日，甲骨文以74亿美元收购Sun，取得了Java版权；**
- 2011年7月28日，Oracle发布Java 7.0 正式版；
- **2014年3月18日，Oracle发布Java SE 8.0(目前最主流版本)；**
- 2017年9月21日，Oracle发布Java SE 9.0；
- 2018年3月21日，Oracle发布Java SE 10.0；
- **2018年9月25日，Oracle发布Java SE 11；**
- 2019年3月20日，Oracle发布Java SE 12；
- 2019年9月，Oracle发布Java SE 13；
- 2020年3月，Oracle发布Java SE 14；
- 2020年9月，Oracle发布Java SE 14；
- 2021年3月，Oracle发布Java SE 16；
- **2021年9月，Oracle发布Java SE 17；**

- 2022年3月，Oracle发布Java SE 18；
- 2022年9月，Oracle发布Java SE 19；
- .....Java继续在大步前进.....

当然，对于以上历史，大家了解即可，不用去记这些东西。

### 三. Java版本(重点)

在上面的章节介绍中，壹哥给大家提到了Java的很多版本，并且这些不同的版本所具有的功能也不尽相同。所以就有小白来问了，我们学习Java时到底该选择哪个版本呢？是越新越好吗？壹哥认为很有必要就这个问题给大家解释一下。

在Java的发展过程中，有很多不同的表述方式，有时候会给初学者造成一些误解。但请你记住，所谓的Java版本，其实就是指JDK的版本，两者是一回事！

#### 1. JDK版本

Oracle把JDK分成了两种维护情况，**即短期支持版本和长期支持版本**，我们先来看看 [Oracle官方发布的JDK路线图](#)：

Release	GA Date	Premier Support Until	Extended Support Until	Sustaining Support
7 (LTS)	July 2011	July 2019	July 2022*****	Indefinite
8 (LTS)**	March 2014	March 2022	December 2030*****	Indefinite
9 (non-LTS)	September 2017	March 2018	Not Available	Indefinite
10 (non-LTS)	March 2018	September 2018	Not Available	Indefinite
11 (LTS)	September 2018	September 2023	September 2026	Indefinite
12 (non-LTS)	March 2019	September 2019	Not Available	Indefinite
13 (non-LTS)	September 2019	March 2020	Not Available	Indefinite
14 (non-LTS)	March 2020	September 2020	Not Available	Indefinite
15 (non-LTS)	September 2020	March 2021	Not Available	Indefinite
16 (non-LTS)	March 2021	September 2021	Not Available	Indefinite
17 (LTS)	September 2021	September 2026****	September 2029****	Indefinite
18 (non-LTS)	March 2022	September 2022	Not Available	Indefinite
19 (non-LTS)***	September 2022	March 2023	Not Available	Indefinite
20 (non-LTS)***	March 2023	September 2023	Not Available	Indefinite
21 (LTS)***	September 2023	September 2028	September 2031	Indefinite

对于短期支持版本(non-LTS)而言，Oracle只会提供6个月的支持维护；而对于长期支持版本(LTS)，则提供8年的支持维护。所以根据这一规划，Oracle每隔6个月，就会发布一个大版本，每个季度则发布一个中间特性版本。另外Oracle还承诺，新的JDK发布周期会严格遵循时间点，将于每年的3月份和9月份发布，中间不会跳票。

现在正常情况下，每隔6个月就会有一个短期维护版本(non-LTS)发布出来，比如JDK 9、10、12、13、14、15、16；然后每隔3年，就会发布一款得到8年长期支持维护的JDK版本，比如JDK



8、11、17。所以小伙伴们，其实你没必要下载使用JDK9/10/12等非-LTS版本，应该使用JDK 8、11、17这样的LTS版本。但目前绝大多数主流企业使用的版本是JDK 8，但不会使用最新的版本，而且Oracle宣布从JDK 9.0之后开始针对企业收费，所以最新版的JDK使用情况待议。

## 2. 版本选择

既然JDK版本有这么多，作为小白，你到底该选择哪个版本的JDK呢？是直接学习JDK 18还是19？或者等几个月学习JDK 21？壹哥告诉你，如果你是一个新手小白，其实没必要追求学习最新的JDK！

首先每一个JDK版本推出时，都存在着很多未知的bug，所以企业在开发时，都会选择最稳定的版本！现在主流的企业级应用都在使用JDK 8.0正式版，并且JDK 8.0及之前版本都是免费的，而JDK 9.0以后则开始收费(针对企业收费，非个人)。所以公司考虑到稳定性和费用等问题，都在使用JDK 8.0，这才是主流！

另外对初学者来说，JDK的新特性其实对小白来说意义并不大。这些新特性一般是针对性能有较大提升，但你作为一个小白，连最基本的用法都不会呢，现在去追求这些新特性没必要。等你把JavaSE学完之后，以后自己再针对性地提升即可。请先把精力放在最核心的地方，要不然基础还没搞扎实，却一心贪多求快，很容易出现“空中楼阁”的状况。

另外有些培训机构，宣称会重点讲解JDK 12、JDK 10、JDK 9等版本的内容，不过是哗众取宠。这些版本官方都停止维护和下载了，你讲这些有啥用。

所以，为了贴合最主流的企业要求，壹哥的本套教程案例都是基于JDK 8来进行讲解的！如果你的JDK版本比我的版本高，当然是可以直接兼容运行我的案例的，不用担心！

## 四. 配套视频

至此，壹哥通过本文给大家简单地介绍了一下Java的历史。其实对于今天的内容，大家简单看看，把重点内容稍微记一下就好，不用在这些地方花费太多精力。

开场白已结束，在下一篇文章中，壹哥会带各位先熟悉Java的运行机制和JVM虚拟机工作原理，然后搭建出学习Java时的必备环境——《JDK安装及环境变量配置》，敬请各位期待哦！

-----正片已结束，来根事后烟-----  
-----

如果你不习惯阅读技术文章，或是对文中的技术概念不能很好地理解，可以来看看壹哥帮你筛选出的视频教程。如果你自己去筛选学习视频，会浪费很长的时间，且找出来的视频也未必适合你。所



以壹哥专门为大家找到了与本文配套的Java学习视频，[视频链接如下](#)：

<https://player.bilibili.com/player.html?bvid=BV1Ja411x7XB&p=5&page=5>

另外如果你独自学习觉得有很多困难，可以私信壹哥，[【加入壹哥的学习互助群】](#)，我们大家一起交流学习。

## 五. 今日作业

### 1. 第一题：

此处为语雀投票卡片，点击链接查看：

[https://www.yuque.com/\\_\\_workers/yiyigesun/hkm11k/pyb9vu/pdf#CCNZ4](https://www.yuque.com/__workers/yiyigesun/hkm11k/pyb9vu/pdf#CCNZ4)

### 2. 第二题：

此处为语雀投票卡片，点击链接查看：

[https://www.yuque.com/\\_\\_workers/yiyigesun/hkm11k/pyb9vu/pdf#vtivu](https://www.yuque.com/__workers/yiyigesun/hkm11k/pyb9vu/pdf#vtivu)

### 3. 第三题：

Java分为哪几个平台体系？有什么区别？请在评论区写出你的答案哦！

### 4. 第四题：

目前Java的长期维护版本有哪些？请在评论区写出你的答案，并说出你希望学习的JDK版本吧。