Table of Contents

个人简介

大家可以叫我沈老师、也可以叫我易风；我从事IT软件行业10年+，爱好技术，爱好写代码。闲暇之余喜欢旅游、爬爬山、打打球、或者在家里研究一下厨艺。

个人亮点

- 全面的技术视野

  10年以上技术开发管理经验，负责前公司基础平台和部分核心业务系统且一直保持编码状态(基础架构/业务核心代码)，熟悉后端/中间件/大前端/DevOps/微服务/容器等技术栈。

  毕业后近15年一直在一家软件公司，经历并成功主导了公司从C/S到B/S、ASP到Java、传统软件到互联网的技术转型。

- 熟悉ToB/ToC业务

  至今负责过40个以上业务类项目的规划设计及核心开发工作，涉及ToB、ToC行业。

- 熟悉支付/金融/电商领域

  负责项目很多涉及支付/金融/电商，因此对这三个领域特别熟悉。

- 擅长研发效能管理

  曾负责的项目基本都高质量按期或提前交付，对设计规划/研发/测试/部署流程有较好的把控。

从业经历

- `2005~2019` 某软件公司 CTO
- `2019至今` 自由职业(咨询培训/团队组建/架构设计/开发)

技术栈

- 架构设计· 面向对象/微服务/DDD
- 设计工具· Axure/Sketch/蓝湖/磨刀
- 编程语言· Java/JavaScript/C++/PHP/Python/C#/SQL
- Java技术栈· Java/Apache/Guava/Spring/MyBatis等
- 数据库· MySQL/MSSQL/Oracle
- 缓存系统· Redis
- 消息队列· RocketMQ/RabbitMQ
- Test· 单元测试/集成测试/接口测试
- DevOps· LinuxShell/Git/CI/Maven/Jenkins/Docker/云效
- 大前端· jQuery/React/Vue/App(安卓/iOS/混合模式)
- 中间件· Nginx/Jetty/Tomcat/Undertow
- 微服务· SpringCloud/Dubbo/Thrift
- 微信生态· 公众号/小程序/微信支付/腾讯云
- 阿里生态· 生活号/小程序/支付宝/淘宝天猫/阿里云
- 第三方服务· 短信/存储/推送/第三方登录分享/短视频/直(点)播等

为什么来做培训

在前公司负责招聘面试时，也和很多培训机构合作过，发现很多培训机构还是按10年前的套路在进行培训、还是SSH三大框架，培训的老师也基本没多少项目经验；最后的实战项目都叫的很大，比如某某电商系统、某某电信运营系统，但实际真正能从中学到多少?

基于这些原因，我感觉目前培训这一块还是有可以改进的地方，我想我可以为大家带来一个不一样的培训，当然收获也会不一样。

所以下边第一节内容，我首先会讲 Java 技术栈，我们要学什么，怎么学。学习目的有两个：提高自身技术素养、面试通关。

Java 语言简介

Java 历史

我们简单了解一下 Java 的发展历史中几个比较重要的时间点。

- `1990年12月`，sun成立了一个由 Jeams Golsling博士（Java之父）领导的绿色计划“Green Team”项目，该项目的目的是开发一种能够在各种消费性电子产品（如机顶盒、冰箱、收音机等）上运行的程序架构。
- `1991年`，Java 语言前身，Oak(橡树）诞生了。
- `1995年5月23日`，Oak 改名 Java，并在 SunWorld 大会上正式发布 Java 1.0 版本。
- `1996年1月23日`，JDK 1.0 发布，Java 语言有了第一个正式版本的运行环境。
- `2004年9月30日`，JDK 1.5 发布，工程代号 Tiger。从 JDK 1.2 开始，Java 在语法层面上的变化就一直比较小，而 JDK 1.5 在 Java 语法易用性上做出了非常大的改进。比如：自动装箱/拆箱、泛型、动态注解、枚举、可变长参数、遍历循环（foreach）等。在虚拟机和 API 层面上，这个版本改进了 Java 的内存模型、提供了 java.util.concurrent 并发包等。
- `2006年12月11日`，JDK 1.6发布，工程代号 Mustang（野马）。在这个版本中，Sun 终结了从 JDK 1.2 开始已经有8年历史的J2EE、

J2SE、J2ME的命名方式，启用Java SE 6、Java EE 6、Java ME 6的命名方式。

- `2009年` Oracle 以74亿美元收购了 sun。
- `2011年7月`，Java SE 7 发布。
- `2014年3月`，Java SE 8 发布。
- `2017年7月`，Java SE 9 发布。
- `2018年3月21日`，Java SE 10 发布。
- `2018年9月25日`，Java SE 11 发布。
- `2019年3月`，Java SE 12 发布。

Java 语言特性

Java 语言特性

Java 是一门面向对象的编程语言，他有下面一些特点：

- 简单性
- 面向对象
- 网络技能
- 健壮性
- 安全性
- 体系结构中立
- 可移植性
- 解释性
- 高性能
- 多线程
- 动态性

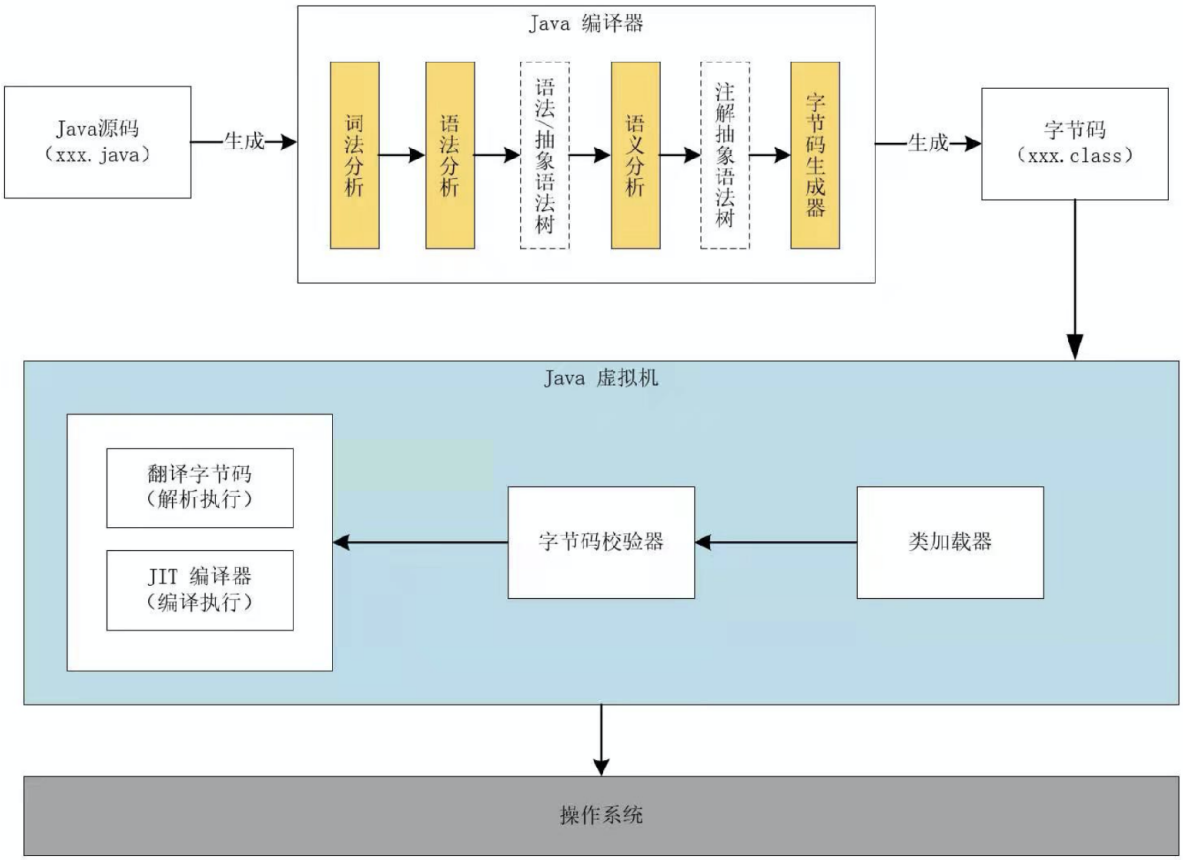
Java / JDK / JRE / JVM

Java 是一门语言，而 JDK 是 Java 语言开发工具包，它包括：编译器、Java 运行环境（JRE，Java Runtime Environment）、JVM（Java 虚拟机）、监控和诊断工具。

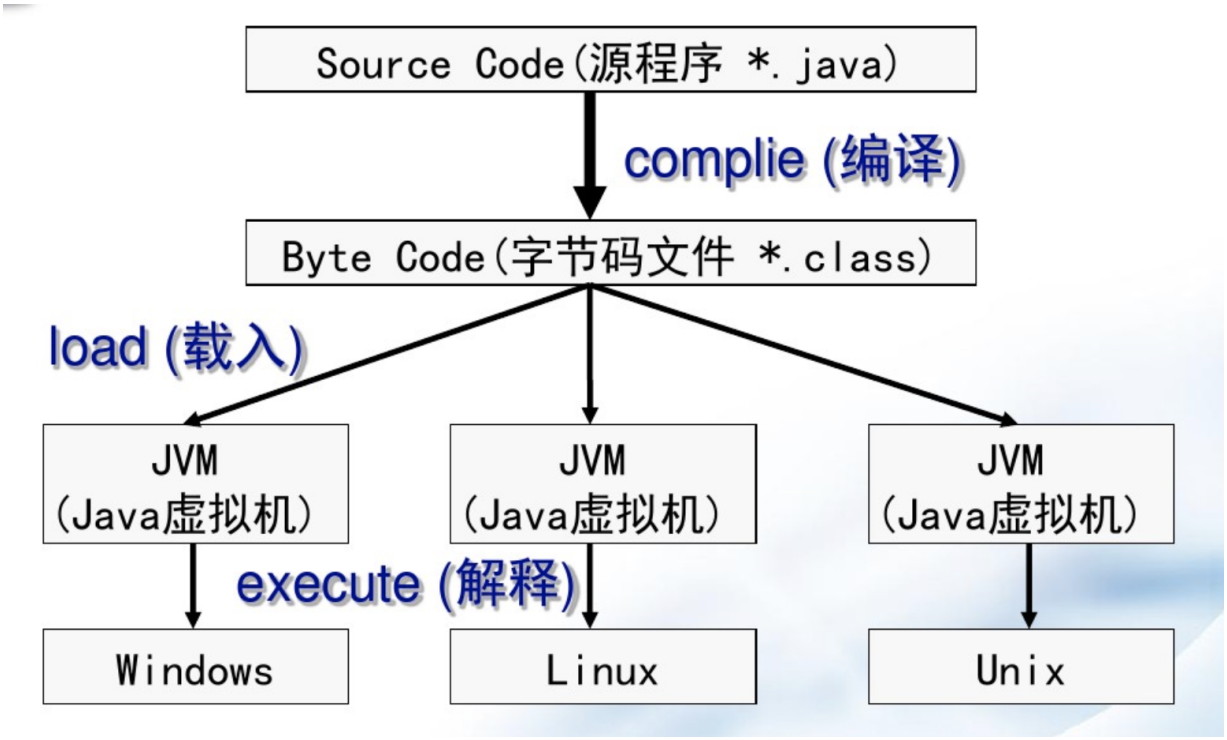
那么 Java 语言是怎么在机器上执行，并实现一次编写、到处执行（Write Once，Run Anywhere）。

1. 通过编译器将 Java 源代码（.java文件）编译成字节码（.class文件）。
2. 将 class 文件放置到 Java 虚拟机中。
3. Java 虚拟机使用类加载器（Class-Loader）加载 class 文件。
4. 类加载完成后，会进行字节码校验，字节码校验通过后 JVM 解释器会将字节码翻译成机器码交由操纵系统执行。但并不是所有的代码都会解释执行，在主流 Java 版本中，如 JDK 8 实际是解释和编译混合的一种模式。像 Oracle Hotspot JVM 内置了 JIT compiler（Just In Time，动态编译器），他能够在运行时将热点代码编译为机器码，然后再进行执行，以提高执行效率，这时字节码就变成了编译执行。

Java 程序执行流程图如下：



那么 Java 语言是如何实现跨平台，一次编写、到处执行的？他是通过 JVM 虚拟机来屏蔽了不同服务器类型之间的差异，而 Java 语言运行在 JVM 之上，如下图：


Source Code(源程序 *.java)
complie (编译)
Byte Code(字节码文件 *.class)
load (载入)
JVM
(Java虚拟机)
JVM
(Java虚拟机)
JVM
(Java虚拟机)
execute (解释)
Windows
Linux
Unix

学什么

本次课程将侧重于基础理论的深入学习，主要包括 Java基础 + JDK源码、Web 基础、Spring+MyBatis、MySQL。

- Java基础 + JDK源码

  学习 Java 基础的同时深入相应的 JDK 源码，了解源码的实现理论。为什么要学习 JDK 源码的实现？请大家看一下下边代码的执行结果，实际运行一下，看看是不是和自己想象的一样？

```
Integer num1 = 127;
Integer num2 = 127;
System.out.println(num1 == num2);

Integer num3 = 128;
Integer num4 = 128;
System.out.println(num3 == num4);

Integer num5 = 127;
Integer num6 = new Integer(127);
System.out.println(num5 == num6);

int num7 = 127;
Integer num8 = 127;
System.out.println(num7 == num8);

int num9 = 128;
Integer num10 = 128;
System.out.println(num9 == num10);

Double d1 = 127D;
Double d2 = 127D;
System.out.println(d1 == d2);

Double d3 = 128D;
Double d4 = 128D;
System.out.println(d3 == d4);

System.out.println(2 * 0.1 == 0.2);
System.out.println(3 * 0.1 == 0.3);
```

为什么会这样？我们来看一下 Integer 的源码实现（节选了部分）：

```
  public static Integer valueOf(int i) {
      if (i >= IntegerCache.low && i <= IntegerCache.high)
          return IntegerCache.cache[i + (-
IntegerCache.low)];
      return new Integer(i);
  }
```

```
  private static class IntegerCache {
      static final int low = -128;
      static final int high;
      static final Integer cache[];

      static {
          // high value may be configured by property
          int h = 127;
          String integerCacheHighPropValue =

sun.misc.VM.getSavedProperty("java.lang.Integer.IntegerCac
he.high");
          if (integerCacheHighPropValue != null) {
              try {
                  int i =
parseInt(integerCacheHighPropValue);
                  i = Math.max(i, 127);
                  // Maximum array size is
Integer.MAX_VALUE
                  h = Math.min(i, Integer.MAX_VALUE - (-
low) -1);
              } catch( NumberFormatException nfe) {
                  // If the property cannot be parsed into
an int, ignore it.
              }
          }
          high = h;

          cache = new Integer[(high - low) + 1];
          int j = low;
          for(int k = 0; k < cache.length; k++)
              cache[k] = new Integer(j++);

          // range [-128, 127] must be interned (JLS7
5.1.7)
```

```
            assert IntegerCache.high >= 127;
        }

        private IntegerCache() {}
    }
```

- Web 基础

  学习 HTTP 协议，深入理解 Servlet/Filter/Session/Cookie/Session/JDBC 实现原理。其中一个实战中我们不使用任何第三方类库，自己实现一个简单的但是功能齐全的Web应用，以加深对相关理论的理解。

- Spring+MyBatis

  以Spring+MyBatis为基础的开发架构的使用，会对后端进行合理分层开发，以及如何测试（单元测试、集成测试）、常用设计模式的应用。常用分层法：

  - controller

    接口层，对外暴露接口，可以是 Web、APP、小程序、公众号、第三方应用等。本身没有业务逻辑，直接调用 Service 层。

  - service

    负责参数检验、Manager层的调用、跨领域的业务聚合、幂等事务等工作。

  - manager

“单领域层”，负责单个业务对象的逻辑。

- dao

  数据操作层，对 Manager 层提供数据读写/存储操作。

- mapper

  MyBatis XML 和数据库实体的映射。

- repository

  SpringJPA 数据访问。

- entity

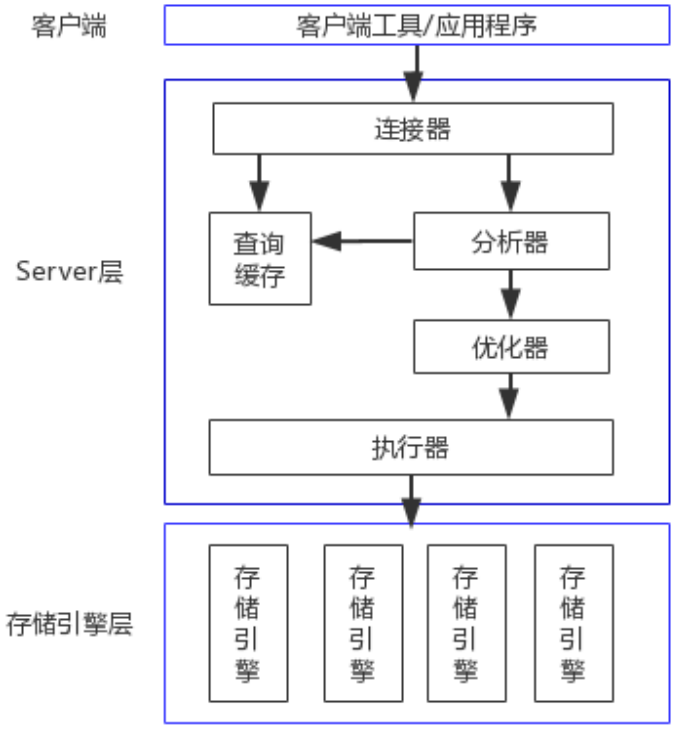
  Java 对象和数据库表的映射。

- MySQL

  深入了解我们常用的 MySQL 相关原理

  - MySQL 逻辑架构
  - 锁：读写锁、锁粒度、表锁、行级锁
  - 事务：ACID原则、事务隔离级别、死锁
  - 日志：重做日志 redo log、归档日志 binlong、两阶段提交
  - 多版本并发控制
  - 存储引擎：InnoDB
  - 索引：索引类型、索引优点、索引策略、索引和锁

    MySQL 逻辑架构图如下：



因为本次课程主要侧重于 Java 后端，因此对于前端来说，我只会着重讲一下 JS 基础，而对于 CSS/HTML 部分只简单介绍一下。

大前端技术流派：

- `原生JS`：jQuery/Bootstrap
- `Vue`：element-ui
- `React`：ant.design
- `微信小程序`
- `APP`：Android/iOS/混合

本次课程对于前端部分，我会选择一个小程序前端框架和一个Vue框架来直接使用。

怎么学

怎么学，就两个字：“多练”。主要从以下几个方面进行：

- `技术博客`：每学完一部分，将所学内容根据自己的理解再写一遍，建立自己的技术博客。
- `思维导图`：根据自己的理解建立一张 Java 栈技术思维导图，每学一部分将所学部分加入思维导图。可以使用 XMind 或者 https://processon.com/。
- `学习小组`：有时间的同学可以相互成立5人左右规模的学习小组，课后练习可以相互 Review 代码，可以定期（比如一周）来自己总结讲一下最近学习的内容。（可以使用zoom视频会议软件）
- `自己玩`：任何知识，你学的再好，不在实践中应用，100%会忘记，只是忘记的时间早晚而已。老师讲的案例大概率不是你感兴趣的，因此你自己想一个要做的项目，比如：个人账单记录、背单词、好友闲置物品买卖等。实现方式：微信小程序+LeanCloud。
- `作业提交`：作业提交方式会以 git 方式进行。

本次课程的案例，都会选择咱们会用到的一些内容来做出来，大家参与，大家来用。比如下边列举的一些，可以选择其中几项：

- 作业 Review
- 知识问答
- 模拟面试
- 信息聚合

- 闲置物品买卖

书单

- `Java Language and Virtual Machine Specifications`

  https://docs.oracle.com/javase/specs/index.html

- 《Java编程思想（第4版） [thinking in java]》

  https://item.jd.com/10058164.html

- 《Java核心技术 卷I：基础知识（原书第10版）》

  https://item.jd.com/12037418.html

- 《Effective Java中文版（原书第3版）》

  https://item.jd.com/12507084.html

你应该知道的一些网站

- `CSDN` https://www.csdn.net/
- `开源中国` https://www.oschina.net/
- `github` https://github.com/
- `StackOverflow` https://stackoverflow.com/
- `SegmentFault` https://segmentfault.com/
- `v2ex` https://www.v2ex.com/
- `掘金` https://juejin.im/
- `腾讯云社区` https://cloud.tencent.com/developer
- `云栖社区` https://yq.aliyun.com/
- `开发者头条` https://toutiao.io/
- `InfoQ` https://www.infoq.cn/
- ……

面试问答

Java 语言有哪些特点

- 面向对象
- 跨平台
- 执行性能好，运行效率高
- 有大量 API 扩展
- 多线程
- 安全性好，自带验证机制
- GC

JDK / JRE / JVM 有什么区别

JRE: Java Runtime Environment 的简称，Java 运行环境，为 Java 的运行提供了所需环境。 JDK: Java Development Kit 的简称，Java 开发工具包，提供了 Java 的开发环境和运行环境。 JVM：Java Virtual Machine 的简称，Java 虚拟机，是一种计算设备的规范，是一个虚拟出来的计算机，所有的 Java 程序以及实现了 JVM 规范的语言都可以运行在 JVM 上。

JDK 包含了 JRE，同时还包含了 javac、调试和分析工具，主要是 Java 开发者使用。JRE 是 Java 运行的最小单元，一般安装在 Java 服务器上。JVM 是 Java 程序运行的载体，只有通过 JVM，Java 程序才能正常运行。

Java 是解释执行吗?

这个说法不太准确。Java 的源代码，首先通过 Javac 编译成为字节码（bytecode），然后，在运行时，通过 Java 虚拟机（JVM）内嵌的解释器将字节码转换成为最终的机器码。但是对于热点代码，比如 Oracle JDK 提供的 Hotspot JVM，都提供了 JIT（Just-In-Time）编译器（动态编译器），JIT 能够在运行时将热点代码编译成机器码，这种情况下部分热点代码就属于编译执行，而不是解释执行了。

课后练习

- 建立自己的技术博客
- github账号
- 申请微信小程序个人账号
- LeanCloud账号 https://www.leancloud.cn/