

# 第一章 Java入门引导与内容

---

## 1.1 课程内容

---

### 1.1.1 学习课程的三个阶段

1. 培养编程思想，体会编程的乐趣
2. 提升编程能力，将编程融入到生活中
3. 分析需求，使用代码实现定制功能

### 1.1.2 课程说明

1. 以全方位培养编程思想、提升编程能力作为核心目标，助力同学们达到能够精准剖析业务需求，并熟练运用代码实现的专业水平。
2. 全程采用项目导向式教学，让同学们在攻克一个个项目的过程中，收获满满的成就感，深切领略编程的独特乐趣。
3. 项目体系从基础起步，逐步拓展规模。每个项目都会被细致拆解为多个功能模块，随着功能的不断丰富与完善，各类编程知识点自然而然地融入其中，帮助同学们循序渐进，以轻松高效的方式完成知识积累与能力进阶。
4. 我们鼓励同学们亲自创建专属项目，从无到有精心培育，就如同悉心照料一只宠物，在这个过程中，感受每一次成长与进步带来的喜悦。

## 1.2 就业方向

---

### 1. Java 开发工程师

- 主要负责使用 Java 语言进行各类软件系统的设计、开发与维护，包括企业级应用、Web 应用、桌面应用等。
- 例如开发一个电商网站的后台系统，负责处理用户注册登录、商品管理、订单处理等功能。

### 2. Java Web 开发工程师

- 专注于 Web 应用程序的开发，熟练掌握 Servlet、JSP 等技术，与前端开发人员协作，实现网站的动态交互功能。
- 比如开发一个在线教育平台的课程展示、学习记录等 Web 功能。

### 3. Java 软件工程师

- 参与整个软件项目的生命周期，从需求分析、设计、编码到测试和维护，确保软件的质量和性能。
- 以开发一款企业办公自动化软件为例，负责实现流程审批、文档管理等功能模块。

### 4. Java 架构师

- 负责设计和规划复杂的软件系统架构，制定技术方案和技术选型，指导开发团队进行技术研发。
- 如在设计大型分布式系统时，确定采用微服务架构，选择合适的框架和技术来保证系统的高可用性和可扩展性。

### 5. Android 开发工程师

- 由于 Java 是 Android 应用开发的主要语言之一，可从事 Android 移动应用的开发工作，为智能手机和平板电脑等设备开发各种应用程序。
- 比如开发一个健身类的 Android 应用，实现运动记录、课程推荐等功能。

### 6. 大数据开发工程师

- 利用 Java 操作 Hadoop、Spark 等大数据处理框架，进行数据的采集、存储、处理和分析。
- 例如从海量的用户行为数据中提取有价值的信息，为企业的决策提供支持。

#### 7. 云计算工程师

- 参与云计算平台的开发与维护，使用 Java 与云平台的 API 进行交互，实现云服务的部署和管理。
- 像在阿里云等平台上，利用 Java 开发自定义的云函数，实现特定的业务逻辑。

#### 8. 软件测试工程师

- 运用 Java 编写测试脚本和测试工具，对软件进行功能、性能、自动化等方面的测试，确保软件的质量。
- 例如使用 Selenium 等框架，用 Java 编写自动化测试脚本，对 Web 应用进行功能测试。

#### 9. 技术支持工程师

- 为用户提供技术支持和维护服务，解决用户在使用 Java 软件过程中遇到的问题，对软件进行优化和升级。
- 当企业用户反馈某个 Java 应用出现故障时，负责排查问题并进行修复。

#### 10. 游戏开发工程师

- 利用 Java 的游戏开发框架，如 LibGDX 等，开发 2D 或简单的 3D 游戏。
- 开发一款类似于《植物大战僵尸》的休闲游戏，实现游戏场景、角色、交互等功能。

同学们可以根据以上介绍，选择一个自己的目标岗位，逐步接近。

## 1.3 Java的应用领域

### 1. 企业级应用开发

- **大型商业系统**：许多企业的核心业务系统，如银行的金融交易系统、电信运营商的计费系统等，都需要处理大量的数据和高并发的业务请求，Java 凭借其强大的性能和稳定性，能够满足这些系统的需求。
- **企业资源规划（ERP）系统**：像 SAP、Oracle E-Business Suite 等知名的 ERP 系统，部分功能就是基于 Java 开发的，用于整合企业的财务、采购、销售、生产等各个环节的业务流程。
- **客户关系管理（CRM）系统**：Salesforce 等 CRM 系统也会使用 Java 来实现客户信息管理、销售流程跟踪、市场营销等功能，帮助企业更好地管理客户关系，提高客户满意度。

### 2. Web 开发

- **门户网站**：各大新闻网站、政府网站、企业官网等通常会使用 Java 来构建后台服务，如处理用户请求、生成动态页面内容、管理网站数据等。
- **电子商务平台**：像淘宝、京东等大型电商平台，其后台的商品管理、订单处理、用户管理等核心业务逻辑很多都是用 Java 实现的，以保证系统在高并发情况下的稳定运行。
- **Web 应用框架**：Spring MVC、Struts 等 Java Web 框架为 Web 开发提供了便捷的开发模式和工具，使得开发人员能够快速构建功能丰富的 Web 应用。

### 3. 大数据处理

- **数据采集与存储**：在大数据领域，Java 常用于编写数据采集程序，从各种数据源（如数据库、日志文件、传感器等）采集数据，并将数据存储到 Hadoop 分布式文件系统（HDFS）等大数据存储系统中。
- **数据处理与分析**：Hadoop、Spark 等大数据处理框架都是基于 Java 实现的，开发人员可以使用 Java 编写 MapReduce 任务或 Spark 应用程序，对海量数据进行处理和分析，提取有价值的信息。
- **数据可视化**：借助 Java 的图形界面库和数据可视化工具，如 JFreeChart、Processing 等，可以将处理后的数据以图表、图形等形式展示出来，方便用户理解和决策。

### 4. 移动应用开发

- **Android 应用开发**: Java 是 Android 应用开发的主要语言之一, 通过 Android SDK 和 Java 语言, 开发人员可以创建各种类型的 Android 应用, 包括社交应用、游戏、办公应用等。
- **跨平台移动开发**: 一些跨平台移动开发框架, 如 Apache Cordova、Xamarin 等, 也支持使用 Java 进行开发, 使得开发人员能够一次编写代码, 在多个移动平台上运行。

## 5. 游戏开发

- **桌面游戏开发**: 可以使用 Java 的游戏开发框架, 如 Slick2D、LWJGL 等, 开发 2D 或简单的 3D 桌面游戏, 实现游戏的图形渲染、音效处理、用户交互等功能。
- **网页游戏开发**: Java Applet 曾经是网页游戏开发的常用技术, 虽然现在由于安全和性能等问题逐渐被淘汰, 但在一些旧的网页游戏或特定场景中仍有应用。此外, Java 也可以用于开发网页游戏的后台服务, 如用户管理、游戏数据存储等。

## 6. 分布式系统与云计算

- **分布式系统开发**: Java 的分布式计算框架, 如 Apache Dubbo、Spring Cloud 等, 为构建分布式系统提供了便捷的工具和技术支持, 使得开发人员能够将大型系统拆分成多个微服务, 实现系统的高可扩展性和高可用性。
- **云计算平台**: 许多云计算平台, 如 OpenStack、Cloud Foundry 等, 都使用 Java 作为主要的开发语言, 提供云基础设施管理、云服务部署等功能。

## 7. 人工智能与机器学习

- **数据预处理**: 在人工智能和机器学习项目中, Java 可以用于数据的预处理工作, 如数据清洗、数据标注、数据转换等, 为后续的模式训练提供高质量的数据。
- **机器学习算法实现**: 虽然 Python 在机器学习领域应用更为广泛, 但 Java 也有一些机器学习库, 如 Weka、Deeplearning4j 等, 开发人员可以使用 Java 来实现一些基本的机器学习算法和深度学习模型。
- **人工智能应用开发**: Java 可以用于开发人工智能应用的后端服务, 如自然语言处理、图像识别等应用的服务器端程序, 与前端应用进行交互, 提供人工智能服务。

# 1.4 Java的历史发展

## 1. 起源与孕育 (20 世纪 90 年代初)

- 1991 年, Sun Microsystems 公司的詹姆斯·高斯林 (James Gosling) 等人开始着手开发一种新的编程语言, 最初被命名为 “Oak”, 旨在用于嵌入式系统, 如智能家电等。
- Oak 语言具有跨平台、面向对象等特点, 但在当时并没有引起太大的关注, 因为它在市场上没有找到合适的应用场景。

## 2. Java 的诞生 (1995 年)

- 1995 年, 随着互联网的兴起, Sun 公司看到了 Oak 语言在网络应用方面的潜力, 将其重新命名为 “Java”, 并正式发布。
- Java 以其 “一次编写, 到处运行” 的特性, 迅速在互联网开发领域崭露头角, 它允许开发人员编写一次代码, 就可以在不同的操作系统上运行, 大大降低了开发成本和难度。

## 3. JDK 1.0 - 1.4 时代 (1996 - 2002 年)

- 1996 年, Java Development Kit (JDK) 1.0 发布, 标志着 Java 开发环境的正式形成。JDK 1.0 提供了基本的类库和开发工具, 为 Java 开发人员提供了必要的基础。
- 随后, JDK 1.1 增加了内部类、反射等重要特性, 进一步丰富了 Java 的功能。
- 1998 年, JDK 1.2 发布, 这是一个重要的版本, 引入了 Java 2 平台, 将 Java 分为三个版本: J2SE (Java 2 Platform, Standard Edition)、J2EE (Java 2 Platform, Enterprise Edition) 和 J2ME (Java 2 Platform, Micro Edition), 分别面向不同的应用领域。
- 2002 年, JDK 1.4 发布, 增加了正则表达式、NIO (New I/O) 等功能, 提升了 Java 的性能和功能。

## 4. JDK 5 - 7 时代 (2004 - 2011 年)

- 2004 年，JDK 5 发布，带来了一系列重大的改进，如泛型、注解、自动装箱 / 拆箱、枚举等特性，使 Java 的编程更加方便和安全，提高了代码的可读性和可维护性，也标志着 Java 在工业级应用方面的进一步成熟。
- 2006 年，Sun 公司将 Java 开源，这一举措极大地推动了 Java 的发展，吸引了更多的开发者参与到 Java 的开发和维护中。
- 2009 年，Oracle 公司收购了 Sun Microsystems，成为 Java 的新东家。
- 2011 年，JDK 7 发布，引入了菱形运算符、字符串在 switch 语句中的使用等新特性，进一步完善了 Java 语言。

#### 5. Java 8 的变革 (2014 年)

- 2014 年，Java 8 发布，这是 Java 发展历程中的一个重要里程碑。它引入了 Lambda 表达式和 Stream API，使 Java 在函数式编程方面有了很大的突破，大大提高了代码的简洁性和可读性，同时也提升了数据处理和并行计算的能力。
- 此外，Java 8 还增加了默认方法、方法引用等特性，为接口的演进和扩展提供了更好的支持。

#### 6. 后续发展 (2017 年及以后)

- 2017 年，Java 9 发布，带来了模块化系统，将 Java 平台拆分成多个模块，提高了系统的可维护性和安全性，同时也优化了 Java 的内存管理和性能。
- 2018 年，Java 10 发布，引入了局部变量类型推断等特性，使代码更加简洁。
- 2019 年，Java 11 发布，它是一个长期支持 (LTS) 版本，增加了 HTTP 客户端、Epsilon 垃圾回收器等新功能，进一步提升了 Java 的性能和稳定性。
- 2021 年，Java 17 发布，也是一个 LTS 版本，带来了模式匹配、密封类等新特性，增强了 Java 语言的表达能力和安全性。
- 2023 年，Project Loom 正式落地，它为 Java 带来了虚拟线程等特性，使得 Java 能够更好地处理百万级并发架构，提升了 Java 在高并发场景下的性能和效率。

## 课后练习

---

思考：

1. 想一下自己学习Java后最想做的项目
2. 说一下自己在学习Java的过程中最想收获什么（编程思维的提升？代码能力的提升？还是其他？）