



用微课学“程序猿”系列
学习 Java 的第一本书

●智慧职教平台 ●微课 ●案例源码 ●教学课件 ●课程级资源

用微课学 Java 开发基础

主编 唐亮 王洋

高等教育出版社



YONG WEIKE XUE Java KAIFA JICHU

用微课学
Java 开发基础

主编 唐亮 王洋

高等教育出版社·北京

图书在版编目 (C I P) 数据

Java 开发基础 / 唐亮, 王洋主编. -- 北京: 高等教育出版社, 2016. 7
(用微课学)
ISBN 978-7-04-045117-7

I. ① J… II. ①唐… ②王… III. ① Java 语言 - 程序设计 - 高等学校 - 教材 IV. ① TP312

中国版本图书馆 CIP 数据核字 (2016) 第 070163 号

策划编辑 张值胜 责任编辑 张值胜 封面设计 杨立新 版式设计 王艳红
插图绘制 杜晓丹 责任校对 李大鹏 责任印制 毛斯璐

出版发行	高等教育出版社	网 址	http://www.hep.edu.cn
社 址	北京市西城区德外大街4号		http://www.hep.com.cn
邮政编码	100120	网上订购	http://www.hepmall.com.cn
印 刷	北京鑫丰华彩印有限公司		http://www.hepmall.com
开 本	787mm×1092mm 1/16		http://www.hepmall.cn
印 张	14.5	版 次	2016年7月第1版
字 数	360千字	印 次	2016年7月第1次印刷
购书热线	010-58581118	定 价	35.00元
咨询电话	400-810-0598		

本书如有缺页、倒页、脱页等质量问题, 请到所购图书销售部门联系调换

版权所有 侵权必究 >

物料号 45117-00

III 内容简介

本书是高等教育出版社与达内时代科技集团（以下简称达内集团）联合出品的程序设计类新形态一体化教材，由达内集团诸多具有丰富开发及授课经验的一线讲师编写。本书以互联网企业实际用人的要求为导向，总结近几年国家应用型本科院校与示范性高职院校软件技术专业教学改革经验及达内集团在 IT 培训行业十多年的经验编写而成。

本书注重 Java 基础教学部分，全书分为 10 章，包括 Java 开发环境和语言基础、Java 语法基础、选择结构、循环结构、数组、结构化程序设计、Java 面向对象编程、Java 核心 API、异常处理、集合等内容。每一章都由基础知识、阶段案例、练习总结等模块组成，通过不同模块间的互相辅助来搭建 Java 程序设计基础所需掌握的知识结构。

与本书配套的数字课程已在“智慧职教”（www.icve.com.cn）网站上线，学习者可登录网站进行学习；也可通过扫描书中二维码观看教学视频，详见“郑重声明”页资源服务提示。

本书可作为应用型本科院校、高等职业院校软件技术专业 Java 课程的配套教材，也可作为 Java 开发爱好者的学习和参考用书。

Java

III 丛书编委会

主任:

韩少云 达内教育集团

副主任: (以姓氏笔画为序)

王世来 浙江海洋学院东海科学技术学院

叶时平 树人大学

刘益和 内江师范学院

张平安 深圳信息职业技术学院

张志凌 牡丹江大学

赵宝江 佳木斯大学

秦 锋 安徽工业大学

凌 云 浙江工商大学

常中华 青岛职业技术学院

眭碧霞 常州信息职业技术学院

软件技术专业委员: (以姓氏笔画为序)

王 军 沈阳化工大学

王国贞 河北工业职业技术学院

王桂霞 内蒙古师范大学

王渝俊 四川科技职业技术学院

王 婷 湘潭大学

邓曦辉 晋中学院

甘诚志 宜春学院

吕振雷 黄河水利职业技术学院

朱 强 中原工学院信息商务学院

刘中华 辽宁机电职业学院

刘庆杰 防灾科技学院

刘彦宾 遵义师范学院

许海成 红河学院

李文斌 河北地质大学

李存华 淮海工学院

李庆年 南宁学院

李 林 重庆电子工程职业学院

李 忠 宜宾学院

杨 喜 吉首大学

何 淼 南京信息职业技术学院

张 伟 枣庄学院

张 婷 唐山学院

陈 哲 商丘职业技术学院

陈 琳 长江大学

邵雪航 黑龙江工商学院

范剑波 宁波工程学院

赵向军 江苏师范大学

赵 芳 新乡学院

赵竞雄 华北科技学院

姚 湘 无锡太湖学院

耿焕同 南京信息工程大学滨江学院

徐庆生 楚雄师范学院

黄先珍 菏泽学院

黄星寿 河池学院

商信华 信阳农林学院

梁 丰 浙江万里学院

程世辉 河南教育学院

程曾平 湖北汽车工业学院

焦 战 辽宁轻工职业学院

赖俊峰 内蒙古工业大学

谭 靖 华北理工大学轻工学院

魏海平 辽宁石油化工大学

III 总序

一直以来，我们都在探索信息技术（IT）教学的新方法。如何把看似复杂深奥的编程知识教给完全没有基础的学员，让编程不再难学，这是我们在过去的十年中一直追寻的目标。为此，我们尝试了一系列的方法去为零基础的学员讲解编程技术。最初，我们以知识点为导向，将编程技术分解为若干知识点，力图使我们的课程完整地覆盖所有的知识点，但我们发现这样的培训方式收效甚微，大多数人能背下这些知识点，但不能灵活地运用它们。我们知道，要编写一个完整的程序涉及方方面面的知识，也就是说知识其实是张网。后来，我们又希望能用案例和项目贯穿课程的始终。使用案例来讲解知识点，能很好地帮助学员理解和运用知识，但我们同时也发现，案例的设计和难度的把控，以及与知识点的契合程度很难把握。我们似乎找到了教学的法门，但却又不知如何去运用好这个法门。

最后，我们放弃了一切已知的思维定势，而是从最根本的人本认知需求出发。我们认为，良好的教育应该是以人为本的教育，是发现知识和运用知识的过程。以人为本就是以人的认知曲线为依据，跟随人们发现知识的过程来设计案例。经过长期的实践和不断的摸索，我们整理了一套基于认知曲线的案例教学法。我们认为，一个初学编程的人的认知曲线分为单词、语句、小段逻辑、完整程序等几个认知阶段，而且在不同的认知阶段，学员的关注点不同。如何将学员的关注点与要学习的知识有机地结合起来，我们想到了按照学员的认知曲线去设计这些案例，这才是符合人的学习规律的，是以人为本的课程设计。

这套教材正是基于我们这一教学方法的具体产物。我们希望通过一系列按需设计的案例的贯穿，完整地完成一个有一定技术价值的程序，通过这一系列的程序来学习和理解知识的运用过程。因此，本套教材在编写初期就采用了人本认知的方式对所有的知识点进行了重构，所有章节内容的讲解方式一律遵循“知识介绍—知识运用—案例实战—知识总结”这样一套学习方案。读者通过浏览目录会发现，在每个章节中都存在一个或多个以【案例】标示的节点，每一个案例都可延伸拓展为一个应用程序。而在全书的知识点讲解中还涉及大量的代码段的讲解，这其中又存在多个小案例，这些小案例的数量多达几十个。这样的小案例也方便读者对知识有更加全面的认识。这些大大小小的案例都是经过精心设计的。我们坚信，通过这样一种案例式驱动的教学方式，既可以摆脱传统知识点学习的枯燥，又可以很快地让读者掌握每一个知识点。

本系列教材的另一个突出特色是，作为“用微课学”系列教材，我们为每一个章节中的重要知识点录制了一段微课视频。这些视频都是我们在实际授课中的经验总结和教学方法呈现。我们希望这些微课视频能够帮助读者更进一步准确理解所学知识。同时，本系列教材还提供了整体设计、单元设计、电子课件、项目案例、源代码、工程资源等丰富的电子资源供读者使用。



III 前言

Java 是一种可以撰写跨平台应用程序的面向对象的程序设计语言。Java 技术具有卓越的通用性、高效性、平台移植性和安全性，是一门真正做到“一次编译，到处运行”的高级语言。多年来，Java 语言一直深受计算机开发者的喜爱，因此无论是大学生还是行业领军者，都熟悉 Java 语言。目前，在全球云计算、大数据、移动互联网迅猛发展的产业环境下，Java 语言更具备了显著优势和广阔前景。

本套书分为《Java 开发基础》和《Java 高级开发》两本。《Java 开发基础》分为 10 章，内容从 Java SE 基础知识讲起，适用于在校大学生、编程爱好者和零基础学者。具体包括 Java 开发环境和语言基础、Java 语法基础、选择结构、循环结构、数组、结构化程序设计、Java 面向对象编程、Java 核心 API、异常处理、集合。每一章均包含基础知识、阶段案例、练习、总结等内容并配套若干微课视频。

《Java 高级开发》内容包括 Java 桌面图形界面开发、Java IO、Java 多线程、Java 网络编程、JDBC、XML 和 JSON、Java 反射和注释等。

通过基础部分和高级部分内容的学习，读者可全面掌握 Java 开发技术，独立开发 Java 相关应用。

本套书以介绍 Java 开发基础和培养 Java 开发基本技能为目标，注重 Java 开发基础技术的应用。读者通过对本套书的学习，可以强化分析问题和解决问题的能力，激发创新实践兴趣，真正做到学以致用。

本书凝聚了我们多年的教学和实践经验，但由于水平有限，疏漏之处在所难免。欢迎广大读者提出宝贵意见。

编 者

2016 年 2 月 19 日

Java

III 目录

第 1 章 Java 开发环境和语言基础	1
1.1 Java 简介	2
1.1.1 Java 的历史	2
1.1.2 3G 时代对 Java 的影响	2
1.1.3 Java 平台	3
1.1.4 Java 的特点	4
1.2 搭建 Java 开发环境	5
1.2.1 JDK 简介及安装	5
1.2.2 下载 JDK API 文档	6
1.2.3 配置环境变量	7
1.2.4 下载 Eclipse 开发环境	9
1.3 编写第一个 Java 程序	10
1.3.1 新建工程	10
1.3.2 新建包和类	12
1.3.3 定义 main 方法	13
1.3.4 错误和异常的调试	14
本章小结	15
第 2 章 Java 语法基础	17
2.1 数据类型	18
2.1.1 不同的数据类型	18
2.1.2 Java 常用数据类型	18
2.2 Java 基本类型之间的转换	21
2.2.1 为什么要使用数据类型转换	21
2.2.2 基本数据类型转换方式	21
2.2.3 数值运算时的自动类型转换	24
2.3 变量	25
2.3.1 变量的含义	25
2.3.2 变量命名规则	26
2.3.3 变量的访问	27
2.4 运算符	28
2.4.1 赋值运算符	28
2.4.2 算术运算符	29
2.4.3 关系运算符	31
2.4.4 逻辑运算符	32
2.4.5 位运算符	34
2.4.6 条件（三目）运算符	37
2.4.7 运算符优先级	37
2.5 从控制台输入数据	38
2.6 产生随机数据	40
2.7 【案例】String 类的应用	41
2.7.1 案例描述	41
2.7.2 案例分析	41
2.7.3 案例代码	41
2.8 【案例】某航空公司航班班期查询程序	42
2.8.1 案例描述	42
2.8.2 案例分析	42
2.8.3 案例代码	42
2.9 【案例】摄氏温度与华氏温度的转换	42
2.9.1 案例描述	42
2.9.2 案例分析	43
2.9.3 案例代码	43
本章小结	44
第 3 章 选择结构	45
3.1 if 条件选择结构	46
3.1.1 基本 if 条件结构	46
3.1.2 多重 if 条件结构	48
3.1.3 嵌套 if 条件结构	51
3.2 switch 条件选择结构	52
3.3 【案例】出租车计费程序设计	55
3.3.1 案例描述	55
3.3.2 案例分析	55
3.3.3 案例代码	55
3.4 【案例】某电信公司手机通话计费程序	57

3.4.1 案例描述	57	4.5.1 几种循环的使用场景	74
3.4.2 案例分析	57	4.5.2 嵌套循环	75
3.4.3 案例代码	58	本章小结	76
3.5 【案例】个人所得税计算	59	第 5 章 数组	79
3.5.1 案例描述	59	5.1 数组的定义和使用	80
3.5.2 案例分析	59	5.1.1 什么是数组	80
3.5.3 案例代码	59	5.1.2 数组的定义	80
3.6 【案例】学生成绩等级判断	61	5.1.3 数组的初始化	81
3.6.1 案例描述	61	5.1.4 数组的访问	83
3.6.2 案例分析	61	5.1.5 二维数组的定义与初始化	86
3.6.3 案例代码	61	5.2 数组排序问题的研究	87
本章小结	62	5.2.1 冒泡排序	87
第 4 章 循环结构	63	5.2.2 插入排序	88
4.1 循环结构	64	5.2.3 Arrays.sort() 排序	88
4.1.1 什么是循环	64	5.2.4 3 种排序算法的效率比较	89
4.1.2 while 循环	64	5.3 【案例】输出唐诗	89
4.1.3 do-while 循环	66	5.3.1 案例描述	89
4.1.4 for 循环	67	5.3.2 案例分析	90
4.1.5 break 和 continue 的使用	70	5.3.3 案例代码	90
4.2 【案例】求多项式的和值	71	5.4 【案例】扫雷游戏的随机布雷程序	91
4.2.1 案例描述	71	5.4.1 案例描述	91
4.2.2 案例分析	72	5.4.2 案例分析	91
4.2.3 案例代码	72	5.4.3 案例代码	92
4.3 【案例】求 1 ~ 100 以内的所有素数	72	5.5 【案例】扫雷游戏的统计地雷数程序	93
4.3.1 案例描述	72	5.5.1 案例描述	93
4.3.2 案例分析	72	5.5.2 案例分析	93
4.3.3 案例代码	73	5.5.3 案例代码	94
4.4 【案例】猜数字游戏	73	5.6 【案例】扑克牌游戏的随机发牌程序	95
4.4.1 案例描述	73	5.6.1 案例描述	95
4.4.2 案例分析	73	5.6.2 案例分析	95
4.4.3 案例代码	73	5.6.3 案例代码	96
4.5 循环扩展	74	本章小结	97

第 6 章 结构化程序设计	99
6.1 方法的定义与调用.....	100
6.1.1 方法的定义.....	100
6.1.2 方法的调用.....	101
6.2 方法的调用分析.....	101
6.2.1 方法调用的过程分析.....	101
6.2.2 方法调用的内存分析.....	102
6.3 方法调用的特殊案例.....	103
6.4 【案例】成绩统计程序.....	104
6.4.1 案例描述.....	104
6.4.2 案例分析.....	105
6.4.3 案例代码.....	105
6.5 【案例】酒店前台管理程序.....	106
6.5.1 案例描述.....	106
6.5.2 案例分析.....	106
6.5.3 案例代码.....	106
6.6 【案例】计算一手扑克牌的级别.....	107
6.6.1 案例描述.....	107
6.6.2 案例分析.....	107
6.6.3 案例代码.....	108
6.7 【案例】扫雷游戏的递归排雷程序.....	109
6.7.1 案例描述.....	109
6.7.2 案例分析.....	109
6.7.3 案例代码.....	109
本章小结.....	110
第 7 章 Java 面向对象编程	111
7.1 面向对象.....	113
7.1.1 对象.....	113
7.1.2 面向对象简介.....	113
7.1.3 类的抽象.....	113
7.1.4 对象的创建.....	114
7.1.5 对象属性的自动初始化.....	116
7.2 构造器.....	117
7.2.1 Java 中的构造器.....	117
7.2.2 对象实例化的过程.....	117
7.2.3 this 关键字.....	119
7.2.4 构造方法重载.....	120
7.2.5 默认构造器.....	120
7.3 对象中的方法.....	121
7.3.1 依赖对象存在.....	121
7.3.2 方法重载.....	122
7.3.3 方法签名.....	123
7.3.4 方法调用原则.....	123
7.3.5 方法中参数的传递规则.....	124
7.4 【案例】圆.....	129
7.4.1 案例描述.....	129
7.4.2 案例分析.....	129
7.4.3 案例代码.....	129
7.5 访问修饰符.....	130
7.6 继承.....	132
7.6.1 继承的定义.....	132
7.6.2 方法重写.....	133
7.6.3 继承中多态的体现.....	134
7.6.4 引用类型转换.....	135
7.6.5 继承中的构造器.....	136
7.6.6 对象的实例化过程.....	137
7.7 Java Bean 规范.....	139
7.8 static 关键字.....	140
7.8.1 static 修饰属性.....	140
7.8.2 static 修饰方法.....	142
7.8.3 静态代码块.....	143
7.9 final 关键字.....	143
7.9.1 final 修饰类.....	143
7.9.2 final 修饰方法.....	144

- 7.9.3 final 修饰变量 144
- 7.9.4 final static 145
- 7.10 抽象类和接口 145
 - 7.10.1 抽象类的定义 145
 - 7.10.2 接口的定义 147
- 7.11 接口的意义 147
 - 7.11.1 对单一继承关系的补充 147
 - 7.11.2 降低系统耦合 148
 - 7.11.3 面向接口编程 149
- 7.12 【案例】栈的抽象 149
 - 7.12.1 案例描述 149
 - 7.12.2 案例分析 149
 - 7.12.3 案例代码 149
- 7.13 【案例】人类的抽象 150
 - 7.13.1 案例描述 150
 - 7.13.2 案例分析 150
 - 7.13.3 案例代码 150
- 7.14 【案例】即时战略游戏的类结构设计 151
 - 7.14.1 案例描述 151
 - 7.14.2 案例分析 152
 - 7.14.3 案例代码 152
- 7.15 内部类 152
 - 7.15.1 成员内部类 152
 - 7.15.2 静态内部类 153
 - 7.15.3 局部内部类 153
 - 7.15.4 匿名内部类 154
- 7.16 【案例】使用内部类构建链表 154
 - 7.16.1 案例描述 154
 - 7.16.2 案例分析 155
 - 7.16.3 案例代码 155
- 本章小结 156

- 第 8 章 Java 核心 API 157
 - 8.1 String 字符串 158
 - 8.1.1 String 基础知识 158
 - 8.1.2 String 是 final 类型 160
 - 8.1.3 String 中常用的 API 161
 - 8.1.4 StringBuffer 类和 StringBuilder 类 163
 - 8.2 Object 类 165
 - 8.2.1 什么是 Object 165
 - 8.2.2 Object 类中的方法 165
 - 8.3 Java 中的日期类 167
 - 8.3.1 Date 类 167
 - 8.3.2 Calendar 类 169
 - 8.3.3 日期的输入与输出 170
 - 8.4 包装类 171
 - 8.4.1 包装类的认识 171
 - 8.4.2 自动装箱和拆箱 173
 - 本章小结 174
- 第 9 章 异常处理 175
 - 9.1 异常概述 176
 - 9.2 异常处理 177
 - 9.2.1 Java 异常类说明 177
 - 9.2.2 异常处理机制 178
 - 9.3 异常处理语句 178
 - 9.3.1 try/catch 语句块 178
 - 9.3.2 多个 catch 179
 - 9.3.3 finally 语句块 180
 - 9.4 throws 关键字 182
 - 9.5 Exception 常用的 API 183
 - 9.5.1 printStackTrace 183
 - 9.5.2 getMessage 183

9.5.3	getCause	183	10.3.3	使用增强型 for 循环遍历集合元素	198
9.6	throw 关键字	183	10.4	Set 集合	199
9.7	自定义异常类	185	10.5	List 集合	201
9.8	异常处理规则	186	10.5.1	List 接口	201
9.8.1	不要过度使用异常	186	10.5.2	Iterator 接口	204
9.8.2	不要使用过于庞大的 try 块	187	10.5.3	ArrayList 类	205
9.8.3	避免使用 Catch All 语句	188	10.5.4	LinkedList 类	205
9.8.4	不要忽略捕获到的异常	188	10.6	【案例】贪食蛇	206
本章小结		189	10.6.1	案例描述	206
第 10 章 集合		191	10.6.2	案例分析	207
10.1	集合框架概述	192	10.6.3	案例代码	207
10.2	Collection 集合	192	10.7	Map 集合	210
10.2.1	add 方法	192	10.7.1	Hashtable 类	211
10.2.2	contains 方法	193	10.7.2	HashMap 类	211
10.2.3	size、clear 和 isEmpty 方法	194	10.8	泛型	212
10.2.4	addAll 和 containsAll 方法	194	10.8.1	泛型概念的提出	212
10.3	Iterator	196	10.8.2	什么是泛型	213
10.3.1	hasNext、next 方法	196	10.8.3	自定义泛型接口、类和方法	215
10.3.2	remove 方法	197	10.8.4	类型通配符	216
			本章小结		219

第 1 章 Java 开发环境和语言基础



本章重点

本章主要介绍程序以及 Java 的发展历史；讲解计算机语言的运行机制，并介绍 Java 的运行机制；介绍 JRE 的概念和它的安装，并以一个简单的 MyFirstJava 为例，介绍 Java 应用程序编写的基本方法和使用 Eclipse 工具开发 Java 程序的基本流程；介绍 Java 初学者常常会犯的一些错误，以及相应的原因和解决方法；最后，介绍 Java 程序中的注释以及编码规范。



本章资源

1. 文本 课程标准
2. 文本 课程整体设计
3. 文本 第 1 章 章节设计
4. PPT 第 1 章 Java 开发环境和语言基础
5. 微视频 001 搭建 Java 开发环境
6. 微视频 002 Eclipse 创建第一个 Java 程序



1.1 Java 简介

1.1.1 Java 的历史

Java 语言的历史可以追溯到 1991 年。当时，Sun 公司（Sun Microsystems）成立了一个 Green 的项目组，致力于数字家电之间的通信和协作。James Gosling（Java 之父）作为该项目的负责人。面对 Green 计划，Gosling 需要开发一种全新的语言，该语言必须简洁、健壮。更重要的是，该语言可以屏蔽硬件设备之间的差别从而使代码具备更好的可移植性，毕竟相比于 PC（个人计算机）平台而言，数字家电设备具有更高的平台差异性。于是，Java 语言应运而生，不过它最初的名字并不是 Java 而是 Oak（橡树）。

Java 语言真正被世人了解是从 1995 年开始的。互联网技术的高速发展对 Java 语言起了巨大的推动作用。Java 语言迅速成为使用广泛的编程语言。从这个时候起，Java 语言所涉及的领域已经远远超越了当初设计它的目标。Java 已经由单纯的语言成长为通用的平台技术标准。

Java 是开放的技术。Sun 公司于 1998 年成立了 JCP（Java Community Process）。JCP 是一个开放的国际组织，用来维护和发展 Java 技术规范。JCP 成员可以提交 JSR（Java Specification Requests）——“Java 规范请求”，通过特定的程序，经 JCP 执行委员会（Executive Committee）批准后，可以正式纳入到下一个版本的 Java 规范中。

JCP 分为两个执行委员会：一个负责 Java SE 和 Java EE 方面（SE/EE EC），另一个负责 Java ME 方面（ME EC）。每个执行委员会投票成员的服务期限为 3 年；其中有 10 个批准席位、5 个开放席位，还有 1 个固定席位，该席位以前归 Sun 公司所有，现在则变成了 Oracle。

在 Java 开放社区的努力下和服务器及软件提供商的支持下，Java 技术走过了蓬勃发展的 16 年，Java 语言是在基础性的 C 语言之外拥有程序员最多、使用最广泛的语言。如图 1-1 所示为 TIOBE 开发语言排行榜（2015 年年初更新数据）。TIOBE 开发语言排行榜（<http://www.tiobe.com>）每月更新一次，依据的指数由世界范围内的资深软件工程师和第三方供应商提供，其结果作为当前业内程序开发语言的流行使用程度的有效指标。

2015年4月	2014年4月	同期比	编程语言	市场份额	同期比
1	1		C	16.642%	-0.89%
2	2		Java	15.580%	-0.83%
3	3		Objective-C	6.688%	-5.45%
4	4		C++	6.636%	+0.32%
5	5		C#	4.923%	-0.65%
6	6		PHP	3.997%	+0.30%
7	9	▲	JavaScript	3.629%	+1.73%
8	8		Python	2.614%	+0.59%
9	10	▲	Visual Basic .NET	2.326%	+0.46%

图 1-1 编程语言应用排行榜

1.1.2 3G 时代对 Java 的影响

3G（指第三代移动通信技术）时代的来临，意味着手机将享受更高速的数据传输服务，

可以更好地实现在全球范围内的无线漫游，并在处理图像、音乐、视频流等多媒体数据能力方面有着显著的提升，可以更好地支持包括在线浏览、电子邮件、即时通信、全球定位、电话会议、电子商务等多种信息服务。3G 时代造就了“移动商务应用领域”。这是对“传统商务应用”的扩展，将为企业带来巨大的商机。

3G 行业兴起为商业应用开创了一个新的时代；这是继 PC 时代、互联网时代之后的又一个 IT 从业人员的“黄金时代”。在互联网时代，Java 语言已经是使用广泛的服务器端语言。随着 3G、物联网时代的到来，Java 语言并不会“过时”，相反 Java 语言会在新的业务领域有着更辉煌的发展前景。

2007 年，Google 推出了开放的智能移动操作系统——Android（安卓）。在 Google 及摩托罗拉、HTC、三星、索爱、LG 等主流手机厂商的鼎力推动下，Android 迅速成长为流行的移动智能平台。Android 通过内置虚拟机技术支持 Java 并提供了完善的基于 Java 的应用层开发框架。从此，Java 语言日趋成为 3G 时代的主流开发语言。

1.1.3 Java 平台

1999 年，Sun 公司发布了基于 Java 的 3 个平台技术标准：J2SE、J2EE 和 J2ME（2005 年之后它们分别被更名为 Java SE、Java EE 和 Java ME），Java 从此迎来了属于自己的时代。

Java SE（Java Platform, Standard Edition），称之为“Java 平台标准版”，是 Java 平台的基础。Java SE 包含了运行 Java 应用所需要的基础环境和核心类库。除此之外，Java SE 还定义了基于桌面应用的基础类库，通过使用这些类库，可以编写出类似于 Office 那样丰富多彩的桌面应用。

Java EE（Java Platform, Enterprise Edition），称之为“Java 平台企业版”。Java EE 构建在 Java SE 基础之上，用于构建企业级应用。所谓企业级应用是指那些为商业组织、大型企业而创建的应用系统。例如，电信的“计费系统”、银行的“网银系统”、企业中的“客户关系管理系统”等。这些系统与个人使用的单机桌面系统不同，它们部署、运行在结构复杂的服务器环境中；往往需要处理海量的数据；需要遵守通用的数据传输协议和数据表示；需要维护复杂而多变的业务逻辑；需要应对巨大的用户访问量、必须具备可靠的安全性和健壮性。Java EE 为了解决企业应用中需要解决的各种问题提供了众多组件标准和服务规范，如 Servlet/JSP 和 EJB。

Java ME（Java Platform, Micro Edition），称之为 Java 平台微型版。Java ME 为机顶盒、移动电话和 PDA 等嵌入式消费电子设备提供的 Java 解决方案。也许只有 Java ME 才符合 Java 语言创建时的初衷。随着 3G 移动智能平台时代的到来，特别是以 Java 为核心编程语言的 Android 智能平台的迅速普及，Java ME 未来的发展似乎不容乐观。

绝大多数 Java 初学者都是从 Java SE 开始入手，从官方网站下载 Java SE 发行版，然后根据文档资料了解 Java 的基本运行原理，学习各种各样的类库。例如，如何使用 Java 语言处理字符串、如何处理日期、如何应用 Java SE 提供的各种数据结构编写算法较为复杂的程序、如何访问数据库、如何读写 XML 数据、如何应用各种安全策略等。待这些必备的基础掌握之后，他们其中的很多人并不会继续研究 Java SE 中所涉及的 Java 桌面应用开发

部分，而是转向了市场前景更广阔的 Java EE 方向。

1.1.4 Java 的特点

Java 的语言特点一般描述为：简单、面向对象、跨平台、安全以及多线程。

1. 简单

设计 Java 的初衷是为了能构建一个无需深奥的专业训练就可以进行编程的系统，但也要符合一定的编程标准惯例，所以 Java 在设计上尽可能地接近当时的 C++，但同时又剔除了 C++ 中很少使用、难以理解的和易混淆的一些特性（比如头文件、指针、操作符重载等语法与操作），Java 语法更像 C++ 语法的一个“纯净”版本。

2. 面向对象

面向对象设计是一种程序设计技术。它将重点放在对象（即数据）和对象接口上。例如，一个“面向对象”的木匠始终关注的是椅子，其次是使用的工具，一个“非面向对象”的木匠首先会考虑使用什么工具。本质上，Java 面向对象的能力与 C++ 是一样的。Java 与 C++ 的主要不同点在于多继承，Java 中取而代之的是较简单的接口概念。相对于传统的面向过程语言（C、Basic 和 Pascal 等），面向对象程序设计语言（C++、Java 和 C# 等）在实现大型复杂项目时更加有效。

面向对象的基本思想是从现实世界中客观存在的事物（即对象）出发来构建软件系统，并在系统中尽可能地应用人类的自然思维方式，强调以事物为中心来思考问题和认识问题，并根据这些事物的本质特点，把它们抽象地表示为系统中的类，作为系统中的基本构成单元，使客观世界的事物在计算机系统中保持相互关联的本来面貌。

抽象、继承、封装、多态是面向对象方法的 4 个基本特征。

（1）抽象：是将现实世界中的事物描述为系统中类、对象及方法的过程，在这个过程中去除了不相关数据和信息，保留的数据用来实现系统特定功能。

（2）继承：是面向对象实现软件复用的重要手段，利用继承，人们可以基于已存在的类构建一个新类，子类继承已存在的类就是复用父类的方法和属性。除此之外，子类还可以添加一些新的方法和属性来满足新的需求。

（3）封装：是将对象的实现细节隐藏起来（用户无需知道这些细节），只提供一些公共的方法将对象的功能展现出来。

（4）多态：多态使 Java 更有生命和鲜活起来。多态指子类对象可以直接赋值给父类变量，但在运行时依然表现出子类的特征。Java 引用变量有两种类型，分别是编译时类型和运行时类型，编译时类型由声明类型决定，运行时类型由赋值对象的类型决定。如果编译时类型和运行时类型不一致，就会出现所谓多态。

3. 跨平台

Java 通过 JVM（Java 虚拟机）实现了跨平台技术，Java 源代码（*.java）经过 Java 的编译器编译成 Java 字节码（*.class），执行 Java 字节码，Java 字节码经过 JVM 的解释为具体平台的具体指令并执行。不同平台有不同的 JVM，主流平台都提供了 JVM（比如

Windows、UNIX、主流手机操作系统)。所以 Java 字节码可以在所有平台上解释执行。在这个意义上 Java 是跨平台的。也就是说, Java 的字节码是跨平台的。

Java 是跨平台的, JVM 不是跨平台的(需要强调的是,没有 JVM, Java 是不能运行的)。

Java 的设计初衷是“一处编译,四处运行”,通过在 JVM 中运行编译好的 .class 文件屏蔽掉操作系统之间的差异,从而实现跨平台。但现实是因为一些系统间的差异,Java 没有实现完全的跨平台(比如当 Windows 系统开发的程序移植到 Linux 上时会有一些小 Bug),因此有人戏称 Java 的跨平台是“一次编译,到处调试”。

4. 安全

Java 不支持指针,避免了指针操作错误及欺骗访问;严格的编译和字节码装载于检验机制。在任何情况下,蒙骗 Java 的安全机制都十分困难,现在发现 Bug 的技术越来越强,从一开始 Java 就设计成能够防范各种袭击,其中包括:

- (1) 运行时堆栈溢出(蠕虫等病毒常用的袭击手段)。
- (2) 在自己处理空间之外破坏内存。
- (3) 未经授权读写文件。
- (4) 许多安全特性也不断加入 Java 中。

5. 多线程

相比较其他语言编写多线程应用,Java 多线程处理更具魅力的是它的便捷性和简单性。

所有软件都具有多线程(比如,打开 QQ 软件可以同时与许多人一起聊天,而且互不影响,这称为并发,实际上就是多线程)。

以上知识点在初学阶段无需强行记忆,但是在找工作阶段会以面试笔试题形式出现,所以要注重逐渐理解。

1.2 搭建 Java 开发环境

1.2.1 JDK 简介及安装

1. JDK 简介

JDK 是 Java 语言的软件开发工具包,主要用于移动设备、嵌入式设备上的 Java 应用程序。目前的最新版本是 Java SE 8.0(1.8.0),其结构如图 1-2 所示。

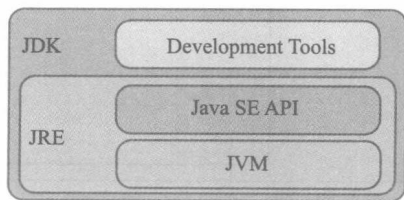
(1) JDK (Java Development kit), 是 Java 开发工具包。

(2) JRE (Java Runtime Environment), 是 Java 运行时环境。

(3) JVM (Java Virtual Machine), 是 Java 虚拟机。

Java SE API, 就是 Java 系统核心类库(拿来即用,包括网络、I/O、GUI 等)。

在整个 JDK 中, JVM 是不跨平台的,其余都是跨平台的(即所有 .class 文件是跨平台的),如果仅运行 Java 程序,则只需要在用户系统上安装 JRE 即可;如果还要开发,则需要完整安装 JDK。



微视频 001
搭建 Java 开发环境

图 1-2
JDK 结构