

少年人工智能教育课堂

Java 程序设计 第一课

EASY JAVA PROGRAMMING

赵军 编著



多媒体视频教学
和扫码阅读

- >>> 难易适中，循序渐进
- >>> 范例多样，完整解析
- >>> 图文并茂，优化学习
- >>> 综合范例，活用语法
- >>> 轻松上手，编写自己的
第一行代码



清华大学出版社

Java 程序设计 第一课



赵军 编著

清华大学出版社
北京

内 容 简 介

本书面向中学生和初学者，书中的内容以基础语法为主，再导入一些简单的面向对象基本概念，希望读者可以通过有趣、多样的范例程序轻松学会Java程序设计语言。

本书共分7章，内容包括初识Java语言、Java的数据处理、运算符与表达式、条件式流程控制、循环流程控制、数组与字符串、Java的类方法。本书提供完整的范例程序代码，希望降低初学者学习的障碍。另外，各章结束前也提供了综合范例程序的练习机会，希望能帮助读者活用各章所传授的语法。各章的最后还安排了课后习题，可用于检验学习成效。

本书封面贴有清华大学出版社防伪标签，无标签者不得销售。
版权所有，侵权必究。侵权举报电话：010-62782989 13701121933

图书在版编目(CIP)数据

Java程序设计第一课/赵军编著.—北京：清华大学出版社，2018
ISBN 978-7-302-51094-9

I. ①J… II. ①赵… III. ①JAVA语言—程序设计 IV. ①TP312.8

中国版本图书馆CIP数据核字(2018)第195630号

责任编辑：夏毓彦

封面设计：王翔

责任校对：闫秀华

责任印制：丛怀宇

出版发行：清华大学出版社

网 址：<http://www.tup.com.cn>，<http://www.wqbook.com>

地 址：北京清华大学学研大厦A座

邮 编：100084

社总机：010-62770175

邮 购：010-62786544

投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn

质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn

印 装 者：北京密云胶印厂

经 销：全国新华书店

开 本：190mm×260mm

印 张：13

字 数：291千字

版 次：2018年10月第1版

印 次：2018年10月第1次印刷

定 价：49.00元

产品编号：079692-01

前言

人工智能技术的未来就是信息技术的未来，而“程序设计”或称为“编程”是学习人工智能技术最重要的基础工具，从小建立编程逻辑思维，通过编程实践培养解决问题的能力，是将来人才综合素质的评估条件之一。程序设计已经列入中学的信息技术课程，即便是非计算机或信息类专业的人才，编程也是必备的基础能力之一。

Java 是新一代面向对象的高级程序设计语言，具有“支持 Web”的功能，常用于专业级 Web 应用的开发和移动应用的开发，并成为企业构建 Web 数据库的最佳开发工具之一。只有当我们具备了逻辑编程的坚实基础和通过编程实践来解决问题的能力，才能进一步学习人工智能的知识理论类的课程、应用拓展类课程、创新设计类的课程。

市面上关于 Java 程序设计的书琳琅满目，主要编写的内容通常适用于大专院校的程序设计语言课程，这类书的定位不太适合中学生和初学者。而本书编写的内容则是面向中学生和初学者的，在这种思路的指导下，本书讲述的内容以基础语法为主，再导入一些简单的面向对象基本概念，希望学习者可以通过有趣、多样的范例程序轻松学会 Java 程序设计语言。

再者，市面上有关 Java 的书注重理论讲解的偏多，在完整范例程序的考虑上稍显不足，对初学者而言，需要让读者有更多实际动手进行程序设计的机会。因此，本书提供完整的范例程序代码，希望降低初学者学习的障碍。另外，各章结束前也提供了综合范例程序的练习机会，希望能帮助读者活用各章所传授的语法。各章的最后也安排了课后习题，可用于检验学习成效。

本书非常适合作为学习 Java 的入门教材，笔者希望初学者可以通过本书快速学懂 Java 程序设计语言，迈出运用功能强大的 Java 语言编程的第一步。

读者可以从以下地址（注意区分数字与英文字母大小写）下载所有范例程序的源代码、教学 PPT 和全程视频文件：

<https://pan.baidu.com/s/1Ejom12HKrEebk3evKz8nmg>

也可以扫描右方的二维码下载，如果下载有问题，或者对本书有任何疑问与建议，请联系 booksaga@126.com，邮件主题为“Java 程序设计第一课”。



最后，为了便于读者和同学们在学习中进行讨论和交流，编者建立了“程序设计第一课讨论群”（QQ 群），大家可以在群里提出问题互相讨论，编者将会对重点问题进行解答。QQ 群号为 801630455，也可以直接扫描进群群的二维码加入：



本书主要由赵军编著，参与本书编写的人员还有张明、王国春、施妍然、王然等。由于编者水平和经验所限，书中可能存在疏漏和不足之处，希望得到大家的批评指正。

编者

2018 年 7 月

目 录

第 1 章 Java 语言

1.1 什么是程序设计语言	2	我的第一个 Java 程序	
1.1.1 机器语言	2	范例程序: CH01_01.java	14
1.1.2 汇编语言	3	1.4.2 程序代码的编译与运行	15
1.1.3 高级语言	3	1.4.3 程序的调试	16
1.1.4 非过程性语言	4	1.5 Java 程序的基本结构	17
1.1.5 人工智能语言	4	1.5.1 main() 函数	17
1.2 面向对象程序设计概念	5	1.5.2 System.out.println 函数与 注释	18
1.3 认识 Java	7	1.6 综合范例程序—— 学生学籍信息的输出	18
1.3.1 Java 的特色与优点	7	学生学籍信息的输出	
1.3.2 Java 环境的下载与简介	9	范例程序: CH01_02.java	19
1.3.3 JDK 的安装与设置	9	本章重点回顾	19
1.3.4 Eclipse 工作环境	11	课后习题	20
1.4 第一个 Java 程序	12		
1.4.1 程序代码的编写	14		

第 2 章 Java 的数据处理

2.1 变量与常数	23	2.1.3 常数	27
2.1.1 变量声明	23	硬盘容量转换程序	
变量声明的实践		范例程序: CH02_02.java	28
范例程序: CH02_01.java	24	2.2 基本数据类型	29
2.1.2 变量的命名规则	26	2.2.1 整数类型	29

整数声明变量	
范例程序: CH02_03.java 30
2.2.2 浮点数类型 32
单精度与双精度浮点数	
范例程序: CH02_04.java 33
浮点数科学记数表示法	
范例程序: CH02_05.java 35
2.2.3 布尔类型 36
声明并设置布尔值	
范例程序: CH02_06.java 36
2.2.4 字符类型 37
三种不同的字符声明方式	
范例程序: CH02_07.java 38
2.3 转义字符 39
转义字符的简单应用	
范例程序: CH02_08.java 39
2.4 格式化输出功能 40
2.4.1 格式化高级输出的设置	... 41
格式化输出的实例	
范例程序: CH02_09.java 42
八进制与十六进制表示法	
范例程序: CH02_10.java 44
2.4.2 从键盘输入数据 45
从键盘读取字符	
范例程序: CH02_11.java 45
Scanner 对象输入的应用	
范例程序: CH02_12.java 47
2.5 数据类型转换 49
摄氏温度与华氏温度转换器	
范例程序: CH02_13.java 50
2.6 综合范例程序 51
转义序列的应用	
范例程序: CH02_14.java 51
本章重点回顾 52
课后习题 53

第 3 章 运算符与表达式

3.1 表达式与运算符 56	3.6 递增与递减运算符 64
3.2 赋值运算符 57	递增与递减运算符的实际应用范例	
3.3 算术运算符 57	范例程序: CH03_04.java 65
币值兑换程序		3.7 位运算符 67
范例程序: CH03_01.java 58	3.7.1 位逻辑运算符 67
3.4 关系运算符 59	3.7.2 位位移运算符 69
关系运算符运算		位运算符的综合运用	
范例程序: CH03_02.java 60	范例程序: CH03_05.java 70
3.5 逻辑运算符 61	3.8 复合赋值运算符 71
关系与逻辑运算符的求值		复合赋值运算符的实际应用	
范例程序: CH03_03.java 63	范例程序: CH03_06.java 72

3.9 条件运算符····· 73	3.11 综合范例程序····· 78
条件运算符的使用	成绩统计小帮手
范例程序: CH03_07.java ····· 74	范例程序: CH03_09.java ····· 78
3.10 运算符优先级····· 75	本章重点回顾····· 79
运算符优先级的实际运用	课后习题····· 80
范例程序: CH03_08.java ····· 76	

第 4 章 条件式流程控制

4.1 流程控制简介····· 83	快餐店点餐程序
4.2 选择结构····· 85	范例程序: CH04_05.java ····· 98
4.2.1 if 条件语句····· 86	分数段判断
消费满额赠送来店礼品	范例程序: CH04_06.java ··· 100
范例程序: CH04_01.java ····· 87	4.3 综合范例程序 1——
4.2.2 if-else 条件语句····· 88	闰年计算器····· 102
奇偶数判断器	闰年计算器
范例程序: CH04_02.java ····· 90	范例程序: CH04_07.java ··· 103
4.2.3 if else if 条件语句····· 92	4.4 综合范例程序 2——
消费额折扣回馈	简易计算器制作····· 104
范例程序: CH04_03.java ····· 93	简易计算机制作
阶梯电价查询程序	范例程序: CH04_08.java ··· 104
范例程序: CH04_04.java ····· 94	本章重点回顾····· 106
4.2.4 switch 选择语句····· 96	课后习题····· 106

第 5 章 循环流程控制

5.1 for 循环····· 109	5.2 while 循环····· 114
数字累加计算	正因数求解
范例程序: CH05_01.java ··· 111	范例程序: CH05_03.java ··· 115
九九乘法表	判断循环执行次数
范例程序: CH05_02.java ··· 113	范例程序: CH05_04.java ··· 117

数字反向输出		求解最大公约数	
范例程序: CH05_05.java ...	119	范例程序: CH05_08.java ...	125
5.3 流程跳离指令	120	5.5 综合范例程序 2 ——	
5.3.1 break 指令	121	密码验证器	126
break 指令的应用		密码验证器	
范例程序: CH05_06.java ...	121	范例程序: CH05_09.java ...	126
5.3.2 continue 指令	122	本章重点回顾	128
使用 continue 指令跳过本轮次的循环		课后习题	129
范例程序: CH05_07.java ...	123		
5.4 综合范例程序 1 ——			
求解最大公约数	125		

第 6 章 数组与字符串

6.1 数组简介	132	6.2.3 字符串数组	148
6.1.1 一维数组	132	字符串数组的应用	
寻找数组中的最大值		范例程序: CH06_07.java ...	148
范例程序: CH06_01.java ...	134	字符串数组与学生成绩的计算	
学生成绩的计算与输出		范例程序: CH06_08.java ...	149
范例程序: CH06_02.java ...	135	6.3 String 类的常见方法	151
6.1.2 二维数组	137	6.4 综合范例程序 1 ——	
二维数组的应用		冒泡排序法	153
范例程序: CH06_03.java ...	138	冒泡排序法	
二阶行列式		范例程序: CH06_09.java ...	155
范例程序: CH06_04.java ...	139	6.5 综合范例程序 2 ——	
6.1.3 多维数组	141	成绩段分布图的制作	156
三维数组的应用范例		成绩段分布图的制作	
范例程序: CH06_05.java ...	142	范例程序: CH06_10.java ...	157
6.2 字符与字符串	144	本章重点回顾	159
6.2.1 字符声明	144	课后习题	160
6.2.2 创建字符串	145		
String 类与字符串			
范例程序: CH06_06.java ...	146		

第 7 章 Java 的类方法

7.1 函数的基本概念	162	7.3 递归函数	175
7.1.1 创建类方法	162	求解 n 阶乘方法	
7.1.2 方法的调用	164	范例程序: CH07_07.java ...	176
“Hello! 我是函数!” 方法		7.4 Math 类的常见方法	178
范例程序: CH07_01.java ...	164	彩票游戏猜猜猜	
方法范例——数字比大小		范例程序: CH07_08.java ...	179
范例程序: CH07_02.java ...	165	7.5 综合范例程序 1——	
7.1.3 参数传递方式	167	汉诺塔游戏	180
传值调用的范例		汉诺塔游戏	
范例程序: CH07_03.java ...	168	范例程序: CH07_09.java ...	181
7.2 数组参数传递	169	7.6 综合范例程序 2——	
7.2.1 一维数组的传递	169	万年历的设计	182
传址调用的范例		万年历的设计	
范例程序: CH07_04.java ...	170	范例程序: CH07_10.java ...	182
7.2.2 多维数组传递	171	本章重点回顾	185
二维数组与参数传递		课后习题	186
范例程序: CH07_05.java ...	171		
三维数组与参数传递			
范例程序: CH07_06.java ...	173		
附录 习题答案	188		

第 1 章

Java 语言

本章重点

- 认识各代程序设计语言的特性
- 了解编译型语言与解释型语言
- 学习面向对象程序设计全新概念
- Java 语言的特色与优点
- Java 环境的建立
- 编写第一个 Java 程序



Java 语言原名为 Oak，源自 1991 年美国 Sun 公司的 GREEN 计划，不过这个计划并未受到消费者的青睐，反而意外地搭上 1993 年因特网兴起的浪潮，于是在 1995 年，Sun 公司正式对外界发表并改名为 Java。初期以能在浏览器上执行的 Java 小程序（Applet）备受瞩目，但随着 Java 开发群体的壮大，Java 也逐渐在各个领域发展壮大。Java 是新一代面向对象程序设计的高级语言，具有“支持 Web”的功能，常被应用于专业级 Web 应用开发和移动应用开发，并成为企业构建 Web 数据库的最佳开发工具。

1.1 什么是程序设计语言



程序设计语言发展的历史已有半个世纪之久，种类不少，如果包括实验、教学或科学研究的相关用途，问世的程序设计语言可能有上百种之多，由早期的机器语言发展至今，已经迈入第 5 代自然语言，不过每种语言都有其发展的背景及目的。

程序设计语言是一种人类用来和计算机沟通的语言，也是用来指挥计算机运算或工作的指令集合，可以将人类的思考逻辑转换成计算机能够了解的语言。每一代语言都有其特色，任何一种语言都有其专用的语法、特性、优点及相关应用的领域。按照其发展演变过程，分类如图 1-1 所示。



图 1-1

下面分别认识各代程序设计语言的特性。

1.1.1 机器语言

机器语言（Machine Language）是早期的程序设计语言，任何程序在执行前都必须被转换为机器语言，由 1 和 0 两种符号构成。机器语言写法如下：

```
10111001（设置变量 A）  
00000010（将 A 设置为数值 2）
```

不过计算机制造商往往因为计算机硬件设计的不同而开发不同的机器语言。这样不但使用不方便、可读性低，也不易于维护，并且不同机器平台的编码方式也不尽相同。

1.1.2 汇编语言

汇编语言（Assembly Language）指令比机器码指令看起来稍有“意义”一些，但与机器语言仍然是一一对应的对应关系，因此与机器语言一样被归类为低级语言，只是它在编写上比机器语言容易多了。每一种系统的汇编语言都不一样，就 PC 而言，用的是 80×86 的汇编语言。例如，MOV 指令代表设置变量内容，ADD 指令代表加法运算，SUB 指令代表减法运算。汇编语言写法范例如下：

```
MOV  A , 2   ( 将变量 A 的数值内容设置为 2 )
ADD  A , 2   ( 将变量 A 加上 2 后，将结果再存回变量 A 中，如 A=A+2 )
SUB  A , 2   ( 将变量 A 减掉 2 后，将结果再存回变量 A 中，如 A=A-2 )
```

1.1.3 高级语言

对一般人来说，纯粹用汇编语言完成一个程序仍然是一件相当困难的事情。所谓高级语言，就是比汇编语言的语句更容易看懂的程序设计语言。高级语言的指令和语句更接近日常生活中常使用的文字或符号，我们编程时所需要做的就是变量声明以及程序流程的控制。例如，Fortran 语言是世界上第一个开发成功的高级语言，更是历久弥新，现在仍有许多研究机构用来解决工程与科学计算上的问题。早期非常流行的 Basic 语言易学易懂，非常适合初学者了解程序语言的工作方式。目前流行的高级语言有 C、C++、Java、Visual Basic、Python。

用高级语言编写而成的程序代码必须经过编译器或解释器翻译为计算机能解读与执行的机器语言，称为执行文件，才能被 CPU 执行。因此从高级语言转换的方式来看，可以分为编译型语言与解释型语言两种。

1. 编译型语言

在程序开始执行前，编译型语言必须使用编译器（Compiler）来将源代码程序转换为机器可读取的可执行文件或目标程序，不过编译器必须先把源代码程序读入主存储器后才可以开始编。编译后的目标程序（object file）可直接对应成机器码，故可在计算机上直接执行，不需要每次执行都重新“翻译”，执行速度自然较快。例如，C、C++、Pascal、Fortran、Java 语言都属于编译型语言。

2. 解释型语言

在程序开始执行前，解释型语言的源代码程序可以通过解释器（Interpreter）将程序一行接一行地读入，逐行“解释翻译”并交由计算机执行，如果在解释的过程中发生错误，就会立刻停止，不会产生目的文件或可执行文件。由于每次执行时都必须再“解释”一次，因此执行速度较慢，效率也较低，例如 Basic、LISP、Prolog、Python 等语言都采用解释执行的方法。

1.1.4 非过程性语言

“非过程性语言”（Non-Procedural Language）也称为第 4 代语言（Fourth Generation Language, 4GL），特点是编程者不必描述数据存储的细节，只需要将步骤写出来，且不必管计算机如何执行，也不需要理解计算机的执行过程，这种语言减轻了用户设计程序的负担。例如，数据库的结构化查询语言（Structural Query Language, SQL）就是一种第 4 代语。SQL 语言写法范例如下：

```
DELETE FROM employees
WHERE employee_id = 'C800312' AND dept_id = 'R01';
```

1.1.5 人工智能语言

人工智能语言称为第 5 代语言，或称为自然语言（Natural Language），是程序设计语言发展的终极目标，为用户提供以一般人类语言的语句直接和计算机进行对话，向计算机发出问题，而不必考虑程序的语法与规则，所以

自然语言必须有人工智能（Artificial Intelligence, AI）技术的发展作为保障。

1.2 面向对象程序设计概念



面向对象程序设计（Object-Oriented Programming, OOP）是一种全新的程序设计概念，主要精神是将存在于日常生活中举目可见的对象（object）概念应用在软件设计的开发模式中，以一种更生活化的设计概念来进行程序设计和软件开发，重点是强调程序代码的可读性（Readability）、可重复使用性（Reusability）与扩展性（Extension）。

什么是对象与类

在现实生活中充满了各种形形色色的物体，每个物体都可视为一种对象。任何面向对象程序设计方法中最主要的单元都是对象（Object），我们可以通过对象的外部行为（behavior）及内部状态（state）模式来进行详细的说明和描述。行为代表此对象对外所显示出来的运行方法，状态则代表对象内部的各种属性，如图 1-2 所示。



图 1-2

对象可以看成是一种抽象概念或具体的东西，其中包括“属性”（Attribute）与相关的“方法”（Method）。“属性”用来描述对象的基本特征与其所属的性质，就是对象的静态外观描述。例如，一个人的属性可能会包括姓名、住址、年龄、出生年月日等，或者是一辆汽车引擎的马力、排气量等。“方法”则是对象的动作与行为，指对象中的动态响应方式，例如车子可以开动、停止、加速、减速等。

通常对象并不会凭空产生，它必须有一个可以依据的原型（Prototype），而这个原型就是一般在面向对象程序设计语言中的“类”（Class）。类是具有相同原型及行为的对象集合，是许多对象共同特征的描述。例如小明与小华都属于“人类”这个类，他们都有出生年月日、血型、身高、体重等类的属性。以汽车为例来说明，汽车有很多品牌，如宝马、奔驰、大众、雪铁龙、丰田等，它们都属于汽车类。

在类中包含“属性”和“方法”，这些属性和方法都可以提供给此类的对象使用。简单地说，对于类与对象之间的关系，可以将“类”看成是“对象”的模型、模块，“对象”则是“类”实际制作后的成品，如图 1-3 所示。当我们创建一个类后，要使用类所定义的属性与方法，必须通过对象，因为类是一个蓝图，对象是蓝图所产生的实例（instance），唯有通过实例才能使用所定义的类。

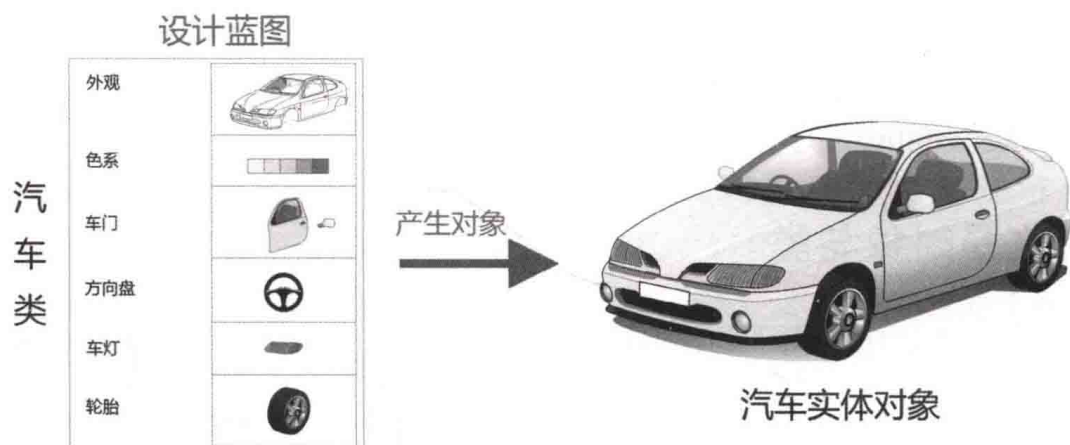


图 1-3

在面向对象程序设计中，可以通过类的继承行为来定义一个新的类以继承现有的类。在继承关系中，被继承者称为“基类”或“父类”，而继承者则称为“派生类”或“子类”，如图 1-4 所示。

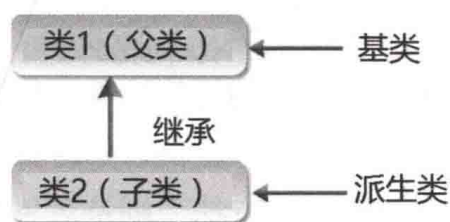


图 1-4

继承（Inheritance）除了可重复使用之前所开发过的类之外，最大的好处在于维持对象的封装性（Encapsulation），因为继承时不容易改变已经设计完整的类，这样可以减少继承时类设计发生错误。

注 意

封装是一种信息隐藏 (Information Hiding) 的重要概念, 也就是将对象的数据和实现的方法等信息隐藏起来, 让用户只能通过方法主题 (Method) 来使用对象本身, 而不能更改对象里所隐藏的信息。例如, 许多人都了解汽车的内部构造等信息, 却能够通过汽车提供的油门和刹车等轻而易举地驾驶汽车。



1.3 认识 Java

Java 的语法与风格十分接近 C/C++ 语言, 除了保持有 C++ 语言面向对象技术的核心外, 舍弃了 C++ 中容易引起错误的指针, 同时拥有跨平台、面向对象程序设计语言等特性。随着因特网应用程序的发展, Java 语言现在已经超越了 C/C++ 语言, 成为热门的网络开发语言之一, 主要用于因特网系统上应用程序的开发, 范围涵盖因特网、网络通信、电子商务、手机游戏以及智能的通信设备传输系统。

1.3.1 Java 的特色与优点

Java 是一种融合了面向对象程序设计概念的高级语言, 经过多次版本修正、更新后, 逐渐成为一种功能完备的程序设计语言。Java 语言能够如此受欢迎, 主要是因为它拥有相当多的特色, 下面就为大家简单说明 Java 的主要特色与优点。

1. 简单性

Java 语法源于 C++ 语言, 因此它的指令和语法十分简单, 删除了许多不容易理解和容易让人混淆的 C++ 功能, 只要你能了解简单英文单词与语法的概念, 就能进行程序设计并完成运算处理的工作; 同时 Java 语言采用垃圾回收机制 (Garbage Collection), 对于程序中不再使用的资源, 系统会自动释放其占用的内存空间, 减少程序设计者自行管理内存资源不足的困扰。