

Java编程的逻辑 (Java核心技术系列)

作者：马俊昌

Java 核心技术系列

Java编程的逻辑

马俊昌 著

ISBN: 978-7-111-58772-9

本书纸版由机械工业出版社于2018年出版，电子版由华章分社（北京华章图文信息有限公司，北京奥维博世图书发行有限公司）全球范围内制作与发行。

版权所有，侵权必究

客服热线：+ 86-10-68995265

客服信箱：service@bbbvip.com

官方网址：www.hzmedia.com.cn

新浪微博 @华章数媒

微信公众号 华章电子书（微信号：hzebook）

[目录](#)

[读者评论](#)

[前言](#)

[第一部分 编程基础与二进制](#)

[第1章 编程基础](#)

[1.1 数据类型和变量](#)

[1.2 赋值](#)

[1.3 基本运算](#)

[1.4 条件执行](#)

[1.5 循环](#)

[1.6 函数的用法](#)

[1.7 函数调用的基本原理](#)

[第2章 理解数据背后的二进制](#)

[2.1 整数的二进制表示与位运算](#)

[2.2 小数的二进制表示](#)

[2.3 字符的编码与乱码](#)

[2.4 char的真正含义](#)

[第二部分 面向对象](#)

[第3章 类的基础](#)

[3.1 类的基本概念](#)

[3.2 类的组合](#)

[3.3 代码的组织机制](#)

[第4章 类的继承](#)

[4.1 基本概念](#)

[4.2 继承的细节](#)

[4.3 继承实现的基本原理](#)

[4.4 为什么说继承是把双刃剑](#)

[第5章 类的扩展](#)

[5.1 接口的本质](#)

[5.2 抽象类](#)

[5.3 内部类的本质](#)

[5.4 枚举的本质](#)

[第6章 异常](#)

[6.1 初识异常](#)

[6.2 异常类](#)

[6.3 异常处理](#)

[6.4 如何使用异常](#)

[第7章 常用基础类](#)

[7.1 包装类](#)

[7.2 剖析String](#)

[7.3 剖析StringBuilder](#)

[7.4 剖析Arrays](#)

[7.5 剖析日期和时间](#)

[7.6 随机](#)

[第三部分 泛型与容器](#)

[第8章 泛型](#)

[8.1 基本概念和原理](#)

[8.2 解析通配符](#)

[8.3 细节和局限性](#)

[第9章 列表和队列](#)

[9.1 剖析ArrayList](#)

[9.2 剖析LinkedList](#)

[9.3 剖析ArrayDeque](#)

[第10章 Map和Set](#)

[10.1 剖析HashMap](#)

[10.2 剖析HashSet](#)

[10.3 排序二叉树](#)

[10.4 剖析TreeMap](#)

[10.5 剖析TreeSet](#)

[10.6 剖析LinkedHashMap](#)

[10.7 剖析EnumMap](#)

[10.8 剖析EnumSet](#)

[第11章 堆与优先级队列](#)

[11.1 堆的概念与算法](#)

[11.2 剖析PriorityQueue](#)

[11.3 堆和PriorityQueue的应用](#)

[第12章 通用容器类和总结](#)

[12.1 抽象容器类](#)

[12.2 Collections](#)

[12.3 容器类总结](#)

[第四部分 文件](#)

[第13章 文件基本技术](#)

[13.1 文件概述](#)

[13.2 二进制文件和字节流](#)

[13.3 文本文件和字符流](#)

[13.4 文件和目录操作](#)

[第14章 文件高级技术](#)

[14.1 常见文件类型处理](#)

[14.2 随机读写文件](#)

[14.3 内存映射文件](#)

[14.4 标准序列化机制](#)

[14.5 使用Jackson序列化为JSON/XML/MessagePack](#)

[第五部分 并发](#)

[第15章 并发基础知识](#)

[15.1 线程的基本概念](#)

- [15.2 理解synchronized](#)
- [15.3 线程的基本协作机制](#)
- [15.4 线程的中断](#)
- [第16章 并发包的基石](#)
- [16.1 原子变量和CAS](#)
- [16.2 显式锁](#)
- [16.3 显式条件](#)
- [第17章 并发容器](#)
- [17.1 写时复制的List和Set](#)
- [17.2 ConcurrentHashMap](#)
- [17.3 基于跳表的Map和Set](#)
- [17.4 并发队列](#)
- [第18章 异步任务执行服务](#)
- [18.1 基本概念和原理](#)
- [18.2 线程池](#)
- [18.3 定时任务的那些陷阱](#)
- [第19章 同步和协作工具类](#)
- [19.1 读写锁ReentrantReadWriteLock](#)
- [19.2 信号量Semaphore](#)
- [19.3 倒计时门栓CountDownLatch](#)
- [19.4 循环栅栏CyclicBarrier](#)
- [19.5 理解ThreadLocal](#)
- [第20章 并发总结](#)
- [20.1 线程安全的机制](#)
- [20.2 线程的协作机制](#)
- [20.3 容器类](#)
- [20.4 任务执行服务](#)
- [第六部分 动态与函数式编程](#)
- [第21章 反射](#)
- [21.1 Class类](#)
- [21.2 应用示例](#)
- [21.3 反射与泛型](#)
- [第22章 注解](#)
- [22.1 内置注解](#)
- [22.2 框架和库的注解](#)
- [22.3 创建注解](#)
- [22.4 查看注解信息](#)
- [22.5 注解的应用：定制序列化](#)
- [22.6 注解的应用：DI容器](#)
- [第23章 动态代理](#)
- [23.1 静态代理](#)
- [23.2 Java SDK动态代理](#)
- [23.3 cglib动态代理](#)
- [23.4 Java SDK代理与cglib代理比较](#)
- [23.5 动态代理的应用：AOP](#)
- [第24章 类加载机制](#)
- [24.1 类加载的基本机制和过程](#)
- [24.2 理解ClassLoader](#)
- [24.3 类加载的应用：可配置的策略](#)
- [24.4 自定义ClassLoader](#)
- [24.5 自定义ClassLoader的应用：热部署](#)
- [第25章 正则表达式](#)
- [25.1 语法](#)
- [25.2 Java API](#)
- [25.3 模板引擎](#)
- [25.4 剖析常见表达式](#)
- [第26章 函数式编程](#)
- [26.1 Lambda表达式](#)
- [26.2 函数式数据处理：基本用法](#)
- [26.3 函数式数据处理：强大方便的收集器](#)
- [26.4 组合式异步编程](#)
- [26.5 Java 8的日期和时间API](#)

读者评论

我觉得你的文章跟一般Java教程最大的不同在于，你把各个知识点的“为什么”都解释得很清楚，非常对味，非常感谢。很多网上教程，都是直接教如何做的，主要是动手能力。可是做完了还是云里雾里。结合你的文章，一下子就通透了。

——Hannah

老马说编程，太好了。把神秘的编程，通俗地讲解，使编程者认识了本质。每个专题的介绍都是深入浅出，有分析，有总结，有详细例子，真是爱不释手之宝书。

——张工荣成

其实老马写的东西网上都有大把的类似文章，但是老马总是能把复杂的东西讲得深入浅出，把看似简单的东西分析得细致深入！

——VitaminChen

文章比其他文章的亮点：有情景带入，重点突出，让人耳目一新，读起来很方便。感谢辛苦付出。

——hellojd

虽然我使用Java多年，可是阅读作者的文章仍然觉得受益匪浅。并发总结得很好，对前面讲的并发知识作了很好的总结和梳理。

——彭越

我不是初学者，依然能从这里学到很多东西。对不了解原理的非初学者来说，像回头捡落下的宝贝似的。关于编码，之前一直云里雾里的，找了几篇文章都没读进去。你的讲解浅显易懂！

——Keyrei

用平实的语言把计算机科学的思维方法由浅入深，娓娓道来，让人如沐春风，醍醐灌顶。这里面没有复制、粘贴的拼凑，更没有生硬古怪的翻译腔，文章中句句都能感觉到老马理解、实践、贯通后表达出来的逻辑严密周全和通透流畅。

——杜鹏

最近从PHP转Java，从您的文章学到了很多知识，很系统地重构了对计算机以及程序语言的认知，很感谢。

——房飞

多线程一直连概念也模糊，阅读后真的受益匪浅！异常处理，看着简单，刚开始学习时，自己也是胡乱try和throw，不过到开发时，才体会到正确处理的重要性。感谢这篇文章。比起学习使用庞大的框架，我觉得基础知识是更重要的，对于一个知识点的理解，细细琢磨，知道实现原理，也是一种收获。

——Chain

前言

为什么要写这本书

写一本关于编程的书，是我大概15年前就有的一个想法，当时，我体会到了编程中数据结构的美妙和神奇，有一种收获的喜悦和分享的冲动。这种收获是我反复阅读教程十几遍，花大量时间上机练习调试得到的，这是一个比较痛苦的过程。我想，如果把我学到的知识更为清晰易懂地表达出来，其他人不就可以掌握编程容易一些，并体会到那种喜悦了吗？不过，当时感觉自己学识太浅，要学习的东西太多，想一想也就算了。

触发我开始写作是在2016年年初，可汗学院的事迹震撼了我。可汗学院的创始人是萨尔曼·可汗，他自己录制了3000多个短视频，主要教中小學生基础課。他为每门课程建立了知识地图，地图由知识点组成，知识点之间有依赖关系。每个知识点都有一个视频，每个视频10分钟左右，他的讲解清晰透彻，极受欢迎。比尔·盖茨声称可汗是他最欣赏的老师，邀请其在TED发表演讲，同时投资可汗成立了非营利机构可汗学院，可汗也受到了来自谷歌等公司的投资。可以说，可汗以一己之力推动了全世界的教育。

我就想，我可不可以学习可汗，为计算机编程教育做点事情？也就是说，为编程的核心知识建立知识地图，从最基础的概念开始，分解为知识点，一个知识点一个知识点地讲解，每一个知识点都力争清晰透彻，阐述知识点是什么、怎么用、有什么用途、实现原理是什么、思维逻辑是什么、与其他知识点有什么关系等。可汗的形式是视频，但我想先从文字总结开始。我希望表达的是编程的通用知识，但编程总要用一个具体语言，我想就用我最熟悉的Java吧。

过去十几年，Java一直是软件开发领域最主流的语言之一，在可以预见的未来，Java还将是最主流的语言之一。但关于Java编程的书比比皆是，也不乏经典之作，市场还需要一本关于Java编程的书吗？甚至，还需要编程的书吗？如果需要，需要什么样的书呢？

关于编程的需求，我想答案是肯定的。过去几十年，IT革命深刻地改变了人们的生活，但这次革命还远远没有停止，在可以预见的未来，人工智能等前沿技术必将进一步改变世界，而要掌握人工智能技术，必须先掌握基本编程技术。人工智能在我国已经上升为国家战略。2017年7月，国务院印发了《新一代人工智能发展规划》，其中提到“实施全民智能教育项目，在中小学阶段设置人工智能相关课程，逐步推广编程教育”，未来，可能大部分人都需要学习编程。

关于编程的书是很多，但对于非计算机专业学生而言，掌握编程依然是一件困难的事情。绝大部分教程以及培训班过于追求应用，读者学完之后虽然能照着例子写一些程序，但却懵懵懂懂，知其然而不知其所以然，无法灵活应用，当希望进一步深入学习时，发现大部分专业书籍晦涩难懂，难以找到通俗易懂的与学过的应用相结合的进阶原理类书籍。

即使计算机专业的学生，学习编程也不容易。学校开设了很多理论课程，但学习理论的时候往往感觉比较枯燥，比如二进制、编码、数据结构和算法、设计模式、操作系统中的线程和文件系统知识等。而学习具体编程语言的时候，又侧重学习的是语法和API。学习计算机理论的重要目的是为了更好编程，但学生却难以在理论和编程之间建立密切的联系。

这样，我的想法基本就确定了，用Java语言写一本帮助理解编程到底是怎么回事的书，尽量用通俗易懂的方式循序渐进地介绍编程中的主要概念、语法和类库，不仅介绍用法和应用，还剖析背后的实现原理，以与基础理论相结合，同时包含一些实用经验和教训，并解释一些更为高层的框架和库的基本原理，以与实践应用相结合，在此过程中，融合编程的一些通用思维逻辑。

我有能力写好吗？我并不是编程大师，但我想，可汗也不是每个领域的大师，但他讲授了很多领域的知识，的确帮助了很多。过去十几年我一直从事编程方面的工作，也在不断学习和思考，我想，只要用心写，至少会给一些人带来一点帮助吧。

于是，我在2016年3月创建了微信公众号“老马说编程”，开始发布系列文章“计算机程序的思维逻辑”。每一篇文章对我都是一个挑战，每一个知识点我都花大量时间用心思考，反复琢磨，力求表达清晰透彻，做到最好。写作是一个痛苦和快乐交织的过程，最痛苦的就是满脑子都是相关的内容，但就是不知道该怎么表达的时候，而最快乐的就是写完一篇文章的时候。令人欣慰的是，这些文章受到了大量读者的极高评价，他们的溢美之词、自发分享和红包赞赏进一步增强了我写作的信心和动力。到2017年7月底，共写了95篇文章，关于Java编程的基本内容也就写完了。

在写作过程中，很多读者反馈希望文章可以尽快整理成书，以便阅读。2016年9月，机械工业出版社的高婧雅女士联系到了我，商讨出版的可能，在她的鼎力帮助和出版社的大力支持下，就有了大家看到的这本书。

本书特色

本书致力于帮助读者真正理解Java编程。对于每个语言特性和API，不仅介绍其概念和用法，还分析了为什么要有这个概念，实现原理是什么，背后的思维逻辑是什么；对于类库，分析了大量源码，使读者不仅知其然，还知其所以然，以透彻理解相关知识点。

本书虽然是Java语言描述，但以更为通用的编程逻辑为主，融入了很多通用的编程相关知识，如二进制、编码、数据结构和算法、设计模式、操作系统、编程思维等，使读者不仅能够学习Java语言，还可以提升整体的编程和计算机水平。

本书不仅注重实现原理，而且重视实用性。本书介绍了很多实践中常用的技术，包含不少实际开发中积累的经验和教训，使读者可以少走一些弯路。在实际开发中，我们经常使用一些高层的系统程序、框架和库，以提升开发效率，本书也介绍了如何利用基本API开发一些系统程序和框架，比如键值数据库、消息队列、序列化框架、DI（依赖注入）容器、AOP（面向切面编程）框架、热部署、模板引擎等，讲解这些内容的目的不是为了“重新发明轮子”，而是为了帮助读者更好地理解和应用高层的系统程序与框架。

本书高度注重表述，尽力站在读者的角度，循序渐进、简洁透彻，从最基本的概念开始，一步步推导出更为高级的概念，在介绍每个知识点时，都会尽力先介绍用法、示例和应用，再分析实现原理和思维逻辑，并与其他知识点建立联系，以便读者能够容易地、全面透彻地理解相关知识。

本书侧重于Java编程的主要概念，绝大部分内容适用于Java 5以上的版本，但也包含了最近几年Java的主要更新，包括Java 8引入的重要更新——Lambda表达式和函数化编程。

读者对象

本书面向所有希望进一步理解编程的主要概念、实现原理和思维逻辑的读者，具体来说有以下几种。

初中级Java开发者：本书采用Java语言，侧重于剖析编程概念背后的实现原理和内在逻辑，同时包含很多实际编程中的经验教训，所以，对于Java编程经历不多，对计算机原理不太了解、对Java的很多概念一知半解的开发人员，阅读本书的收益可能最大，通过本书可以快速提升Java编程水平。而零基础

础Java开发者，可跳过原理性内容阅读。

非Java语言的开发者：本书不假设读者有任何Java编程基础，系统、全面、细致地讲述了Java的语法和类库，给出了很多示例。另外，本书介绍了很多编程的通用概念、知识、数据结构、设计模式、算法、实现原理和思维逻辑。同时，全书的讨论都尽量站在一个通用的编程语言角度，而非Java语言特定的角度。通过阅读本书，读者可以快速学习和掌握Java，建立与其他语言之间的联系，提升整体编程思维和水平。

中高级Java开发者：经验丰富的Java开发者阅读本书的收获也会很大，可以通过本书对编程有更为系统、更为深刻的认识。

如何阅读本书

本书分为六大部分，共26章内容。

第一部分（第1～2章）介绍编程基础与二进制。第1章介绍编程的基础知识，包括数据类型、变量、赋值、基本运算、条件执行、循环和函数。第2章帮助读者理解数据背后的二进制，包括整数的二进制表示与位运算、小数计算为什么会出错、字符的编码与乱码。

第二部分（第3～7章）介绍面向对象。第3章介绍类的基础知识，包括类的基本概念、类的组合以及代码的基本组织机制。第4章介绍类的继承，包括继承的基本概念、细节、实现原理，分析为什么说继承是把双刃剑。第5章介绍类的一些扩展概念，包括接口、抽象类、内部类和枚举。第6章介绍异常。第7章剖析一些常用基础类，包括包装类、String、StringBuilder、Arrays、日期和时间、随机。

第三部分（第8～12章）介绍泛型与容器及其背后的数据结构和算法。第8章介绍泛型，包括其基本概念和原理、通配符，以及一些细节和局限性。第9章介绍列表和队列，剖析ArrayList、LinkedList以及ArrayDeque。第10章介绍各种Map和Set，剖析HashMap、HashSet、排序二叉树、TreeMap、TreeSet、LinkedHashMap、LinkedHashSet、EnumMap和EnumSet。第11章介绍堆与优先级队列，包括堆的概念和算法及其应用。第12章介绍一些抽象容器类，分析通用工具类Collections，最后对整个容器类体系从多个角度进行系统总结。

第四部分（第13～14章）介绍文件。第13章主要介绍文件的基本技术，包括文件的一些基本概念和常识、Java中处理文件的基本结构、二进制文件和字节流、文本文件和字符流，以及文件和目录操作。第14章介绍文件处理的一些高级技术，包括一些常见文件类型的处理、随机读写文件、内存映射文件、标准序列化机制，以及Jackson序列化。

第五部分（第15～20章）介绍并发。第15章介绍并发的传统基础知识，包括线程的基本概念、线程同步的基本机制synchronized、线程协作的基本机制wait/notify，以及线程的中断。第16章介绍并发包的基石，包括原子变量和CAS、显式锁与显式条件。第17章介绍并发容器，包括写时复制的List和Set、ConcurrentHashMap、基于跳表的Map和Set，以及各种并发队列。第18章介绍异步任务执行服务，包括基本概念和实现原理、主要的实现机制线程池，以及定时任务。第19章介绍一些专门的同步和协作工具类，包括读写锁、信号量、倒计时门栓、循环栅栏，以及ThreadLocal。第20章对整个并发部分从多个角度进行系统总结。

第六部分（第21～26章）介绍动态与函数式编程。第21章介绍反射，包括反射的用法和应用。第22章介绍注解，包括注解的使用、创建，以及两个应用：定制序列化和DI容器。第23章介绍动态代理的用法和原理，包括Java SDK动态代理和cglib动态代理以及一个应用：AOP。第24章介绍类加载机制，包括类加载的基本机制和过程，ClassLoader的用法和自定义，以及它们的应用：可配置的策略与热部署。第25章介绍正则表达式，包括语法、Java API、一个简单的应用（模板引擎），最后剖析一些常见表达式。第26章介绍Java 8引入的函数式编程，包括Lambda表达式、函数式数据处理、组合式异步编程，以及Java 8的日期和时间API。

对于有一定经验的读者，可以挑选感兴趣的章节直接阅读。而对于初学者，建议从头阅读，但对于一些比较深入的原理性内容，以及一些比较高级的内容，如果理解比较困难可以跳过，有一定实践经验后再回头阅读。任何读者都可以将本书作为一本案头参考书，以备随时查阅不确定的概念、用法和原理。

勘误和支持

由于笔者的水平有限，编写时间仓促，书中难免会出现一些错误或者不准确的地方，恳请读者批评指正。如果读者有更多的宝贵意见，欢迎关注我的微信公众号“老马说编程”，可在后台留言，在“关于”部分也有最新的微信和QQ群信息，欢迎加入讨论，我会尽量提供满意的解答。同时，读者也可以通过邮箱swiftma@sina.com联系我。期待得到你们的真挚反馈，在技术之路上互勉共进。

致谢

感谢我的微信公众号“老马说编程”、掘金、开发者头条和博客园技术社区的广大读者，他们的极高评价、自发分享和红包赞赏让我备受鼓舞，更重要的是，他们指出了很多文章中的错误，使我可以及时修正。

感谢掘金和开发者头条技术社区，他们经常推荐我的文章，使更多人可以看到。

感谢我在北京理工大学学习时的老师和同学们，在老师的教导和同学们的探讨中，我掌握了比较扎实的计算机基础，特别是我的已故恩师古志民教授，古教授指导我完成了本科到博士的学业，他严谨认真的学术态度深深地影响了我。

感谢我工作以来的领导和同事们，由于他们的言传身教，我得以不断提高自己的技术水平。

感谢机械工业出版社的编辑高婧雅，在一年多的时间中始终支持我的写作，她的帮助和建议引导我顺利完成全部书稿。

特别致谢

特别感谢我的爱人吴特和儿子久久，我为写作这本书，牺牲了很多陪伴他们的时间，但也正因为有了他们的付出与支持，我才能坚持写下去。

特别感谢我岳父母，特别是我的岳母，不遗余力地帮助我们照顾儿子，有了他们的帮助和支持，我才有时间和精力去完成写作工作。

特别感谢我的父母，他们在困难的生活条件下，付出了巨大的汗水与心血，将我养育成人，使我能够完成博士学业，他们一生勤劳朴素的品质深深地影响了我。

特别感谢我的兄长马俊杰，他一直是我成长路上的指明灯，也是从他的耐心讲解中我第一次了解到了计算机的基本工作机制。

谨以此书献给我最亲爱的家人，以及众多热爱编程技术的朋友们！

第一部分 编程基础与二进制

·第1章 编程基础

·第2章 理解数据背后的二进制

第1章 编程基础

我们先来简单介绍何谓编程，以及编出来的程序大概是什么样子。

计算机是个机器，这个机器主要由CPU、内存、硬盘和输入/输出设备组成。计算机上跑着操作系统，如Windows或Linux，操作系统上运行着各种应用程序，如Word、QQ等。

操作系统将时间分成很多细小的时间片，一个时间片给一个程序用，另一个时间片给另一个程序用，并频繁地在程序间切换。不过，在应用程序看来，整个机器资源好像都归它使用，操作系统给它制造了这种假象。对程序员而言，编写程序时基本不用考虑其他应用程序，做好自己的事就可以了。

应用程序看上去能做很多事情，能读写文档、能播放音乐、能聊天、能玩游戏、能下围棋等，但本质上，计算机只会执行预先写好的指令而已，这些指令也只是操作数据或者设备。所谓程序，基本上就是告诉计算机要操作的数据和执行的指令序列，即对什么数据做什么操作，比如：

- 1) 读文档，就是将数据从磁盘加载到内存，然后输出到显示器上；
- 2) 写文档，就是将数据从内存写回磁盘；
- 3) 播放音乐，就是将音乐的数据加载到内存，然后写到声卡上；
- 4) 聊天，就是从键盘接收聊天数据，放到内存，然后传给网卡，通过网络传给另一个人的网卡，再从网卡传到内存，显示在显示器上。

基本上，所有数据都需要放到内存进行处理，程序的很大一部分工作就是操作在内存中的数据。那具体如何表示和操作数据呢？本章介绍一些基础知识，具体分为7个小节。

数据在计算机内部都是二进制表示的，不方便操作，为了方便操作数据，高级语言引入了数据类型和变量的概念，这两个概念我们在1.1节介绍。

表示了数据后，1.2节介绍能对数据进行的第一个操作：赋值。

数据有了初始值之后，1.3节介绍可以对数据进行的一些基本运算，计算机之所以称为“计算”机，是因为最初发明它的主要目的也是运算。

为了编写有实用功能的程序，只进行基本运算是远远不够的，至少需要对操作的过程进行流程控制。流程控制有两种：一种是条件执行；另外一种循环。我们分别在1.4节和1.5节介绍。

为了减少重复代码和分解复杂操作，计算机程序引入了函数和子程序的概念，我们分别在1.6节和1.7节介绍函数的用法和函数调用的基本原理。

1.1 数据类型和变量

数据类型用于对数据归类，以便于理解 and 操作。对Java语言而言，有如下基本数据类型。

- 整数类型：有4种整型byte/short/int/long，分别有不同的取值范围；
- 小数类型：有两种类型float/double，有不同的取值范围和精度；
- 字符类型：char，表示单个字符；
- 真假类型：boolean，表示真假。

基本数据类型都有对应的数组类型，数组表示固定长度的同种数据类型的多条记录，这些数据在内存中连续存放。比如，一个自然数可以用一个整数类型数据表示，100个连续的自然数可以用一个长度为100的整数数组表示。一个字符可以用一个char类型数据表示，一段文字可以用一个char数组表示。

Java是面向对象的语言，除了基本数据类型，其他都是对象类型。对象到底是什么呢？简单地说，对象是由基本数据类型、数组和其他对象组合而成的一个东西，以方便对其整体进行操作。比如，一个学生对象，可以由如下信息组成。

- 姓名：一个字符串数组；
- 年龄：一个整数；
- 性别：一个字符；
- 入学分数：一个小数。

日期在Java中也是一个对象，内部表示为整型long。

世界万物都是由元素周期表中的基本元素组成的，基本数据类型就相当于化学中的基本元素，而对象就相当于世界万物。

为了操作数据，需要把数据存放到内存中。所谓内存程序看来就是一块有地址编号的连续的空间，数据放到内存中的某个位置后，为了方便地找到和操作这个数据，需要给这个位置起一个名字。编程语言通过变量 这个概念来表示这个过程。

声明一个变量，比如int a，其实就是在内存中分配了一块空间，这块空间存放int数据类型，a指向这块内存空间所在的位置，通过对a操作即可操作a指向的内存空间，比如a=5这个操作即可将a指向的内存空间的值改为5。

之所以叫“变”量，是因为它表示的是内存中的位置，这个位置存放的值是可以变化的。

虽然变量的值是可以变化的，但变量的名字是不变的，这个名字应该代表程序员心目中这块内存空间的意义，这个意义应该是不变的。比如，变量int second表示时钟秒数，在不同时间可以被赋予不同的值，但它表示的始终是时钟秒数。之所以说应该，是因为这不是必需的，如果一定要为一个名为age的变量赋予身高的值，计算机也拿你没办法。

重要的话再说一遍！变量就是给数据起名字，方便找不同的数据，它的值可以变，但含义不应变。再比如说一个合同，可以有4个变量：

- first_party: 含义是甲方；
- second_party: 含义是乙方；
- contract_body: 含义是合同内容；
- contract_sign_date: 含义是合同签署日期。

这些变量表示的含义是确定的，但对不同的合同，它们的值是不同的。初学编程的人经常使用像a、b、c、hehe、haha这种无意义的名字。在此建议为变量起一个有意义的名字吧！通过声明变量，每个变量赋予一个数据类型和一个有意义的名字，我们就告诉了计算机要操作的数据。

有了数据，如何对数据进行操作呢？我们先来看对数据能做的第一个操作：赋值。

欢迎访问：电子书学习和下载网站 (<https://www.shgis.cn>)

文档名称：《Java编程的逻辑》(Java核心技术系列) 马俊昌.epub

请登录 <https://shgis.cn/post/1841.html> 下载完整文档。

手机端请扫码查看：

