

普通高等院校计算机基础教育“十三五”规划教材

Java程序设计基础

Java CHENGXU SHEJI JICHU

罗恩韬 李文 扈乐华 主 编
黄丽韶 郭力勇 杜 敏 段华斌 黄伟国 副主编



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

普通高等院校计算机基础教育“十三五”规划教材

Java程序设计基础

罗恩韬 李文 扈乐华 主 编
黄丽韶 郭力勇 杜 敏 副主编
段华斌 黄伟国



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书使用开发环境是 JDK 1.8+Eclipse 4.7, 主要针对 Java 语言开发初学者, 重点讲解 Java 语言的基础知识。全书共 10 章, 主要介绍 Java 基本语法知识, 控制语句, 方法, 数组与字符串, 类和对象, 封装、继承和多态, 抽象类和接口, 异常处理机制, GUI 编程。内容由浅入深, 并辅以实例说明, 逐步引领读者学习 Java 语言程序设计的各个知识点。

本书适合作为普通高等院校 Java 程序设计课程的教材, 也可作为 Java 入门者的参考用书。

图书在版编目 (CIP) 数据

Java 程序设计基础/罗恩韬, 李文, 扈乐华主编. — 北京: 中国铁道出版社, 2017.9

普通高等院校计算机基础教育“十三五”规划教材
ISBN 978-7-113-23598-7

I. ①J… II. ①罗… ②李… ③扈… III. ①JAVA 语言-程序设计-高等学校-教材 IV. ①TP312.8

中国版本图书馆 CIP 数据核字 (2017) 第 215317 号

书 名: Java 程序设计基础
作 者: 罗恩韬 李 文 扈乐华 主编

策 划: 包 宁
责任编辑: 刘丽丽 徐盼欣
封面设计: 刘 颖
责任校对: 张玉华
责任印制: 郭向伟

读者热线: (010) 63550836

出版发行: 中国铁道出版社 (100054, 北京市西城区右安门西街 8 号)

网 址: <http://www.tdpress.com/51eds/>

印 刷: 三河市宏盛印务有限公司

版 次: 2017 年 9 月第 1 版 2017 年 9 月第 1 次印刷

开 本: 787 mm×1 092 mm 1/16 印张: 13.75 字数: 328 千

书 号: ISBN 978-7-113-23598-7

定 价: 38.00 元

版权所有 侵权必究

凡购买铁道版图书, 如有印制质量问题, 请与本社教材图书营销部联系调换。电话: (010) 63550836

打击盗版举报电话: (010) 51873659

中国铁道出版社

» 前言



Java 是当今最为流行的程序开发语言之一，具有与平台无关、简单高效、多线程、安全和健壮等特点，广泛应用于企业级 Web 应用开发和移动应用开发。

要学好程序设计，首先要有兴趣。本书在结构上作了精心安排，以知识点和具体实例相结合的方式介绍所有内容。在对细节的逐步深入过程中，通过对问题的求解提升读者的学习兴趣。

本书在对知识点进行分析和归纳的同时，对引例作扩展或改变，逐步形成更全面、复杂的实例，让读者通过对比加强对概念的理解，从而达到举一反三的学习效果。考虑到知识的连贯性，各章节会在内容和实例上有所联系，以涵盖各知识点并拓宽读者思路。

本书的编写目的在于，进一步深化读者对基本概念的理解，提高读者综合应用能力，使读者在掌握 Java 面向对象程序设计核心理论与编程思想、技巧的同时，养成良好的编程习惯。本书所配备的例题清晰直观、循序渐进，并通过通俗易懂且逻辑性强的讲解巩固知识点。

本书编者长期从事教学工作，积累了丰富的经验，其“实战教学法”取得了很好的效果。本书具有以下特点：

1. 注重基础性

本书内容注重基础性，深入浅出，并在每章后面安排了大量的习题，帮助学生学习的每一个知识点。

2. 兼顾流行性

本书讲解的是 Java 开发过程中最流行的方法，可培养学生良好的编程风格和编程习惯。

3. 适合教学

书中每章内容安排适当，符合教学要求，教师可以根据具体情况选用，也可以进行适当增减。

本书共分 10 章：第 1 章为 Java 概述，介绍 Java 的基本知识（包括 Java 发展历史，Java 的特点和基本原理）；第 2 章为 Java 基本语法知识，重点介绍标识符及关键词、数据类型、常量与变量，以及运算符与表达式；第 3 章为 Java 控制语句，介绍 Java 的三大基本结构化程序；第 4 章为方法，重点介绍方法的基本概念，包括方法声明、方法调用、参数传递、方法重载、局部变量和包；第 5 章为数组与字符串，首先介绍数组的基本概念，包括一维数组和二维数组，其次介绍字符串相关类；第 6 章为类和对象，主要

介绍类和对象的基本概念；第 7 章为封装、继承和多态，介绍封装、继承和多态三大特征之间的概念以及联系；第 8 章为抽象类和接口，重点介绍抽象类和接口的特点以及主要作用，应灵活掌握其在程序中的使用；第 9 章为 Java 异常处理机制，重点介绍异常处理机制的基本原理、异常处理的过程，以及异常处理的三种方式，最后介绍自定义异常的实现；第 10 章为 GUI 编程，重点介绍图形用户界面编程，利用编程人员对图形用户界面编程的兴趣，将前面 9 章的内容结合起来，将所有内容融合成一个实例。

本书由罗恩韬、李文、扈乐华任主编，黄丽韶、郭力勇、杜敏、段华斌、黄伟国任副主编。具体编写分工如下：第 1、2 章由李文编写，第 3、4 章由罗恩韬编写，第 5 章由黄丽韶编写，第 6 章由扈乐华编写，第 7 章由郭力勇编写，第 8 章由杜敏编写，第 9 章由段华斌编写，第 10 章由黄伟国编写。全书由罗恩韬、李文、扈乐华统稿。本书在编写过程中参阅了许多优秀的同类教材以及网上资料，在此向其作者表示衷心的感谢。

由于时间仓促和编者水平有限，书中疏漏和不妥之处在所难免，敬请读者批评指正。

编 者

2017 年 7 月

» 目 录



❖ 第1章 Java概述	1
1.1 Java简介	1
1.1.1 Java的历史	1
1.1.2 Java的特点	2
1.1.3 Java运行基本原理	4
1.1.4 Java程序的类型	4
1.2 Java中的OOP	5
1.2.1 OOP的基本思想	5
1.2.2 OOP的发展过程	6
1.2.3 OOP的特点	7
1.3 Java开发环境	9
1.3.1 JDK环境配置	9
1.3.2 JDK常用命令	12
1.3.3 Eclipse配置	16
1.3.4 Eclipse主界面	18
1.3.5 用Eclipse开发Java应用程序	20
小结	22
习题	22
❖ 第2章 Java基本语法知识	24
2.1 标识符及关键词	24
2.1.1 标识符	24
2.1.2 关键词	25
2.1.3 语句及注释	25
2.2 数据类型	25
2.2.1 基本数据类型	25
2.2.2 引用数据类型	26
2.3 常量与变量	26
2.3.1 常量	26
2.3.2 变量	28
2.4 运算符与表达式	29



2.4.1	算术运算符及表达式	29
2.4.2	关系运算符及表达式	31
2.4.3	逻辑运算符及表达式	31
2.4.4	位运算符及表达式	32
2.4.5	赋值运算符及表达式	32
2.4.6	其他运算符及表达式	33
2.4.7	运算符的优先级与结合性	33
2.4.8	数据类型转换	34
小结		35
习题		36
第3章	Java控制语句	40
3.1	顺序结构语句	40
3.1.1	表达式语句	40
3.1.2	块语句	40
3.2	选择结构语句	41
3.2.1	if语句	41
3.2.2	switch语句	47
3.2.3	选择结构语句的嵌套	50
3.3	循环结构语句	53
3.3.1	while循环语句	53
3.3.2	do...while循环语句	54
3.3.3	for循环语句	56
3.4	其他控制语句	58
3.4.1	break语句	58
3.4.2	continue语句	58
小结		59
习题		60
第4章	方法	69
4.1	方法声明	69
4.2	方法调用	70
4.3	参数传递	71
4.4	方法重载	72
4.5	局部变量	73
4.6	包	74
4.6.1	package语句	74
4.6.2	import语句	75

小结	75
习题	76
第5章 数组与字符串	81
5.1 数组	81
5.1.1 一维数组	81
5.1.2 多维数组	83
5.2 字符串相关类	86
5.2.1 String类	86
5.2.2 StringBuffer类	89
5.2.3 StringBuilder类	90
小结	90
习题	91
第6章 类和对象	99
6.1 引入实例	99
6.2 类	100
6.2.1 类的定义	100
6.2.2 成员变量	100
6.2.3 成员方法	101
6.3 对象	101
6.3.1 对象的声明和创建	102
6.3.2 对象的使用	103
6.4 类的封装	104
6.4.1 访问控制属性	104
6.4.2 设置类的访问控制属性	104
6.4.3 设置类成员的访问控制属性	104
6.5 静态成员	106
6.5.1 静态变量	106
6.5.2 静态方法	108
小结	109
习题	110
第7章 封装、继承和多态	119
7.1 封装	119
7.2 继承	120
7.2.1 继承的基本概念	120
7.2.2 子类对象的构造过程	121
7.2.3 继承中的构造方法	121



7.2.4	方法重写	121
7.2.5	this关键字	122
7.2.6	super关键字	124
7.3	多态	126
7.3.1	多态的概念	126
7.3.2	多态的实现	126
7.3.3	对象转型	127
7.4	equals()方法	128
	小结	129
	习题	130
第8章	抽象类和接口	139
8.1	抽象类	139
8.2	接口	142
8.2.1	接口的定义	142
8.2.2	实现接口	142
8.2.3	继承接口	143
8.2.4	接口的特征	143
8.2.5	比较抽象类与接口	144
8.3	final关键字	145
8.4	常用类	146
8.4.1	基本数据类型包装类	146
8.4.2	Math类	147
8.4.3	Random类	148
8.4.4	System类	149
8.4.5	Runtime类	149
8.4.6	Date类	150
8.4.7	SimpleDateFormat类	150
8.4.8	Calendar类	150
	小结	152
	习题	153
第9章	Java异常处理机制	163
9.1	Java异常的基础知识	163
9.2	Java异常类类图	164
9.3	Java异常处理机制	165
9.4	定义和使用异常类	167
9.5	运行时异常和受检查异常	168

9.6 Java异常处理的原则和技巧	169
小结	169
习题	170
 第10章 GUI编程	175
10.1 Swing介绍	175
10.2 Swing控件	176
10.2.1 JApplet	176
10.2.2 ImageIcon和JLabel	177
10.2.3 JTextField	179
10.2.4 AbstractButton	181
10.2.5 JButton	181
10.2.6 JCheckBox	182
10.2.7 JRadioButton	184
10.2.8 JComboBox	186
10.2.9 JTabbedPane	187
10.2.10 JScrollPane	189
10.2.11 JTree	191
10.2.12 JTable	192
10.3 Swing事件处理	192
10.4 Swing事件监听器	195
10.5 Swing Layout布局	196
10.5.1 BorderLayout	196
10.5.2 FlowLayout	198
10.5.3 GridLayout	199
10.5.4 GridBagLayout	201
10.5.5 CardLayout	204
10.5.6 BoxLayout	205
小结	206
习题	207
 参考文献	210

第1章 Java概述



Java 是 20 世纪 90 年代出现的完全面向对象的程序设计语言,体现了计算机编程的新方法、新思想。本章首先介绍面向对象程序设计的基本概念、特点和基本思想;然后介绍面向对象的 Java 程序设计语言的发展概况、特点、运行机制和运行环境;最后简单介绍 Java 集成编程工具 Eclipse。



1.1 Java 简介

Java 是美国 Sun 公司(已于 2009 年被 Oracle 公司收购)研制的一种程序设计语言。在高级语言已经非常丰富的背景下,Java 脱颖而出,独树一帜,在瑞士 TIOBE 公司每月发布的程序开发语言排行榜中,Java 连续多年名列榜首,说明了人们对 Java 的喜爱程度。

1.1.1 Java 的历史

1994 年,美国 Sun 公司成立了 Green 项目开发小组,旨在研制一种能对家用电器进行控制和通信的分布式代码系统,当时这套系统被命名为 Oak,这就是 Java 的前身。

1994 年前后,正是 Internet 特别是 Web 的大发展时期, Sun 公司的研究人员发现 Oak 的许多特性更适合网络编程,于是在这方面进行了一系列改进和完善,并获得了成功。1995 年初, Sun 公司要给这种语言申请注册商标,由于 Oak 已经被人注册,必须要为这种语言找到一个新的名字。在公司召开的命名征集会上, Mark Opperman 提出 Java 这个名字。据说, Mark Opperman 是因品尝咖啡时得到灵感的。Java 是印度尼西亚爪哇岛的英文名称,该岛因盛产高质量的咖啡而闻名,常被用来当做优质咖啡的代名词。Mark Opperman 的这个提议,得到了所有人的认可和律师的通过, Sun 公司用 Java 这个名字进行了注册,并以一杯热气腾腾的咖啡作为标志, Java 由此诞生。

Java 从诞生到今天,不断进行改进和更新。其发展历程大致可以分成以下几个阶段。

1. 诞生期——Java 1.0 和 Java 1.1

Java 1.0 的出现是为了帮助开发人员建立运行环境并提供开发工具。1996 年 1 月 23 日, Sun 公司发布了第一个 Java 开发工具 JDK 1.0, JDK(Java Development Kit), JDK 1.0 由运行环境(Java Runtime Enviroment, JRE)及开发工具(即 JDK)组成,其中运行环境又包括 Java 虚拟机(Java Virtual Machine, JVM)、API(Application Programming Interface, 应用程序接口)和发布技术。1997 年, Sun 公司对 1.0 版本进行了较大的改进,推出了 JDK 1.1 版本,其中增加了 JIT



(Just-In-Time) 编译器。

2. 发展期——Java 1.2 和 Java 1.3

1998 年 12 月, Java 1.0 诞生近三年后, Sun 公司推出 Java 1.2, 并将其改名为 Java2, 且把 Java 1.2 以后的版本统称为 Java2, 同时将 JDK 1.2 改名为 J2SDK (SDK 的全称为 Software Development Kit, 意为软件开发工具), 从此 Java 进入了快速发展的阶段。1999 年, Sun 公司发布 Java 的三个版本: 标准版 (Java2 Standard Edition, J2SE)、企业版 (Java2 Enterprise Edition, J2EE) 和微型版 (Java2 Micro Edition, J2ME), 以适应不同的应用开发要求。2000 年, JDK 1.3 发布。Java 1.3 在 Java 1.2 取得成功的基础上进行了一些改进, 主要是对 API 做了改进和扩展。

3. 成熟期——Java 1.4、Java 1.5 和 Java 1.6

自 Java2 平台开始, Java 的发展日趋成熟稳定, 此后的 Java 1.4、Java 1.5 和 Java 1.6 主要在分布式、稳定性、可伸缩性、安全性和管理方面进行了改进和提高。Java 1.4 比 Java 1.3 的运行效率提高了一倍。而从 Java 1.5 版本开始, Java 1.5 改名为 Java 5.0, J2SE 1.5 改名为 J2SE 5.0, 更好地反映出了 J2SE 的成熟度和稳定性; Java 1.6 (J2SE 6.0) 则更强调管理的易用性, 为外部管理软件提供更多的交互信息, 并更好地集成了图形化用户界面。从 2004 年开始, 为了更加突出 Java 本身, 而不是 Java 的某个版本编号, Java 的三个版本陆续更名, 去掉其中的编号 2, J2SE、J2EE、J2ME 更名为 Java SE、Java EE 和 Java ME。

经过不断完善和发展, Java 已经得到业界的广泛认可, 主要体现在工业界认可、软件开发商青睐和编程人员欢迎等几方面。

① 工业界认可。目前绝大部分计算机企业 (包括 IBM、Apple、DEC、Adobe、Silicon Graphics、HP、Oracle、Toshiba 以及 Microsoft 等公司) 都购买了 Java 的许可证, 用 Java 开发相应的产品。这说明 Java 已得到了工业界的认可。

② 软件开发商青睐。除购买 Java 许可证, 用 Java 开发新产品以外, 众多的软件厂商还在自己已有的产品上增加 Java 接口, 以使自己的产品支持 Java 的应用。例如 Oracle、Sybase、Versant 等数据库厂商开发了 CGI 接口, 使得这些数据库支持 Java 开发。

③ 编程人员欢迎。Java 的一个重要特点是其网络编程能力, 因而成为网络时代编程人员最欢迎的程序设计语言之一, 各行业对掌握 Java 的人员需求量也非常大。

上述事实说明, Java 是一种得到广泛应用并有很好的发展前景的程序设计语言。

1.1.2 Java 的特点

Java 之所以能够受到如此众多的好评并拥有如此迅猛的发展速度, 与其本身的特点是分不开的。Java 的主要特点如下:

1. 面向对象设计

面向对象设计是 Java 的标志特性。作为一种纯粹的面向对象程序设计语言, Java 不再支持面向过程的设计方法, 而是从面向对象的角度思考和设计程序。Java 通过创建类和对象来描述和解决问题, 支持封装、继承、重载、多态等面向对象特性, 提高了程序的可重用性和可维护性。

2. 简单易用

Java 最初的产生源于对家用电器的控制，其设计以简单易用、规模小为原则。一方面，Java 的语法非常简单，它不再使用其他高级程序设计语言中诸如指针运算、结构、联合、多维数组、内存管理等复杂的语言现象，降低了程序编写的难度；另一方面，Java 提供了极为丰富的类库，封装了各种常用的功能，程序设计人员无须对这些常用的功能自行编写程序，只要直接调用即可，尽可能降低了程序设计人员的工作量。

3. 平台无关性

Java 的平台无关性主要体现在三方面。首先，Java 运行环境是 Java 虚拟机，Java 虚拟机负责解释编译后的 Java 代码并将其转换成特定系统的机器码，再由机器加以执行。Java 虚拟机屏蔽了具体平台的差异性，用 Java 编写的应用程序无须重新编译就可以在不同平台的 Java 虚拟机上运行，实现了平台无关性。其次，Java 的数据类型被设计成不依赖于具体机器。例如，整数总是 32 位，长整数总是 64 位。这样，Java 基本数据类型及其运算在任何平台上都是一致的，不会因平台的变化而改变。第三，Java 核心类库与平台无关，对类库的调用，不会影响 Java 的跨平台性。

4. 安全性和健壮性

Java 去除了指针和内存管理等易出错的操作，在程序设计上增强了安全性。而且，Java 作为网络开发语言，提供了多层保护机制增强安全性，例如，不允许 Applet 运行和读/写任何浏览器端机器上的程序等。此外，Java 注重尽早发现错误。Java 编译器可以检查出很多开发早期的错误，增强了程序设计的安全性和健壮性。

5. 性能优异

Java 可以在运行时直接将目标代码翻译成机器指令，充分地利用硬件平台资源，从而可以得到较高的整体性能。另外，与 Java 有关的厂商在不断完善 Java 的 JIT 编译器技术，旨在提高 Java 的运行速度。从基准测试来看，Java 的运行速度超过了典型的脚本语言，越来越接近 C 和 C++。

6. 分布式

分布式是指在由网络相连的不同平台上，可以在独立运行时间内运行不同程序。Java 作为一种强大的网络开发语言，其能力主要体现在开发分布式网络应用。Java 语言本身的特点很适合开发基于 Internet 的分布式应用程序，并且提供了完备的适应分布式应用的程序库。Java 支持 TCP/IP 协议及其他协议，可以通过 URL (Uniform Resource Locator, 统一资源定位符) 实现对网络上其他对象的访问，实现分布式应用。

7. 多线程

Java 支持多线程，允许在程序中并发地执行多个指令流或程序片段，以更好地利用系统资



源,提高程序的运行效率。Java 不仅支持多线程,而且对线程划分了优先级,以更好地支持系统的交互和实时响应能力。此外,Java 具备线程同步功能,确保了计算结果的可预测性,有助于对程序进行更好的控制。

1.1.3 Java 运行基本原理

Java 程序的运行机制与 C/C++等程序设计语言有较大的差别,这种差别也是保证 Java 具有更强动态性和平台无关性的基础。概括来说,Java 的运行有三个步骤:编写、编译和运行。

① 编写是指利用编辑器生成 Java 程序代码,形成 Java 源文件。Java 程序以.java 为扩展名。一个 Java 应用程序中可能会包括多个 Java 的类,这些类可以放在同一个 Java 源文件中,也可以为每一个类分别编写一个源文件。

② 编译是指 Java 编译器将编辑好的 Java 源程序转换成 JVM 可以识别的字节码的过程。字节码是一种独立于操作系统和机器平台的中间代码,用二进制形式表示,由 JVM 解释后才能在机器上运行。编译成功后,Java 编译器生成扩展名为.class 的字节码文件。如果一个 Java 源程序中包含了多个类,编译后会生成多个对应的.class 文件。

③ 运行是指 JVM 将编译生成的.class 字节码文件翻译为与硬件环境及操作系统匹配的代码,并运行和显示结果。JVM 可以将 Java 字节码程序和具体的操作系统及硬件区分开,而不用考虑程序文件要在何种平台上运行,从而保证了 Java 语言的平台无关性和动态性。

图 1-1 所示是 Java 程序编写、编译和运行的过程。



图 1-1 Java 程序编写、编译和运行的过程

1.1.4 Java 程序的类型

Java 支持开发 4 种基本类型的程序,分别是 Java 应用程序 (Java Application)、Java 小应用程序 (Java Applet)、服务器端小程序 (Java Servlet) 以及可重用 Java 组件 JavaBean。这 4 种类型的 Java 程序都遵循 Java 的基本编程结构,并且都要在 Java 虚拟机上运行,它们的表现形式都是 Java 的类。

1. Java 应用程序

Java 应用程序是指完整的、可以独立运行的 Java 程序。一个 Java 应用程序由一个或多个类组成。Java 应用程序经过编译之后,可在 Java 虚拟机上独立运行,完成一定的功能。在组成 Java 应用程序的类中,必须有一个类中包含有 main()方法 (或称 main()函数),该方法是 Java 的内置方法,作用是提供 Java 应用程序的入口。Java 虚拟机从 main()方法开始执行 Java 应用程序。包含 main()方法的类称为 Java 应用程序的主类 (简称主类)。在编写 Java 源程序时,如果将 Java 应用程序所包含的多个类同时写在一个文件中,则该文件名必须和主类的类名保持一致,并以.java 为扩展名。如果将不同的类分别写在不同的文件中,通常将源文件命名为与其包含的类名相同,并以.java 为扩展名。

2. Java 小应用程序

Java 小应用程序简称 Applet，是一种嵌在 HTML 页中由 Web 浏览器激活 Java 虚拟机来运行的程序。也就是说，Applet 本身不能独立运行，必须以 Web 浏览器为其容器才能运行，因此，可以简单地将 Applet 理解成由 Web 浏览器来执行的程序。Applet 部署在服务器端，当用户访问嵌入了 Applet 的网页时，相应的 Applet 被下载到客户端的机器上执行。Applet 通常用来在网页上实现与用户的交互功能或者实现动态的多媒体效果，使得网页更具活力。能够执行 Applet 的浏览器必须支持 Java。

3. 服务器端小程序

服务器端小程序简称 Servlet，是一种用 Java 编写的服务器端程序。Servlet 以 Web 服务器为容器，靠 Web 服务器来加载和运行。和 Applet 一样，Servlet 本身不能独立运行；但与 Applet 不同的是，Applet 在客户端运行，Servlet 在服务器端运行。Servlet 的作用是接收、处理客户端的请求并将响应发送到客户端，从而实现客户端计算机与服务器端计算机之间的交互。利用 Servlet 技术，可以扩展 Web 服务器能力，充分利用 Web 服务器上的资源（如文件、数据库、应用程序等）。能够执行 Servlet 的服务器必须支持 Java。

4. JavaBean

JavaBean 是一种用 Java 编写的可重用的软件组件，目前尚没有统一的中文译名。JavaBean 本身不能独立运行，必须以 Java 应用程序、Applet、Servlet 或者 JavaBean 为容器才能运行。JavaBean 有两种类型，一种是可视化的 JavaBean，另一种是非可视化的 JavaBean。可视化的 JavaBean 具有图形界面，可以包括窗体、按钮、文本框、报表元素等。非可视化的 JavaBean 不包括图形界面，主要用来实现业务逻辑或封装业务对象。可视化的 Javabeen 是 JavaBean 的传统应用，随着网络的兴起，非可视化的 JavaBean 应用越来越广泛，它与 JSP (Java Server Pages) 技术相结合，成为当前开发 Web 应用的主流模式。



1.2 Java 中的 OOP

面向对象程序设计 (Object-Oriented Programming, OOP) 是计算机软件技术发展过程中的一个重大飞跃，它能更好地适合软件开发在规模、复杂性、可靠性和质量、效率上的需求，因而被广泛应用，并逐渐成为当前的主流程序设计方法。

1.2.1 OOP 的基本思想

面向对象程序设计代表了一种全新的程序设计思路和表达、处理问题的方法。在解决问题的过程中，面向对象程序设计以问题中所涉及的各种对象为主要线索，关心的是对象以及对象之间的相互关系，以符合人们日常的思维习惯来求解问题，降低、分解了问题的难度和复杂性，提高了整个求解过程的可控性、可监测性和可维护性，从而能以较小的代价和较高的效率对问题进行求解。简言之，面向对象程序设计的特点是使用对象模型对客观世界进行抽象，分析出



事物的本质特征,从而对问题进行求解。面向对象程序设计的思想认为世界是由各种各样具有各自运动规律和内部状态的对象组成的,不同对象之间的相互通信和作用构成了现实世界,因此,人们应当按照现实世界本来的面貌理解世界,直接通过对象及其相互关系来反映世界,这样建立起来的系统才符合世界本来的面貌,才会对现实世界的变化有很好的适应性。所以,面向对象方法强调程序系统的结构应当与现实世界的结构相对应,应当围绕现实世界中的对象来构造程序系统。

所谓对象,是指现实世界的实体或概念在计算机程序中的抽象表示。具体地说,程序设计中的对象是指具有唯一对象名和一组固定对外接口的属性和操作的集合,它用来模拟组成或影响现实世界问题的一个或一组因素。其中,对象名是用于区别对象的标识;对象的对外接口是在约定好的运行框架和消息传递机制下与外界进行通信的通道;对象的属性表现了它所处的状态;对象的操作(也称方法)是用来改变对象状态的特定功能。

具体地说,面向对象程序设计的思想主要体现在如下几方面:

① 面向对象程序设计的核心和首要问题是标识对象,而不是标识程序中的功能(函数/过程)。从面向对象程序设计的角度来看,对象作为现实世界中事物的基本组成部分,是系统框架中最稳定的因素,对象描述清楚了,就能够很容易地找出它们之间的关系,从而发现它们之间的相互作用,进而解决问题。

② 正是由于把标识对象作为解决问题的出发点,面向对象程序设计在整体上说是一种自底向上的开发方法。面向对象的基本思想将程序看作众多协同工作的对象所组成的集合,这些对象相互作用构成系统完整的功能,因此在设计开发程序时,面向对象的方法按照标识对象、定义对象属性和操作、明确对象之间事件驱动和消息关系,最后形成程序的整体结构顺序进行。

③ 同任何其他应用系统开发一样,面向对象程序设计在概念模式与系统组成模式上的一致性,使得面向对象程序设计过程中的各个阶段是一种自然平滑的过渡,各阶段的界限不是那么明显。系统分析阶段的结果能够直接映射成系统设计的概念,系统设计阶段的结果可以方便地翻译成实施阶段的程序组件,反之亦然。这样,系统设计和开发人员就能够容易地跟踪整个系统开发过程,了解各个阶段所发生的变化,不断对各个阶段进行完善。

总之,面向对象程序设计方法更符合人们对客观世界的认识规律,开发的软件系统易于维护、理解、扩充和修改,并支持软件的复用。从 20 世纪 90 年代开始,面向对象程序设计的方法逐渐成为软件开发的主流方法。

1.2.2 OOP 的发展过程

面向对象程序设计方法作为一种程序设计规范,其发展与程序设计语言的发展密切相关。事实上,最早的面向对象程序设计的一些概念正是由一些特定语言机制体现出来的。

20 世纪 50 年代后期,为了解决 FORTRAN 语言编写大型软件时出现的变量名在不同程序段中的冲突问题,ALGOL 语言设计者采用了“阻隔”(Barriers)的方式来区分不同程序段中的变量名,在程序设计语言 ALGOL60 中用 Begin...End 为标识对程序进行分段,以便区分不同程序段中的同名变量,这也是首次在编程语言中出现保护(Protection)和封装(Encapsulation)的思想。

20 世纪 60 年代,挪威科学家 O. J. Dahl 和 K. Nygaard 等采用了 ALGOL 语言中的思想,设计出用于模拟离散事件的程序设计语言 Simula 67。与以往程序设计语言不同,Simula 67 从一个

全新的角度描述并理解客观事实,首次在程序设计中将数据和与之对应的操作结合成一个整体,提出“封装”的概念,它的类型结构和以后的抽象数据类型基本是一样的。尽管 Simula 67 还不是真正的面向对象程序设计语言,但它提出的思想标志着面向对象技术正式登上历史舞台。真正的面向对象程序设计语言是由美国 Alan Keyz 主持设计的 Smalltalk 语言。Smalltalk 这个名字源自 Talk Small (少说话),意思是可以通过很少的工作量完成许多任务。Smalltalk 在设计中强调对象概念的统一,引入了对象、对象类、方法、实例等概念和术语,采用了动态联编和单继承机制。用 Smalltalk 编写的程序具有封装、继承、多态等特性,由此奠定了面向对象程序设计的基础。20 世纪 80 年代以后,美国 Xerox 公司推出了 Smalltalk-80,引起人们的广泛重视。

Smalltalk 语言的出现引发了学术界对面向对象程序设计的广泛重视,随之涌现出了很多面向对象的系统分析与设计方法,诞生了一系列面向对象的语言,如 C++、Eiffel、Ada 和 CLOS 等。其中,C++不仅继承了 C 语言易于掌握、使用简单的特点,而且增加了众多支持面向对象程序设计的特性,促进了面向对象程序设计技术的发展。

20 世纪 90 年代,美国 Sun 公司提出的面向对象的程序设计语言 Java,被认为是面向对象程序设计的一次革命。Java 去除了 C++中为了兼容 C 语言而保留的非面向对象的内容,使程序更加严谨、可靠、易懂。尤其是 Java 所特有的“一次编写、多次使用”的跨平台优点,使得它非常适合在 Internet 应用开发中使用。

从面向对象程序设计语言的发展历程可以看出,面向对象程序设计语言是经过研究人员的不断改进与优化,才形成了今天的模样。正是由于这种语言更好地适应了软件开发过程中规模、复杂性、可靠性和质量、效率上的需求,并且在实践中得到了检验,才逐渐成为当前主流的程序设计方法。

1.2.3 OOP 的特点

面向对象程序设计有许多特点,这里重点介绍其主要特点。

1. 抽象

抽象 (Abstract) 是日常生活中经常使用的一种方法,即去除被认识对象中与主旨无关的部分,或是暂不予考虑的部分,而仅仅抽取与认识目的有关的、实质性的内容加以考察。在计算机程序设计中所使用的抽象有两类:一类是过程抽象;另一类是数据抽象。

过程抽象将整个系统的功能划分为若干部分,强调功能完成的过程和步骤。面向过程的软件开发方法采用的就是这种抽象方法。使用过程抽象有利于控制、降低整个程序的复杂程度,但是这种方法本身自由度较大,难于规范化和标准化,操作起来有一定难度,质量上不易保证。

数据抽象是与过程抽象不同的抽象方法,它把系统中需要处理的数据和这些数据上的操作结合在一起,根据功能、性质、作用等因素抽象成不同的抽象数据类型。每个抽象数据类型既包含数据,也包含针对这些数据的授权操作,是相对于过程抽象而言更为严格、也更为合理的抽象方法。

面向对象程序设计的主要特点之一,就是采用了数据抽象的方法来构建程序的类、对象和方法。在面向对象程序设计中使用的数据抽象方法,一方面可以去除与核心问题无关的细节,使开发工作可以集中在关键、重要的部分;另一方面,在数据抽象过程中,对数据操作的分析、辨别和定义可以帮助开发人员对整个问题有更深入、准确的认识,最后抽象形成的抽象数据类