

P ROGRAMMING IN C/C++ C/C++程序设计

宋晓宇 主编

赵艳平 杨艳春 李世伟 等参编



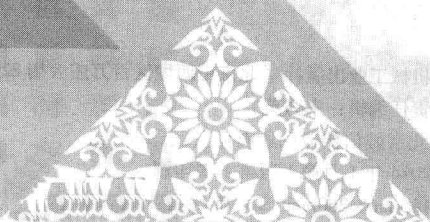
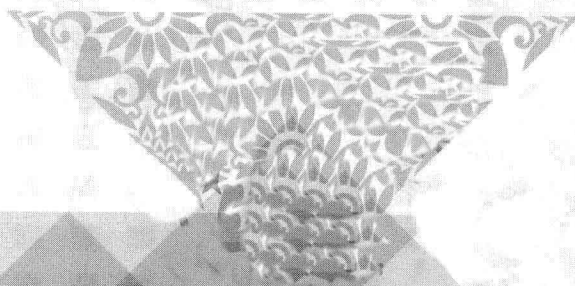
机械工业出版社
China Machine Press

计算机基础教育课程体系规划教材

P ROGRAMMING IN C/C++ C/C++程序设计

宋晓宇 主编

赵艳平 杨艳春 李世伟 等参编



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

C/C++ 程序设计 / 宋晓宇主编. —北京: 机械工业出版社, 2014. 1
(计算机基础教育课程体系规划教材)

ISBN 978-7-111-45157-0

I. C… II. 宋… III. C 语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2013) 第 295116 号

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问 北京市展达律师事务所

本书针对初学者的特点, 采取“提出问题—分析问题—解决问题—归纳提高”的教学模式, 突出对学习者的计算思维、编程实践能力的培养与训练。全书共分为 12 章, 全面系统地介绍了 C/C++ 语言的基本概念、语法及程序设计方法, 详细地讲解了 C/C++ 中的数据类型、运算符与表达式、基本控制语句、数组、函数、指针、类和对象、继承和派生、多态性、虚函数、输入/输出流等内容。

本书定位准确、结构合理、概念清晰、逻辑严密、例题丰富、循序渐进, 符合学习者的认知规律, 适合作为高等学校开设程序设计基础课程的教材, 也可作为工程技术人员、自学人员以及参加全国计算机等级考试 (二级 C/C++ 程序设计) 的人员的参考书。

机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码 100037)

责任编辑: 朱秀英

北京诚信伟业印刷有限公司印刷

2014 年 1 月第 1 版第 1 次印刷

185mm × 260mm · 18.75 印张

标准书号: ISBN 978-7-111-45157-0

定 价: 35.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88378991 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzjsj@hzbook.com

Preface 前言

程序设计是高等学校计算机基础教育的重要内容和入门课程，C/C++ 语言以其功能丰富、表达能力强、应用面广等特点，在整个计算机基础教育课程设计中占有重要地位。本书是根据教育部《关于进一步加强高等学校计算机基础教学的意见》的基本要求，参考《高等学校计算机基础教学发展战略研究报告暨计算机基础课程教学基本要求》编写而成的。

本书注重对学生编程实践和问题求解能力的培养，以 Visual C++ 为平台，在学习 C/C++ 语言基础知识的同时，通过大量精选的例题和习题将程序设计的基本思想和方法介绍给学生。全书共分 12 章，涉及 C/C++ 的基本数据类型、运算符和表达式、程序控制结构、数组和字符串处理、函数和模块化程序设计、指针、结构体和共用体、面向对象程序设计基础等内容。前 9 章以面向过程程序设计方法为出发点，介绍 C 语言和程序设计的基础知识；第 10~12 章是面向对象程序设计方法基础，介绍了 C/C++ 语言中类和对象、继承和派生以及多态性等知识。其中，带 * 的节内容偏难，供老师选讲。全书在内容安排上实现了从结构化程序设计方法到面向对象程序设计方法的过渡，注重知识的系统性和连贯性。

全书由宋晓宇主编，其中杨艳春编写了第 1~2 章及附录 A~C，李世伟编写了第 3、6、7 章，宋晓宇编写了第 4、9、12 章，赵艳平编写了第 5、8 章，黎扬编写了第 10、11 章，高德勇参与编写了 8.5~8.10 节及 10.7 节并参与了书稿的校对工作。在本书的编写过程中，兰州交通大学电信学院软件工程系教师给予了大力的支持与帮助，在此表示衷心的感谢！

因编者水平有限，书中难免有错误和不妥之处，敬请专家和读者提出宝贵意见，编者邮箱：sxy9998@126.com。

编者

2013 年 11 月

教学建议 Suggestion

教学章节	教学要求	课时
第1章 绪论	了解程序设计方法及 C++ 语言的发展 理解算法的概念、熟练掌握流程图表示方法	2
	掌握 C++ 程序上机步骤	2
第2章 C++ 语言基础知识	掌握 C++ 语言的基本数据类型、运算符及表达式	6
第3章 顺序结构程序设计	掌握 C++ 语言数据的输入/输出 熟练掌握顺序结构程序设计	4
第4章 选择结构程序设计	熟练掌握 if 语句的三种形式、if 的嵌套 掌握多分支选择 switch 语句	6
第5章 循环结构程序设计	熟练掌握 while 循环、do-while 循环及 for 循环 掌握 break、continue 语句 掌握循环的嵌套	8
第6章 数组	熟练掌握一维数组、二维数组的定义及使用 掌握字符数组、字符变量的定义及使用	6
第7章 指针	掌握指针的定义及使用	2
	掌握用指针变量访问数组 了解指数组和多级指针	4
第8章 函数与编译预处理	掌握函数的定义及调用方法 理解变量的存储类别 掌握编译预处理命令的使用	8
第9章 自定义数据类型	掌握结构体变量、共用体的定义和使用 了解链表及其基本操作 掌握类型标识符 typedef 的使用	6
第10章 面向对象程序设计基础	掌握类的定义和对对象的声明 理解构造函数和析构函数 掌握类成员访问权限和访问方法	4
第11章 继承和多态	掌握类的继承及其实现、类的多态及其实现 掌握对象动态创建与撤销的方法	4
第12章 C++ 语言的流类库	了解 C++ 语言的流类库 掌握文件流、字符串流及其基本操作	2
总课时		64

说明:

- 1) 建议课堂教学全部在多媒体机房内完成, 实现“讲-练”结合。
- 2) 建议教学分为核心知识技能模块和技能提高模块, 其中核心知识技能模块建议教学学时为 64~80, 技能提高模块建议教学学时为 16~24, 不同学校可以根据各自的教学要求和计划学时数对教学内容进行取舍。

推荐阅读



Linux系统应用与开发教程 第2版

作者：刘海燕 等 ISBN：978-7-111-30474-6 定价：29.00元

Visual Basic程序设计教程 第3版

作者：邱李华 等 ISBN：978-7-111-33368-5 定价：33.00元

数据库原理与应用教程 第3版

作者：何玉洁 等 ISBN：978-7-111-31204-8 定价：29.80元

数据库原理及应用

作者：王丽艳 等 ISBN：978-7-111-40997-7 定价：33.00元

数据库与数据处理：Access 2010实现

作者：张玉洁 等 ISBN：978-7-111-40611-2 定价：35.00元

Visual C++ .NET程序设计教程 第2版

作者：郑阿奇 等 ISBN：978-7-111-40084-4 定价：36.80元

计算机网络教程（第2版）

作者：熊建强 等 ISBN：978-7-111-38804-3 定价：39.00元

C语言程序设计：问题与求解方法

作者：何勤 ISBN：978-7-111-40002-8 定价：36.00元

C语言程序设计：理论与实践

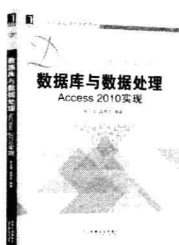
作者：孙浩 等 ISBN：978-7-111-36959-2 定价：35.00元

推荐阅读



大学计算机基础

作者：陈明 王锁柱 主编 ISBN：978-7-111-43767-3 定价：35.00元



数据库与数据处理：Access 2010实现

作者：张玉洁 孟祥武 编著 ISBN：978-7-111-40611-2 定价：35.00元



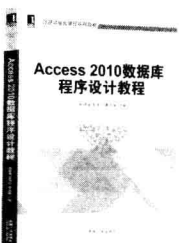
数据库原理及应用

作者：王丽艳 郑先锋 刘亮 编著 ISBN：978-7-111-40997-7 定价：33.00元



计算机网络教程 第2版

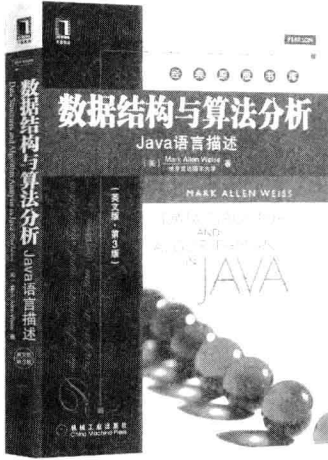
作者：熊建强 黄文斌 彭庆喜 主编 ISBN：978-7-111-38804-3 定价：39.00元



Access 2010数据库程序设计教程

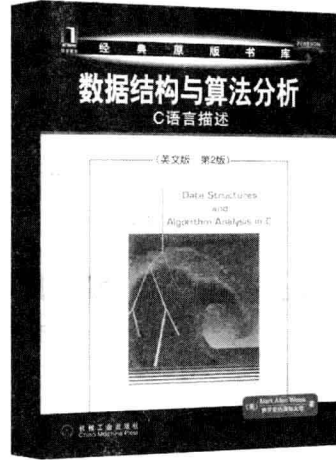
作者：熊建强 吴保珍 黄文斌 主编 ISBN：978-7-111-43681-2 定价：39.00元

推荐阅读



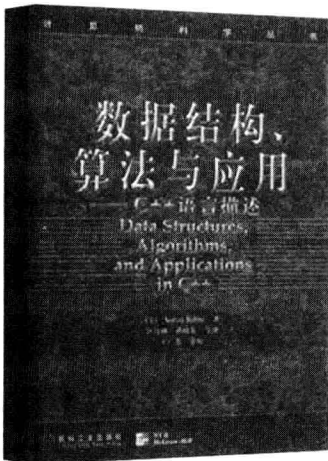
数据结构与算法分析：Java语言描述（英文版·第3版）

作者：Mark Allen Weiss ISBN：978-7-111-41236-6 定价：79.00元



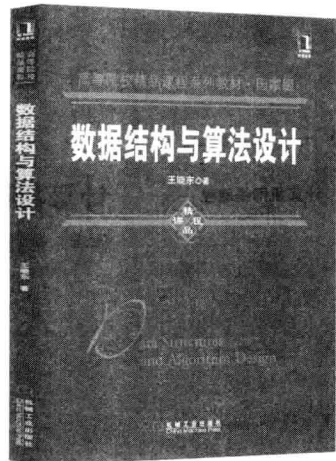
数据结构与算法分析：C语言描述（英文版·第2版）

作者：Mark Allen Weiss ISBN：978-7-111-31280-2 定价：45.00元



数据结构、算法与应用：C++语言描述

作者：Sartaj Sahni ISBN：7-111-07645-1 定价：49.00元



数据结构与算法设计

作者：王晓东 ISBN：978-7-111-37924-9 定价：29.00元

Contents 目 录

前 言	
教学建议	
第 1 章 绪论	1
1.1 程序设计语言概述	1
1.1.1 机器语言	1
1.1.2 汇编语言	2
1.1.3 高级语言	2
1.2 程序设计方法	3
1.2.1 结构化程序设计方法	3
1.2.2 面向对象程序设计方法	4
1.2.3 两种程序设计方法的比较	5
1.3 算法及其表示	5
1.3.1 算法	6
1.3.2 算法的表示方法	7
1.4 C++ 语言的发展及特点	12
1.4.1 C++ 语言的发展	12
1.4.2 C++ 语言的特点	12
1.5 C++ 语言程序	13
1.5.1 C++ 语言程序举例	13
1.5.2 C++ 语言程序的构成	15
1.6 C++ 语言集成开发环境	16
1.6.1 C++ 语言程序调试步骤	16
1.6.2 在 Visual C++ 6.0 环境中 开发 C++ 语言程序	17
1.6.3 Visual C++ 6.0 程序调试 常见错误	22
习题 1	23
第 2 章 C++ 语言基础知识	27
2.1 C++ 语言的字符集与词汇	27
2.1.1 C++ 语言字符集	27
2.1.2 C++ 语言词汇	27
2.2 C++ 语言的数据类型	28
2.3 常量与变量	29
2.3.1 变量	30
2.3.2 常量	32
2.4 运算符与表达式	35
2.4.1 算术运算符与算术表达式	37
2.4.2 关系运算符与关系表达式	39
2.4.3 逻辑运算符与逻辑表达式	40
2.4.4 逗号运算符与逗号表达式	41
2.4.5 条件运算符与条件表达式	42
2.4.6 赋值运算符与赋值表达式	43
2.4.7 其他运算符	44
2.5 数据类型转换	45
2.5.1 隐式类型转换	45
2.5.2 显式类型转换	46
2.5.3 赋值转换	46
习题 2	47
第 3 章 顺序结构程序设计	53
3.1 C++ 语言的语句	53
3.2 数据的输入与输出	54
3.2.1 输入/输出流的基本操作	54
3.2.2 输入/输出流的格式控制	56
3.3 顺序结构程序举例	60
习题 3	62
第 4 章 选择结构程序设计	65
4.1 if 语句	65
4.1.1 if 语句的省略格式	65
4.1.2 if-else 语句格式	66
4.1.3 if-else if-else 语句格式	68
4.1.4 if 语句的嵌套	72
4.1.5 if 语句与条件表达式的关系	75
4.1.6 if 语句程序举例	75
4.2 switch 语句	78

4.2.1	switch 语句格式	78	第7章 指针	143
4.2.2	switch 语句程序举例	80	7.1 指针和地址	143
4.2.3	if 语句与 switch 语句的 比较	83	7.1.1 指针变量的定义	144
习题4	83	7.1.2 指针变量的初始化	144
第5章 循环结构程序设计	89	7.1.3 指针变量的引用	144
5.1	概述	89	7.1.4 几种特殊的指针	147
5.2	while 语句	89	7.2 指针的运算	149
5.3	do-while 循环	94	7.3 指针和数组	150
5.4	for 语句	97	7.3.1 指向一维数组的指针变量	151
5.4.1	for 语句的基本形式	97	7.3.2 通过指针变量引用数组 元素	151
5.4.2	for 循环程序举例	99	7.3.3 指向多维数组的指针和 指针变量	152
5.4.3	三种循环的比较	103	7.4 利用字符指针处理字符串	155
5.4.4	三种循环语句的选择	103	7.5 二级指针和指针数组	156
5.5	break 和 continue 语句	104	7.5.1 二级指针	156
5.5.1	break 语句	104	7.5.2 指针数组	156
5.5.2	continue 语句	106	习题7
5.6	循环的嵌套	107	第8章 函数与编译预处理
习题5	112	8.1 函数	164
第6章 数组	120	8.1.1 函数的定义	165
6.1 一维数组	120	8.1.2 函数的调用	167	
6.1.1	一维数组的定义	120	8.1.3 函数的参数	167
6.1.2	一维数组元素的引用	121	8.1.4 函数的返回值	168
6.1.3	一维数组的初始化	122	8.1.5 对被调函数的声明	168
6.1.4	一维数组的存储	123	8.2 参数传递方式	169
6.1.5	一维数组程序举例	123	8.2.1 值传递	169
6.2 二维数组	128	8.2.2 地址传递	171	
6.2.1	二维数组的定义	128	8.2.3 引用参数	174
6.2.2	二维数组元素的引用	129	8.3 函数程序举例	175
6.2.3	二维数组的初始化	129	8.4 函数的嵌套调用	176
6.2.4	二维数组程序举例	130	8.5 函数的递归调用	177
6.3 字符数组	132	*8.6 内联函数	179	
6.3.1	字符数组的定义	132	*8.7 函数的重载	180
6.3.2	字符数组的初始化	132	*8.8 函数模板	181
6.3.3	字符数组元素的引用	133	*8.9 带默认参数的函数	183
6.3.4	字符数组的输入/输出	133	8.10 指针函数和函数指针	184
6.3.5	字符串处理函数	135	8.10.1 指针函数	184
习题6	138		

8.10.2	函数指针	185	10.2.1	类和对象	233
8.11	变量的作用域和存储类型	186	10.2.2	属性和方法	233
8.11.1	局部变量	186	10.2.3	封装	234
8.11.2	全局变量	188	10.2.4	继承	234
8.11.3	变量的存储类别	190	10.2.5	多态	235
8.12	编译预处理	194	10.2.6	消息传递	235
8.12.1	宏定义	194	10.3	类的声明和对象的定义	235
8.12.2	文件包含	199	10.3.1	类的声明	235
8.12.3	条件编译	200	10.3.2	对象的定义	236
习题8		201	10.4	构造函数与析构函数	237
第9章 自定义数据类型		214	10.4.1	构造函数	237
9.1	结构体类型与结构体变量的定义	214	10.4.2	拷贝构造函数	239
9.1.1	结构体类型的声明	214	10.4.3	析构函数	240
9.1.2	结构体变量的定义	215	10.5	成员函数的声明方式	241
9.2	结构体变量的初始化与引用	215	10.5.1	内联成员函数声明方式	241
9.3	结构体数组	216	10.5.2	成员函数原型与函数体 定义分离的声明方式	241
9.4	指向结构体变量的指针	217	10.6	类成员的访问方式	243
9.5	链表	218	10.6.1	在类之内访问类成员	243
9.5.1	new、delete 运算符	218	10.6.2	this 指针	244
9.5.2	链表的概念	219	10.6.3	在类之外访问类成员	244
9.5.3	创建链表	220	10.7	静态成员	246
9.5.4	插入链表节点	220	10.7.1	静态数据成员	246
9.5.5	删除链表节点	221	10.7.2	静态成员函数	248
9.6	共用体和枚举类型	222	10.8	程序举例	249
9.6.1	共用体类型	222	习题10		251
9.6.2	枚举类型	223	第11章 继承和多态		253
9.7	类型定义符 typedef	224	11.1	继承与派生	253
9.7.1	用 typedef 定义数据类型	224	11.1.1	子类的声明	254
9.7.2	用 typedef 定义函数指针 类型	225	11.1.2	访问控制	255
习题9		225	11.1.3	子类的构造函数	256
第10章 面向对象程序设计基础		231	11.2	动态对象与对象指针	261
10.1	从面向过程到面向对象	231	11.2.1	对象的动态创建与 销毁	261
10.1.1	面向过程方法的基本 特征	231	11.2.2	父类指针指向子类对象的 问题	262
10.1.2	面向对象方法的产生和 发展	232	11.3	虚函数与多态	264
10.2	面向对象程序设计的原理和基本 概念	232	习题11		269

第 12 章 C++ 语言的流类库	270	12.3 字符串流	278
12.1 输入/输出流及流类库	270	12.3.1 字符串流对象的定义及 初始化	279
12.1.1 输入/输出流的概念	270	12.3.2 字符串流的输入与输出 ...	279
12.1.2 流类库	270	12.3.3 字符串流的赋值	280
12.2 文件流	271	12.3.4 字符串流的比较	280
12.2.1 文件	271	习题 12	281
12.2.2 定义文件流对象	272	附录 A ASCII 码表	285
12.2.3 文件的打开与关闭	272	附录 B C++ 语言的关键字	286
12.2.4 输出文件流	273	附录 C C++ 语言的常用库函数	287
12.2.5 输入文件流	274	参考文献	289
12.2.6 文件流定位	277		

【本章要点】

- 程序设计方法
- 算法及其表示
- C++ 语言程序开发过程及上机调试步骤

自然语言（如汉语）是人类交流和表达思想的工具，自从人类进入信息社会以来，计算机作为人类的得力助手进入了社会生活的方方面面。于是人类怎样驾驭计算机，如何与计算机进行“交流”也成为一个问题。其实，人与计算机“交流”，也需要借助于一种“语言”来实现，这种语言就是计算机语言，它是一种人与计算机双方都能理解的语言。计算机语言由一套固定的符号和语法规则组成。人们要利用计算机解决问题，就必须用计算机语言来告诉计算机“做什么”和“怎样做”，这个过程就是程序设计过程。因此，计算机语言是人与计算机进行信息交流的工具，也称为程序设计语言或编程语言。

1.1 程序设计语言概述

程序设计语言（programming language）是一组用来编写计算机程序的符号语言，它由一组预先定义的指令、数据类型、语法规则构成。人使用程序设计语言描述解决问题的方法，即编写程序，供计算机阅读和执行，达到借助计算机解决特定问题的目的。从计算机问世以来，程序设计语言也在不断地发展变化着，按其发展过程，计算机程序设计语言可分为机器语言、汇编语言和高级语言。

1.1.1 机器语言

机器语言是最早使用的编程语言，它的每条指令都是完全由 0 和 1 组成的二进制代码。机器语言对应计算机的指令系统或指令集，它是最底层的程序设计语言。机器语言编写的程序中，每一条指令都是二进制形式的指令代码，计算机硬件可以直接识别。机器语言是面向机器的，不同的计算机硬件（主要是 CPU）其机器语言是完全不同的，因此机器语言的通用性很差。

由于机器语言程序是直接针对计算机硬件编写的，因此它的执行效率比较高，可以充分发挥计算机的性能。但是，用机器语言编写程序，即使是一个简单的算式，也要用很多条指令才能表达出来，因而难以阅读、理解、记忆，且不易查错。

机器语言的特点：

- 1) 机器语言是计算机能够直接识别并执行的唯一一种语言。
- 2) 机器语言是直接面向机器的语言。

3) 机器语言的指令记忆、程序编写和阅读等都很困难。

1.1.2 汇编语言

在 20 世纪 50 年代中期,人们开始用一些“助记符”来代替机器语言指令中的操作符和操作数,这种用助记符表示的语言称为汇编语言。汇编语言是一种符号语言,它在一定程度上解决了机器语言指令难以记忆、编程困难的问题,但它和机器语言一样也是面向机器的。使用汇编语言编写的程序,计算机不能直接执行,必须借助一种工具,将它翻译成机器语言目标程序,这种工具就称为汇编程序,翻译的过程称为汇编。

使用机器语言和汇编语言都需要对计算机的内部结构有较深的了解,因此用它们编写程序不仅劳动强度大而且程序的通用性差。通常将这两种语言称为“计算机低级语言”。

汇编语言的优点:

1) 能够直接对硬件(寄存器、存储单元和 I/O 端口)进行操作,并利用硬件的特点和指令系统提供的各种寻址方式,编写出高质量的程序。这种程序占用内存少,执行速度快,因此多用于编写系统程序和实时处理程序,并经常被高级语言嵌入使用。

2) 能够直接对位、字节、字、字节串以及字串等数据类型进行操作,为数据处理提供了灵活手段。

汇编语言的缺点:

1) 使用汇编语言,需要程序员了解计算机的硬件结构和指令系统,因此增加了编程难度。

2) CPU 不同,其汇编指令也不相同,因此程序的通用性差。

3) 使用汇编语言编写的程序与数学模型之间的对应关系不直观。

1.1.3 高级语言

高级语言是相对于汇编语言而言的,它是接近自然语言和数学公式的编程语言,基本脱离了机器的硬件系统,用人们更易理解的方式编写程序。高级语言与计算机的硬件结构及指令系统无关,它有更强的表达能力,可方便地表示数据的运算和程序的控制结构,能更好地描述各种算法,而且容易学习,编写出的程序易于阅读、理解、修改,移植性和通用性较好,因而得到了迅速的普及和发展。

1957 年,第一种完全脱离机器硬件的高级语言 FORTRAN 问世了。几十年来,共有几百种高级语言出现,其中影响较大、使用较普遍的有 FORTRAN、COBOL、LISP、C、PROLOG、C++、VC、VB、Delphi、Java、C#等。

高级语言的出现使得计算机程序设计语言不再过度地依赖某种特定的机器或环境。这是因为高级语言在不同的平台上会被翻译成不同的机器语言程序,而不是直接被机器执行。计算机并不能直接接受和执行用高级语言编写的源程序,源程序在输入计算机时,通过“翻译程序”翻译成目标程序,计算机才能识别和执行。这种“翻译”通常有两种方式,即编译方式和解释方式。

编译方式是指在源程序执行之前,就将程序源代码“翻译”成目标代码(机器语言),因此其目标程序可以脱离语言环境独立执行,使用比较方便、效率较高。但应用程序一旦需要修改,必须先修改源代码,再重新编译生成新的目标文件才能执行。

解释方式是应用程序源代码一边由相应语言的解释器“翻译”成目标代码(机器语言),一

边执行,因此程序的执行效率较低,而且不生成可独立执行的目标文件,应用程序不能脱离程序的解释系统。但这种方式比较灵活,可以动态地调整、修改应用程序。

计算机语言的发展经历了从机器语言到结构化程序设计语言,再到面向对象程序设计语言的过程。面向过程的语言致力于用计算机能够理解的逻辑来描述需要解决的问题及解决问题的具体方法、步骤。用这类语言编程时,程序不仅要说明做什么,还要详细地告诉计算机如何做,即程序需要详细描述解题的过程、步骤和细节,面向过程的语言种类繁多,如 FORTRAN、BASIC、Pascal、Ada、C 等。20 世纪 60 年代中后期,软件越来越多,规模越来越大,而软件的生产基本上是各自为战,缺乏科学规范的系统规划与测试、评估标准。这种落后的软件生产方式无法满足迅速增长的计算机软件需求,从而导致软件开发与维护过程中出现一系列严重问题的现象。

20 世纪 80 年代初开始,在软件设计方法上,产生了一次革命,其成果就是面向对象的程序设计语言的出现。面向对象程序设计(Object Oriented Programming, OOP)语言与以往各种编程语言的根本区别是程序设计思维方法不同,面向对象程序设计可以更直接地描述客观世界存在的事物(即对象)及事物之间的相互关系。面向对象技术强调的基本原则是直接面对客观事物本身进行抽象并在此基础上进行软件开发,将人类的思维方式与表达方式直接应用在软件设计中。目前,典型的纯面向对象的语言有 Java 和 SmallTalk,典型的既面向过程又面向对象的混合型语言有 C++ 语言。需要指出的是,计算机程序设计语言的高级和低级并不说明语言的优劣,它只表明了这种语言离硬件的远近,同时也暗含了程序设计的难易和执行效率的高低。

高级语言具有如下特点:

1) 高级语言是完全独立于机器的通用语言。使用这种语言编写计算机程序,可以完全不考虑机器的结构特点,不必了解机器的指令系统,所编写的程序可以在各种机器上通用。

2) 高级语言是面向过程或面向对象的语言。用高级语言编写的程序与问题本身的数学模型之间有着良好的对应关系,因此程序的编写与阅读都很方便。

3) 计算机不能直接识别和执行高级语言源程序。使用高级语言编写的源程序,必须要经过翻译程序的处理,将之翻译成对应的机器语言程序,然后机器才能执行。

高级语言的缺点是翻译后生成的目标代码往往比较长,占用内存多,执行时间长,因此不适合于需要实时处理的应用。

1.2 程序设计方法

随着大容量存储器的出现以及计算机应用范围的扩大,程序的编制越来越困难,程序的大小以算数级数递增,而程序的逻辑控制难度则以几何级数递增,这样人们不得不考虑程序设计方法。在程序设计的过程中,除了要求程序逻辑正确,能被计算机理解并执行外,还涉及程序的可读性、可靠性、可维护性以及程序的效率等方面的问题。在改善整个程序设计过程、提高程序质量等方面,程序设计方法都起着重要的作用。在软件的发展过程中,涌现出很多的程序设计方法,下面主要讨论两种程序设计方法。

1.2.1 结构化程序设计方法

1. 结构化程序设计的总体思想

结构化程序设计又称为面向过程的程序设计,它的总体思想是采用模块化结构,自顶向下、

逐步求精。即首先把一个复杂的大问题分解为若干相对独立的小问题，如果小问题仍较复杂，则可以把这些小问题再继续分解成若干子问题，这样不断地分解，使得小问题或子问题简单到能够直接用程序的三种基本控制结构表达为止；然后，对应每一个小问题或子问题编写出一个功能上相对独立的程序块来，这种程序块被称为模块，每个模块各个击破，最后再统一组装，这样，对一个复杂问题的解决就变成了对若干个简单问题的求解。这就是自顶向下、逐步求精的程序设计方法。

确切地说，模块是程序对象的集合，模块化就是把程序划分成若干个模块，每个模块完成一个确定的子功能，把这些模块集中起来组成一个整体，就可以完成对问题的求解。这种用模块组装起来的程序被称为模块化结构程序。在模块化结构程序设计中，采用自顶向下、逐步求精的设计方法便于问题的分解和模块的划分，所以，它是结构化程序设计的基本原则。

2. 结构化程序设计的特征

结构化程序设计的特征有：

- 1) 以三种基本控制结构的组合来描述程序。
- 2) 整个程序采用模块化结构。
- 3) 有限制地使用 goto 转移语句。
- 4) 以控制结构为单位，每个控制结构只有一个入口、一个出口，各单位之间接口简单、逻辑清晰。
- 5) 采用一定的书写格式使程序结构清晰，易于阅读。

3. 结构化程序设计的缺点

在结构化程序设计中，整个程序的功能是通过模块之间的相互调用实现的。在实际应用中存在一些缺点：

- 1) 将数据和数据处理过程分离成相互独立的实体，当数据结构发生变化时，所有相关的处理都要进行相应的修改，因此，程序代码的可重用性较差。
- 2) 对于图形用户的应用，开发起来比较困难，而图形界面越来越被人们广泛使用。
- 3) 用户的要求难以在系统分析阶段准确定义，从而使系统在交付使用时产生许多问题。

1.2.2 面向对象程序设计方法

在程序设计方法的发展过程中，每一次重大突破都使得程序员可以应对更大的复杂性问题。在这条道路上迈出的每一步中，新的方法都运用和发展了以前的方法中最好的理念。今天，许多项目的规模又进一步发展，为了解决这个问题，面向对象程序设计方法应运而生。发明面向对象程序设计方法的主要出发点是弥补结构化程序设计方法中的一些缺点。面向对象程序设计方法把数据看做程序开发中的基本元素，并且不允许它们在系统中自由流动。它将数据和操作这些数据的函数紧密联系在一起，并保护数据不会被外界的函数意外修改。面向对象程序设计在程序设计方法中是一个新的概念，对于不同的人可能意味着不同的内容。面向对象程序设计的定义是“面向对象程序设计是一种方法，这种方法为数据和函数提供共同的独立内存空间，这些数据和函数可以作为模板以便在需要时创建类似模块的副本。”从以上定义可以看到，一个对象被认为是计算机内存中的一个独立区间，在这个区间中保存着数据和能够访问数据的一组操作。因为内存区间是相互独立的，所以对象可以不经修改就应用于多个不同的程序中。

面向对象程序设计方法以对象为基础，它最主要的特点和成就是利用特定的软件工具直接

完成从对象客体的描述到软件结构之间的转换。与结构化程序设计不同，面向对象程序设计方法是一种运用对象、类、继承、封装、消息传递、多态性等概念来构造系统的软件开发方法。它的应用解决了传统结构化开发方法中客观世界描述工具与软件结构的不一致性问题，缩短了开发周期，解决了从分析和设计到软件模块结构之间多次转换映射的繁杂过程。

1.2.3 两种程序设计方法的比较

面向对象程序设计方法克服了结构化程序设计中存在的问题。在面向对象程序设计方法出现以前，结构化程序设计方法是程序设计的主流。在结构化程序设计方法中，问题被看做一系列需要完成的任务，函数（在此泛指例程、函数、过程）用于完成这些任务，解决问题的焦点集中于函数。其中函数是面向过程的，即它关注如何根据规定的条件完成指定的任务。在多函数程序中，许多重要的数据被放置在全局数据区，这样它们可以被所有的函数访问。每个函数都可以具有它们自己的局部数据。这种结构很容易造成全局数据在无意中被其他函数改动，因而程序的正确性不易保证。面向对象程序设计方法的出发点之一就是弥补结构化程序设计方法中的缺点，即对象是程序的基本元素，它将数据和操作紧密地联系在一起，并保护数据不会被外界的函数意外修改。

表 1-1 在基本思想、代表语言等方面进一步比较了结构化程序设计方法与面向对象的程序设计方法的异同。

表 1-1 结构化程序设计方法与面向对象的程序设计方法比较

项目	结构化设计方法	面向对象设计方法
基本思想	自顶向下设计过程库，逐步求精，分而治之	自底向上设计类库
代表语言	BASIC、FORTRAN、C 等	Visual Basic、Java、C++ 等
处理问题的出发点	面向过程	面向问题
控制程序的方式	通过设计调用或返回程序	通过“事件驱动”来激活和运行程序
可扩展性	功能变化会危及整个系统，扩展性差	只需修改或增加操作，而基本对象结构不变，扩展性好
重用性	不好	好
层次结构的逻辑关系	用模块的层次结构概括模块和模块之间的关系和功能	用类的层次结构来体现类之间的继承和发展
运行效率	相对高	相对低

1.3 算法及其表示

计算机是一种具有内部存储能力的自动、高效的电子设备，它最本质的使命就是执行指令所规定的操作。如果我们需要计算机完成什么工作，只要将其步骤用诸条指令的形式描述出来，并把这些指令存放在计算机的内存中，机器就会自动逐条顺序执行指令，全部指令执行完成就得到预期的结果。这种可以被连续执行的一条条指令的集合称为计算机程序。也就是说，程序是计算机指令的序列，编制程序的过程就是为计算机安排指令序列。

一个程序应包括以下两方面内容：

- 对数据的描述，即在程序中要指定数据的类型和数据的组织形式。
- 对数据的操作，即操作步骤，也就是算法。