

计 算 机 科 学 与 技 术 系 列 教 材

面向对象程序设计 (C++)

主 编 张 俊 王 庆 春



WUHAN UNIVERSITY PRESS
武汉大学出版社

计算机科学与技术系列教材

面向对象程序设计 (C++)

主 编 张 俊 王庆春
副主编 吕 品 王 邯 吕 涛



WUHAN UNIVERSITY PRESS

武汉大学出版社

图书在版编目(CIP)数据

面向对象程序设计(C++)/张俊,王庆春主编. —武汉:武汉大学出版社,2009.7

计算机科学与技术系列教材

ISBN 978-7-307-07069-1

I. 面… II. ①张… ②王… 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(2009)第087766号

责任编辑:支 笛 责任校对:刘 欣 版式设计:支 笛

出版发行:武汉大学出版社 (430072 武昌 珞珈山)

(电子邮件:cbs22@whu.edu.cn 网址:www.wdp.com.cn)

印刷:湖北睿智印务有限公司

开本:787×1092 1/16 印张:22.25 字数:557千字

版次:2009年7月第1版 2009年7月第1次印刷

ISBN 978-7-307-07069-1/TP·335 定价:26.00元

版权所有,不得翻印;凡购买我社的图书,如有缺页、倒页、脱页等质量问题,请与当地图书销售部门联系调换。

计算机科学与技术系列教材

编 委 会

主 任：何炎祥，武汉大学计算机学院院长，教授

副 主 任：康立山，中国地质大学（武汉）计算机学院院长，教授

陆际光，中南民族大学计算机科学学院院长，教授

编 委：（以姓氏笔画为序）

王江晴，中南民族大学计算机科学学院副院长，教授

王春枝，湖北工业大学计算机学院副院长，教授

牛冀平，黄冈师范学院计算机系主任，副教授

石曙东，湖北师范学院计算机科学与技术系主任，教授

朱 英，桂林电子工业学院计算机系副教授

孙扬波，湖北中医学院信息技术系信息管理与信息系统教研室主任

刘腾红，中南财经政法大学信息学院副院长，教授

陈少平，中南民族大学电信学院副院长，教授

杜友福，长江大学计算机科学学院院长，教授

陆 迟，江汉大学数学与计算机科学学院计算机系主任，副教授

闵华松，武汉科技大学计算机科学与技术学院副院长，副教授

陈佛敏，咸宁学院信息工程学院计算机系主任，副教授

陈建新，孝感学院计算机科学系主任，副教授

李禹生，武汉工业学院计算机与信息工程系副主任，教授

李晓林，武汉工程大学计算机科学与工程学院副院长，副教授

张焕国，武汉大学计算机学院教授

张唯佳，湖北省信息产业厅信息化推进处处长

余敦辉，湖北大学数学与计算机科学学院计算机系副主任

肖 微，湖北警官学院信息技术系副教授

钟 珞，武汉理工大学计算机科学与技术学院院长，教授

钟阿林，三峡大学电气信息学院计算机系主任

姜洪溪，襄樊学院电气信息工程系副主任，副教授
桂 超，湖北经济学院计算机与电子科学系副主任，副教授
黄求根，武汉科技学院计算机科学学院院长，教授
阎 菲，湖北汽车工业学院计算中心主任，副教授
韩元杰，桂林电子工业学院计算机系教授
谢坤武，湖北民族学院信息工程学院计算机系主任，副教授
戴光明，中国地质大学（武汉）计算机学院副院长，教授
魏中海，华中农业大学理学院计算机系副教授

执行编委：林 莉，武汉大学出版社计算机图书事业部主任
支 笛，武汉大学出版社计算机图书事业部编辑

内 容 简 介

本书综合考虑了“关于进一步加强高等学校计算机基础教学意见”中 C++ 语言程序设计基础的大纲要求，以及中国计算机科学与技术学科教程和计算机学科专业规范中关于程序设计基础、算法和复杂性、程序设计语言、软件工程领域中的相关知识单元要求，并结合多年来在面向对象程序设计和 C++ 语言教学实践中的经验编写而成。

本书以面向对象程序设计思想和方法为主线，以 C++ 语言为载体，详细讲述面向对象程序设计与 C++ 语言中的主要内容：面向对象思想和方法、数据类型及基本计算、函数与程序结构、类与对象、运算符重载、模板与 STL、继承与派生、虚函数与多态性、C++ 的 I/O 流、异常处理等。全书共分为 10 章，体系编排完整，内容结构合理，例题丰富，注重反映 C++ 语言的新规范、新技术和新发展。

本书以培养设计、分析、建模和应用能力为目的，遵循计算机学科专业规范要求，适合作为计算机科学与技术及相关专业的课程教材，也可供读者自学使用和参考。

前言

面向对象程序设计思想和方法是现今主流的程序设计方法，它能够更好地对现实世界中的各种数据、概念进行建模和抽象。此外，应用面向对象程序设计思想和方法，能够更好地组织和管理大型程序项目，并有利于继承发展程序设计领域中的各种杰出的智慧和闪亮的思想，例如各种程序库和设计模式。

C++语言作为当今最流行的一种高级程序设计语言，它完全兼容 C 语言，支持结构化程序设计，同时也支持面向对象程序设计。C++程序设计在运行效率、语法与语义、组件与类库、代码与资源等方面都有着其他程序语言所不能比拟的显著优势。

本书综合考虑了“关于进一步加强高等学校计算机基础教学意见”中 C++语言程序设计基础的大纲要求，以及中国计算机科学与技术学科教程和计算机学科专业规范中关于程序设计基础、算法和复杂性、程序设计语言、软件工程领域中的相关知识单元要求，并结合多年来在面向对象程序设计和 C++语言教学实践中的经验编写而成。

本套教材分为《面向对象程序设计(C++)》和《面向对象程序设计(C++)学习与实践》。《面向对象程序设计(C++)》教材以 C++语言为载体，结合 C++语言的各种特性和要点，重点讲授面向对象程序设计的思想和方法。《面向对象程序设计(C++)学习与实践》是与《面向对象程序设计(C++)》配套的教材，包括模拟试题、试验指导、程序调试基础三部分。本套教材以面向对象方法和 C++语言的学习与实践为主，重在培养学生的分析、设计、抽象和应用能力。

本教材体系编排完整，内容结构合理，例题丰富，对于面向对象和 C++程序设计中的重要内容，如面向对象思想和方法、数据类型及基本计算、函数与程序结构、类与对象、运算符重载、模板与 STL、继承与派生、虚函数与多态性、C++的 I/O 流、异常处理等，都有详细讲解，同时强调重要概念。各章节所选择的例题贴合重点，同时注重反映 C++语言的新规范、新技术和新发展。本教材中的示例程序全部在 Visual C++ 2005 上调试通过，程序代码符合 C++标准。

本书第 1、2 章由王庆春编写，第 3、4 章由吕品编写，第 5、6 章由张俊编写，第 7、8 章由王邯编写、第 9、10 章由吕涛编写。在本书编写过程中，得到了张彦铎教授、江世宏教授的热情指导，他们根据自己丰富的教学经验提出了大量宝贵的意见，在此表示衷心的感谢！同时感谢李晓林、王忠、姬涛等老师的热情支持！

虽然经过了多年的教学实践，但是由于计算机科学与技术的迅速发展，加之编者水平有限，本教材编写中难免存在错误和不足之处，我们诚恳期待并接受读者的批评和指正，以供今后进一步完善。

编 者

2008 年 12 月

目 录

第1章 面向对象程序设计概述	1
1.1 程序设计概述	1
1.1.1 计算机系统	1
1.1.2 算法与程序	4
1.1.3 程序设计	8
1.2 面向对象程序设计	11
1.2.1 面向对象程序设计的基本特点	11
1.2.2 面向对象的软件工程	13
1.2.3 UML(Unified Modeling Language)	14
本章小结	16
习题1	16
第2章 数据类型及其运算	18
2.1 基本数据类型与基本运算	18
2.1.1 基本数据类型	18
2.1.2 基本输入/输出	22
2.1.3 表达式与基本运算	24
2.1.4 应用举例	32
2.2 语句与控制结构	34
2.2.1 语句	34
2.2.2 控制结构	34
2.2.3 转向语句	40
2.2.4 应用举例	43
2.3 复合数据类型	44
2.3.1 数组	44
2.3.2 指针	48
2.3.3 引用	53
2.3.4 字符串	54
2.3.5 应用举例	56
本章小结	57
习题2	58
第3章 函数与程序结构	60

3.1 函数	60
3.1.1 函数概述	60
3.1.2 C++新增的函数机制	62
3.1.3 函数嵌套调用及递归调用	69
3.1.4 函数应用	74
3.1.5 应用举例	82
3.2 存储类、作用域和程序结构	85
3.2.1 变量的存储类	85
3.2.2 标识符的作用域	88
3.2.3 编译预处理	90
3.2.4 程序的多文件结构	92
3.2.5 应用举例	92
本章小结	94
习题 3	95

第 4 章 类与对象	96
4.1 从结构到类	96
4.1.1 结构类型概述	96
4.1.2 结构类型与结构变量定义	97
4.1.3 结构应用	99
4.2 类的定义	104
4.2.1 类的定义	104
4.2.2 定义对象	105
4.2.3 访问类的成员	106
4.2.4 类的程序结构	108
4.3 对象的定义	109
4.3.1 构造函数	109
4.3.2 析构函数	112
4.3.3 三个特殊的构造函数	113
4.3.4 this 指针	121
4.3.5 应用举例	124
4.4 类的复合	126
4.4.1 类的复合	126
4.4.2 应用举例	129
4.5 const 关键字	130
4.5.1 常见用法	130
4.5.2 应用举例	132
4.6 new/delete 运算符	133
4.6.1 基本用法	133
4.6.2 高级用法	136



4.7 友元和静态	138
4.7.1 友元函数和友元类	138
4.7.2 静态成员	141
4.7.3 应用举例	144
本章小结	145
习题 4	147
第 5 章 运算符重载	149
5.1 概述	149
5.1.1 基本概念	149
5.1.2 运算符重载的语法规则	151
5.1.3 运算符重载的形式	152
5.2 成员函数形式的运算符重载	155
5.2.1 复数类 Complex	155
5.2.2 有理数类 Rational	163
5.3 友元函数形式的运算符重载	169
5.3.1 友元函数形式的语法	169
5.3.2 重载流插入运算符和提取符	174
5.4 几个运算符的重载	176
5.4.1 重载增量/减量运算符	176
5.4.2 重载下标运算符	179
5.4.3 重载函数调用运算符	181
5.4.4 重载转换运算符	184
5.5 综合应用举例	186
本章小结	193
习题 5	195
第 6 章 模板与 STL	196
6.1 模板概述	196
6.1.1 数据类型的参数化	196
6.1.2 模板的初认识	198
6.2 函数模板	199
6.2.1 函数模板的定义	199
6.2.2 函数模板的实例化	203
6.2.3 函数模板的重载	206
6.2.4 函数模板对数据类型的需求	207
6.3 类模板	209
6.3.1 类模板的定义	209
6.3.2 类模板的实例化	214
6.4 STL 概述	216

6.4.1 STL 及其组件	216
6.4.2 常用头文件	217
6.5 函数对象与算法	218
6.5.1 函数对象	218
6.5.2 STL 算法	221
6.6 容器	227
6.6.1 string 类	227
6.6.2 vector 容器类	230
6.6.3 list 容器类	232
本章小结	235
习题 6	236
第 7 章 继承与派生	237
7.1 基本概念	237
7.1.1 概述	237
7.1.2 继承的机制	240
7.1.3 继承与复合	241
7.1.4 继承的语法	243
7.1.5 应用举例	247
7.2 访问控制	249
7.2.1 继承的三种方式	249
7.2.2 公有继承应用举例	252
7.3 派生类的构造函数	253
7.3.1 派生类的构造函数	253
7.3.2 派生类对象构造和析构的顺序	259
7.4 几个问题	263
7.4.1 冲突: 成员名限定法	263
7.4.2 成员覆盖: 支配规则	264
7.4.3 赋值兼容规则	266
7.4.4 虚基类	271
7.4.5 类模板的继承与派生	273
7.4.6 应用举例	274
本章小结	277
习题 7	278
第 8 章 虚函数与多态性	280
8.1 概述	280
8.1.1 程序关联的两种方式	280
8.1.2 问题的引出	281
8.1.3 多态性	282

8.2 虚函数	283
8.2.1 虚函数的定义	284
8.2.2 虚析构函数	290
8.2.3 应用举例	291
8.3 抽象类	292
8.3.1 纯虚函数	292
8.3.2 抽象类	294
8.3.3 应用举例	295
本章小结	297
习题 8	298
第 9 章 C++ 的 I/O 流	300
9.1 概述	300
9.1.1 流与缓冲区	300
9.1.2 ios 基类	301
9.1.3 标准流对象	301
9.2 标准 I/O 流	302
9.2.1 标准输出流	302
9.2.2 标准输入流	305
9.3 格式化 I/O	310
9.3.1 流格式状态标志位	310
9.3.2 成员函数	312
9.3.3 流操纵算子	313
9.4 文件 I/O 流	314
9.4.1 基本概念	314
9.4.2 文件操作	315
9.4.3 应用举例	320
9.5 字符串 I/O 流	323
9.5.1 基本概念	323
9.5.2 字符串输出流	323
9.5.3 字符串输入流	324
本章小结	325
习题 9	326
第 10 章 异常处理	327
10.1 概述	327
10.2 C++ 的异常处理	327
10.2.1 C++ 异常处理基础	328
10.2.2 异常处理的执行过程	329
10.2.3 异常的类型匹配规则	330



10.2.4 异常规范	331
10.2.5 异常的重新抛出	332
10.3 异常处理与构造函数和析构函数	333
10.3.1 异常抛出时局部对象的释放	333
10.3.2 在构造函数中抛出异常	334
10.3.3 在析构函数中抛出异常	335
10.4 C++标准库中的异常处理	336
10.4.1 exception 基类	336
10.4.2 逻辑异常与运行时异常	337
本章小结	338
习题 10	339
参考文献	340



第1章 面向对象程序设计概述



面向对象程序设计是一种应用对象、类、继承、封装、消息传送、多态性等概念来构造软件的开发方法。面向对象设计作为一种主流的软件开发思想和方法，已经引起人们越来越广泛的关注和重视。面向对象方法是一种把面向对象的思想应用于软件开发的过程中，指导软件开发活动，建立在“对象”概念基础上的系统方法学。

通过本章的学习，应了解计算机系统的组成，了解算法与程序的一些概念，了解程序设计的思想和方法。要求掌握面向对象方法学的概念和特点，掌握类、对象、实例、方法等面向对象的基本概念。

1.1 程序设计概述

1.1.1 计算机系统

一个完整的计算机系统是由计算机硬件系统和计算机软件系统两大部分构成。计算机硬件是由计算机系统电子、机械等物理部件所构成的各种计算机的实体部件和设备的总称，是计算机完成各项工作的物质基础，是看得见、摸得着的，是实实在在存在的实体。计算机软件是在计算机硬件设备上运行的各种程序及相关文档的总称。计算机系统的组成如图 1.1 所示。

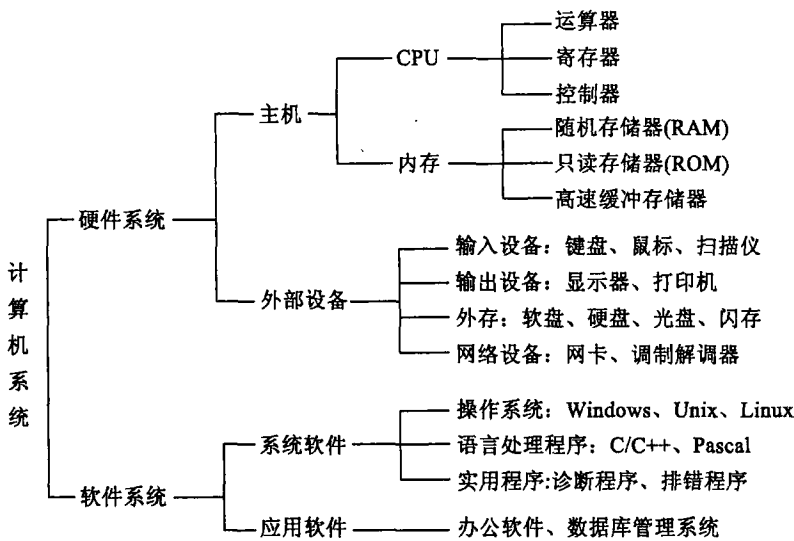


图 1.1 计算机系统

没有安装任何软件的计算机通常称为“裸机”，裸机是无法工作的。反之，如果计算机硬件脱离了计算机软件，那么它就成了一些无任何功能的电子、机械设备。所以，计算机硬件与计算机软件之间是相互支持、相互促进发展的。

1. 计算机硬件系统

计算机硬件系统是由构成计算机所有实体部件的集合。虽然计算机的种类繁多，功能差异也较大，但它们的硬件组成及基本结构是相似的。计算机硬件系统的组成通常是由运算器、存储器、控制器和输入/输出设备五大部分组成。如图 1.2 所示。

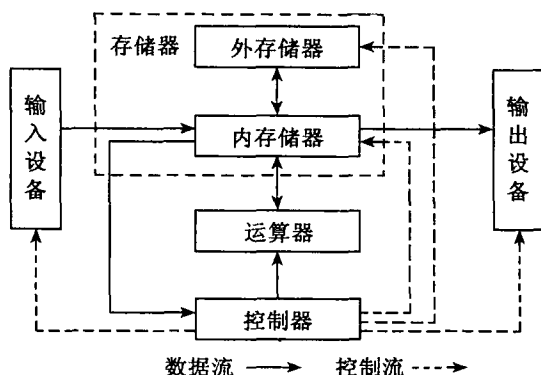


图 1.2 计算机硬件系统

(1) 运算器

运算器是用于加工、处理数据的部件，主要完成对数据的算术运算和逻辑运算。其中，算术运算包括：加、减、乘、除及它们的复合运算。逻辑运算包括：比较运算、逻辑加运算、逻辑乘运算、逻辑反运算等。运算器通常由若干个算术逻辑单元 ALU 和一系列的寄存器组成。

(2) 存储器

存储器是计算机的记忆部件，它的主要功能是存放程序和数据。程序和数据在计算机中以二进制（“0”或“1”）的形式存放在存储器中。存储器是由许多存储单元组成的，存储器中将一个字节的 8 位称为一个存储单元。每个存储单元对应一个编号，用二进制编码表示，称为存储单元地址。向存储器中存数据或者从存储器中取数据都是通过地址找到相应的存储单元来进行的。存储器所存储的总数称为存储容量，通常用单位 KB 来表示，存储容量越大，表示计算机记忆储存的信息就越多。

(3) 控制器

控制器是计算机的控制部件，它控制各部件协调统一的工作，并使整个机器连续地、有条不紊地完成对指令的分析和执行。控制器每次从存储器读取一条指令，经过分析，产生一串操作命令，发向各个部件，进行相应的操作。接着从存储器取出下一条指令，再执行这条指令，从而完成控制各部件协调统一的工作。通常把取指令的一段时间叫做取指周期，而把执行指令的一段时间叫做执行周期。因此，控制器反复交替地处在取指周期与执行周期之中，直至程序执行完毕。

(4) 输入/输出设备



输入设备是从计算机外部向计算机内部输入信息的部件,其功能是将程序、数据及相关信息转换成计算机能接收并识别的信息形式。目前常用的输入设备是键盘、鼠标器、扫描仪等。

输出设备是将计算机的处理结果传送到外部供计算机用户使用的设备,其功能是将计算机内部的二进制形式的相关信息转换成人们所能识别的信息形式,如字符、文字、图形、图像、声音等。目前常用的输出设备有显示器、打印机、绘图仪等。

计算机的输入/输出设备通常称为 I/O 设备。

2. 计算机软件系统

计算机软件是由程序和有关的文档组成,程序是指令序列的符号表示,文档是软件开发过程中建立的技术资料。程序是软件的主体,一般保存在存储介质(如软盘、硬盘和光盘)中,以便在计算机上使用。文档对于使用和维护软件尤其重要,随着软件产品发布的文档主要是使用手册,其中包含了该软件产品的功能介绍、运行环境要求、安装方法、操作说明和错误信息说明等。某个软件要求的运行环境是指运行它至少应有的硬件和其他软件的配置,也就是说,在计算机系统层次结构中,它是该软件的下层(内层)至少应有的配置(包括对硬件的设备和指标要求、软件的版本要求等)。

系统软件是管理、监控和维护计算机资源的软件,是用来扩大计算机的功能、提高计算机的工作效率、方便用户使用计算机的软件,人们借助于软件来使用计算机。系统软件是计算机正常运转不可缺少的,一般由计算机生产厂家或专门的软件开发公司研制,出厂时写入 ROM 芯片或存入磁盘(供用户选购)。任何用户都要用到系统软件,其他程序都要在系统软件的支持下运行。计算机软件按用途可分为系统软件和应用软件。

(1) 系统软件

系统软件主要包括:操作系统软件(软件的核心)、各种语言处理程序、服务性程序(支撑软件)和数据库管理系统等。

操作系统是系统软件的核心,是由指挥与管理计算机系统运行的程序模块和数据结构所组成的一种大型的软件系统,是管理计算机软件、硬件资源的一个平台,没有它,任何计算机都无法正常运行。其功能是管理计算机的软硬件资源和数据资源,为用户提供高效、全面的服务。在计算机发展史上曾出现过许多不同的操作系统,其中最为常见的操作系统有: DOS、Windows、Linux、Unix 和 OS/2 等。

语言处理系统包括机器语言、汇编语言和高级语言等。这些语言处理程序除个别常驻在 ROM 中可以独立运行外,其他都必须在操作系统的支持下运行。

机器语言是指机器能直接识别的语言,也是早期使用的一种语言。它是由“1”和“0”组成的一组程序代码指令,输入计算机后能直接运行,但由于机器语言比较难记,因此,就产生了汇编语言。

汇编语言是由一组与机器语言指令一一对应的符号指令和简单语法组成的语言,它用符号来表示。汇编语言程序需要由一种“翻译”程序来将它翻译成机器语言程序,这种翻译程序称为汇编程序。由于汇编语言与具体的机器密切相关,因此,就产生了高级语言。

高级语言比较接近日常用语,对具体机器依赖性低,可适用于各种机器,是人们普遍使用的语言。同样,高级语言程序也需要由一种“翻译”程序来将它翻译成机器语言程序才能运行。常用的高级语言有: C、C++等,目前高级语言种类已达数十种。

数据库管理系统是以一定的组织方式进行组织和存储并具有相关性的数据集合系统。数

数据库管理系统是在计算机上实现数据库技术的系统软件，由它来实现用户对数据库的建立、管理、维护和使用等。目前比较流行的数据库管理系统软件有 SQL Server 和 Oracle 等。

(2) 应用软件

应用软件主要包括：用户程序与应用软件包。

用户程序是为了解决用户的特定问题而开发的软件，如：票务管理系统、人事管理系统、财务管理系统等。同时，编制用户程序应充分利用计算机系统的种种现成软件，在系统软件和应用软件包的支持下可以更加方便、有效地研制用户专用程序，随着计算机应用领域的不断扩展而与日俱增。

应用软件包是为实现某种特殊功能而经过精心设计的、结构严密的独立系统，是一套满足同类应用的许多用户所需要的软件。如：Microsoft 公司发布的 Office 家族应用软件包就包含了 Word（字处理）、Excel（电子表格）、PowerPoint（幻灯片）、Access（数据库管理）等应用软件，是实现办公自动化的很好的应用软件包。还有日常使用的杀毒软件瑞星、金山毒霸以及各种游戏软件等。

1.1.2 算法与程序

算法是在有限步骤内求解某一问题所使用的一组定义、求解步骤和明确的求解规则，即计算机解题的过程。在这个过程中，无论是形成解题的思路还是写程序，都是在实施某种算法，即实现推理的算法或实现操作的算法。算法是程序设计的精髓，程序设计的实质就是构造解决问题的算法。总之，算法与程序可用下列公式来表示：

程序=算法+数据结构+程序设计方法+语言工具与环境

1. 算法

计算机解题的过程即为算法，因此，一个算法应该具有以下五个重要的特征：

- ①确切性：算法的每一步骤必须有确切的定义和规则。
- ②可行性：算法原则上能够精确地运行，即可行性。
- ③有穷性：一个算法必须保证执行有限步之后正确结束。
- ④输入：一个算法有 0 个或多个输入，以刻画运算对象的初始情况。
- ⑤输出：一个算法有一个或多个输出，能反映对输入数据加工后的结果。没有输出的算法是毫无意义的。

算法的通常表示方法有：自然语言、程序语言、流程图、伪代码等。

(1) 自然语言

自然语言就是人们日常使用的语言，可以是汉语、英语或其他语言。用自然语言表示通俗易懂，但文字冗长，容易出现“歧义性”。自然语言表示的含义往往不严格，要根据上下文才能判断其正确含义。此外，用自然语言描述包含分支和循环的算法，不很方便。因此，除了很简单的问题以外，一般不用自然语言描述算法。

(2) 程序语言

程序语言是用于编写计算机程序的语言。语言的基础是一组记号和一组规则。根据规则由记号构成的记号串的总体就是程序语言。在程序设计语言中，这些记号串就是程序。程序设计语言包含三个方面，即语法、语义和语用。语法表示程序的结构或形式，亦即表示构成程序的各个记号之间的组合规则，但不涉及这些记号的特定含义，也不涉及使用者。语义表示程序的含义，亦即表示按照各种方法所表示的各个记号的特定含义，但也不涉及使用者，