

可下载教学资料

<http://www.tup.tsinghua.edu.cn>



高等学校教材
计算机应用

面向对象程序设计 与 Visual C++ 6.0 教程

陈天华 编著

清华大学出版社



高等学校教材
计算机应用

面向对象程序设计 与 Visual C++ 6.0 教程

陈天华 编著

清华大学出版社
北京

内 容 简 介

本书将 C++ 语言和应用 Visual C++ 6.0 设计 Windows 应用程序紧密结合在一起,全面系统地讲述了 C++ 语言的基本概念、语法和面向对象程序设计的方法及应用,并对 C++ 面向对象语言的抽象性、封装性、继承性与多态性进行了全面介绍,内容包括 C++ 语法、函数、类与对象、数组与指针、继承与派生、多态性、模板、I/O 流库及异常处理机制。在此基础上,介绍了用 Visual C++ 6.0 开发 Windows 应用程序的基本原理与概念,以及各种典型 Windows 应用程序的设计方法。各章均配有丰富的例题和习题。内容安排上循序渐进,讲解深入浅出,力求突出重点、面向应用、提高能力、解决问题。

为便于学习,《面向对象程序设计与 Visual C++ 6.0 教程题解与实验指导》将与本书配套出版。本书可作为高等院校计算机、电子技术、通信、信息工程、自动化、电气及相关专业的面向对象程序设计课程教材,也可以作为工程技术人员或其他相关人员的参考书。

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

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。

本书防伪标签采用特殊防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

面向对象程序设计与 Visual C++ 6.0 教程/陈天华编著. —北京:清华大学出版社,2006.1

(高等学校教材·计算机应用)

ISBN 7-302-12311-X

I. 面… II. 陈… 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(2005)第 160100 号

出版者:清华大学出版社 地 址:北京清华大学学研大厦
http://www.tup.com.cn 邮 编:100084
社总机:010-62770175 客户服务:010-62776969

组稿编辑:郑寅莹

文稿编辑:陶萃渊

印装者:清华大学印刷厂

发行者:新华书店总店北京发行所

开 本:185×260 印张:27 字数:671 千字

版 次:2006 年 1 月第 1 版 2006 年 1 月第 1 次印刷

书 号:ISBN 7-302-12311-X/TP·7900

印 数:1~3000

定 价:33.00 元

改革开放以来,特别是党的十五大以来,我国教育事业取得了举世瞩目的辉煌成就,高等教育实现了历史性的跨越,已由精英教育阶段进入国际公认的大众化教育阶段。在质量不断提高的基础上,高等教育规模取得如此快速的发展,创造了世界教育发展史上的奇迹。当前,教育工作既面临着千载难逢的良好机遇,同时也面临着前所未有的严峻挑战。社会不断增长的高等教育需求同教育供给特别是优质教育供给不足的矛盾,是现阶段教育发展面临的基本矛盾。

教育部一直十分重视高等教育质量工作。2001年8月,教育部下发了《关于加强高等学校本科教学工作,提高教学质量的若干意见》,提出了十二条加强本科教学工作提高教学质量的措施和意见。2003年6月和2004年2月,教育部分别下发了《关于启动高等学校教学质量与教学改革工程精品课程建设工作的通知》和《教育部实施精品课程建设提高高校教学质量和人才培养质量》文件,指出“高等学校教学质量和教学改革工程”是教育部正在制定的《2003—2007年教育振兴行动计划》的重要组成部分,精品课程建设是“质量工程”的重要内容之一。教育部计划用五年时间(2003—2007年)建设1500门国家级精品课程,利用现代化的教育信息技术手段将精品课程的相关内容上网并免费开放,以实现优质教学资源共享,提高高等学校教学质量和人才培养质量。

为了深入贯彻落实教育部《关于加强高等学校本科教学工作,提高教学质量的若干意见》精神,紧密配合教育部已经启动的“高等学校教学质量与教学改革工程精品课程建设工作”,在有关专家、教授的倡议和有关部门的大力支持下,我们组织并成立了“清华大学出版社教材编审委员会”(以下简称“编委会”),旨在配合教育部制定精品课程教材的出版规划,讨论并实施精品课程教材的编写与出版工作。“编委会”成员皆来自全国各类高等学校教学与科研第一线的骨干教师,其中许多教师为各校相关院、系主管教学的院长或系主任。

按照教育部的要求,“编委会”一致认为,精品课程的建设工作从开始就要坚持高标准、严要求,处于一个比较高的起点上;精品课程教材应该能够反映各高校教学改革与课程建设的需要,要有特色风格、有创新性(新体系、新内容、新手段、新思路,教材的内容体系有较高的科学创新、技术创新和理念创新的含量)、先进性(对原有的学科体系有实质性的改革和发展、顺应并符合新世纪教学发展的规律、代表并引领课程发展的趋势和方向)、示范性(教材所体现的课程体系具有较广泛的辐射性和示范性)和一定的前瞻

性。教材由个人申报或各校推荐(通过所在高校的“编委会”成员推荐),经“编委会”认真评审,最后由清华大学出版社审定出版。

目前,针对计算机类和电子信息类相关专业成立了两个“编委会”,即“清华大学出版社计算机教材编审委员会”和“清华大学出版社电子信息教材编审委员会”。首批推出的特色精品教材包括:

(1) 高等学校教材·计算机应用——高等学校各类专业,特别是非计算机专业的计算机应用类教材。

(2) 高等学校教材·计算机科学与技术——高等学校计算机相关专业的教材。

(3) 高等学校教材·电子信息——高等学校电子信息相关专业的教材。

(4) 高等学校教材·软件工程——高等学校软件工程相关专业的教材。

(5) 高等学校教材·信息管理与信息系统。

清华大学出版社经过近 20 年的努力,在教材尤其是计算机和电子信息类专业教材出版方面树立了权威品牌,为我国的高等教育事业做出了重要贡献。清华版教材经过 20 多年的精雕细刻,形成了技术准确、内容严谨的独特风格,这种风格将延续并反映在特色精品教材的建设中。

清华大学出版社教材编审委员会
E-mail: dingl@tup.tsinghua.edu.cn

随着信息技术和计算机科学的发展,计算机技术已渗透到各学科的研究和应用之中,C++语言不再像诞生的初期,只被少数专业开发人员使用,而已经被各专业的工程技术人员广泛应用于国民经济的各行各业之中。

面向对象程序设计方法所强调的基本原则之一是直接面对客观世界中存在的问题进行软件开发,使软件开发方法更符合人类的思维习惯。由于面向对象编程语言所具有的优点,目前它已经成为开发大型软件的主流方法,而 C++ 是面向对象的程序设计语言中应用最广泛的一种,成为了国内外高等院校程序设计的一门专业必修课程,同时也是编程人员最广泛使用的工具。学好 C++, 可以很容易地触类旁通 Java、C# 等其他语言。Visual C++ 是具有强大功能的可视化开发工具,它将面向对象、网络技术、事件驱动、数据库及应用程序向导完美地结合在一起,使用户可以快捷、可视化地开发应用程序,它已经成为基于 Windows 应用程序开发的主流平台。本书较好地实现了将 C++ 面向对象编程语言与可视化工具的结合,力求使学生具有良好的程序设计素养和能力。

许多学生虽学过 C++, 却疏于编程,作者在教学中深刻地认识到了这一点。要学好程序设计,学生不仅需要掌握编程语言,也需要掌握基本的数据结构和程序设计方法,才能更好地分析问题和解决问题。面向对象程序设计方法是软件分析、设计和实现的一种新方法,本书以面向对象的程序设计方法贯穿始终,不仅详细介绍了 C++ 语言本身,而且剖析了常用的数据结构和算法,着重从程序设计方法的角度介绍语法及应用,力求使读者既能熟练掌握 C++ 程序设计语言,也能具有运用面向对象方法解决实际问题的能力。

本书共 12 章,从内容上可以分为三大部分。第一部分(第 1 章~第 3 章)是面向对象程序设计的基本概念和基本方法,介绍从 C 语言到 C++ 语言的过渡及 C++ 语法。第二部分(第 4 章~第 10 章)是 C++ 语言实现面向对象程序设计的基本方法,通过对概念和原理的准确描述,并结合典型的例题,由浅入深地介绍 C++ 的类与对象、数组与指针、继承与派生、多态性、模板、I/O 流库、异常处理机制等概念,通过实例掌握面向对象程序设计的原理、思想和方法内核。第三部分(第 11 章~第 12 章)是 Visual C++ 6.0 平台下 Windows 应用程序的开发,在介绍 Windows 程序设计的基本理论与概念的基础上,详细介绍了包括输入输出处理(文本输入输出及绘图)、菜单、工具栏、状态栏、对

话框、控件及数据库应用程序的设计方法,通过典型的实例和详细的步骤,掌握基于 MFC 的各种典型 Windows 应用程序设计的方法,为 Windows 程序的深入应用奠定坚实的基础。

本书作者一直从事面向对象程序设计及相关的教学与科研工作,主讲过程序设计方面的多门课程,深刻了解学生在学习中的难点和对教材内容的需求。本书凝集了作者多年教学和科研实践经验,全书以面向对象的思维贯穿始终,选材新颖,注重内容的科学性、适应性和针对性,符合当今计算机科学的发展趋势。本书设计了许多与实际有关的例题和习题,并且它们彼此相关,环环相扣。全部程序都在 Visual C++ 6.0 调试通过,并给出了程序运行结果。全部程序风格统一,对关键性语句进行了注释,对类名、函数名等标识符的命名做到“见名知义”,且绝大多数程序给出了设计要点分析。

本书内容深入浅出,将复杂的概念用简洁浅显的语言来讲述,使读者可轻松入门,循序渐进地提高,在有限的学时中,全面掌握基本理论和基础知识。在此基础上,再进一步通过实验熟练掌握开发环境的使用以及程序设计的技巧和方法。面向对象程序设计课程是一门既要求理论,又强调实践的课程。希望读者认真实践教材的每一道例题与习题。

为方便读者使用本书,《面向对象程序设计与 Visual C++ 6.0 教程题解与实验指导》将与本书配套出版,与此同时,还提供与教材配套的电子教案及教材的全部源程序。

本书可以作为高等院校计算机、电子技术、通信、信息工程、自动化、电气及相关专业的面向对象程序设计课程教材,也可作为 IT 业工程技术人员或其他相关人员的参考书。

使用本教材约需 80 学时,其中实验 30 学时左右,各学校可根据实际情况和内容安排学时。在本书的写作过程中得到了清华大学计算机与信息管理中心蒋东兴主任的大力支持和帮助,此外,周玉英、陈茜、丁灿飞、文静、陈鸣红、吴玘中、周海英、宋义召、樊星、谢娇颖、杨成、王蜀毅、林欣欣、王娟、许飞、倪国英等同志在文稿录入和校对方面承担了许多工作,在此表示衷心的感谢。

由于作者水平有限,缺点和疏漏之处在所难免,恳请读者批评指正。欢迎读者对本书提出任何意见和建议。作者的联系方式如下:

cth188@sina.com, cth188@hotmail.com

作 者

2005 年 8 月

第 1 章 面向对象程序设计概述	1
1.1 计算机程序设计语言的发展	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
1.1.4 面向对象语言	3
1.2 面向对象程序设计的基本概念	4
1.2.1 面向对象方法的产生	4
1.2.2 面向对象的方法与结构	4
1.2.3 类与对象的概念	5
1.2.4 消息与事件的概念	6
1.3 面向对象程序设计的特点	7
1.3.1 抽象性	7
1.3.2 封装性	8
1.3.3 继承性	8
1.3.4 多态性	9
1.3.5 C++ 的面向对象特性	10
1.4 面向对象的程序设计语言	11
1.4.1 混合型面向对象语言——C++	11
1.4.2 Java 语言	12
1.4.3 Visual Basic 语言	12
1.5 Visual C++ 概述	13
1.5.1 Visual C++ 6.0	13
1.5.2 Visual C++ .NET	14
1.6 面向对象的软件开发	16
1.6.1 面向对象分析	16
1.6.2 面向对象设计	16

1.6.3	面向对象编程	17
1.6.4	面向对象测试	18
1.6.5	面向对象软件维护	18
1.7	本章小结	18
1.8	思考与练习题	19
第 2 章	C++ 程序设计基础	21
2.1	C++ 语言概述	21
2.1.1	C++ 的产生	21
2.1.2	C++ 的特点	22
2.1.3	C++ 的字符集	22
2.1.4	C++ 程序的组成	23
2.2	数据类型	26
2.2.1	基本数据类型	26
2.2.2	常量	27
2.2.3	变量	30
2.2.4	符号常量	34
2.3	运算符与表达式	35
2.3.1	表达式	35
2.3.2	运算符及性质	36
2.3.3	运算符的优先级	42
2.3.4	混合运算时数据类型的转换	43
2.4	简单的输入与输出	45
2.4.1	I/O 流	45
2.4.2	插入运算符和提取运算符	45
2.4.3	简单的 I/O 格式	46
2.5	程序的基本控制结构	46
2.5.1	顺序结构	46
2.5.2	选择结构	46
2.5.3	循环结构	52
2.6	自定义数据类型	56
2.6.1	结构类型	56
2.6.2	联合类型	58
2.6.3	枚举类型	60
2.6.4	typedef 的应用	62
2.7	本章小结	63
2.8	思考与练习题	64

第 3 章 函数	66
3.1 函数的定义与使用	66
3.1.1 函数的定义	67
3.1.2 函数的调用	67
3.1.3 函数原型	69
3.2 函数的参数传递	70
3.2.1 传值调用	71
3.2.2 传地址调用	72
3.2.3 引用调用	73
3.3 函数的嵌套与递归调用	74
3.3.1 函数的嵌套调用	74
3.3.2 函数的递归调用	75
3.4 函数的其他问题	78
3.4.1 内联函数	78
3.4.2 函数重载的概念	79
3.4.3 带默认参数的函数	80
3.5 C++ 的系统函数	82
3.6 作用域与存储类型	83
3.6.1 作用域	83
3.6.2 存储类型	86
3.6.3 生存期	86
3.7 全局变量与局部变量	87
3.7.1 全局变量	87
3.7.2 局部变量	87
3.8 头文件与多文件结构	88
3.8.1 头文件	89
3.8.2 多文件结构	89
3.9 编译预处理	90
3.9.1 宏定义	91
3.9.2 文件包含指令	91
3.9.3 条件编译	91
3.10 本章小结	93
3.11 思考与练习题	93
第 4 章 类与对象	95
4.1 类	95
4.1.1 类的定义	95
4.1.2 访问控制	97

4.2	成员函数	98
4.2.1	成员函数的定义	98
4.2.2	内联成员函数	99
4.3	对象	100
4.3.1	对象的定义	101
4.3.2	类成员的访问	101
4.4	构造函数与析构函数	104
4.4.1	构造函数	104
4.4.2	析构函数	106
4.4.3	复制构造函数	111
4.4.4	浅复制与深复制	116
4.5	静态成员	120
4.5.1	静态成员的定义与引用	120
4.5.2	静态数据成员	120
4.5.3	静态成员函数	122
4.6	友元	124
4.6.1	友元的作用	124
4.6.2	友元函数	124
4.6.3	友元类	126
4.7	类作用域及对象的生存期	127
4.7.1	类作用域	127
4.7.2	对象的生存期	128
4.8	名空间	129
4.8.1	名空间的定义	129
4.8.2	名空间成员的访问	130
4.8.3	名空间的应用	130
4.9	常类型	132
4.9.1	常引用	132
4.9.2	常对象	133
4.9.3	常对象成员	134
4.10	本章小结	137
4.11	思考与练习题	138
第 5 章	数组与指针	143
5.1	数组	143
5.1.1	一维数组	143
5.1.2	二维数组	148
5.1.3	对象数组	151
5.2	指针	153

5.2.1	内存空间的访问方式	154
5.2.2	指针变量的定义	154
5.2.3	“*”和“&”运算符	155
5.2.4	指针的赋值	156
5.3	指针运算	157
5.3.1	指针的算术运算	157
5.3.2	指针的关系运算	158
5.4	指针与数组	159
5.4.1	一维数组元素的指针表示	159
5.4.2	二维数组元素的指针表示	161
5.4.3	指针数组	162
5.5	指针与函数	163
5.5.1	指针作为函数参数	163
5.5.2	指针型函数	165
5.5.3	指向函数的指针	166
5.6	指针与字符串	167
5.6.1	使用指针访问字符串	167
5.6.2	字符指针作为函数参数	168
5.7	对象指针	169
5.7.1	对象指针的概念	169
5.7.2	类数据成员的指针	170
5.7.3	类成员函数的指针	170
5.7.4	this 指针	172
5.8	动态内存分配	173
5.8.1	new 运算符	173
5.8.2	delete 运算符	174
5.9	本章小结	176
5.10	思考与练习题	177
第 6 章	继承与派生	179
6.1	继承与派生的概念	179
6.1.1	继承的层次结构	179
6.1.2	继承的作用	180
6.2	派生类	181
6.2.1	派生类的定义	181
6.2.2	派生类的生成过程	185
6.3	访问权限控制	185
6.3.1	公有继承	186
6.3.2	私有继承	188

6.3.3	保护继承	189
6.4	衍生类的构造函数和析构函数	192
6.4.1	衍生类的构造函数	192
6.4.2	衍生类的析构函数	195
6.5	多继承	196
6.5.1	多继承的定义	196
6.5.2	多继承的构造函数	197
6.5.3	多继承的析构函数	199
6.5.4	虚基类	202
6.6	赋值兼容规则	205
6.7	程序实例	208
6.8	本章小结	213
6.9	思考与练习题	214
第7章	多态性	217
7.1	多态性的实现类型	217
7.2	联编	218
7.2.1	静态联编	218
7.2.2	动态联编	220
7.3	虚函数	220
7.3.1	虚函数的声明	221
7.3.2	虚函数的调用	221
7.3.3	虚析构函数	224
7.4	抽象类	226
7.4.1	纯虚函数的定义	227
7.4.2	抽象类的使用	227
7.5	函数重载	229
7.6	运算符重载	231
7.6.1	运算符重载规则	231
7.6.2	运算符重载为成员函数	232
7.6.3	运算符重载为友元函数	240
7.7	综合设计举例	242
7.8	本章小结	246
7.9	思考与练习题	246
第8章	模板	248
8.1	模板概述	248
8.2	函数模板	249
8.2.1	函数模板的定义	249

8.2.2	函数模板的使用	250
8.2.3	模板函数的生成	252
8.3	类模板	252
8.3.1	类模板的定义	252
8.3.2	类模板的使用	254
8.3.3	类模板的友元	256
8.4	STL 简介	259
8.4.1	STL 和 C++ 标准库	260
8.4.2	容器	261
8.4.3	算法	262
8.4.4	迭代器	263
8.4.5	函数对象	264
8.5	STL 应用实例	265
8.6	本章小结	269
8.7	思考与练习题	269
第 9 章	I/O 流库	271
9.1	I/O 流库的层次结构	271
9.1.1	I/O 流的概念	271
9.1.2	ios 类的层次关系	272
9.1.3	streambuf 类的层次关系	273
9.2	输入与输出的格式控制	273
9.2.1	ios 类成员函数的格式控制	273
9.2.2	操作符函数的格式控制	277
9.2.3	自定义操作符函数的格式化	278
9.3	文件的输入与输出	280
9.3.1	文件的打开与关闭	281
9.3.2	ifstream、ofstream 和 fstream 类	282
9.3.3	文本文件的输入与输出	283
9.3.4	二进制文件的输入与输出	286
9.4	自定义数据类型的输入与输出	289
9.4.1	输出运算符重载	289
9.4.2	输入运算符重载	291
9.5	本章小结	293
9.6	思考与练习题	293
第 10 章	异常处理	296
10.1	异常处理的任务与思想	296
10.1.1	传统的异常处理方法	296

10.1.2	C++ 的异常处理机制	298
10.2	C++ 异常处理的实现	298
10.2.1	异常处理的语法	299
10.2.2	异常处理接口	302
10.3	异常处理中的构造与析构	303
10.4	C++ 标准异常类	305
10.5	本章小结	309
10.6	思考与练习题	309
第 11 章	Visual C++ 应用简介	311
11.1	Visual C++ 编程基础	311
11.1.1	Visual C++ 环境介绍	311
11.1.2	Visual C++ 的菜单功能	313
11.1.3	Visual C++ 的工具栏	317
11.1.4	项目和项目工作区	318
11.2	Windows 编程概述	319
11.2.1	多任务和多线程	319
11.2.2	事件驱动原理	320
11.2.3	Windows 编程的基本概念	320
11.2.4	GDI 简介	323
11.2.5	Windows 资源	323
11.2.6	常用数据结构	324
11.2.7	Windows 标识符的命名方法	326
11.3	用 MFC 向导创建 Windows 应用程序	326
11.3.1	MFC 类库简介	327
11.3.2	MFC 类库的常用类	327
11.3.3	MFC 应用程序的类型	331
11.3.4	用 AppWizard 向导创建 MFC 应用程序	332
11.4	本章小结	338
11.5	思考与练习题	338
第 12 章	MFC 典型应用程序设计	339
12.1	消息处理机制	339
12.1.1	MFC 消息的种类	339
12.1.2	MFC 消息的映射机制	340
12.2	应用程序分析	341
12.2.1	AppWizard 向导生成的类和文件	342
12.2.2	应用程序的运行机制	343
12.3	输入与输出处理程序	345

12.3.1	文本输出程序	345
12.3.2	键盘输入处理程序	347
12.3.3	鼠标处理程序	351
12.4	菜单应用程序设计	355
12.4.1	菜单简介	355
12.4.2	菜单资源编辑器	355
12.4.3	菜单应用实例	357
12.5	工具栏应用程序设计	364
12.5.1	工具栏简介	364
12.5.2	使用资源编辑器创建工具栏	364
12.5.3	工具栏应用实例	366
12.6	状态栏应用程序设计	367
12.6.1	状态栏的实现	367
12.6.2	状态栏应用实例	368
12.7	对话框应用程序设计	372
12.7.1	对话框的分类	372
12.7.2	常用对话框类	373
12.7.3	对话框的常用函数	374
12.7.4	对话框应用实例	374
12.8	控件应用程序设计	382
12.8.1	控件简介	382
12.8.2	常用控件类	383
12.8.3	创建控件	383
12.8.4	控件应用实例	383
12.9	数据库应用程序设计	390
12.9.1	ODBC 技术	390
12.9.2	ADO 技术	391
12.9.3	数据库应用实例	391
12.10	本章小结	406
12.11	思考与练习题	407
附录 A	Visual C++ 程序的调试方法	408
附录 B	ASCII 码字符集	413
参考文献	414

面向对象程序设计概述

面向对象程序设计是一种全新的软件工程技术,它使程序设计方法更符合人类的思维方式,更能直接地描述客观世界。通过提高代码的可重用性、可扩充性和程序自动生成功能来提高编程效率,并且大大减少软件维护的开销,目前,面向对象技术已经被越来越多的软件技术人员所接受和采用。本章首先概要介绍面向对象程序设计语言的产生和特点以及面向对象程序设计的基本概念与基本特征,并介绍面向对象与面向过程程序设计的区别、面向对象的软件工程,同时介绍目前流行的 Java、.NET 技术等面向对象程序设计语言。

1.1 计算机程序设计语言的发展

计算机系统包括硬件和软件系统。计算机从诞生以来,功能越来越强大,其强大的功能不仅仅是因为它拥有基于微电子技术的硬件系统,同时,也是因为计算机具有越来越强大和完善的软件系统。软件简单地说就是程序,计算机的工作是由程序(系统软件和应用软件)控制的,而程序是一系列程序设计语言指令的集合,没有程序,计算机就不可能工作。在自然界,语言是表达思维的工具,或者说思维是通过语言来表达的。计算机程序设计语言也同样如此,它是描述人类思维、使计算机按人的意愿工作的载体和工具。因此,计算机语言指由一系列字符组成、具有描述问题的能力、且计算机能识别和执行的代码或指令集。

1.1.1 机器语言

自从 1946 年 2 月人类历史上第一台名为 ENIAC 的电子计算机诞生以来,已有近 60 年历史,计算机科学与技术得到了迅速发展,计算机越来越普及,其应用领域已渗透到社会的各个层面,成为了人类经济活动与社会活动不可缺少的重要工具。

计算机是按照人的意图进行计算和工作的。我们必须以计算机所能接受的语言——计算机程序设计语言与其通信,告诉计算机对什么数据进行怎样的运算或进行怎样的操作等。第一代计算机语言是机器语言,它是一种依赖于计算机硬件的语言,即不同的计算机有不同的机器语言。机器语言由一系列机器指令组成。在每一条指令中要规定机器做什么运算(由操作码指示)和对哪个存储单元中的数据进行运算(由地址码指示)。而且,数据和指令必须分别存放,即存放在不同单元中。