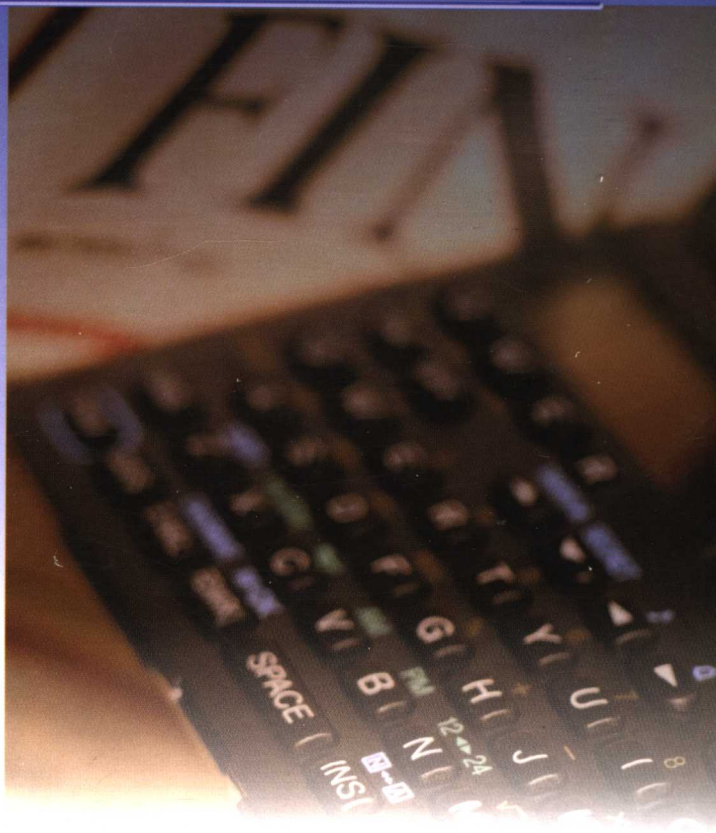


高等学校计算机基础教育教材精选



姚普选 仇国巍 编著

程序设计教程

(Visual C++)

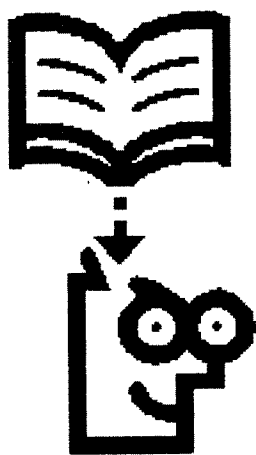


清华大学出版社

高等学校计算机基础教育教材精选

程序设计教程 (Visual C++)

姚普选
仇国巍
编 著



清华大学出版社

清华大学出版社
北 京

内 容 简 介

本书以 Visual C++, .NET 为工具,循序渐进地介绍了计算机程序设计的基本思想和常用的程序设计方法,并简明扼要地介绍了与之相关的算法、数据结构以及其他方面的知识。本书主要内容包括程序设计基础知识、算法基本常识、数据类型知识和应用、程序结构知识和应用、面向对象程序设计方法、用户界面设计方法、图形图像处理及多媒体播放程序设计方法,以及数据文件的使用和数据库应用程序设计方法等。

本书讲求文字的准确性、思想的连贯性、方法的实用性和内容的先进性,对于重要的名词、抽象的概念、常用的技术,以及具有递进关系的系列内容的讲解,都根据教学活动中的实际情况进行了精心地安排。本书的编写体系考虑到不同层次读者的需求,可作为高等院校计算机程序设计课程的教材,也可供学习程序设计的其他人员使用。

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

程序设计教程(Visual C++)/姚普选,仇国巍编著. —北京:清华大学出版社,2005.7

(高等学校计算机基础教育教材精选)

ISBN 7-302-11134-0

I. 程… II. ①姚… ②仇… 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 054797 号

出 版 者:清华大学出版社

<http://www.tup.com.cn>

社 总 机:010-62770175

地 址:北京清华大学学研大厦

邮 编:100084

客户服务:010-62776969

组稿编辑:焦 虹

文稿编辑:汪汉友

印 刷 者:北京市清华园胶印厂

装 订 者:三河市新茂装订有限公司

发 行 者:新华书店总店北京发行所

开 本:185×260 印张:25 字数:589千字

版 次:2005年7月第1版 2005年7月第1次印刷

书 号:ISBN 7-302-11134-0/TP·7358

印 数:1~5000

定 价:29.00元

出版说明

—— 高等学校计算机基础教材精选 ——

在教育部关于高等学校计算机基础教育三层次方案的指导下,我国高等学校的计算机基础教育事业蓬勃发展。经过多年的教学改革与实践,全国很多学校在计算机基础教育这一领域中积累了大量宝贵的经验,取得了许多可喜的成果。

随着科教兴国战略的实施以及社会信息化进程的加快,目前我国的高等教育事业正面临着新的发展机遇,但同时也必须面对新的挑战。这些都对高等学校的计算机基础教育提出了更高的要求。为了适应教学改革的需要,进一步推动我国高等学校计算机基础教育事业的发展,我们在全中国各高等学校精心挖掘和遴选了一批经过教学实践检验的优秀教学成果,编辑出版了这套教材。教材的选题范围涵盖了计算机基础教育的三个层次,面向各高校开设的计算机必修课、选修课,以及与各类专业相结合的计算机课程。

为了保证出版质量,同时更好地适应教学需求,本套教材将采取开放的体系和滚动出版的方式(即成熟一本、出版一本,并保持不断更新),坚持宁缺勿滥的原则,力求反映我国高等学校计算机基础教育的最新成果,使本套丛书无论在技术质量上还是在文字质量上均成为真正的“精选”。

清华大学出版社一直致力于计算机教育用书的出版工作,在计算机基础教育领域出版了许多优秀的教材。本套教材的出版将进一步丰富和扩大我社在这一领域的选题范围、层次和深度,以适应高校计算机基础教育课程层次化、多样化的趋势,从而更好地满足各学校由于条件、师资和生源水平、专业领域等的差异而产生的不同需求。我们热切期望全国广大教师能够积极参与到本套丛书的编写工作中来,把自己的教学成果与全国的同行们分享;同时也欢迎广大读者对本套教材提出宝贵意见,以便我们改进工作,为读者提供更好的服务。

我们的电子邮件地址是: jiaoh@tup.tsinghua.edu.cn;联系人:焦虹。

清华大学出版社

前言

程序设计教程(Visual C++)

计算机程序设计是计算机基础教育的重点和基础。作为高等院校的学生,尤其是将来要成为工程技术人员理工科学生,不能只满足于使用别人设计好的软件,而应该具有一定的程序设计能力。

实际上,程序设计并不是一件容易的事情。就现状而言,要编写程序来解决某个实际问题,首先要理解问题本身的逻辑结构和工作方式,然后再考虑选用哪种程序设计语言和工具。而在程序设计过程中,还要考虑问题在计算机中如何表示?应该采用哪种算法?按照什么方法和步骤来编制和调试程序等等。这涉及程序设计语言的语法规则、程序设计工具的使用方法、算法设计策略、数据结构常识、程序设计的一般方法和技巧等多方面的知识和技能。要使学生在有限的时间内掌握程序设计技术,首先要有能够将这些知识和技能有机地融合在一起的教材。

多年来,围绕着计算机程序设计课程,积累了一大批各具特色的教材和参考书,其中不乏立意深刻、内容丰富的教材。但是,当程序设计发展到面向对象设计方法、可视化设计环境之后,因为需要解决的实际问题的范围越来越广、程序设计工具越来越复杂、程序设计手段更为丰富多彩等各种原因,有关的书籍大都有所侧重,或者详细地介绍程序设计语言和软件开发工具的使用方法,或者系统地讨论程序设计的一般方法,而将这两者有机地结合在一起并适合于教学的书籍较少。有鉴于此,笔者编写了这本《程序设计教程(Visual C++)》。

本书选择了核心的程序设计技术以及 Visual C++ .NET 软件开发环境的常用功能,由浅入深地进行了详细的讲解,力图使读者在有限的时间内,对课程的相关知识有一个清晰、完整的理解。本书中对重要的概念和方法一般都先在例题或讲解中以极易理解的简单方式加以运用,然后在其后的某个章节集中讲解,这样既起到了分散难点的作用,便于初学者理解和掌握,又避免了因刻意分散难点而形成相关内容的割裂,消除了读者不便查阅、不易形成完整印象的弊病。同时,本书注重知识和技能的合理调配,力图避免因强调某些方面、忽略其他方面所造成的程序设计能力整体上的缺失。

本书可作为高等院校程序设计课程的教材,也可作为程序设计工作者的参考书。采用本书作为教材的程序设计课程以 64~72(包括上机时数)学时为宜。本书每章都配备了内容丰富的习题,不同类型的读者可根据自己的实际情况选做部分习题。

本书第 1 章~第 6 章由姚普选编写,第 7 章~第 9 章由仇国巍编写,并由姚普选统稿。

程序设计技术博大精深,其内容绝非一本书所能包括。而且,书的编写不可避免的要受到作者的思想水平、时间、篇幅等种种限制。因此,本书传达的信息是否到位或者是否得体,还要经过读者的检验。望广大读者批评指正。

姚普选

2005 年 2 月

目录

程序设计教程(Visual C++)

第 1 章 程序设计基础知识	1
1.1 程序设计的概念	1
1.2 程序设计语言	4
1.2.1 程序设计语言的种类	4
1.2.2 高级语言	6
1.2.3 C++ 语言的背景和特点	8
1.3 C++ 程序的基本结构	9
1.3.1 C++ 程序实例	9
1.3.2 注释	10
1.3.3 编译预处理	10
1.3.4 函数	11
1.3.5 数据的输入输出	12
1.4 Visual C++ .NET 开发环境及应用程序	12
1.5 C++ 程序设计实例	17
1.5.1 控制台应用程序实例	18
1.5.2 图形用户界面应用程序实例	21
1.6 程序设计的一般步骤	25
习题 1	29
第 2 章 算法与程序设计	31
2.1 算法的概念	31
2.2 算法的特性	34
2.3 算法的流程图表示	35
2.4 算法的结构	38
2.4.1 算法的三种基本结构	38
2.4.2 基本结构的本质属性	41
2.4.3 N-S 结构化流程图	42
2.5 算法的 C++ 程序实现	44
2.5.1 顺序结构的 C++ 程序	44

2.5.2	选择结构的 C++ 程序	45
2.5.3	循环结构的 C++ 程序	47
2.5.4	循环结构和选择结构的嵌套	49
2.6	算法与数据结构	51
2.6.1	数据类型的意义	51
2.6.2	数据结构的概念	52
2.6.3	数据结构与算法	54
2.7	结构化程序设计	56
2.7.1	结构化程序	56
2.7.2	结构化程序设计方法	57
习题 2		61

第 3 章 基本数据类型与表达式 63

3.1	标识符和名字	63
3.1.1	C++ 字符集	63
3.1.2	标识符	64
3.1.3	名字	64
3.2	数据类型的概念	65
3.2.1	C++ 中的数据类型	65
3.2.2	基本类型的计算机表示	66
3.3	基本数据类型	68
3.3.1	数字、字符及字符串	68
3.3.2	基本类型变量的声明和引用	70
3.3.3	无符号数和 const 常量	72
3.3.4	用 typedef 定义类型	73
3.4	算术表达式	74
3.4.1	基本算术表达式	74
3.4.2	数据类型转换	75
3.4.3	自增和自减运算符	77
3.5	逻辑表达式	78
3.5.1	关系表达式	78
3.5.2	逻辑表达式	79
3.6	C++ 中的特殊表达式	80
3.6.1	赋值表达式	81
3.6.2	条件表达式	82
3.6.3	逗号表达式	83
3.6.4	位运算表达式	83
3.7	运算符的优先顺序	86

3.8 C++ 中的数据输入输出	87
3.8.1 输入流 cin	87
3.8.2 输出流 cout	90
3.8.3 输入输出函数 printf()和 scanf()	95
习题 3	96

第 4 章 C++ 程序结构 99

4.1 C++ 语言中的语句	99
4.2 选择结构	101
4.2.1 if 语句的使用	101
4.2.2 if 语句的嵌套	103
4.2.3 switch 语句的使用	105
4.3 循环结构	107
4.3.1 while 语句	107
4.3.2 do...while 语句	108
4.3.3 for 语句	109
4.3.4 循环结构的嵌套	112
4.4 非正常流程控制	114
4.4.1 goto 语句和语句标号	114
4.4.2 break 语句和 continue 语句	115
4.4.3 exit 函数和 abort 函数	117
4.5 函数的定义和调用	118
4.5.1 函数的定义	118
4.5.2 函数的调用	120
4.5.3 函数的声明	122
4.5.4 内联函数	123
4.6 函数的参数	124
4.6.1 参数的值传递方式	124
4.6.2 变量的引用类型与函数参数	126
4.6.3 带有默认参数的函数	127
4.7 函数的嵌套调用和递归	128
4.7.1 函数的嵌套调用	128
4.7.2 函数的递归	130
4.8 变量的作用域和生存期	135
4.8.1 局部变量和全局变量	135
4.8.2 变量的存储类别	137
4.9 程序的多文件组织	141
4.10 编译预处理	143



4.10.1	#define 宏定义指令	143
4.10.2	#include 文件包含命令	146
4.10.3	条件编译	149
习题 4		151
第 5 章	用户自定义数据类型	154
5.1	数组	154
5.1.1	一维数组	154
5.1.2	二维数组	158
5.1.3	数组作为函数参数	162
5.2	结构体	164
5.3	枚举型变量	168
5.4	指针	173
5.4.1	地址与指针的概念	173
5.4.2	指针的定义和使用	175
5.4.3	动态存储分配	178
5.5	指针与数组	180
5.5.1	一维数组和指针	180
5.5.2	二维数组和指针	184
5.5.3	指针的数组	186
5.5.4	指向数组的指针	188
5.6	字符串处理	189
5.6.1	字符数组	189
5.6.2	字符指针	192
5.6.3	字符串类型	194
5.7	指针与函数	197
5.7.1	返回指针的函数	197
5.7.2	指向函数的指针	198
习题 5		200
第 6 章	面向对象程序设计	203
6.1	面向对象程序设计的概念	203
6.2	类和对象	206
6.2.1	类的定义	206
6.2.2	类中成员函数的定义	208
6.2.3	对象	210
6.2.4	构造函数与析构函数	212
6.2.5	const 对象与 const 成员函数	215

6.2.6	对象与指针	217
6.2.7	类的静态成员	218
6.2.8	友元	219
6.3	方法重载	222
6.3.1	函数重载	222
6.3.2	运算符重载	224
6.4	继承	227
6.4.1	基类和派生类	227
6.4.2	多重继承	229
6.4.3	派生类的构造函数	230
6.4.4	派生类的三种继承方式	232
6.4.5	继承与组合	237
6.5	多态性及虚函数	239
6.5.1	派生类对象替换基类对象	240
6.5.2	虚函数	241
6.5.3	纯虚函数和抽象类	244
6.6	模板	246
6.6.1	函数模板	246
6.6.2	类模板	248
6.7	C++ 的 I/O 流类和对象	251
6.7.1	C++ 流类	251
6.7.2	插入和提取运算符	254
6.8	异常处理	256
6.9	命名空间	259
6.9.1	命名空间的使用	260
6.9.2	标准命名空间 std	261
习题 6		262

第 7 章	用户界面设计	264
7.1	建立 Visual C++ .NET 工程	264
7.1.1	Visual C++ .NET 工程种类	264
7.1.2	基于 MFC 的应用程序类型	265
7.2	基于 MFC 的应用程序概述	267
7.2.1	文档/视图结构	267
7.2.2	客户区重绘	270
7.2.3	资源的管理	272
7.3	各类消息处理方法	275
7.3.1	菜单及工具栏命令	276



7.3.2	鼠标消息处理	279
7.3.3	键盘消息处理	282
7.3.4	定时器消息处理	283
7.4	对话框	286
7.4.1	对话框的创建	286
7.4.2	对话框的调用	288
7.4.3	对话框中的控件	294
7.4.4	通用对话框	298
7.4.5	基于对话框的应用	300
习题 7		302
第 8 章	图形图像与多媒体应用程序	303
8.1	设备环境类	303
8.1.1	CPaintDC 类	304
8.1.2	CClientDC 类	305
8.2	绘图工具类	306
8.2.1	画笔的使用	307
8.2.2	画刷的使用	308
8.2.3	库存绘图对象	311
8.2.4	字体的使用	311
8.3	坐标映射方式	315
8.4	常用绘画模式	316
8.5	数据显示与字符串	319
8.6	几何图形绘制	321
8.7	位图显示及图片框	325
8.7.1	客户区位图显示	325
8.7.2	图片框控件	329
8.8	多媒体应用	331
8.8.1	声音文件播放	331
8.8.2	网页浏览	332
8.8.3	媒体播放	335
习题 8		338
第 9 章	数据文件与数据库应用程序	340
9.1	数据管理方式的变迁	340
9.2	数据文件的读写	341
9.2.1	标准 C++ 中的文件流类	342
9.2.2	用 MFC 文件类读写文件	346



9.2.3 序列化方式读写文件.....	351
9.3 数据库系统概念	356
9.3.1 数据库系统构成.....	356
9.3.2 关系型数据库.....	358
9.4 数据库操作语言——SQL	360
9.5 数据库应用程序开发	362
9.5.1 ODBC 数据源.....	363
9.5.2 通过向导建立应用程序.....	364
9.5.3 利用 MFC ODBC 类编程	370
9.5.4 利用列表控件显示记录.....	377
习题 9	382
参考文献	384

计算机的基本工作原理,就是用人们预先编制好并存放在计算机内部的程序来控制计算机自动运行,完成程序所规定的任务。因而,使用计算机的基本方式就是编写程序和运行程序。这种将预定的任务用程序表达出来的整个过程叫做程序设计。

程序设计工作要使用程序设计语言或将程序设计语言、编辑工具、调试工具等集成在一起的软件开发工具来完成,因而,了解程序的一般形式和基本设计方法,掌握程序设计语言或软件开发工具的使用方法是程序设计人员首当其冲的任务。

1.1 程序设计的概念

为了使计算机能按照人的意图来完成特定的任务,就必须先让计算机接受和执行人给出的命令和数据。用于向计算机发出操作命令,在计算机和人之间互相传递数据的指令序列或语句序列称为程序。

“程序”和“软件”常被当做同义词来使用,“程序设计”和“软件开发”有时也被不加区分地使用。一般来说,编写较小的程序常称为程序设计,而大型程序设计则更多地称为软件开发。

1. 计算机的基本工作方式

计算机种类繁多,在规模、处理能力、价格、复杂程度等方面都有很大差别,但各种计算机的基本原理是相同的。数学家冯·诺依曼(von Neumann)曾经提出过数字计算机设计的一些基本思想,其中最重要的一点是:内存程序控制原理是计算机的基本工作原理。

程序就是为解决一个信息处理任务而预先编制的工作执行方案,是由一串 CPU 能够执行的基本指令组成的序列,每条指令规定了计算机应该执行什么操作(如加、减、乘、判断等)以及执行时所需要的数据。例如,从存储器读取一个数送到运算器就是一条指令,从存储器读取一个数并和运算器中原有的数相加也是一条指令。

当要求计算机执行某项任务时,就设法把完成任务的解决方法分解成一个一个的步骤,使每个步骤完成的工作可由若干条指令来完成,并将这些指令所构成的程序送入计算机,以二进制代码的形式存放在存储器中(习惯上把这一过程叫做程序设计)。一旦程序

被“启动”，计算机严格地一条条地分析执行程序中的指令，便可以逐步地自动完成这项任务。

程序存储功能使计算机变成了一种自动运行的机器，将程序存入计算机并启动它，计算机就可以独立地工作，以电子速度一条条地执行指令。虽然每条指令能够完成的工作很简单，但通过成千成万条指令的执行，计算机就能够完成非常复杂，意义重大的工作。

2. 大脑解题的基本方式

电子计算机是人所设计制造的增强人类信息处理能力的工具，相当于人的头脑功能的延伸，自然具备人类处理问题的特点。因此，电子计算机的基本工作方式与人处理问题的基本方式十分相似。

人们在日常生产生活中要解决各种各样的问题，尽管不同种类问题的内容和解决方法千差万别，但解决问题的基本过程大致可归结为三步。

(1) 接受原始信息。解决一个问题时，首先要通过眼耳等感官感知所论问题的原始信息，并将其记忆在大脑的相应功能区，或者通过手口等感官转而记录在纸、录音机、录像机这样的外部设备上。

(2) 分析处理信息。解决问题最关键的步骤是通过大脑并借助其他工具和手段对已获取的信息进行综合分析处理，主要包括以下几个方面。

① 综合分析问题的相关信息(包括原始信息和其他辅助信息)，建立起必要的信息之间的总体联系(如数量关系、逻辑关系等)，并将其记忆在纸、录音机、录像机这样的外部设备上或大脑中。

② 运用某些基本信息，主要是在解决该问题之前已记忆在大脑中的经验、方法、技巧、知识等，将上述信息之间的总体联系进行必要的数学推演或逻辑推理，从而得出所需的结果信息，即中间结果和最终结果，并将其记忆在纸、录音机、录像机这样的外部设备上或大脑中。

③ 根据原始信息、相关信息和基本信息核验所得结果信息，特别是最终结果的可靠性、合理性和正确性。必要时，还可借助外部设备来进行。

(3) 表达出最终答案。经过核验确认是正确的最终结果将通过感官输出。例如，通过报表的形式向有关部门汇报、通过讲演的形式进行宣传，或通过书籍、电子文档等形式提供给公众。

上述人处理问题的基本方式可形象地表现出来，如图 1-1 所示。

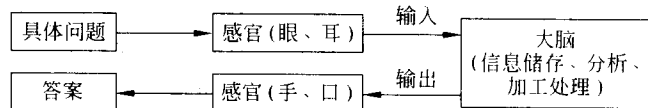


图 1-1 人解题的基本方式

3. 程序的例子

在计算机上计算一个题目之前，先要编制相应的程序，然后运行程序，程序便会控制

计算机自动运行,直至算完这个题目。如果不考虑程序的细节,可以将程序看成是一个函数 $f(x)$,它接收原始数据集合 x ,经过运算处理,最后产生出结果数据集合 y 。即

$$y = f(x)$$

例 1-1 根据下面的函数,由已知的 x 值计算 y 值。

$$y = \begin{cases} 2x+1, & x \geq 0 \\ -x/2, & x < 0 \end{cases}$$

与人脑解题的方式相似,在计算机上求解这个问题的程序要完成的功能为:接受用户在键盘(或其他输入设备)上输入的 x 值;再根据 x 值所在的范围,分别调用不同的数学式计算 y 值;最后输出 y 值。相应的 C++ 程序的主要部分如下:

```
void main()
{
    float x,y;
    cin>>x;
    if (x>=0)
        y=2*x+1;
    else
        y=-x/2;
    cout<<y<<endl;
}
```

该程序的主要部分包含在一个名为 main,由一对花括号“{”和“}”定界的函数中。这个函数称为主函数^①,其中包括一些由 C++ 语言编写的语句,这些语句可以按人脑处理问题的方式划分为三部分:数据的输入部分、运算部分和运算结果的输出部分。

(1) 数据的输入及其定义部分。

输入语句

```
cin>>x;
```

负责输入程序中要用到的原始数据。该语句的功能是:等待用户从标准输入设备(键盘)上输入一个数据,并将它赋给变量 x 。

C++ 语言规定:程序中要用到的所有变量都必须先定义,然后才能使用。故在这个赋值语句之前,先用类型定义语句

```
float x,y;
```

定义变量 x 以及后面要用到的变量 y 。

(2) 运算部分。

以 if 开头的条件语句,即

^① ANSI/ISO 标准 C++ 要求 main() 函数定义为 int 型,有的操作系统要求在执行了一个程序后向操作系统返回一个数值。C++ 的实际处理方法是:如果程序正常运行,则向操作系统返回 0,否则返回 -1。实际使用的 C++ 编译系统不一定都遵守这个规定。本例中按 C 语言的习惯,将 main() 函数的返回值定义为 void(空)类型,且函数体中不包括返回数值的语句。这在 Visual C++ .NET 中是允许的。

```

if (x >= 0)
    y = 2 * x + 1;
else
    y = -x / 2;

```

负责程序中的实际运算,该语句的功能是:判断 x 的值所属的范围,如果 $x \geq 0$,则按 $y = 2x + 1$ 计算 y 值,如果 $x < 0$,则按 $y = -x/2$ 计算 y 值。

在这个语句中,嵌入了两个赋值语句:

```

y = 2 * x + 1;
y = -x / 2;

```

其功能都是计算等号右边表达式的值,并将计算的结果赋给等号左边的变量。

(3) 运算结果输出部分。

输出语句

```

cout << y << endl;

```

负责输出运算结果,即输出按照给定的分段函数和已知的 x 值计算得到的 y 值。语句的功能是:将 y 值按默认的格式输出到标准输出设备(屏幕)上。

该程序在执行后,屏幕显示输入提示符(一个闪烁的短线),等待用户输入,当用户输入一个数字(如 10.9)并按 Enter 键后,屏幕显示运算结果,如:

22.8

1.2 程序设计语言

使用计算机解决问题时,必须用某种“语言”来和计算机进行交流。具体地说,就是利用某种计算机程序设计语言提供的命令来编制程序,并把程序存储在计算机的存储器中,然后计算机在这个程序的控制下运行,达到解决问题的目的。

程序设计语言种类繁多,它们在形式、应用范围、对计算机硬件和软件环境的适应性等各个方面都各有侧重,用户可以根据自己的需要和已有的条件选用。

1.2.1 程序设计语言的种类

程序描述了计算机处理数据、解决问题的过程,这是程序的本质。但程序对数据和问题的描述方式却是多种多样的。随着计算机技术的不断进步,程序设计语言的形式和种类也在不断地发展变化。按照程序设计语言发展的先后,大体上可将其分为三类:机器语言、汇编语言和高级语言。

1. 机器语言

能被计算机直接理解和执行的指令称为机器指令,它在形式上是由“0”和“1”构成的

