



高等职业教育“十一五”规划教材
计算机类核心课程教改项目成果系列教材

Visual C++ 程序设计案例教程

谭建辉 主编

- 经典案例，突显实用
- 特色分析，易学易懂
- 完备理论，牢建基础
- 综合实训，拓展能力

案例导向

理论讲授

自主实践



成就自我能力的提升



科学出版社

免费提供电子课件

高等职业教育“十一五”规划教材

计算机类核心课程教改项目成果系列教材

Visual C++程序设计案例教程

谭建辉 主编

李 红 甘汪雨 副主编

科学出版社

北 京

内 容 简 介

本书系统地介绍了面向对象程序设计技术及 C++ 语言的相关知识, 内容包括面向对象程序设计技术的概念和特征、C++ 语言基础、类和对象、派生与继承、多态性、模板、输入/输出流、综合案例等。

本书注重基本概念, 从案例应用出发, 分析案例, 引入知识, 突出重点, 叙述清楚, 深入浅出, 论述详尽, 使读者既能深刻领会面向对象程序设计思想, 了解面向对象程序设计的特征, 又能掌握 C++ 语言的编程及实际应用。

本书可作为高职高专院校计算机及相关专业的教学用书, 也可供所有计算机用户参考使用。

图书在版编目 (CIP) 数据

Visual C++ 程序设计案例教程/谭建辉主编. —北京: 科学出版社, 2010
(高等职业教育“十一五”规划教材·计算机类核心课程教改项目成果系列教材)

ISBN 978-7-03-029696-2

I. ①V… II. ①谭… III. ①C 语言—程序设计—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2010) 第 238559 号

责任编辑: 孙露露 / 责任校对: 王万红
责任印制: 吕春珉 / 封面设计: 耕者设计工作室

科学出版社出版

北京东黄城根北街 16 号
邮政编码: 100717

<http://www.sciencep.com>

铭浩彩色印装有限公司 印刷

科学出版社发行 各地新华书店经销

*

2011 年 2 月第 一 版 开本: 787×1092 1/16
2011 年 2 月第一次印刷 印张: 16 3/4
印数: 1—3000 字数: 382 000

定价: 26.00 元

(如有印装质量问题, 我社负责调换<环伟>)

销售部电话 010-62134988 编辑部电话 010-62135763-8212

版权所有, 侵权必究

举报电话: 010-64030229; 010-64034315; 13501151303



前言

面向对象程序设计技术是程序设计方法学的一场革命。C++语言是一种面向对象程序设计语言，且与C语言兼容。Visual C++是应用程序集成开发环境，Visual C++ 6.0是开发Windows应用程序的理想工具。本书是一本可以系统学习和迅速掌握C++语言的实用教程。

本书基于案例实现过程介绍Visual C++程序设计的基础知识和操作技能。在内容编排上由简单到复杂，案例讲解深入浅出，通俗易懂，便于自学。

本书写作特色如下。

一是详略得当，主次分明。

C++语言中包含的内容较多，由于篇幅的限制不可能面面俱到，必须有所舍弃。书中对于非重点或较复杂的内容略讲，如数组部分重点讲解一维和二维数组，三维数组只介绍概念；在结构与联合部分，重点讲解结构，而联合的内容着重讲解与结构的区别。

二是例题、案例生动，实用性强。

本书针对C++语言的特点，精选重点，强化主要概念，图文并茂地讲解每个重要知识点，并配以容易理解的程序实例，以例题释义，总结知识，便于理解和应用。本书还安排有收银机，二进制与十进制相互转换，学生成绩录入、排序并输出，图书管理，人员信息管理，日期程序，通讯录等实用小程序或案例，通过案例引入、案例分析、知识准备、案例实现的结构呈现出来，引导读者很快走入程序设计的知识殿堂，使读者不再感觉学习C++很累很难。

本书全部程序都在Visual C++ 6.0上调试通过，并给出了程序运行结果，便于读者自学。

三是配有实训练习题及参考答案，学练结合。

本书每章末都有适当的实训练习题，题目典型、多样，覆盖本章知识点。通过练习，可以达到巩固强化所学知识和技能的作用。

四是本书配套教学资源丰富。

本书配有教学课件、例题和实训题目的源程序代码及习题参考答案，可到科学出版社网站（www.abook.cn）下载。

C++程序设计语言是计算机专业的一门专业基础课。本课程建议总学时为 72 学时，理论课 36 学时，实训课 36 学时。本书可作为高职高专院校计算机及相关专业的教学用书，也可供所有计算机用户参考使用。本书具体学时分配见下表。

章序号	章名称	学时数
第 1 章	C++与面向对象程序概述	4 学时
第 2 章	数据类型与表达式	4 学时
第 3 章	数据输入/输出与控制语句	10 学时
第 4 章	函数	8 学时
第 5 章	数组、指针与字符串	12 学时
第 6 章	类与对象	10 学时
第 7 章	继承与派生	8 学时
第 8 章	多态性	4 学时
第 9 章	模板	4 学时
第 10 章	输入/输出流	4 学时
第 11 章	综合案例	4 学时

本书由谭建辉、李红、甘汪雨、李连天、敖卓缅、罗明挽、贺慧玲、张三群、周献珍、相景丽、张春丽、王越超共同编写，由谭建辉任主编。在编写本书过程中，我们参考了有关资料（见参考文献），在此谨向相关作者表示衷心的感谢。

由于编者水平有限，书中难免存在疏漏和差错，诚恳地期望广大读者批评指正。



目 录

第 1 章 C++与面向对象程序概述	1
1.1 Visual C++ 6.0 编程环境简介	2
1.2 简单的 C++程序设计案例	7
1.3 程序设计语言的发展	8
1.4 C++语言与面向对象程序设计的特点	9
本章小结	10
实练习题	10
第 2 章 数据类型与表达式	12
2.1 C++的基本数据类型	13
2.2 常量与变量	15
2.3 表达式与操作符	19
2.4 位运算符	21
2.5 C++的构造类型（结构、联合和枚举）	22
本章小结	26
实练习题	27
第 3 章 数据输入/输出与控制语句	29
3.1 案例引入（收银机）	30
3.2 案例分析	30
3.3 知识准备	31
3.4 案例实现	41
本章小结	43
实练习题	43
第 4 章 函数	45
4.1 案例引入（二进制与十进制相互转换）	46
4.2 案例分析	47



4.3 知识准备	48
4.4 案例实现	63
本章小结	64
实练习题	65
第5章 数组、指针与字符串	67
5.1 案例引入（学生成绩录入、排序并输出）	68
5.2 案例分析	69
5.3 知识准备	69
5.4 案例实现	93
本章小结	95
实练习题	95
第6章 类与对象	98
6.1 案例引入（图书管理）	99
6.2 案例分析	99
6.3 知识准备	100
6.4 案例实现	121
本章小结	126
实练习题	127
第7章 继承与派生	130
7.1 案例引入（人员信息管理）	131
7.2 案例分析	132
7.3 知识准备	133
7.4 案例实现	151
本章小结	155
实练习题	155
第8章 多态性	157
8.1 案例引入（日期程序）	158
8.2 案例分析	158
8.3 知识准备	158
8.4 案例实现	177
本章小结	179
实练习题	180
第9章 模板	181
9.1 函数模板	182
9.2 模板函数的覆盖	185
9.3 类模板	186
9.4 程序举例	193

本章小结	194
实练习题	195
第 10 章 输入/输出流	196
10.1 案例引入 (通讯录)	197
10.2 案例分析	199
10.3 知识准备	199
10.4 案例实现	218
本章小结	226
实练习题	227
第 11 章 综合案例	228
11.1 系统功能	229
11.2 程序主要结构	229
11.3 程序实现代码	231
11.4 程序运行结果	246
本章小结	250
实练习题	250
附录	251
附录 1 C++语言常用库函数	251
附录 2 ASCII 表	256
参考文献	258



第 1 章

C++与面向对象程序 概述

【知识教学目标】

1. 了解 Visual C++ 6.0 的编程环境;
2. 熟悉 C++源程序的创建、编辑、编译和执行过程;
3. 了解程序设计语言的发展历程;
4. 掌握 C++语言及面向对象程序设计的特点。

【技能培养目标】

1. 能够创建 C++工程项目并添加 C++源文件;
2. 能够编辑、编译、执行简单的 C++程序。



1.1 Visual C++ 6.0 编程环境简介

Visual C++开发环境是一个基于 Windows 操作系统的可视化、面向对象的集成开发环境（Integrated Development Environment, IDE）。在该环境下，用户可以开发有关 C 和 C++的各种应用程序，包括建立、编辑、浏览、保存、编译、链接和调试应用程序等操作，这些操作都可以通过单击菜单选项或工具按钮来完成，使用方便、快捷。它还提供了项目工作区（WorkSpace）、应用程序向导（AppWizard）、类操作向导（ClassWizard）和 WizardBar 等实用编程工具。

1.1.1 Visual C++6.0 界面介绍

Visual C++6.0 集成开发环境的主界面由标题栏、菜单栏、工具栏、工作区（Workspace）窗口、编辑区和输出窗口几部分组成，如图 1-1 所示。

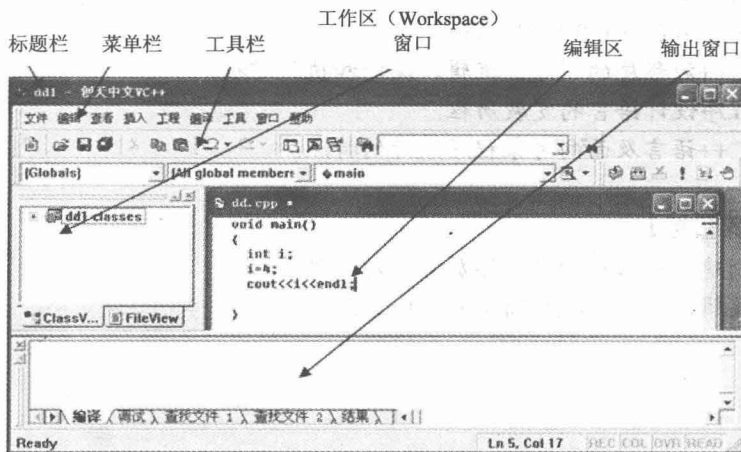


图 1-1 Visual C++ 6.0 集成开发环境主界面

1.1.2 编辑、编译和运行程序

1. 先建立 C++项目再添加文件

(1) 建立 C++项目

选择菜单“文件|新建”（“File|New”），弹出“新建”（“New”）对话框。选择“工程”（“Projects”）标签，并在左侧窗口选择“Win32 Console Application”，输入项目的名称，并选择（或输入）项目存放的文件夹，单击“确定”（“OK”）按钮，如图 1-2 所示。

选择空的项目，单击“完成”（“Finish”）按钮，出现一个显示项目信息的对话框；再单击“确定”（“OK”）按钮，项目创建完毕。

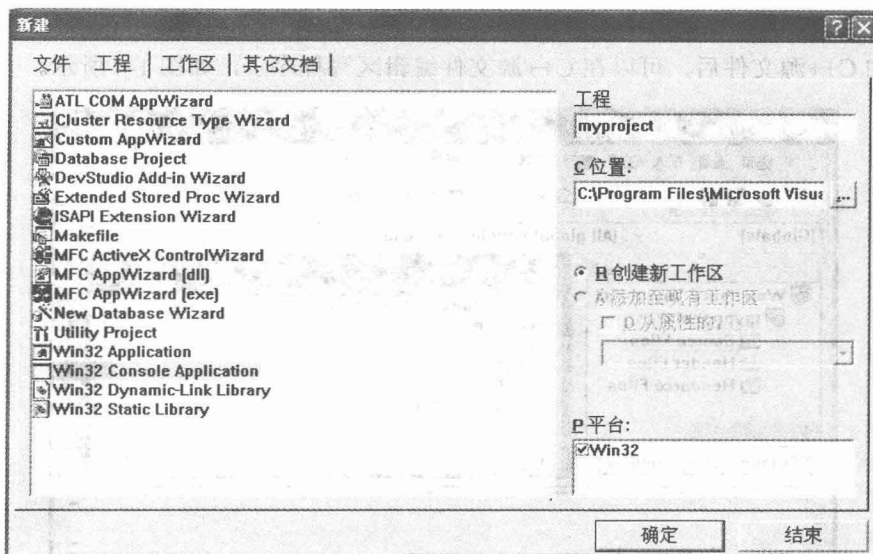


图 1-2 C++工程项目建立

(2) 添加文件

选择菜单“工程|添加到工程|新建”（“Project|Add To Project|New”），又弹出“新建”（“New”）对话框。选择“文件”（“Files”）标签，在左侧窗口选择“C++源程序”（“C++ Source File”），在右侧窗口输入文件名，并单击“确定”（“OK”）按钮，则在项目中添加了一个 C++源文件。如果要建立头文件，则要选择“C/C++头文件”（“C/C++ Header File”），如图 1-3 所示。

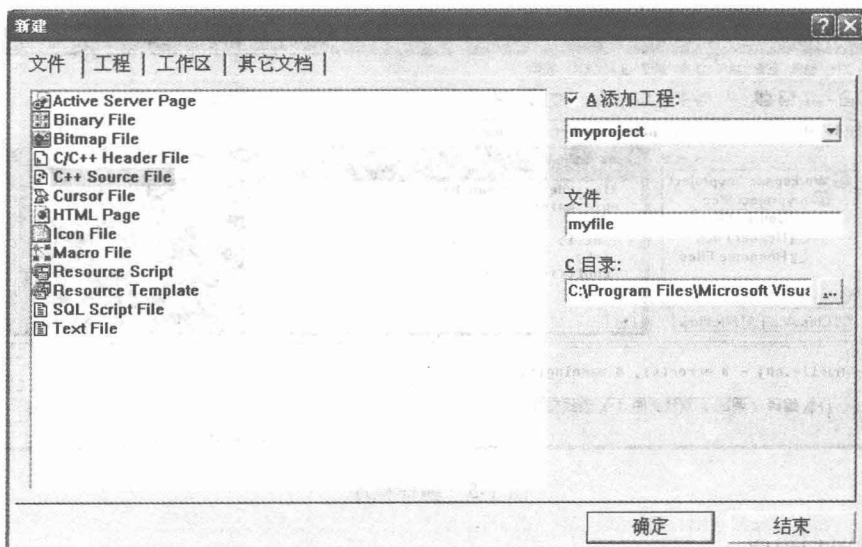


图 1-3 C++源程序建立

(3) 编辑程序

添加 C++ 源文件后, 可以在 C++ 源文件编辑区写程序了, 如图 1-4 所示。



图 1-4 C++源程序编写

(4) 编译程序

单击编译工具栏上的“编译”按钮, Visual C++编译器开始编译程序。如果程序没有语法错误, 就会成功编译; 否则, 就会在 Visual C++下方的输出窗口显示错误信息。根据提示信息修改程序中的错误, 再重新编译, 直到将所有语法错误修改完毕, 如图 1-5 所示。



图 1-5 编译程序

(5) 运行程序

编译通过后, 单击工具栏上的“运行”(感叹号“!”)按钮运行程序, 如图 1-6 所

示。如果经过各种测试,程序运行的输出结果和我们预计的相同,说明程序可以正常工作了。如果运行结果和我们预计的不一致,说明虽然程序没有语法错误,但仍然有逻辑错误,这时就要对程序进行调试了。

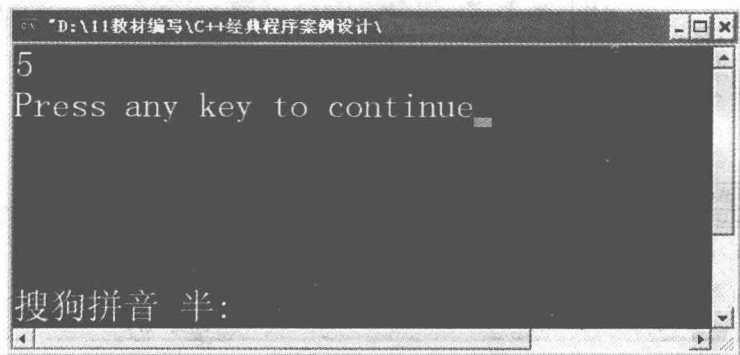


图 1-6 程序运行界面

2. 直接建立 C++源文件

(1) 建立 C++源文件

选择菜单“文件|新建”(“File|New”),弹出“新建”(“New”)对话框,选择“文件”(“Files”)标签。在左侧窗口选择“C++源程序文件”(“C++ Source File”),在右侧窗口输入文件名,并选择文件存放的文件夹,单击“确定”(“OK”)按钮,则建立了一个C++源文件。

(2) 编译程序

输入完源程序代码后,单击编译工具栏上的“编译”按钮,弹出一个信息框,询问是否要为程序创建一个项目工作空间(Project Workspace),单击“是”按钮,开始编译,如图 1-7 所示。

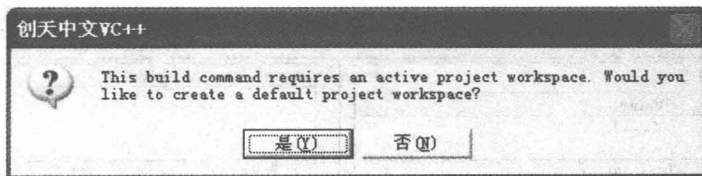


图 1-7 信息框

1.1.3 程序调试

程序运行结果如果与预计的结果不相同,可以使用调试工具栏对程序进行调试。

1. 设置断点

把光标定位在某一行语句,然后单击按钮,设定断点,如图 1-8 所示。

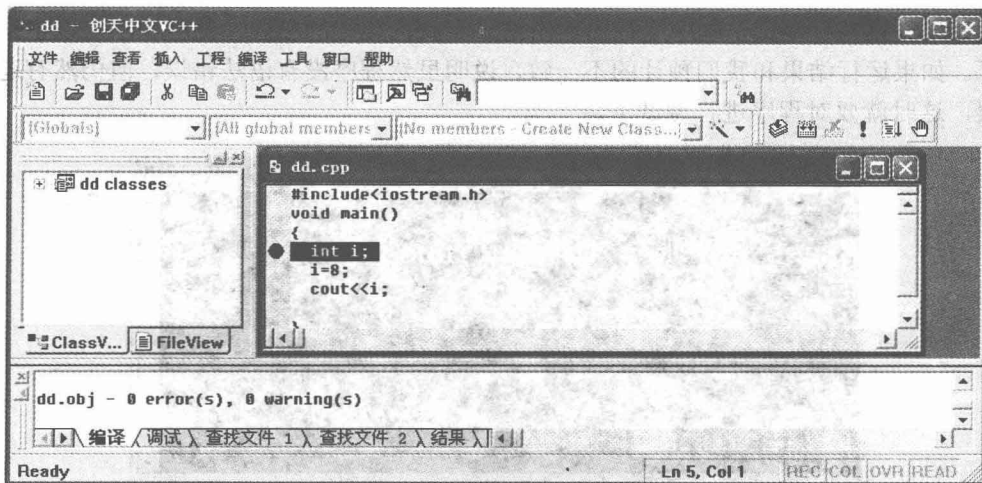


图 1-8 断点设置

2. 单步运行

打开“编译”菜单，选择“调试”，然后选择“Step Into (单步执行)”，如图 1-9 所示。

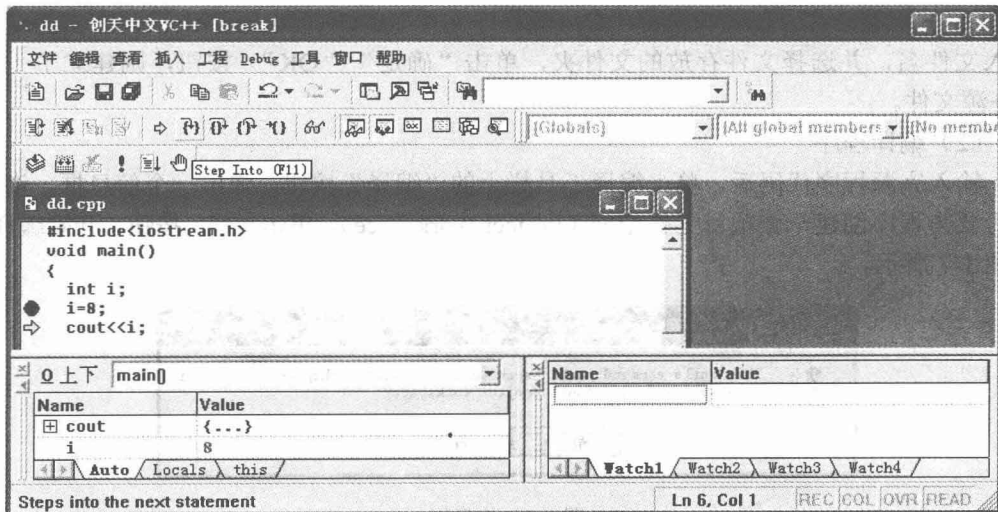


图 1-9 单步执行，观察变量值

3. 观察变量的值

在调试过程中，用户可以通过 Watch 和 Variable 窗口查看当前变量的值，如图 1-9 所示。这些信息可以反映程序运行过程中的状态变化结果的正确与否，可以反映程序是否有错，再加上人工分析，可以发现错误所在。

1.2 简单的 C++ 程序设计案例

1.2.1 cout 标准输出

格式:

```
cout << 待输出的内容 1 << 待输出的内容 2 ...;
```

【例 1-1】 使用 cout 进行简单的输出。

```
#include <iostream.h>
void main()
{
    cout<<"*****"<<endl;
    cout<<"* 您好!                *"<< endl;
    cout<<"* 这是一个简单的 C++程序。 *"<< endl;
    cout<<"*****"<<endl;
}
```

运行结果如图 1-10 所示。



图 1-10 运行结果

1.2.2 cin 标准输入

格式:

```
cin >> 保存数据的变量 1 >>保存数据的变量 2 ...;
```

【例 1-2】 使用 cin 进行简单的输入。

```
#include <iostream.h>
void main()
{
    int a,b;
    cin >> a >> b;
    cout << a << "," << b << endl;
}
```

运行输入 23，按回车键，结果如图 1-11 所示。

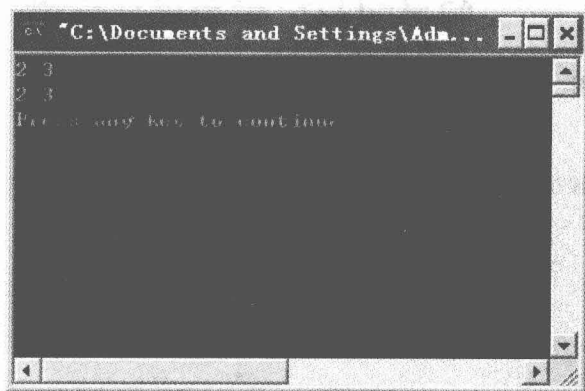


图 1-11 运行结果

1.3 程序设计语言的发展

1. 机器语言

由计算机硬件系统可以识别的二进制指令组成的语言称为机器语言，其由指令码和内存地址组成指令。

2. 汇编语言

汇编语言是用人们比较习惯的符号来代替指令编码。例如，用“ADD”来代替“001”，表示加法操作；用“Move”来代替“010”，表示数据移动。

汇编语言和机器语言都属于低级语言，它们的共同缺点是：依赖于机器，可移植性差；代码冗长，不易于编写大规模程序；可读性差，可维护性差。

3. 高级语言

与汇编语言和机器语言相比，高级语言更接近人类的自然语言，程序可以采用具有一定涵义的数据命名和容易理解的语句编写，如 Basic、Fortran、Algol 60、Pascal 和 C 语言等。

高级语言的发展经历了高级语言编程的初级阶段、结构程序设计阶段和面向对象程序设计阶段。

结构化程序设计的特点是：编写程序时，从解决问题的步骤出发，按照解决问题的过程设计程序，自顶向下、逐步求精。它是一种面向过程的程序设计方法，将程序分解为模块，使得数据和数据处理过程分离。

面向对象的程序设计方法是：将数据和处理数据的过程封装在一起，形成一个有机的整体（即类），更符合人们通常的思维习惯，使得开发的软件产品易重用、易修改、

易测试、易维护、易扩充。

1.4 C++语言与面向对象程序设计的特点

1.4.1 C++语言的特点

C++语言是在C语言的基础上发展而来的,同时它又支持面向对象的程序设计。C++语言主要具有以下特点。

1) 继承自C语言的优点,包括语言简洁、紧凑;使用方便、灵活;拥有丰富的运算符;生成的目标代码质量高,程序执行效率高;可移植性好等。

2) 对C语言进行的改进,包括编译器更加严格,引入引用的概念,引入const常量和内联函数,取代宏定义等。

3) 同时支持面向过程和面向对象的方法。在C++环境下既可以进行面向对象的程序设计,也可以进行面向过程的程序设计。因此,它也具有数据封装和隐藏、继承和多态等面向对象的特征。

1.4.2 面向对象程序设计的特点

1. 面向对象程序设计的基本概念

在分析面向对象的思想之前,先看一个生活中经常接触到的卖报亭。这类卖报亭的特点是:有杂志、报纸销售,杂志和报纸都有价格和数量,而卖报人备有钱箱。现针对卖报亭分析如下。

类:售报服务类

对象:卖亭报

属性(封装的):

报纸

杂志

钱箱

服务(行为):

售报(对外)

清点钱箱(对内)

窗口:传递消息

消息:“请给我一份报纸!”

通常人们把所有的卖报亭归为一种,这就是抽象。面向对象思想与日常活动中的思维方式是相似的,主要有对象、类、继承、封装、消息传递、多态性等概念。

1) 类。具有相同特性(数据元素)和相同行为(函数)的对象的集合,如汽车类、人类等。

2) 对象。是现实世界实际存在的事物,是类的一个具体实例。面向对象程序设计