



新世纪


高等职业教育
计算机类课程规划教材

VC++程序设计

新世纪高等职业教育教材编审委员会组编 主编/张凯 主审/付兴宏



GAODENG ZHIYE JIAOYU JISUANJILEI
KECHENG GUIHUA JIAOCAI

 大连理工大学出版社

12C
3



高等职业教育计算机类课程规划教材

GAODENGZHIYE JIAOYU JISUANJILEI KECHENG GUIHUAJIAOCAI

VC++ 程序设计

新世纪高等职业教育教材编审委员会组编

主 审 符兴宏

主 编/张 凯 副主编/李永昌 张伟强 任继梅



VC++ CHENG XU SHE JI

大连理工大学出版社

DALIAN UNIVERSITY OF TECHNOLOGY PRESS

图书在版编目(CIP)数据

VC++ 程序设计/张凯主编. —大连:大连理工大学出版社,2002.8

高等职业教育计算机类课程规划教材

ISBN 7-5611-2158-X

I . V… II . 张… III . C 语言-程序设计-高等学校:技术学校-教材
IV . TP312

中国版本图书馆 CIP 数据核字(2002)第 052285 号

大连理工大学出版社出版发行
大连市凌水河 邮政编码:116024
电话:0411-4708842 传真:0411-4701466
E-mail: dulp@mail.dlptt.ln.cn
URL: http://www.dulp.com.cn
大连理工印刷有限公司印刷

教材建设指导委员会

主任委员:

戴克敏(大连职业技术学院院长 教授)

副主任委员(以姓氏笔画为序):

王 敏(辽宁商务职业学院院长 教授)

王永申(盘锦职业技术学院院长)

李竹林(河北建材职业技术学院院长 副教授)

范利敏(丹东职业技术学院院长 教授)

宛 力(沈阳电力高等专科学校副校长 教授)

聂云超(渤海船舶职业学院院长 副教授)

曹勇安(黑龙江东亚学团董事长 齐齐哈尔职业学院院长 副教授)

常 佶(内蒙古工业大学副校长 教授)

鞠学孟(吉林财税高等专科学校校长 教授)

会员单位(排名不分先后):

沈阳电力高等专科学校

丹东职业技术学院

大连职业技术学院

辽宁商务职业学院

齐齐哈尔职业学院

青岛大学高等职业技术学院

烟台大学职业技术学院

广西财政高等专科学校

南昌水利水电高等专科学校

山东铝业职业技术学院

河北建材职业技术学院



燕山大学继续教育学院	辽宁大学高等职业技术学院
承德石油高等专科学校	辽宁工程技术大学技术与经济学院
内蒙古工业大学职业技术学院	辽宁工程技术大学职业技术学院
内蒙古财经学院高职教育部	辽宁工学院职业技术学院
内蒙古建筑职业技术学院	辽宁公安司法管理干部学院
呼伦贝尔学院	辽宁经济管理干部学院
包头钢铁学院职业技术学院	辽宁农业管理干部学院
齐齐哈尔大学职业技术学院	辽宁农业职业技术学院
大庆职业技术学院	辽宁省交通高等专科学校
佳木斯大学职业技术学院	辽阳职业技术学院
黑龙江省建筑职业技术学院	辽阳石油化工高等专科学校
牡丹江大学	盘锦职业技术学院
吉林财税高等专科学校	沈阳大学高等职业技术学院
吉林交通职业技术学院	沈阳大学师范学院
吉林粮食高等专科学校	沈阳工业大学高等职业技术学院
吉林大学应用技术学院	沈阳建工学院高等职业技术学院
四平职业大学	沈阳农业大学高等职业技术学院
沈阳师范学院高等职业技术学院	铁岭师范高等专科学校
鞍山钢铁学院职业技术学院	营口高等职业学院
鞍山师范学院职业技术学院	辽宁金融职业技术学院
本溪冶金高等专科学校	沈阳建工学院职业技术学院
渤海船舶职业学院	辽阳信息职业技术学院
朝阳师范高等专科学校	辽宁中医学院职业技术学院
大连大学	沈阳电视大学
大连轻工业学院职业技术学院	沈阳医学院职业技术学院
大连国际商务职业学院	沈阳音乐学院职业艺术学院
大连水产学院职业技术学院	沈阳职工大学
辽宁对外经贸职业学院	大连医学院丹东分院
辽宁机电职业技术学院	
东北财经大学高等职业技术学院	
抚顺师范高等专科学校	
抚顺石油学院高等职业技术学院	
抚顺职业技术学院	
阜新高等专科学校	
锦州师范高等专科学校	
锦州师范学院	
辽宁财政高等专科学校	



新华社

总序

我们已经进入了一个新的充满机遇与挑战的时代，我们已经跨入了21世纪的门槛。

20世纪与21世纪之交的中国，高等教育体制正经历着一场缓慢而深刻的革命，我们正在对传统的普通高等教育理论教学与社会发展的现实需要不相适应的现状作历史性的反思与变革的尝试。

20世纪最后的几年里，高等职业教育的迅速崛起，是影响高等教育体制变革的一件大事。在短短的几年时间里，普通中专教育、普通高专教育全面转轨，以高等职业教育为主导的各种形式的应用型人才培养的教育发展到与普通高等教育等量齐观的地步，其来势之迅猛，迫人深思。

无论是正在缓慢变革着的普通高等教育，还是迅速推进着的应用型人才培养的高等职业教育，都向我们提出了一个同样的严肃问题：中国的高等教育为谁服务，是为教育发展自身，还是为包括教育在内的大千社会？答案肯定而且惟一，那就是教育也置身其中的现实社会。

由此又引发出高等教育的目的问题。既然教育必须服务于社会，它就必须按照不同领域的社会需要来完成自己的教育过程。换言之，教育资源必须按照社会划分的各个专业（行业）领域（岗位群）的需要实施配置，这就是我们长期以来明乎其理而疏于力行的学以致用问题，这就是我们长期以来未能给予足够关注的教育的目的问题。

如所周知，整个社会由其发展所需要的不同部门构成，包括公共管理部门如国家机构、基础建设部门如教育研究机构和各种实业部门如工业部门、商业部门，等等。每一个部门又可作更为具体的划分，直至同它所需要的各种专门人才相对应。教育如果不能按照实际需要完成各种专门人才培养的目标，就不能很好地完成社会分工所赋予它的使命，而教育作为社会分工的一种独立存在就应受到质疑（在市场经济条件下尤其如此）。可以断言，按照社会的各种不同需要培养各种直接有用人才，是教育体制变革的终极目的。



随着教育体制变革的进一步深入,高等院校的设置是否会同社会对人才类型的不同需要一一对应,我们姑且不论。但高等教育走应用型人才培养的道路和走理论型(也是一种特殊应用)人才培养的道路,学生们根据自己的偏好各取所需,始终是一个理性运行的社会状态下高等教育正常发展的途径。

高等职业教育的崛起,既是高等教育体制变革的结果,也是高等教育体制变革的一个阶段性表征。它的进一步发展,必将极大地推进中国教育体制变革的进程。作为一种应用型人才培养的教育,高等职业教育从专科层次起步,进而高职本科教育、高职硕士教育、高职博士教育……当应用型人才培养的渠道贯通之时,也许就是我们迎接中国教育体制变革的成功之日。从这一意义上说,高等职业教育的崛起,正是在为必然会取得最后成功的教育体制变革奠基。

高职教育还刚刚开始自己发展道路的探索过程,它要全面达到应用型人才培养的正常理性发展状态,直至可以和现存的(同时也正处在变革分化过程中的)理论型人才培养的教育并驾齐驱,还需假以时日;还需要政府教育主管部门的大力推进,需要人才需求市场的进一步完善发育,尤其需要高职教学单位及其直接相关部门肯于做长期的坚忍不拔的努力。新世纪高等职业教育教材编审委员会就是由北方地区近百所高职院校和出版单位组成的旨在以推动高职教材建设来推进高等职业教育这一变革过程的联盟共同体。

在宏观层面上,这个联盟始终会以推动高职教材的特色建设为己任,始终会从高职教学单位实际教学需要出发,以其对高职教育发展的前瞻性的总体把握,以其纵览全国高职教材市场需求的广阔视野,以其创新的理念与创新的组织形式,通过不断深化的教材建设过程,总结高职教学成果,探索高职教材建设规律。

在微观层面上,我们将充分依托众多高职院校联盟的互补优势和丰裕的人才资源优势,从每一个专业领域、每一种教材入手,突破传统的片面追求理论体系严整性的意识限制,努力凸现高职教育职业能力培养的本质特征,在不断构建特色教材建设体系的过程中,逐步形成自己的品牌优势。

新世纪高等职业教育教材编审委员会在推进高职教材建设事业的过程中,始终得到了各级教育主管部门(如国家教育部、辽宁省教育厅)以及各相关院校相关部门的热忱支持和积极参与,对此我们谨致深深谢意;也希望一切关注、参与高职教育发展的同道朋友,在共同推动高职教育发展、进而推动高等教育体制变革的进程中,和我们携手并肩,共同担负起这一具有开拓性挑战意义的历史重任。

新世纪高等职业教育教材编审委员会

2001年8月18日

前 言

《VC++ 程序设计》是新世纪高等职业教育教材编审委员会组编的计算机类课程规划教材之一。

Visual C++ 由于支持面向对象程序设计并提供了全面的 MFC 类库,大大降低了软件开发的难度,缩短了开发周期,所以自诞生以来即成为 Windows 环境下最主要的应用开发系统,在程序员中得到了广泛的应用。

高等职业教育培养的是适应生产、建设、管理和服务第一线需要的技术应用型人才,为了适应这一教育目标的需要,我们组织了部分相关高职院校具有 VC++ 程序设计教学经验的一线教师,编写了这部《VC++ 程序设计》教材。

本教材从介绍面向对象程序方法入手,由浅入深并结合大量实例,介绍了使用 Visual C++ 开发 Windows 应用程序的原理及方法,全书由两部分内容构成。

第一部分包括第 1~4 章,简单介绍了面向对象程序设计方法的基本概念,C++ 语言基本语法规则及在 C++ 应用程序中如何实现封装性、继承性和多态性。这一部分内容主要为后续使用向导编写 MFC 应用程序打基础,以够用为度,后面不常用的内容这里不作介绍。为了上机方便,介绍了 Visual C++ 6.0 开发环境,并用一个小例子说明了控制台应用程序的编写方法。

第二部分包括第 5~12 章,首先介绍了 Windows 程序的运行机制和 MFC,然后介绍使用 AppWizard 和 ClassWizard 编写 MFC 应用程序的方法,最后以一个完整的数据库应用程序实例对前面的内容加以概括。

根据高等职业教育和 Visual C++ 本身的特点,本书加强了实际编程能力的训练,从实际程序开发的需要出发,注重培养学生运用基本知识解决实际问题的能力,而不过分追求知识的系统性和完整性。



《VC++ 程序设计》由沈阳电力高等专科学校张凯任主编,辽宁工程技术大学职业技术学院李永昌、丹东职业技术学院张伟强、山东铝业职业技术学院任继梅任副主编,参加本书编写工作的还有丹东职业技术学院李华和辽宁工程技术大学职业技术学院邢亚军。其中第 1,3,4,12 章由张凯编写,第 2 章由任继梅编写,第 5,7 章由张伟强编写,第 6,8,10 章由李永昌编写,第 9 章由邢亚军编写,第 11 章由李华编写。付兴宏通审了全部书稿并提出了有益的改进意见;李卓玲、张小瑞、王建、赵鹏山等同志在编写本教材的过程中提供了很多帮助,在这里对他们表示诚挚的感谢。

高等职业教育在我国还刚刚兴起,如何实现教材的特色建设还处于探索阶段,尽管我们在寻求本教材特色建设的突破上做出了很大的努力,但不足之处恐在所难免,恳切希望各相关高职院校和读者在使用本教材的过程中给予热忱关注,并将意见及时反馈我们,以便修订时改进。

编者

2002 年 7 月



第 1 章 面向对象程序设计	1
1.1 什么是面向对象程序设计	1
1.2 面向对象程序设计中的几个基本概念	4
习题.....	8
第 2 章 Visual C++ 6.0 概述	9
2.1 C++ 编程简介	9
2.2 Visual C++ 6.0 的历史	11
2.3 Visual C++ 6.0 可视化开发环境	12
2.4 Visual C++ 6.0 程序开发的一般步骤	16
习题	20
实训	20
第 3 章 C++ 中的封装	21
3.1 从结构体到类	21
3.2 对象	26
3.3 对象指针	31
3.4 静态成员	33
3.5 友元	35
3.6 对象使用实例	38
习题	41
实训	42
第 4 章 C++ 中的继承性与多态性	44
4.1 C++ 中的继承性	44
4.2 C++ 中的多态性	60
习题	69
实训	70
第 5 章 Windows 应用程序开发基础	71
5.1 Windows 运行机制	71
5.2 匈牙利表示法	74
5.3 API 编程	74
5.4 MFC 编程	82
习题	97

实训	97
第 6 章 AppWizard 与 ClassWizard	98
6.1 AppWizard	98
6.2 ClassWizard	109
6.3 Windows 应用程序实例	116
习题	125
实训	126
第 7 章 键盘与鼠标	128
7.1 键盘消息与键盘消息处理	128
7.2 鼠标消息与鼠标消息处理	136
7.3 键盘与鼠标应用实例	142
习题	153
实训	154
第 8 章 资源文件的使用	155
8.1 资源文件的概念	155
8.2 定制应用程序向导生成的应用程序所调用的资源	156
8.3 自定义资源	158
习题	164
实训	164
第 9 章 Windows 常用控件与对话框	165
9.1 Windows 常用控件	165
9.2 Windows 对话框	177
9.3 控件与对话框使用实例	184
习题	188
实训	188
第 10 章 图形设备接口	189
10.1 CDC 类	189
10.2 图形对象	192
10.3 用 CDC 绘图实例	194
10.4 用 CDC 输出文字实例	198
习题	202
实训	203
第 11 章 文件读写	204
11.1 CFile 类	204
11.2 CFile 派生类	208
11.3 文件操作实例	211
习题	216
实训	216

第 12 章 Visual C++ 在数据库方面的应用	217
12.1 数据库编程概述	217
12.2 MFC 的 ODBC 类简介	219
12.3 ODBC 数据库编程实例	221
习题	236
实训	236

第 1 章

面向对象程序设计

通过本章的学习,理解什么是面向对象程序设计,以及面向对象程序设计所涉及的几个重要概念:类与对象、封装性、继承性、多态性。

1.1 什么是面向对象程序设计

在程序设计与实现中,计算机编程语言由低级语言向高级语言不断地发展,程序设计方法正在从面向过程程序设计走向面向对象程序设计,使得编程语言与自然语言之间以及程序设计方法与实际解决问题方式之间的距离越来越近。这就意味着软件开发人员可以用更接近自然的思维方式,用更少的精力完成同样的工作。C++作为一种当今广泛流行的面向对象程序设计语言,它是从传统的面向过程编程语言——C语言基础上发展演化而来的,了解什么是面向过程程序设计有助于理解面向对象程序设计。

1.1.1 面向过程程序设计

【引例】 用面向过程的程序设计语言——C语言建立银行系统管理软件,以便说明面向过程程序设计的特点。为说明问题,只建立软件的简化模型。银行系统允许顾客开设不同类型的银行账户,其中包括现金账户、支票账户和贷款账户,同时允许顾客存款、取款和转账。用面向过程的方法,首先要建立一个简单的数据结构,可用一个结构体类型来表示:

```
Struct account{
char name[8];           /* 用户姓名 */
unsigned long accountId; /* 账号 */
float balance;         /* 余额 */
float interestYTD;     /* 年利息 */
char accountType;     /* 账户类型(现金、支票、贷款) */
//...
};
```

然后,建立三个函数 MakeDeposit、Withdraw 和 Transfer,分别完成存款、取款和转账三项工作。

```
MakeDeposit(略)       /* 存款 */
{//...}
Withdraw(略)          /* 取款 */
{//...}
Transfer(略)          /* 转账 */
```

```
{//...}
```

然后,建立主函数。

```
main()                /* 定义结构体类型变量,调用三个函数对数据进行处理 */  
{//...}
```

最后,编写每个函数模块相应的程序代码,完成整个设计。

由上面这个例子可以看出,面向过程程序设计的主要思想是功能的分解并逐步求精。当任务非常复杂以至于无法描述时,可以将它按功能拆分为一系列较小的功能模块,直到这些小的功能模块小到易于理解并能用代码实现的程度。面向过程程序设计把程序看做是处理数据的一系列过程,过程或函数被定义为一个接一个顺序执行的一组指令,数据与过程分开存储,编程的关键在于如何按功能分解实际问题以及追踪确定函数之间的调用关系。在上面所述的例子中,把银行系统按功能分解为三个功能模块,在主函数中完成数据的定义及确定函数之间的调用关系,而数据(账户)与过程(三个功能模块)是分离的。

面向过程的程序设计为处理复杂问题提供了有利的手段。然而,到了 20 世纪 80 年代末,随着软件规模越来越大,它的一些缺点越来越突出地表现出来。

第一,面向过程的程序设计方法对问题域的分析是以功能为中心的,而不是以问题域中存在的实际事物为单位,人为地打破了实际事物的原貌及相互关系。这种分析结果不能在软件设计中直接地映射或体现实际问题域中事物的原貌及相互之间的联系。当实际问题复杂,软件规模变大时,函数之间的调用关系将变得异常复杂而难以确定,给软件的后续开发带来很大困难,这些困难有时是无法克服的。面向过程的程序设计方法要求软件设计人员对实际问题有非常深刻的认识,这样才能很好地将软件分解,而这在社会分工越来越细的当今社会,伴随着软件规模的扩大,要做到这一点几乎是不可能的。虽然可以有专业人员配合程序员的工作,但程序员很容易因为对实际问题理解上的不同而造成错误,而这种错误专业人员又很难发现并予以纠正,给软件设计及编程调试带来很多困难。

第二,当数据量或功能增加时,数据与处理这些数据的过程或函数之间的分离使程序变得越来越难以理解,对数据处理能力的要求越强,这种分离所造成的副作用越明显。并且数据因为得不到有效的保护,程序员可以方便地修改数据,例如账目,因为对程序员来说,不过是在修改数据结构中的数字,实际上他可能通过修改数据而取走顾客的钱。

而且,由于数据对所有函数都可见,而这些函数可能是由不同的人完成的,很难确定数据的修改出自哪个函数,给数据的安全保护及程序的调试带来很多困难。

第三,数据结构一旦变化,整个程序都得进行修改。比如,原来只有现金账户、支票账户和贷款账户三种类型,对不同的账号类型,以不同的代码与之对应(在同一个函数中分情况处理,对三种账户类型的处理不会完全相同)。假设现在增加一种退休账户类型,就需要修改所有三个处理函数,增加与之相应的处理代码。虽然各种账户类型的处理不完全相同,但它们是相似的,可能只有很小的差别。面向过程的方法没有提供代码重用机制,相似的功能也要从头编写代码而不能利用已有的相似功能的代码。这样,每当数据结构发生变化时就需要修改整个源程序,给程序的维护带来很多困难。

由于图形界面的广泛应用,程序运行由顺序的过程驱动演变为非线性的事件驱动,软件使用起来越来越方便。由于面向过程程序设计方法无法克服的缺陷,用面向过程的方

法描述与实现软件设计越来越困难,在这种背景下,面向对象的程序设计方法诞生并逐渐流行起来。

1.1.2 面向对象程序设计

【引例】 按照面向对象的程序设计思想,用 C++ 语言设计上面的银行管理系统软件模型。

```
class BankAccount{
private:
    float balance;
    float interestYTD;
    char * owner;
    int account _ number;

public:
    void MakeDeposit(float amount);
    float WithDraw(float amount);
    bool Transfer(BankAccount & to, float amount);
};
//...
int main()
{
    //...                //定义银行账户对象并调用对象的方法
    return 0;
}
```

在面向对象程序设计中,重点在于那些将要被操作的数据,而不是实现这些操作的过程或函数,数据构成了软件分解的基础而不是功能。更重要的是不能将数据与相应的操作看成两个分离的实体,要把它们作为一个完整的实体来看待。

分析银行管理系统,数据当然是银行账号,而存款、取款和转账这些操作是建立在银行账户这个数据基础上的,两者密不可分。当把银行账户和对银行账户的操作定义在一起,银行账户就是一个类,称为银行账户类(BankAccount)。在主函数中建立许多具体的银行账户,作为银行账户类的实例,每一个具体的银行账户就是银行账户类的一个对象。这样可以调用各个对象的功能函数对其私有数据进行操作。与面向过程的程序设计比较,面向对象程序设计有以下几个明显的优点:

第一,面向对象程序设计中的对象和类来源于现实世界,按实际对象分解软件很容易,并且可以按照实际对象之间的关系设计程序中对象之间的关联。反过来说,在面向对象程序设计中可以直接映射实际对象及其相互关系,程序设计空间中的结构与实际问题域中的结构是一致的,程序设计思想就是实际解决问题的思路,程序结构更清晰易懂,程序员与专业人员更易达成一致。

第二,对象的数据不能被外部程序或过程直接访问。如果想修改银行账户中的数据,唯一的办法是在该数据上提供修改操作,即提供一个修改数据的接口。这样,外部对象或过程只能通过这个接口访问这个数据,对象的数据得到了有效的保护。

第三,由于面向对象程序设计引入了代码重用机制,当对软件的要求改变时,并不需要大量的工作。只需在已有程序的基础上对所涉及部分稍作修改就能实现,而不用修

改整个源程序,也不会涉及其他部分。对于新增的账户类型,可以通过继承与重载等手段去解决而不用修改原来的银行账户类。

面向对象程序设计是一种新的程序设计范型。面向对象程序的主要结构特点是:第一,程序一般是由类的定义和类的使用两部分组成,在主程序中定义各个对象并规定它们之间传递消息的规律;第二,程序中的一切操作都是通过向对象发送消息来实现的,对象接收到消息后,启动有关方法完成相应的操作。

1.2 面向对象程序设计中的几个基本概念

1.2.1 对象与类

一般意义上,对象是现实世界中的一个实体,其特征是:

- (1) 每一个对象必须有一个名字以区别于其他对象;
- (2) 用属性描述对象的特征;
- (3) 用一组操作描述对象的行为。

类是对一组具有共同的属性特征和行为特征的对象抽象。例如,由一个一个的人构成人类,而一个人是人类的一个实例(对象)。类和对象之间的关系是抽象和具体的关系,类是对多个相似对象进行综合抽象(抽取共同点)的结果,对象又是类的一个具体实例。

例如,教师王安妮是一个教师对象,她具有如下特征:

对象名:王安妮

对象的属性:

年龄:25

性别:女

婚否:否

学历:硕士

职称:讲师

单位:沈阳电力高等专科学校,信息工程系

...

对象的行为:

学习

讲课

...

一个个像王安妮这样的教师就构成了教师类。

从一般意义上讲,对象是现实世界中的一个实际存在的事物,它可以是有形的,也可以是无形的。对象是构成世界的一个独立单位,它具有自己的静态特征和动态特征。面向对象方法中的对象是由描述其属性的数据和定义在其上面的一组操作组成的实体,是数据单元和过程单元的组合物。类是对一组对象的抽象,这组对象具有相同的属性结构

(都具有某些属性,但属性值不同以区别各个对象)和操作行为,在对象所属的类中要说明这些结构和行为,也可以说类是一种新的自定义的数据类型,它包括对将要创建对象的状态描述和对操作行为的说明。

以前面定义的银行账户类为例,用户姓名、账号、余额、年利息、账户类型就叫做属性,每个银行账户对象都具有这些特征属性,但每个具体银行账户对象的属性值是不同的,例如,每个银行账户对象都有账号,但其值是不同的。又如,每个银行账户对象的余额都必须大于或等于零,这叫做静态属性;而每个银行账户对象余额的具体值是不同的,或一个银行账户对象的余额随着时间的不同也是变化的,这叫做动态属性。静态属性与动态属性之和构成了对象的属性。

存款、取款、转账这些成员函数(为简化,只给出名字)就是操作行为。把属性和定义在属性之上的操作行为结合在一起构成了一种新的数据类型——类。用银行账户类定义的具体变量叫做对象,例如

```
BankAccount a;
```

a 就是属于 BankAccount 类的一个对象。

类是一种新的数据类型,不能给它分配内存,只有具体变量才能占据内存单元。类与对象的关系可以用整形 int 和整形变量 i 来类比。属于银行账户类的所有的银行账户对象具有相同的行为(存款、取款和转账),但是有各自独立的状态(属性不同,例如账号、姓名等等)。

在面向对象程序设计中,类的确定与划分是非常重要的,是软件开发的基础,划分的好坏直接影响软件开发的后续工作和软件产品的质量。类的确定没有统一的标准和固定的方法,基本上依赖设计人员的经验、技巧以及对实际问题的理解程度。所以,一个实际问题,不同的人有不同的分类方法,对这一点要有一个正确的认识。

1.2.2 封装性

1.2.2.1 现实世界中的抽象与封装

抽象与封装是一对密不可分的概念,现实世界中,抽象与封装的例子很多。以电视机为例,只需按几个按钮,电视机就会播放想看的节目,至于按下某一个按钮后,内部如何实现,我们是不关心的,所有的复杂操作过程都被封装在电视机外壳里面,对用户而言,这些复杂的实现过程是隐蔽的,不可见的,这就是封装。

而抽象是封装的前提,它是对大量实例抽取共同性质而形成概念的过程。例如,对各种式样各种原理的电视机的功能及用户的使用需求进行广泛的分析,抽取其共同的功能特征,形成电视机类,进一步决定保留哪几个按钮,既能实现电视机的功能又能使用户易学易用,这就是抽象的过程,它强调了共性,忽略了个性,强调能做什么,而忽略了怎样做。

1.2.2.2 对象的封装

将数据结构和作用于其上的操作组成一个实体,数据的表示方式和对数据的操作细节被隐藏起来,用户只知道如何通过操作接口对该数据进行操作,而不知道是如何进行