高等院校课程设计案例精编

赠送光盘中附有完整的案例源代码

# Visual C++.NET 课程设计案例精编

陈艳华　余　健　雷志军　编著

- 酒店管理系统
- 贸易公司管理系统
- 图像预处理系统
- 多功能绘图系统
- 远程控制系统
- 网络测试系统
- 局域网聊天系统
- 联机帮助与打包

赠送光盘

清华大学出版社

高等院校课程设计案例精编

# Visual C++ .NET 课程设计案例精编

陈艳华　余　健　雷志军　编著

清华大学出版社
北　京

## 内 容 简 介

Visual C++ .NET 是微软公司开发的集成开发环境(IDE)中功能强大的程序设计软件之一，它的应用领域非常广泛，尤其是在网络、图形、数据库方面，已成为各软件公司首选的开发工具。

全书共分 10 章，第 1 章和第 2 章为案例开发所需要的理论基础知识以及相关 API 的说明；第 3 章至第 10 章介绍了 7 个经典案例。其中，第 3 章和第 4 章为数据库方面的案例，分别是酒店管理系统、贸易公司管理系统，涉及 ODBC、ADO 的数据库开发；第 5 章至第 9 章为图形、网络方面的案例，分别是图像预处理系统、多功能绘图系统、远程控制系统、网络测试系统、局域网聊天系统；第 10 章介绍了如何利用 Visual Studio .NET 进行打包、发行。

本书内容翔实、语言简练、思路清晰、图文并茂、理论与实际设计相结合，既适合作为高等院校计算机、自动化、机械、电子等相关专业学生的课程设计指导书，也适合作为开发人员的参考用书。



图书在版编目(CIP)数据

Visual C++ .NET 课程设计案例精编/陈艳华，余健，雷志军编著. —北京：清华大学出版社，2008.6
(高等院校课程设计案例精编)
ISBN 978-7-302-17611-4

Ⅰ. V…　Ⅱ. ①陈…　②余…③雷…　Ⅲ. C 语言—程序设计—高等学校—教学参考资料　Ⅳ. TP312

中国版本图书馆 CIP 数据核字(2008)第 068030 号

**责任编辑**：李春明　闫光龙
**封面设计**：山鹰工作室
**版式设计**：杨玉兰
**责任校对**：周剑云
**责任印制**：孟凡玉
**出版发行**：清华大学出版社　　**地　　址**：北京清华大学学研大厦 A 座
http://www.tup.com.cn　　**邮　　编**：100084
**社　总　机**：010-62770175　　**邮　购**：010-62786544
**投稿与读者服务**：010-62776969，c-service@tup.tsinghua.edu.cn
**质 量 反 馈**：010-62772015，zhiliang@tup.tsinghua.edu.cn
**印 刷 者**：北京密云胶印厂
**装 订 者**：三河市新茂装订有限公司
**经　　销**：全国新华书店
**开　　本**：185×260　**印　张**：25.25　**字　数**：608 千字
**版　　次**：2008 年 6 月第 1 版　　**印　次**：2008 年 6 月第1 次印刷
附光盘 1 张
**印　　数**：1～4000
**定　　价**：42.00 元

# 前　言

课程设计是教学计划和环节中的一个重要部分，通过课程设计，学生能够得到系统的技能训练，巩固和加强所学的专业理论知识，其目的是培养学生的综合运用能力，使学生成为具有扎实理论基础和较强独立动手能力的专业人才。

Visual C++ .NET 作为 Visual Studio .NET 家族中的一分子，是微软公司基于最新的 .NET 思想开发的以 C++语言为基础的可视化集成开发工具，它与 Visual Basic .NET、Visual C# .NET 等一起被称为 Visual Studio .NET 的集成开发环境。

Visual C++ .NET 开发工具是以 C++语言为基础的，此语言的基本语法结构很复杂，往往会给编程爱好者，甚至于专业编程人员在学习 Visual C++ .NET 时造成极大的困难，而使用 Visual C++ .NET 进行系统的开发更是困难重重。因此，本书根据 Visual C++ .NET 使用 C++语言的基本理论进行应用系统开发编写和使用，供广大读者学习 Visual C++ .NET，制作课程设计以及撰写毕业论文时参考。

本书首先介绍了案例开发所需要的理论知识以及相关 API 的说明，使读者能知其然，然后通过 7 个完整的应用系统的制作开发，使读者熟悉 Visual C++ .NET 开发的步骤、方法和技巧，达到知其所以然的目的。本书中提供的应用实例比较有针对性，各案例都独立成章，分别进行讲述，具体内容如下。

案例一：利用存储过程进行数据操作完成酒店管理系统开发，利用皮肤控件进行美化。

案例二：利用数据库技术进行贸易公司管理系统的开发。

案例三：结合科研实际开发图像预处理系统。

案例四：利用 GDI 进行多功能绘图系统开发，对 C++的继承和多态有更深的应用。

案例五：利用 Socket 进行 C/S 模式的远程控制系统的开发。

案例六：利用 Socket 开发的网络测试系统，对与服务器通信的失包率进行统计。

案例七：利用 Socket、多线程、数据库等多种技术开发局域网聊天系统。

案例八：利用 Vusual Studio .NET 创建程序打包和安装的方法。

本书由陈艳华、余健、雷志军、孙福兆、邓会忠主要执笔，参加编写的人员还有穆志维、伍建刚、陈伟、孙守凯、杨立平、吴宏彬、陈思成、张艳超、马秀萍、李双红、孙景辉、方海刚、许钊，在此一并向他们表示感谢。

由于编者水平有限，加上时间仓促，书中难免有一些不足之处，欢迎同行和读者批评指正。

编　者

# 目　录

# 第 1 章　数据库编程基础

## 1.1　关系数据库的基本概念

关系数据库建立在集合代数基础上，应用数学方法来处理数据库中的数据。现实世界中的各种实体以及实体之间的各种联系均用关系模型来表示。换句话说，关系数据库是建立在关系模型基础上的数据库。

1970 年，IBM 公司的研究员 E.F.Codd 发表了题为《大型共享数据库的关系模型》的论文，提出了数据库的关系模型，奠定了关系数据库的理论基础。关系数据库产品一经问世，就以其简单清晰的概念，易懂易学的数据库语言，使用户不需了解复杂的存取路径细节，不需说明“怎么干”，只需指出“干什么”就能操作数据库，从而深受广大用户喜爱，涌现出许多性能优良的商品化关系数据库管理系统，即 RDBMS。著名的 DB2、Oracle、Ingres、Sybase、Informix 等都是关系数据库管理系统。关系数据库产品也从单一的集中式系统发展到可在网络环境下运行的分布式系统，从联机事务处理到支持信息管理、辅助决策，系统的功能不断完善，数据库的应用领域不断扩大。

1999 年，ANSI(美国国家标准协会)和 ISO(国际标准化组织)发布了人们期待已久的 SQL 标准，这个标准被称为 SQL:1999(也称为 SQL3)。SQL:1999 以关系数据模型为基础，定义了如何在关系数据库中存储和操作数据。RDBMS 将 SQL 标准作为其产品的基础，以提供既支持 SQL 又支持关系数据模型的数据库环境。

## 1.2　关 系 范 式

在实现设计阶段，常常使用关系规范化理论来指导关系数据库设计。其基本思想为：每个关系都应该满足一定的规范，从而使关系模式设计合理，达到减少冗余、提高查询效率的目的。

为了建立冗余较小、结构合理的数据库，将关系数据库中关系应满足的规范划分为若干等级，每一级称为一个“范式”。范式的概念最早是由 E.F.Codd 提出的，他从 1971 年相继提出了三级规范化形式，即满足最低要求的第一范式(1NF)，在 1NF 基础上又满足某些特性的第二范式(2NF)，在 2NF 基础上再满足一些要求的第三范式(3NF)。1974 年，E.F.Codd 和 Boyce 共同提出了一个新的范式概念，即 Boyce-Codd 范式，简称 BC 范式。1976 年 Fagin 提出了第四范式(4NF)，后来又有人定义了第五范式(5NF)。至此，在关系数据库规范中建立了一个范式系列：第一范式(1NF)、第二范式(2NF)、第三范式(3NF)、BCNF 、第四范式(4NF)和第五范式(5NF)(它们之间的包含关系如图 1-1 所示)，这 6 种范式一级比一级有更严格的要求，满足不同程度的要求构成不同的范式级别。下面分别介绍各范式的定义。

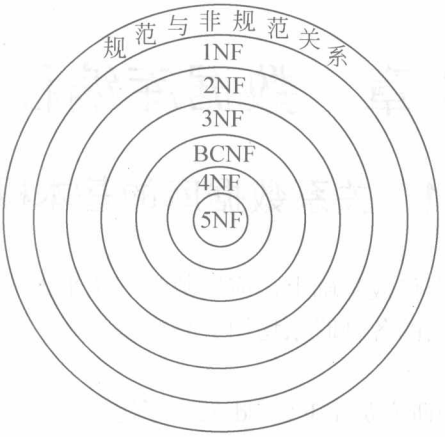


图 1-1　各种范式之间的关系

### 1. 第一范式(1NF)

在关系模式 R 中的每一个具体关系 r 中，如果每个属性值都是不可再分的最小数据单位，则称 R 是第一范式的关系。例如：职工号、姓名、电话号码组成一个表(一个人可能有一个办公室电话号码和一个家里电话号码)，规范成为 1NF 有三种方法。

(1) 重复存储职工号和姓名。这样，关键字只能是电话号码。

(2) 职工号为关键字，电话号码分为单位电话和住宅电话两个属性。

(3) 职工号为关键字，但强制每条记录只能有一个电话号码。

以上三种方法，第一种方法最不可取，按实际情况选取后两种方法。

### 2. 第二范式(2NF)

如果关系模式 R(U，F)中的所有非主属性都完全依赖于任意一个候选关键字，则称关系 R 是属于第二范式的。

例：选课关系 SCI(SNO，CNO，GRADE，CREDIT)，其中 SNO 为学号，CNO 为课程号，GRADE 为成绩，CREDIT 为学分。由以上条件，关键字为组合关键字(SNO，CNO)，使用以上关系模式有以下问题。

(1) 数据冗余。假设同一门课由 40 个学生选修，学分就重复 40 次。

(2) 更新异常。若调整了某课程的学分，相应的元组 CREDIT 值都要更新，有可能会出现同一门课学分不同。

(3) 插入异常。如计划开新课，由于没人选修，没有学号关键字，只能等有人选修才能把课程和学分存入。

(4) 删除异常。若学生已经结业，从当前数据库删除选修记录。某些门课程新生尚未选修，则此门课程及学分记录无法保存。

原因：非关键字属性 CREDIT 仅依赖于 CNO，也就是 CREDIT 部分依赖组合关键字(SNO，CNO)而不是完全依赖。

解决方法：分成两个关系模式 SC1(SNO，CNO，GRADE)，C2(CNO，CREDIT)。新关系包括两个关系模式，它们之间通过 SC1 中的外关键字 CNO 相联系，需要时再进行自

然连接，恢复原来的关系。

### 3. 第三范式(3NF)

如果关系模式 R(U，F)中的所有非主属性对任何候选关键字都不存在传递信赖，则称关系 R 是属于第三范式的。

例如：S1(SNO，SNAME，DNO，DNAME，LOCATION)各属性分别代表学号、姓名、所在系、系名称和系地址。

关键字 SNO 决定各个属性。由于是单个关键字，没有部分依赖的问题，肯定是 2NF。但这关系肯定有大量的冗余，有关学生所在的几个属性 DNO、DNAME 和 LOCATION 将重复存储，插入、删除和修改时也将产生类似的情况。

问题原因：是由于关系中存在传递依赖造成的，即 SNO->DNO，DNO->LOCATION。而 DNO->SNO 却不存在。因此关键字 SNO 对 LOCATION 的决定是通过传递依赖 SNO->LOCATION 实现的。也就是说，SNO 不直接决定非主属性 LOCATION。

解决目地：每个关系模式中不能留有传递依赖。

解决方法：分为两个关系 S(SNO，SNAME，DNO)和 D(DNO，DNAME，LOCATION)。

注意：关系 S 中不能没有外关键字 DNO，否则两个关系之间将失去联系。

### 4. BCNF

如果关系模式 R(U，F)中的所有属性(包括主属性和非主属性)都不传递依赖于 R 的任何候选关键字，那么称关系 R 是属于 BCNF 的，或是关系模式 R。如果每个决定因素都包含关键字(而不是被关键字所包含)，则称 R 是 RCNF 的关系模式。

例：配件管理关系模式 WPE(WNO，PNO，ENO，QNT)各属性分别表示仓库号、配件号、职工号和数量。有以下条件。

(1) 一个仓库有多个职工。

(2) 一个职工仅在一个仓库工作。

(3) 每个仓库里每种型号的配件由专人负责，但一个人可以管理几种配件。

(4) 同一种型号的配件可以分放在几个仓库中。

分析：由以上得 PNO 不能确定 QNT，由组合属性(WNO，PNO)来决定，存在函数依赖(WNO，PNO) ->ENO。由于每个仓库里的每种配件由专人负责，而一个人可以管理几种配件，所以由组合属性(WNO，PNO)才能确定负责人，有(WNO，PNO)-> ENO。因为一个职工仅在一个仓库工作，有 ENO->WNO。由于每个仓库里的每种配件由专人负责，而一个职工仅在一个仓库工作，有(ENO，PNO)->QNT。

找一下候选关键字。因为(WNO，PNO)->QNT，(WNO，PNO)-> ENO，因此(WNO，PNO)可以决定整个元组，是一个候选关键字。根据 ENO->WNO，(ENO，PNO)->QNT，故(ENO，PNO)也能决定整个元组，为另一个候选关键字。属性 ENO、WNO 和 PNO 均为主属性，只有一个非主属性 QNT。它对任何一个候选关键字都是完全函数依赖的，并且是直接依赖，所以该关系模式是 3NF。

分析一下主属性。因为 ENO->WNO，主属性 ENO 是 WNO 的决定因素，但是它本身不是关键字，只是组合关键字的一部分。这就造成主属性 WNO 对另外一个候选关键字

(ENO，PNO)的部分依赖，因为(ENO，PNO)-> ENO，但反过来不成立，而 PNO-> WNO，故(ENO，PNO)-> WNO 也是传递依赖。

虽然没有非主属性对候选关键字的传递依赖，但存在主属性对候选关键字的传递依赖，同样也会带来麻烦。如一个新职工分配到仓库工作，但暂时处于实习阶段，没有独立负责对某些配件的管理任务。由于缺少关键字的一部分，而无法将 PNO 插入到该关系中去。又如某个人不管配件而改去负责安全，则在删除配件的同时该职工也会被删除。

解决办法：分成管理 EP(ENO，PNO，QNT)，关键字是(ENO，PNO)；工作 EW(ENO，WNO)，关键字是 ENO。

缺点：分解后函数依赖的保持性较差。此例中，由于分解，函数依赖(WNO，PNO)->ENO 丢失了，因而对原来的语义有所破坏。没有体现出每个仓库里一种部件由专人负责，有可能出现一个部件由两个或两个以上的人来同时管理。因此，分解之后的关系模式降低了部分完整性约束。

一个关系分解成多个关系，要使得分解有意义，起码的要求是分解后不丢失原来的信息。这些信息不仅包括数据本身，而且包括由函数依赖所表示的数据之间的相互制约。进行分解的目标是达到更高一级的规范化程度，但是分解的同时必须考虑两个问题：无损联接性和保持函数依赖。有时往往不可能做到既有无损联接性，又完全保持函数依赖。要根据需要进行权衡。

第四范式、第五范式是建立在多移植依赖和联系依赖基础上的，限于篇幅，这里不再详细介绍。

## 1.3 数据字典与数据流图

### 1.3.1 数据字典(DD)

数据字典(Data Dictionary，简称 DD)是用来定义数据流图中的各个成分的具体含义的，它以一种准确的、无二义性的说明方式为系统的分析、设计及维护提供了有关元素的一致的定义和详细的描述。它和数据流图共同构成了系统的逻辑模型，是需求规格说明书的主要组成部分。

#### 1. 数据字典的内容以及格式

数据字典的任务是对于数据流图中出现的所有被命名的图形元素在数据字典中作为一个词条加以定义，使得每一个图形元素的名字都有一个确切的解释。

数据字典有以下 4 类条目：数据流、数据项、数据存储和基本加工。

数据字典中所有的定义应是严密的、精确的，不可有含混，不可有二义性。

1) 数据流条目

数据流条目给出了 DD 中数据流的定义，通常列出该数据流的各组成数据项。在定义数据流或数据存储组成时，使用的符号如表 1-1 所示。

表 1-1　数据流条目说明

| 符　号 | 含　义 | 实例及说明 |
| --- | --- | --- |
| = | 被定义为 | |
| + | 与 | x=a+b，表示 x 由 a 和 b 组成 |
| [...\|...] | 或 | x=[a\|b]，表示 x 由 a 或 b 组成 |
| m{...}n 或 {...}mn | 重复 | x=2{a}5，表示 x 中最少出现 2 次 a，最多出现 5 次 a，2、5 为重复次数的上、下限 |
| {...} | 重复 | x={a}，表示 x 由 0 个或多个 a 组成 |
| (...) | 可选 | x=(a)，表示 a 可在 x 中出现，也可不出现 |
| "..." | 基本数据元素 | x="a"，表示 x 是取值为字符 a 的数据元素 |
| .. | 连接符 | x=1..9，表示 x 可取 1 到 9 中任意一个值 |

举例：定义数据流组成及数据项。

机票=姓名+日期+航班号+起点+终点+费用
姓名={字母}
航班号="Y7100"..."Y8100"
终点=[上海|北京|西安]

数据流条目主要内容及举例如下。

数据流名称：订单
别名：无
简述：顾客订货时填写的项目
来源：顾客
去向：加工 1“检验订单”
数据流量：1000 份/每周
组成：编号+订货日期+顾客编号+地址+电话+银行账号+货物名称+规格+数量

2) 数据存储条目

数据存储条目是对数据存储的定义，举例如下。

数据存储名称：库存记录
别名：无
简述：存放库存中所有可供货物的信息
组成：货物名称+编号+生产厂家+单价+库存量
组织方式：索引文件，以货物编号为关键字
查询要求：要求能立即查询

3) 数据项条目

数据项条目是不可再分解的数据单位，其定义格式如下。

数据项名称：货物编号
别名：G-No，G-num，Goods-No
简述：本公司的所有货物的编号
类型：字符串
长度：10

取值范围及含义如下。

第 1 位：进口/国产
第 2～4 位：类别
第 5～7 位：规格
第 8～10 位：品名编号

4) 基本加工条目

基本加工条目是用来说明 DD 中基本加工的处理逻辑的，上层的加工由下层的基本加工分解而来的，只要有了基本加工的说明，就可理解其他加工。举例如下。

加工名：查阅库存
编号：1.2
激发条件：接收到合格订单时
优先级：普通
输入：合格订单
输出：可供货订单、缺货订单
加工逻辑：根据库存记录
IF 订单项目的数量<该项目库存量的临界值>
THEN 可供货处理
ELSE 此订单缺货，登录，待进货后再处理
ENDIF

### 2. 加工逻辑的描述

加工逻辑也称为“小说明”，描述加工逻辑一般用以下三种工具：结构化语言、判定表和判定树。

1) 结构化语言

结构化语言是介于自然语言和形式语言之间的一种半形式语言。结构化语言是在自然语言基础上加了一些限定，使用有限的词汇和有限的语句来描述加工逻辑，它的结构可分成外层和内层两层。

(1) 外层：用来描述控制结构，采用顺序、选择、重复三种基本结构。

(2) 内层：一般是采用祈使语句的自然语言短语，使用数据字典中的名词和有限的自定义词，其动词含义要具体，尽量不用形容词和副词来修饰。

2) 判定表

在有些情况下，数据流图中的某些加工的一组动作信赖于多个逻辑条件的取值，用自然语言或结构化语言都不易清楚地描述出来，而用判定表就能够清楚地表示复杂的条件组合与应做的动作之间的对应关系。

判定表由 4 个部分组成，如表 1-2 所示，构造一张判定表，可采用以下几个步骤。

(1) 提取问题中的条件。

(2) 标出条件的取值。

(3) 计算所有条件的组合数 N。

(4) 提取可能采用的动作或措施。

(5) 制作判定表。

(6) 完善判定表。

表 1-2　判定表结构

| 组成部分 | 详细说明 |
| --- | --- |
| 条件项 | 条件取值的组合 |
| 动作项 | 在各种取值组合下应执行的动作 |
| 条件桩 | 问题的所有条件 |
| 动作桩 | 条件项在各种取值情况下应采取的动作 |

### 3. 判定树

判定树是判定表的变形，一般情况下它比判定表更直观，且易于理解和使用。这 3 种描述加工逻辑的工具各有优缺点，对于顺序执行和循环执行的动作，用结构语言描述；对于存在多个条件复杂组合的判断问题，用判定表和判定树。判定树较判定表直观易读，判定表进行逻辑验证较严格，能把所有的可能性全部考虑到。可将两种工具结合起来，先用判定表打底稿，在此基础上产生判定树。

1) IDEF 方法

IDEF(ICAM Definition 的缩写)方法是美国空军在 1981 年针对集成化计算机辅助制造(Integrated Computer Aided Manufacturing，简称 ICAM)工程项目中用于进行复杂系统分析和设计的方法，是在结构化分析与设计技术的基础上提出来的。IDEF 方法分为 3 部分。

- IDEF0：用来描述系统的功能活动及其联系，建立系统的功能模型。
- IDEF1：用来描述系统的信息及其联系，建立系统的信息模型。
- IDEF2：用来进行系统模拟，建立系统的动态模型。

2) IDEF0 的图形表示

IDEF0 方法采用简单的图形符号和简洁的文字说明，描述系统在不同层次上的功能。在该方法中，将系统功能称为活动，将表示系统功能的图形称为活动图形。在活动图形中，用方框和箭头表示系统的各种活动及相互之间的关系。

3) 建立功能模型的基本方法

(1) 确定建模的范围、观点及目的。

在开始为系统建立模型时，首先要确定建模的立足点，包括范围、观点及目的。范围指所讨论的对象是什么，它的边界和外部接口是什么；观点指从什么角度去考虑所研究的问题；目的指确定所研究问题的意图及理由。

(2) 建立系统的内外关系图——A-0 图。

IDEF0 方法建立的功能模型是一组有层次关系的图形，用以字母 A 开头的编号来标志图形在层次中的位置。先建立系统的内外关系图，用来抽象地描述所研究的问题及其边界或数据接口。图中只有一个活动，活动名概括地描述系统的内容，用进入和离开的箭头表示系统与环境的数据接口，确定系统边界。

(3) 建立顶层图。

把 A-0 图分解为 3～6 个主要部分得到 A0 图，它清楚地表达了 A-0 图在同样信息范围内的细节，从结构上反映了模型的观点，是系统功能模型真正的顶层图。该图中各方框所表示活动的详细含义由低层次的图形说明。

(4) 建立低层次的图形。

按照自顶向下的方法，从 A0 图开始逐层分解，建立一系列的活动图形，直到最低层为止。

## 1.3.2 数据流图(DFD)

数据流图(Data Flow Datagram，简称 DFD)就是采用图形方式来表达系统的逻辑功能、数据在系统内部的逻辑流向和逻辑变换过程，是结构化系统分析方法的主要表达工具。基于计算机的信息处理系统由数据流和一系列的加工构成，这些加工将输入数据流加工为输出数据流。

### 1. 用数据流图描述数据流和加工

1) 数据流图的基本符号

数据流图由 4 种基本符号组成，如图 1-2 所示。


图 1-2 数据流图的基本符号

例：图 1-3 是一个简单的数据流图，它表示数据 X 从源 S 流出，经 P1 加工转换成 Y，接着经 P2 加工转换为 Z，在加工过程中从 F 中读取数据。

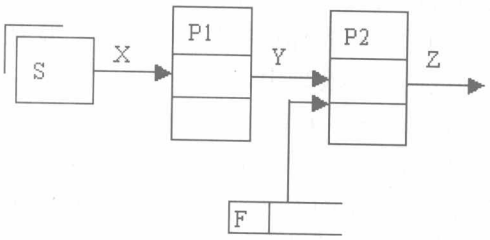


图 1-3 数据流图举例

下面来详细讨论各基本符号的使用方法。

2) 数据流

数据流由一组确定的数据组成。例如，“发票”为一个数据流，它由品名、规格、单位、单价、数量等数据组成。数据流用带有名字的具有箭头的线段表示，名字称为数据流

名，表示流经的数据，箭头表示流向。数据流可以从加工流向加工，也可以从加工流进、流出文件，还可以从源点流向加工或从加工流向终点。

对数据流的表示有以下约定：对流进或流出文件的数据流不需标注名字，因为文件本身就足以说明数据流。而其他数据流则必须标出名字，名字应能反映数据流的含义，数据流不允许同名。两个数据流在结构上相同是允许的，但必须体现人们对数据流的不同理解。例如，图 1-4(a)中的合理领料单与领料单两个数据流，它们的结构相同，但前者增加了合理性这一信息。

两个加工之间可以有几股不同的数据流，这是由于它们的用途不同，或它们之间没有联系，或它们的流动时间不同，如图 1-4(b)所示。

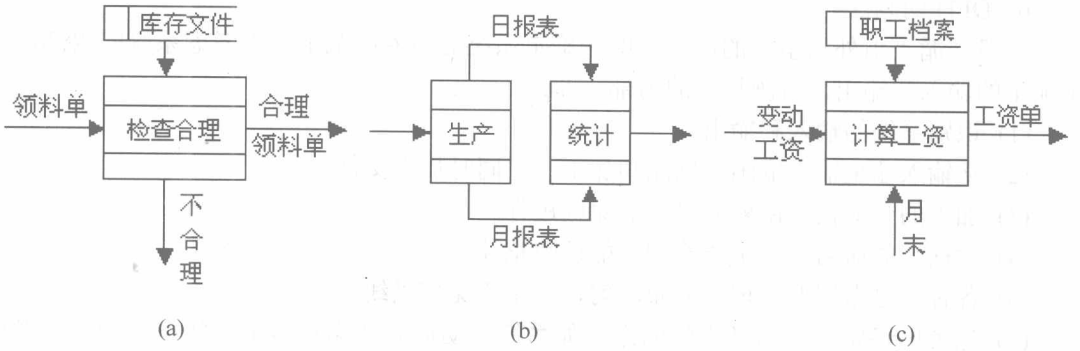


图 1-4　简单数据流图举例

数据流图描述的是数据流而不是控制流，而如图 1-4(c)中，“月末”只是为了激发加工“计算工资”，是一个控制流而不是数据流，所以应从图中删去。

3) 加工处理

加工处理是对数据进行的操作，它把流入的数据流转换为流出的数据流。每个加工处理都应取一个名字表示它的含义，并规定一个编号用来标识该加工在层次分解中的位置。名字中必须包含一个动词，例如“计算”、“打印”等。

对数据加工转换的方式有两种：改变数据的结构，例如，将数组中各数据重新排序；产生新的数据，例如，对原来的数据总计、求平均值等。

4) 文件

文件是存储数据的工具。文件名应与它的内容一致，写在开口长条内。从文件流入或流出数据流时，数据流方向很重要。如果是读文件，则数据流的方向应从文件流出，写文件时则相反；如果是又读又写，则数据流是双向的。在修改文件时，虽然必须先读文件，但其本质是写文件，因此数据流应流向文件，而不是双向。

例如，在图 1-4(a)中，检查合理性加工时，只从库存账目文件中读出库存信息与领料单核对，所以数据流从文件流出，箭头指向加工。

5) 数据源或终点

数据源和终点表示数据的外部来源和去处。它通常是系统之外的人员或组织，不受系统控制。

为了避免在数据流图上出现线条交叉，同一个源点、终点或文件均可在不同位置多次

出现，这时要在源(终)点符号的右下方画小斜线，或在文件符号左边画竖线，以示重复，如图 1-5 所示。


图 1-5　重复的源点、终点或文件

由图 1-4 可见，数据流图可通过基本符号直观地表示系统的数据流程、加工、存储等过程。但它不能表达每个数据和加工的具体、详细的含义，这些信息需要在“数据字典”和“加工说明”中表达。

6) DFD 的画法

一般遵循“由外向里”的原则，即先确定系统的边界或范围，再考虑系统的内部，先画加工的输入和输出，再画加工的内部。即：

(1) 识别系统的输入和输出。

(2) 从输入端至输出端画数据流和加工，并同时加上文件。

(3) 加工的分解按“由外向里”的原则进行。

(4) 数据流的命名，名字要确切，能反映整体。

(5) 各种符号布置要合理，分布均匀，尽量避免交叉线。

(6) 先考虑稳定态，后考虑瞬间态。如系统启动后先考虑正常工作状态，稍后再考虑系统的启动和终止状态。

对于不同的问题，数据流图可以有不同的画法。具体实行时可按下述步骤进行。

(1) 识别系统的输入和输出，画出顶层图。

即确定系统的边界。在系统分析初期，系统的功能需求等还不很明确，为了防止遗漏，不妨先将范围定得大一些。系统边界确定后，越过边界的数据流就是系统的输入或输出，将输入与输出用加工符号连接起来，并加上输入数据来源和输出数据去向就形成了顶层图。

(2) 画系统内部的数据流、加工与文件，画出一级细化图。

从系统输入端到输出端(也可反之)，逐步用数据流和加工连接起来，当数据流的组成或值发生变化时，就在该处画一个“加工”符号。

画数据流图时还应同时画上文件，以反映各种数据的存储处，并表明数据流是流入还是流出文件。

最后，再回过头来检查系统的边界，补上遗漏但有用的输入/输出数据流，删去那些没被系统使用的数据流。

(3) 加工的进一步分解，画出二级细化图。

同样运用“由外向里”方式对每个加工进行分析，如果在该加工内部还有数据流，则可将该加工分成若干个子加工，并用一些数据流把子加工联接起来，即可画出二级细化图。二级细化图可在一级细化图的基础上画出，也可单独画出该加工的二级细化图，二级细化图也称为该加工的子图。

(4) 其他注意事项。

一般应先给数据流命名，再根据输入/输出数据流名的含义为加工命名。名字含义要