

⑤  
16-22

# 通用报表系统的设计思想 An Idea to Design General Table System

魏 虞  
Wei Ying

TP391

(广西计算中心 南宁市 530022)  
(Computing Centre of Guangxi, Nanning, 530022)

A

**摘要** 介绍通用报表系统的主要设计思想,并着重叙述了需解决好的三个问题:表格存储;数据关系描述;表格输出中横表头的自动生成。

**关键词** 通用 报表系统 设计思想

表格存储, 软件, 表格输出,

**Abstract** This paper introduced an idea to design general table system and concentrated on the presentation of three problems to be solved: table storage, description of data relation, automatic generation of output horizontal table.

**Key words** General, table system, idea to design

## 1 前言

对报表的处理,是许多行业、许多部门的基本业务工作,而在报表处理软件的设计中,主要的应该是解决好以下几个问题:(1)表格存储(包括表的数据和表的各描述部分);(2)数据关系描述;(3)表格输出中横表头的自动生成。

下面分别叙述这三个问题。

## 2 表格存储

我们先来看一个报表的格式(如表 1):

表格结构分为四个部分:

(1) 表说明部分,包括标题、副标题、表说明、日期、数据单位和注释

(2) 横表头,包括层号、栏名称、类型、长度;

(3) 纵表头,包括科目、项目、部门、单位、地区及其他等;

(4) 运算公式,包括表内行与行、栏与栏之间的计算和审核关系;表与表之间计算、取数和审核关系。

1995-03-13 收稿。

表说明  
部分

## ××地区预算收入增长情况表

表1

日期：1992年3月8日

单位：千元

横表头	项 目	收 入			支 出		
		1990年	1991年	增长%	1990年	1991年	增长%
纵表头	合 计						
	农林水利气						
	工业交通						
	商 业						
	文 化						
	⋮						

注释一→制表单位：××××

一张表格分为两个库存储：描述库和主库。

## 1. 描述库

## (1) 描述库的结构

段号	段名	类型	宽度	小数位	字段说明
1	标志	C	2		记录内容标志
2	内容	C	70		标题、横表头、公式、注释等内容
3	字型	C	2		字型或类型等
4	表栏数	N	3		表栏数
5	表行数	N	3		表行数
6	每页行	N	3		每页打印行数
7	纵表头	N	2		纵表头指定栏目号
8	打印排数	N	1		打印排数
9	行距	C	1		打印行距

## (2) 表格的各描述部分在描述库中的存放位置

记录号	标志	内容	字型	表栏数	表行数	每页行	纵表头	打印排数	行距
1		标题	×	××	××	××	××	×	×
2		标题续							
3		副标题	×						
4		标题说明	×						
5		日期							
6		单位							

横表头结构	{	BL	栏名称	类型	宽度	小数位	打印宽度	层号
		⋮						
		BL						
公式	{	GS	公式	类型	行数	列数	止行	层号
		⋮						
		GS						
注释	{	ZS	注释	字型				
		⋮						
		ZS						

这里，横表头结构、公式和注释，在描述库中的记录顺序不一定按此排列，其原则是：哪个最后输（或修改），哪个就排在最后。

## 2. 主库

主库的结构是根据横表头结构（类型、宽度、小数位），由系统自动生成。

## 3 数据关系描述

报表中的数据关系是通过公式来描述的，公式的表达形式为：

$$H_i L_j = \text{算术表达式}$$

这里〈算术表达式〉是由单元  $A_i$  和算术运算符组成的，单元  $A_i$  的表达形式为：

$$A_i = [\text{表名:}] H_{x_1} L_{y_1}$$

其中  $H$  表示行， $x_1$  表示行号； $L$  表示栏， $y_1$  表示栏号， $H$ 、 $L$  不一定同时存在，当缺省  $H$ （或  $L$ ）时，表明包括所有行（或栏）。

算术运算符包括：

(、 )	括号	&	连加
*、/	乘、除	+、-	加、减

运算的优先级顺序为：自上而下。至于运算的优先级算法，这是一个专题问题，留待以后讨论。

下面是几条公式的例子：

$$H30 = H1 \& H29 \text{ (表示第 30 行为第 1 行连加到第 29 行)}$$

$$L7 = (L6 - L5) * 100 / L5 \text{ (表示第 7 栏为第 6 栏比第 5 栏数值的增减百分比)}$$

$$H1 = JHB; H17 \text{ (表示第 1 行从表 JHB 中的第 17 行取数而来)}$$

## 4 表格输出中横表头的自动生成

横表头打印格式的自动生成，对于一个通用报表软件来说，是极为重要的，它是报表系统能否实现通用性的关键之一，它应对报表结构维护、修改后，能够自动地作出相应的反映（自动调整）。因此，一个报表的横表头打印格式，其生成依据应尽量少的些、原始些，应该仅仅跟踪输入的横表头结构，而且是实时性的（即需输出时才生成），无需中间过程。

### 1. 横表头结构的描述

设定横表头结构，就是向计算机描述所需要的横表头格式。以前面所述表格为例，其横表头格式如下：

N1	N2			N6		
	N3	N4	N5	N7	N8	N9

其中 N1, N2... 为横表头的文字内容，下面我们就以上图为例，分别叙述横表头结构设定的各部分信息：

#### (1) 层号

分析上述一般表的横表头结构，我们把表的横表头分为几个不同的层次。最上面一层为第一层，即上图中 N1, N2 及 N6，其次是第二层 N3, N4, N5, N7, N8, N9，再次就是第三层，依次类推。我们称最下面一层为根层，N1, N3, N4, N5, N7, N8, N9 均为根层，把层的次序规定为层号，第一层的层号是 1，第二层为 2，第三层为 3，等等。

栏单元：第一层及其覆盖的下属层组成一个栏单元。如上图中共有三个栏单元，分别是：N1；N2, N3, N4, N5；N6, N7, N8, N9。

#### (2) 栏名称

就是打印出报表时，这一栏的名称。栏名称一般要求是汉字。当输入的栏名称内容长度大于该栏所要求的打印宽度时，打印时系统自动将栏名称转头，另打一行。

#### (3) 类型

是指该栏数据的类型，我们将数据类型分为字符型即 C 和数值型即 N 两种。字符型包括汉字、字母、数字和符号，其数据不能参加运算。数值型数据是指反映具体数值的数据，由数码字 0~9 及正 (+)、负 (-) 号组成，可以参加运算。输入数据类型时，只要输 C 或 N 即可，非根层栏则要输入空格。

#### (4) 长度

是指存放数据的宽度。根层栏必需输入 (>0)，非根层栏为 0。

#### (5) 小数位

是指数值数据小数点以后保留的位数。字符型栏不输入。

#### (6) 打印宽度

是指打印时该栏的宽度。根层栏必须输入，且大于或等于该栏数据长度，一般要求输偶数（若为奇数，系统将自动加 1 取偶）；非根层栏输 0，打印宽度为其所属下一层栏目的宽度总和。

根据上图,其横表头结构描述的值可以如下:

层号	栏名称	类型	长度	小数位	打印宽度
1	$N_1$	C	24		24
1	$N_2$				
2	$N_3$	N	10	2	12
2	$N_4$	N	10	2	12
2	$N_5$	N	6	2	6
1	$N_6$				
2	$N_7$	N	10	2	10
2	$N_8$	N	10	2	10
2	$N_9$	N	6	2	6

## 2. 横表头的生成算法

这里,我们主要依据三方面信息:层号、栏名称和打印宽度。至于类型、长度和小数位,那是为产生报表的数据库结构而服务的。对于只有一层的报表,其横表头的打印格式生成是较为容易的,只要根据每栏的打印宽度拼接,并将栏目汉字置入各栏中即成。而对于多层报表,其生成问题就较为复杂些,这要解决好两个问题:一个是层间的分隔线,另一个是各栏汉字的置入。

### (1) 层间分隔线产生算法

我们将第一层与第二层之间的分隔线称为第一条分隔线,第二层与第三层之间称为第二条分隔线,依次类推。由此可见,对一个具有 $n$ 层结构的横表头,它应有 $n-1$ 条分隔线。而只有一层的横表头是没有分隔线的,故其横表头较容易生成。

产生第 $i$ 条分隔线的有关信息是:第 $i$ 层及根层的打印宽度(当前处于第 $i$ 层状态)。其算法如下:

第一步:遍历所有栏(从左到右,由上而下),找到第一个层号为 $i$ 或是根层的栏。

第二步:给出分隔线初始符。当前栏为根层时赋“|”,非根层时则赋“└”。

第三步:当前栏若为根层,转下一步;非根层则转第七步。

第四步:当前栏层号小于等于 $i$ 时,转下一步;层号大于 $i$ 则转第六步。

第五步:上一栏层号大于 $i$ 时,分隔线加“└”;上一栏层号小于等于 $i$ 则加:“|”。然后分隔线加当前栏打印宽度的空符,转第八步。

第六步:分隔线加当前栏打印宽度的“—”,转第八步。

第七步:上一栏层号小于等于 $i$ 时,分隔线加“└”。

第八步:当前栏为最后一栏,转第十一步;否则,跳到下一栏。

第九步:如果上一栏为层号大于 $i$ 的根层,要做以下开关处理:

①当前栏层号为 $i$ 的非根层,分隔线加“+”,回初始状态;

②当前栏层号为 $i+1$ ,分隔线加“└”

③当前栏层号为小于*i*的非根层，分隔线加“┆”，回初始状态；

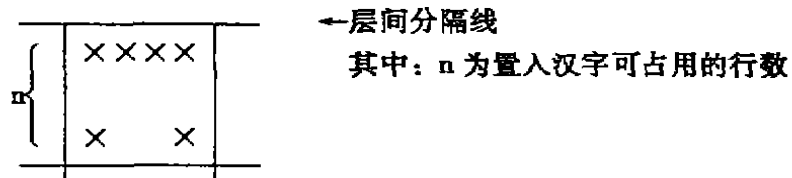
④当前栏层号大于*i*+1，分隔线加“—”；

第十步：找到层号为*i*或是根层的栏，返回第三步。

第十一步：最后一栏层号大于*i*时，分隔线加“┆”；小于等于*i*则加“|”。

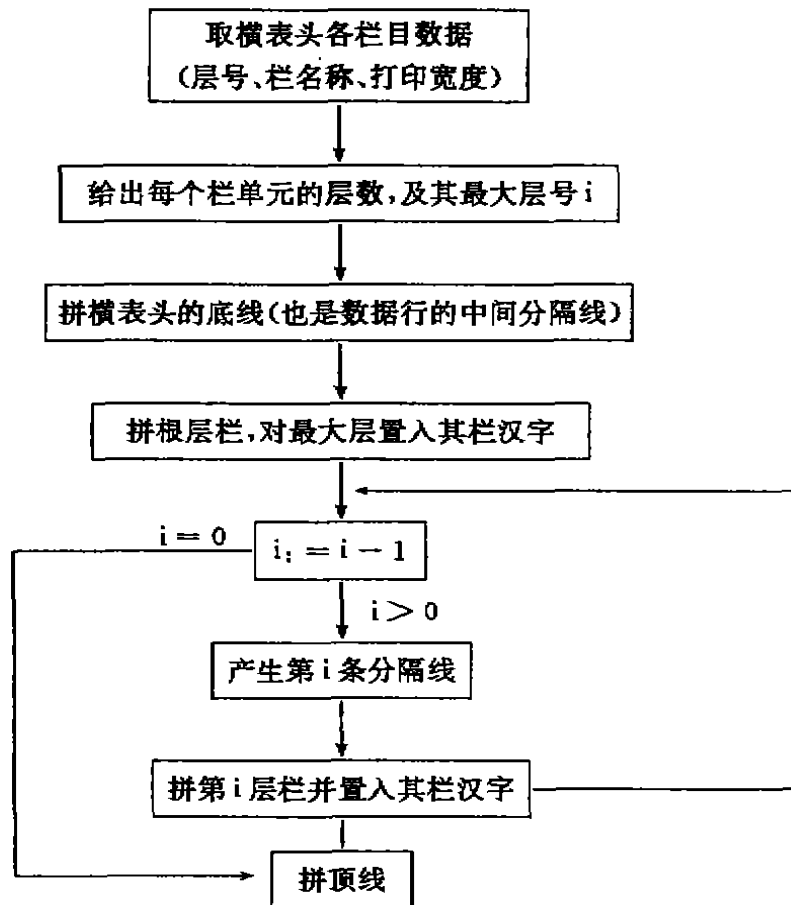
(2) 栏汉字的置入

我们将某一栏的汉字名称置入后的效果示意如下：



3. 横表头的生成流图

这里，横表头是由底往上构造的，其生成流图如下：



另外，表格的标题、标题说明、日期、注释等，在制作表格时就已完成了其输入和各部分字型的定义，打印时设定好有关参数（起始页、每页行数、行距等），这样设计出来的打印

报表程序就具有很好的通用性了。

## 5 各阶段报表软件的主要设计方法

### 1. 第一代设计

(1) 表格的标题(各表)集中存放在一个库。

(2) 每个表的表头各占一个库(用字处理软件如 Wordstar、CCED 等事先制作好横表头的文本文件,然后转换为库文件)。

(3) 主库结构(包括纵表头和表的数据栏)事先造好。

(4) 表的数据来源和关系均放在源程序中反映。

(5) 打印控制命令放在源程序中。

性能:简单、效率高。

存在问题:增加新表、修改表格困难;表的数据来源或关系稍有变化,要改源程序。总的来说:维护性差,生命周期短。

### 2. 第二代设计

(1) 每个表格由两个库存放:描述库和主库。

(2) 主库结构是由描述库中的横表头定义参数自动生成。

(3) 横表头的打印格式(包括画线部分)自动生成。

(4) 数据关系由公式表达。

(5) 打印控制命令可新增或修改。

性能:可维护性好(由用户完成),生命周期长。

存在问题:输入显示不够直观。

### 3. 第三代设计

(1) 用第二代的设计方法初描表格,将表格直接画在屏幕上。

(2) 在屏幕上加以调整、细描,或直接在屏幕上画表(借助鼠标)。

(3) 由系统自动分析、识别,存放表格。

(4) 表格在屏幕上的显示,完全按照原始表格形式,直观反映。

以上的前三条是指制作表格方面的改进,至于其他部分(如描述数据关系等)还是保留第二代的设计。

性能:达到较高的境界。

本文所描述的设计思想属第二代设计。