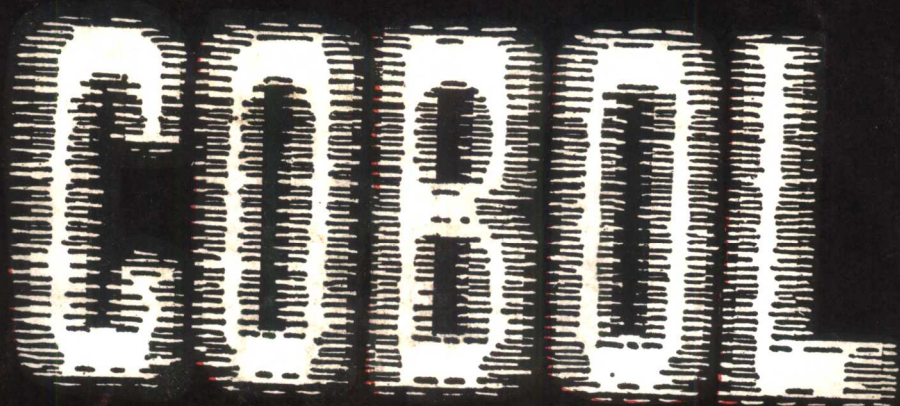


COBOL程序设计

石成孝 王书仁 编

The word "COBOL" is rendered in a bold, white, blocky font with a horizontal line pattern. It is centered within a black rectangular area, which is itself enclosed by a thick red border. The border consists of multiple concentric lines, creating a frame effect. The overall design is minimalist and high-contrast.

4-4-4

校

73-874 73-874
25-619 11-619

653697 716

阅覽已

COBOL 程 序 设 计

石成孝 王书仁 编

中 国 铁 道 出 版 社

1986年·北京

内 容 简 介

本书以PDP-11COBOL文本为依据,介绍了COBOL语言的语法功能及程序设计方法。

全书共十二章,由三部分内容组成。第一部分介绍了COBOL语言的基本概念及程序设计的基本方法;第二部分介绍了COBOL语言的语法功能及程序设计方法;第三部分介绍了结构程序设计的基本概念及应用COBOL语言进行结构程序设计的方法。

本书的特点是举例较多、偏重实用,并且采用了国家标准的COBOL语言名词术语。

本书可供从事计算机数据处理的技术人员及大专院校有关专业师生学习参考。

COBOL程序设计

石成孝 王书仁 编

中国铁道出版社出版

新华书店北京发行所发行

各地新华书店经售

中国铁道出版社印刷厂印

开本: 787×1092毫米★ 印张: 20 字数: 496 千

1986年2月 第1版 第1次印刷

印数: 0001—6,000册 定价: 3.55元

前 言

近年来,我国计算机的应用已不再局限于科学和工程计算领域,开始向着数据处理这个最广阔的应用领域发展。COBOL语言是一种面向数据处理的高级语言,适于处理具有数据量大而运算简单的一类问题,因此得到了广泛的应用。为了满足这种日益发展的需要,我们编写了本书奉献给各行各业从事数据处理业务的读者。

本书在总结我们运用PDP-11/70计算机经验的基础上,并参考了一些国内外有关资料编写的。内容共分三部分。第一章为第一部分,主要介绍了COBOL语言的基本概念及程序设计的基本方法。具有程序设计经验的读者,阅读本书时可以跳过这部分。

第二部分由第二至十一章组成,内容包括:COBOL语言四个部和各功能处理模块的语法功能以及程序设计方法。其依据的语言标准是PDP-11COBOL文本——美国国家标准(ANS COBOL-1974)的一个较大子集。这部分内容不仅适用于在PDP-11计算机上工作的读者,对安装有符合美国国家标准COBOL编译程序的计算机用户,同样具有参考价值(PDP-11COBOL文本与ANS COBOL-1974标准的比较,见附录3)。

第十二章为第三部分,主要介绍了结构程序设计及利用COBOL语言实现结构程序设计等问题。这部分内容无论对初学程序设计还是具有一定程序设计实践的读者都很重要。

本书承蒙全国计算机程序设计语言分技术委员会主任委员吴庆宝审阅,并提出了许多宝贵意见,尤其是她提供的COBOL语言名词术语的国家标准(见附录1),使得本书有幸首先采用国家标准的COBOL术语来编写。对此我们深表谢意。

鉴于我们水平有限,错误和不安之处在所难免,敬请读者批评指正。

编 者

1984年3月于北京

目 录

第一章 概 述	1
第一节 程序实例.....	1
第二节 基本概念.....	4
第三节 COBOL程序的四个部.....	5
第四节 COBOL程序结构.....	13
第五节 COBOL程序的编码规则.....	16
第六节 COBOL语言格式.....	17
第二章 标识部与环境部	20
第一节 标识部.....	20
第二节 环境部结构.....	21
第三节 配置节.....	21
第四节 输入/输出节.....	23
第三章 数 据 部	30
第一节 数据结构与数据属性.....	30
第二节 数据部结构.....	31
第三节 文件描述款.....	33
第四节 数据描述款.....	38
第五节 数据名的限定.....	59
第四章 过 程 部	62
第一节 过程部结构.....	62
第二节 COBOL语句的分类.....	63
第三节 MOVE语句.....	65
第四节 算术运算语句.....	70
第五节 条件语句.....	79
第六节 过程控制语句.....	94
第七节 少量数据的输入/输出语句.....	107
第八节 COPY语句.....	111
第五章 表 处 理	115
第一节 表的概念.....	115

第二节 一维表	116
第三节 多维表	123
第四节 元素数目可变的表	126
第五节 位标与SET语句	127
第六节 SEARCH语句	131
第六章 字符处理	137
第一节 STRING语句	137
第二节 UNSTRING语句	143
第三节 INSPECT语句	154
第七章 报表打印	163
第一节 报表格式及报表输出的一般流程	163
第二节 报表打印的程序控制	165
第三节 报表打印的格式控制	172
第八章 磁带和磁盘文件处理	178
第一节 磁带文件的处理方法	178
第二节 磁盘的物理特性	188
第三节 磁盘文件的组织方式和存取方式	189
第四节 磁盘顺序文件的处理方法	191
第五节 索引文件的处理方法	196
第六节 相对文件的处理方法	212
第七节 I/O操作的出错检测与处理	221
第九章 子程序	227
第一节 调用程序	227
第二节 被调用程序	228
第三节 被调用程序的出口	230
第四节 程序间通信示例	231
第十章 排序	233
第一节 排序键	233
第二节 排序顺序	234
第三节 标准排序方法	235
第四节 PDP-11 COBOL的排序方法	242
第十一章 程序覆盖	249
第一节 目标程序的分段和覆盖	249
第二节 段号	250

第三节	SEGMENT-LIMIT 子句	251
第四节	覆盖程序设计	252
第十二章	结构程序设计	253
第一节	软件的质量标准	253
第二节	什么是结构程序设计	255
第三节	自顶向下程序设计	255
第四节	基本程序结构	264
第五节	逐步求精	276
第六节	程序风格	283
附录一	COBOL语言国家标准名词术语汉英对照表	287
附录二	COBOL语言保留字	297
附录三	COBOL语言元素对照表	301

第一章

概述

COBOL (Common Business-Oriented Language) 是一种通用商业语言, 它是针对数据处理问题而设计的, 是目前世界上使用最多的几种高级程序设计语言之一。据统计, 世界上各种类型的通用计算机中,

大多数都配备COBOL语言编译系统。之所以会出现这种情况, 一方面是数据处理在计算机的应用中占有重要位置; 另一方面是COBOL语言本身所具有的特点。归纳起来, COBOL语言具有以下几方面特点:

(1) COBOL语言是世界上标准化最早的程序设计语言。这种标准化极大地推动了它在商业、企业管理、行政管理等领域的应用。

(2) COBOL语言采用了英语自然语言的书写风格, 用它写的程序类似于一篇英语文章。这样, 对接触数据处理工作的行政人员、财务人员、公司职员等, 只要熟悉英文, 就很容易掌握。

(3) COBOL语言是首先将数据的说明单独作为程序的一个部分(称为数据部)的高级程序设计语言, 并且将数据结构的描述与处理过程(称为过程部)的设计分开, 使程序结构更加清晰, 事后也易于维护。COBOL语言引入了文件的概念, 它能详细、准确地描述数据处理中常见的层次结构, 而这些都是FORTRAN和ALGOL-60语言所无法比拟的。

(4) COBOL语言中揉合了许多常用的数据处理方法, 如数据编辑、制表、排序和检索等。这样就减轻了程序员的工作量, 也减少了程序设计的难度。

(5) 用COBOL语言设计出来的程序有很强的自我说明能力, 有较高的模块化结构, 它便于程序间的交换和程序的维护。

在介绍 COBOL 语言的基本概念及 COBOL 程序设计的基本方法之前, 我们先给出一个程序实例(见程序 1-1)。

第一节 程序实例

通过这个程序实例, 读者可以了解到

COBOL语言的概貌及使用COBOL语言编写程序的基本方法, 作为全面了解COBOL语言及深入学习COBOL程序设计的入门。

程序名: DEPOSIT

```
IDENTIFICATION DIVISION.
PROGRAM-ID. DEPOSIT.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER. XYZ-1.
OBJECT-COMPUTER. XYZ-1.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
```

SELECT CARD-FILE

ASSIGN TO CARD-READER.

SELECT REPORT-FILE

ASSIGN TO PRINTER.

DATA DIVISION.

FILE SECTION.

FD CARD-FILE,

LABEL RECORDS ARE OMITTED,

RECORD CONTAINS 80 CHARACTERS,

DATA RECORD IS CARD-RECORD.

01 CARD-RECORD.

02 C-ACCOUNT PICTURE 999999.

02 C-NAME PICTURE X(18).

02 C-OLD-DEPOSIT PICTURE 9(6)V99.

02 C-NEW-DEPOSIT PICTURE 9(6)V99.

02 C-EXPENDITURE PICTURE 9(6)V99.

02 FILLER PICTURE X(32).

FD REPORT-FILE,

LABEL RECORDS ARE OMITTED,

RECORD CONTAINS 132 CHARACTERS,

DATA RECORD IS REPORT-RECORD.

01 REPORT-RECORD PICTURE X(132).

WORKING-STORAGE SECTION.

77 TEMPORARY PICTURE 9(6)V99.

01 HEADING-LINE.

02	FILLER	PICTURE	X(75)	VALUE	"ACCOUNT NAM
	"E	OLD-DEPOSIT	NEW-DEPOSIT	EXPENDITURE	BALANCE".

02	FILLER	PICTURE	X(57)	VALUE	SPACES.
----	--------	---------	-------	-------	---------

01 DETAIL-LINE.

02	FILLER	PICTURE	XX	VALUE	SPACES.
----	--------	---------	----	-------	---------

02	D-ACCOUNT	PICTURE	9(6).
----	-----------	---------	-------

02	FILLER	PICTURE	XX	VALUE	SPACES.
----	--------	---------	----	-------	---------

02	D-NAME	PICTURE	X(18).
----	--------	---------	--------

02	FILLER	PICTURE	XXX	VALUE	SPACES.
----	--------	---------	-----	-------	---------

02	D-OLD-DEPOSIT	PICTURE	Z(6).99.
----	---------------	---------	----------

02	FILLER	PICTURE	XXX	VALUE	SPACES.
----	--------	---------	-----	-------	---------

02	D-NEW-DEPOSIT	PICTURE	Z(6).99.
----	---------------	---------	----------

02	FILLER	PICTURE	XXX	VALUE	SPACES.
----	--------	---------	-----	-------	---------

02	D-EXPENDITURE	PICTURE	Z(6).99.
----	---------------	---------	----------

02	FILLER	PICTURE	XXX	VALUE	SPACES.
----	--------	---------	-----	-------	---------

02	D-BALANCE	PICTURE	Z(6).99.
----	-----------	---------	----------

02	FILLER	PICTURE	X(56)	VALUE	SPACES.
----	--------	---------	-------	-------	---------

PROCEDURE DIVISION.

细目行。这种过程一直进行到把所有的卡片记录处理完为止。银行储蓄程序流程图如图 1-2 所示。

第二节 基本概念

程序 1-1 是用 COBOL 语言编写的银行储蓄程序。本节仅就这个程序说明 COBOL 语言的基本概念。至于这个程序的设计方法将在以下各节中介绍。

一、字符集

构成一个 COBOL 程序的基本单位是字符，COBOL 语言可以使用的字符集合叫作 COBOL 字符集。ANS COBOL-1974 字符集由下列 51 个字符组成：

- 字母字符：A~Z；
- 数字字符：0~9；
- 专用字符：+（加号）、-（减号或连字符）、*（乘号）、/（除号或斜线）、=（等号）、\$（美元符号）、,（逗号）、;（分号）、.（句号或小数点）、"（引号）、(（左括号）、)（右括号）、>（大于号）、<（小于号）和空格。

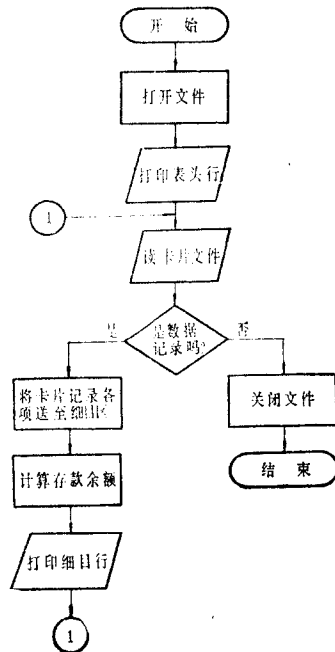


图 1-2 银行储蓄程序流程图

二、分隔符

和自然语言一样，在 COBOL 语言使用的 51 个字符中，有些字符起着标点符号的作用。如空格、逗号、分号和句号等，COBOL 语言将这类符号称作分隔符。空格是使用最多的一个分隔符。在 COBOL 程序中需要使用空格作为分隔符的地方，多个空格和一个空格的作用是等价的。句号不能单独作为分隔符，只有在它们的后边跟有空格时才能成为分隔符。这些分隔符的用法要遵守一定的语法规则。

三、字符串

用分隔符隔开的一个字符或多个字符叫作字符串。字符串可以用字母、数字和专用字符组成。例如，程序 1-1 中的 IDENTIFICATION、DIVISION、SOURCE-COMPUTER、CARD-FILE、C-ACCOUNT、X(32)、DEPOSIT、9(6)V99、ADD、80 等均是字符串。

* 各国在使用 COBOL 语言时，\$ 可以改为本国的货币符号。

四、COBOL 字

COBOL程序中包含很多本身指定要用的词以及程序员自己定义的词,这类词叫做COBOL字(或简称为字)。笼统地讲,COBOL字也是字符串,不过这类字符串在构造方面要遵守一定的规则。

一个COBOL字不能超过30个字符,而且只能用字母和数字两类字符以及连字符“-”来构造。但是,连字符不能出现在COBOL字的开头和末尾。其他专用字符,包括空格均不得在COBOL字中出现。例如,前面列举的字符串中,IDENTIFICATION、DIVISION、SOURCE-COMPUTER、CARD-FILE、C-ACCOUNT、DEPOSIT和ADD等均是合法的COBOL字。

COBOL字分两类:一类叫作保留字,另一类叫作用户定义字。

(1) 保留字:由一定数量的英文单字或缩写字组成的词组,每个词组有固定的含义和用法,程序员只能按COBOL语言规定的格式使用。

(2) 用户定义字:除保留字以外的COBOL字均为用户定义字。用户定义字可以由程序员来定义。在程序中,应该使用用户定义字的地方,一定不要误用保留字。例如,CARD-FILE是一个合法的用户定义字,但是如果用FILE代替CARD-FILE,那么FILE就是一个非法的用户定义字,因为它是一个保留字,不允许作用户定义字使用。

用户定义字一般用来定义程序名、节名、段名、文件名、记录名、数据名、层号等。有关这些术语的含义,将在以后各章中介绍。需要指出的是,除节名、段名和层号以外,所有的用户定义字中必须至少含有一个字母。

第三节 COBOL程序的四个部

COBOL程序的四个部是一个有机的整体,但在职能方面各有分工。标识部提供程序的标识信息,例如程序名。环境部规定程序执行过程中所需要的设备。数据部描述程序处理的文件、记录和数据项的属性。过程部描述计算机应该遵循的处理过程。

一、标识部

标识部是COBOL程序中的第一个也是最简单的一个部,它的作用是规定程序名和识别信息。标识部的部首为:

IDENTIFICATION DIVISION.

即由保留字IDENTIFICATION和DIVISION后跟一个英文句号(简称句号)构成。

标识部由若干段组成,每段用一段首开头。标识部只有一段是必写的,它的段首为:

PROGRAM-ID.

即由保留字PROGRAM-ID后跟一个句号构成。

PROGRAM-ID(程序名)段中有一个程序名款,它是一个用户定义字,用来代表程序名。在程序1—1中,程序名DEPOSIT就是程序名款的具体内容。一般说来,所谓款就是程序特性的一组描述,它必须用句号结尾。款可以是一个字,也可以由几个字组成。但是需要注意,在使用PDP-11计算机时,程序名不许超过9个字符,也不许出现连字符,并且只有

前6个字符有效。

下面给出标识部的两种编码方法。在方法1中，程序名款和段首写在同一行，但段首的句号后边必须至少留一个空格。

方法1 一般形式

```
IDENTIFICATION DIVISION.  
PROGRAM-ID.    DEPOSIT.
```

方法2 替换形式

```
IDENTIFICATION DIVISION.  
PROGRAM-ID.  
    DEPOSIT.
```

方法2是标识部编码的另一种方法，程序名款写在段首的下一行，程序名前面空几格完全是为了便于阅读。

通过这个例子，我们可以看到COBOL程序的编码方法可以灵活多样。

二、环 境 部

紧接标识部后面的是环境部，它指出程序运行时所使用的计算机设备。环境部的部首为：
ENVIRONMENT DIVISION.

即由保留字ENVIRONMENT和DIVISION后跟一个句号构成。

环境部分两节，节中含有段，而段又由款组成（请注意，前面讲的标识部不设节只设段）。环境部的两个节首为：

CONFIGURATION SECTION.（配置节）

INPUT-OUTPUT SECTION.（输入/输出节）

即由保留字CONFIGURATION、INPUT-OUTPUT、SECTION和句号组成。

1. 配置节

配置节规定程序在编译和执行时计算机的配置情况。本节由SOURCE-COMPUTER（源计算机）、OBJECT-COMPUTER（目标计算机）和SPECIAL-NAMES（专用名）三段组成。SOURCE-COMPUTER段内只有一款，用于指出编译程序的计算机名。OBJECT-COMPUTER段内也只有一款，用于指出执行目标程序的计算机名。SPECIAL-NAMES段是一个任选段。COBOL语言之所以规定源计算机和目标计算机，是为了使程序的编译和执行在两台不同的计算机上进行。然而，在大多数情况下，源计算机和目标计算机是同一台计算机。这样一来，两段中的计算机名款则应具有相同的内容。

SOURCE-COMPUTER和OBJECT-COMPUTER段中的计算机名是计算机厂家规定的，属于系统名，也是一种COBOL保留字。在程序1—1中的源计算机名和目标计算机名均为XYZ/-1。

2. 输入/输出节

输入/输出节是一个任选节，如果程序中不使用文件，本节可以省略。

输入/输出节用于规定程序所使用的输入/输出设备。本节由FILE-CONTROL（文件控制）和I-O-CONTROL（输入/输出控制）两段组成。为了区别不同的文件起见，COBOL语言要求给程序使用的每一个文件起一个唯一确定的名字，这样的名字叫做文件

名。文件名是一个用户定义字。在程序 1--1 中，我们用 CARD-FILE 作为卡片输入文件的文件名，REPORT-FILE 作为打印输出文件的文件名。在文件控制段中，为了区别各种不同的文件名，规定一个 SELECT（选择）款。该款的作用是把文件同输入/输出设备联系起来。

SELECT 款由 SELECT（选择）和 ASSIGN（指派）两个子句组成。SELECT 子句指出程序使用的一个文件，文件名跟在 SELECT 的后边。ASSIGN 子句指出文件使用的设备，设备名跟在 ASSIGN TO 的后边。

ASSIGN 子句使用的设备名与具体计算机系统有关，因此一般称作实现名。例如，许多厂家将 CARD-READER 用作卡片输入机的名字，然而在 PDP-11 计算机上，则用字符串“CR:”代表卡片输入机。同样，PDP-11 计算机不用 PRINTER 作为打印机的名字，而用“LP:”代表打印机。因此，ASSIGN TO 后面的实现名必须按规定正确拼写。

在程序 1--1 中，程序使用两个文件，所以有两个 SELECT 款。SELECT 款中的两个子句可以写在同一行中，也可以分别写在两行中。但是，无论怎样书写，句号必须出现在款的末尾。

三、数 据 部

程序处理数据过程一般是先从输入文件读入数据开始，然后处理数据，并把结果作为输出文件的记录输出到外部设备上去。在数据处理过程中，有时还需要一些存放中间结果的数据区。这些数据区的名字、长度、内部组织及其它一些属性，在 COBOL 语言中由数据部加以描述。数据部的部首为：

DATA DIVISION.

即由保留字 DATA 和 DIVISION 后跟一个句号构成。

数据部分三节，节中不分段，直接由款构成。数据部的三个节首为：

FILE SECTION. (文件节)

WORKING-STORAGE SECTION. (工作存储节)

LINKAGE SECTION. (连接节)

即由保留字 FILE、WORKING-STORAGE、LINKAGE、SECTION 和句号组成。

1. 文件节

在环境部中，每一个文件都由一个 SELECT 子句给出了文件名，程序中使用几个文件，环境部就有几个 SELECT 款。与此相对应，环境部每出现一个 SELECT 款，数据部就应有一个文件描述款，而且都应使用相同的文件名。

文件描述款由保留字 FD（文件描述）后跟文件名和若干子句构成。其中有三个重要子句。第一个子句为：

LABEL RECORD IS OMITTED

该子句称作 LABEL RECORD（标号记录）子句，全部由保留字组成。这个子句的含义为“标号记录是省略的”。所谓标号记录是一种专用的描述型记录，供计算机软件系统识别和管理文件使用。

第二个子句，对于文件 CARD-FILE 为：

RECORD CONTAINS 80 CHARACTERS

对于文件REPORT-FILE为:

RECORD CONTAINS 132 CHARACTERS

它们称作 RECORD CONTAINS (记录包含) 子句, 用于指出文件记录是由多少字符组成的。

程序处理的每一种记录必须有一个名字, 这就是记录名, 它是用户定义字。记录名是在文件描述款的最后一个子句中给出的, 它由DATA RECORD IS 三个保留字后跟记录名组成。对于文件CARD-FILE, 该子句为:

DATA RECORD IS CARD-RECORD

对于文件REPORT-FILE, 其文件描述款也有一个类似的子句。

紧接文件描述款的是一组数据描述款, 用来描述文件描述款中规定的文件所属记录的各个数据项。每个数据描述款均用一个编号开头, 这种编号称作层号。层号由阿拉伯数字01~49组成。

与某一特定记录有关的所有数据描述款叫作记录描述款。其中第一个数据描述款的层号是01 (或者1), 接着是记录名和句号。记录名必须和前边文件描述款的 DATA RECORD 子句中出现的记录名相同。例如, 对于文件 CARD-FILE, 其文件描述款的后边必须跟着如下的01层数据描述款:

01 CARD-RECORD.

数据部采用各种不同层号的描述款来描述数据, 完全是为了反映数据的层次结构关系。最高层是文件, 采用FD层的文件描述款来说明; 次高层是文件中的记录, 采用01层的数据描述款来说明; 再次层是记录的数据项, 采用02层的数据描述款来说明; 等等。

程序执行过程中, 往往要使用输入/输出记录当中的一些数据项, 这些数据项必须一一命名。这样编译程序就能根据数据项的属性分配内存, 程序就可以使用数据名访问内存中该数据名代表的存储区。数据项的名字 (即数据名) 是用户定义字, 在同一程序中数据的名字应各不相同。

每个02层描述款必须含有一个数据名或保留字 FILLER。在程序处理过程中, 不被使用的数据项可以不必命名, 一律用 FILLER 来代表。这样的数据项, 叫作充填项。

为了说明各数据项的数据类型和长度, 每个02层描述款使用一个 PICTURE (形象) 子句。该子句由保留字 PICTURE 后跟一个字符串组成。这样的字符串叫形象字符串, 简称为数据项的形象。PICTURE 子句既描述了数据项的类型, 又描述了数据项的长度。对数值类型的数据项而言, 形象字符串由一串 9 组成, 每一个 9 表示数据项中的一个数字位。例如, 数据项 C-ACCOUNT 的形象为 999999, 则表明它是数值型数据项, 长度为 6 位十进制数。为了简化形象字符串的编码, 上述形象字符串可以写成 9(6)。其中 9 后面括号中的数字代表 9 的重复出现次数, 称作重复因子。

对于字符型数据项, 其形象使用一串 X, 或带重复因子的 X 来表示, 每一个 X 表示数据项中的一个字符位置。例如, 数据项 C-NAME 的形象为 X(18), 表示它是 18 个字符长的数据项。

此外, 卡片记录的其余三个数据项 C-OLD-DEPOSIT、C-NEW-DEPOSIT 和 C-CHECK 均具有相同的形象字符串 9(6)V99, 即除 9 以外, 还有一个符号 “V”。因为对于数值型的数据项, 形象中一律不准使用小数点 “.”。如果数据项的内容不是整数, 那么形象中就用字母 “V” 划分出整数部分和小数部分。“V” 叫作虚拟小数点。在定义数据项

的长度时，不能把它计算在内。例如，形象 9(6)V99，表明数据项的长度为 8，整数部分有 6 位数字，小数部分有两位数字。

记录中所有的字符位置必须用 PICTURE 子句描述到。例如，一张卡片有 80 列，各形象子句所描述的字符总数也必须是 80 个字符。

对于 REPORT-FILE 文件，FD 层文件描述款之后只用了一个 01 层数据描述款，并通过 PICTURE 子句说明打印记录是一个 132 个字符长度的数据项。至于打印记录 REPORT-RECORD 的详细数据描述，则在工作存贮节中进行。

2. 工作存贮节

一个 CDBOL 程序，除了在文件节定义输入/输出记录区以外，一般还要定义存放中间结果和常数的数据区。存放中间结果和常数的数据区应在 WORKING-STORAGE (工作存贮节) 中描述。

工作存贮节在数据部文件节之后，它以节首

WORKING-STORAGE SECTION.

开头。节首之后有层号为 77 的数据描述款，它所描述的数据不属于任何记录，故称作独立项。程序 1—1 中的数据项 TEMPORARY 就是这类独立项。

与文件节类似，工作存贮节各数据项的类型与长度也由 PICTURE 子句规定。

工作存贮节不仅用来描述独立项，同时也可以象文件节那样描述记录或组项。例如在程序 1—1 中，我们对打印记录和文件节作了如下描述：

01 REPORT-RECORD PICTURE X(132).

根据这一描述，如何打印出如图 1—1 所示的既有表头又有细目的报表呢？这需要在工作存贮节中加以详细描述。

工作存贮节第一个 01 层数据描述款中的数据名为：

HEADING-LINE.

它是表头记录的名字。隶属于该记录的两个 02 层数据描述款描述这行表头的具体内容。在这些数据描述款中，除了 PICTURE 子句以外，各包含一个 VALUE (值) 子句。VALUE 是保留字，它的后边可以是用户定义的字符串，也可以是某些保留字。该子句的作用是为相应的数据项赋值。

第一个 02 层数据描述款的编码为：

02 FILLER PICTURE X(75) VALUE "ACCOUWT NAM
- "E OLD-DEPOSIT NEW-DEPOSIT CHECK BALANCE".

它描述表头的文字部分。文字的具体内容用 VALUE 子句规定。保留字 VALUE 后边的字符串和下一行句号前边的字符串合在一起构成一个 75 个字符长度的字值常量，与格式编排表的表头内容相符。

字值常量应与数据项的类型相一致。字值常量分字符型和数值型两种。上例就是字符字值常量，至于它的编码规则将在本章第五节中介绍。此外，用十、一、·以及 0~9 构成表示数值的字符串叫作数值字值常量。例如，80、132、-35.6 均是合法的数值字值常量。

第二个 02 层数据描述款说明文字右边还有 57 个字符位置是空白。为了简化编码起见，对于要求其内容全部为空格的数据项，不必采用字符字值常量的形式，而是用保留字 SPACES 来赋值。象 SPACES 一类用于代表字值常量的保留字叫作象征常量。

由于表示记录的内容在程序执行过程中始终不变，因此除 01 层使用一个用户定义的数据

名以外,不被程序直接引用的两个初等项均用保留字定义成充填项。这两个初等项的长度之和为132,正好与文件节定义的打印记录长度一致。

工作存储节第二个01层数据描述款中的数据名为:

DETAIL-LINE.

在这个记录的02层中,数据描述款用来描述细目行的打印格式。

细目行左边是2个字符的空白字段,定义成充填项,用 SPACES 赋初值。第二个02层数据描述款用于描述帐号字段,数据项命名为D-ACCOUNT。因为该项的内容来自卡片的帐号字段,在程序执行过程中不断改变,所以不应用VALUE子句指定初值。接下去的诸02层数据描述款分别描述充填项、姓名数据项(D-NAME)、原存款额数据项(D-OLD-DEPOSIT)…,最后是细目行右边一段空白的充填项。

与卡片类似,打印行的形象字符串对于字符类型的项使用×。例如,细目行第一个充填字段为2个字符,它的形象为××。类似地,D-NAME的形象为×(18)。

对于数值类型的项,其形象可以用9来描述。这时打印的结果全部是数字。例如,对于图象9(6),如果要打印的数据为123456,那么输出的结果是123456;如果要打印的数据为008253,那么输出的结果也是008253。

为了使打印的报表清晰易读,可以在数据项的形象中使用某些特殊字符,对输出结果进行编辑。所谓编辑,是指加工输出结果的过程,而加工规则则由编辑字符决定。形象中带有编辑字符的数据项,称之为编辑类型数据项。

例如,细目行记录的D-OLD-DEPOSIT等四项的形象均为Z(6).99,其中“Z”和“.”就是编辑字符。若卡片输入的原存款额为001024.00,经过编辑输出的结果则为1024.00,即在有效数字前面形成两个空格并在整数和小数之间出现了一个实际小数点。

编辑字符“Z”叫作零抑制符,其作用是把有效数字左边的零用空格代替。编辑字符“.”叫作插入小数点。对于非整数数据,要想在输出数据中打印出小数点,我们就应将实际小数点“.”放在输出数据项的形象中。

编辑字符的种类繁多,原因是COBOL语言是面向事务处理的计算机语言,要求这种语言具有丰富的打印报表格式的能力。适当组合使用编辑字符可以得到清晰、美观的输出报表。关于编辑字符的种类和使用规则,在有关章节详述。值得指出的是,编辑字符只能出现在输出数据项中,卡片输入项和参与算术运算项的形象中决不能出现编辑字符。输入数据项和参与算术运算的数据项,其形象总是非编辑类型的。

在计算编辑类型数据项的长度时,我们应该把所有的字符位置都考虑进去。例如,Z(9).99表示数据项占12个字符位置。基于这种算法,我们可以验证02层的细目行记录的长度为132,它与文件节的报表记录长度一致。此外,COBOL语言的语法规则规定,编辑字符“.”不应出现在形象字符串的末尾。这一规定保证了形象字符串中的小数点不会起到款末尾句号的作用。

3. 连接节

连接节用于描述子程序同主程序之间交换的数据。或者说,连接节是在被调用的程序中用于描述调用与被调用程序之间传递的参数,它的结构形式和工作存储节相同,但是在连接节中不能给数据名赋初值。有关连接节的使用,详见第九章。