



中华人民共和国国家标准

GB/T 4092.12—1992

程序设计语言 COBOL 程序间通信模块

**Programming language COBOL
Inter-program communication module**

1992-08-04 发布

1993-05-01 实施

国家技术监督局 发布

程序设计语言 COBOL
程序间通信模块

GB/T 4092.12—1992

代替 GB 4092.12—1983

Programming language COBOL
Inter-program communication module

1 引言

1.1 功能

程序间通信模块提供了一个程序与另一个或多个程序通信的功能。这种通信通过下面两个方面提供：**(a)**在一个运行单位内把控制从一个程序转到另一个程序的能力。**(b)**在程序间传送参数的能力使某些数据值对被调用程序可用。

1.2 级别特征

1 级程序间通信提供了把控制转到一个或多个名字在编译时是已知的程序的功能以及在这些程序之间的数据共享的功能。

2 级程序间通信提供了把控制转到一个或多个名字在编译时是未知的程序的功能，以及确定控制将转到的程序的目标存储空间的可用的能力。2 级程序间通信也提供了外部属性、全程名和源程序的嵌套。

1.3 语言概念

1.3.1 嵌套的源程序

一个 COBOL 源程序是一个语法上正确的 COBOL 语句集。一个 COBOL 源程序可以包含其他 COBOL 源程序，而这些被包含的程序可以引用包含它们的程序的某些资源。

当一个程序 B 包含在另一个程序 A 中，它可以是被直接包含或间接包含。如果没有被包含于 A 又包含 B 的程序，则程序 B 被直接包含于程序 A 中。如果存在一个被包含于 A 中又包含 B 的程序，则程序 B 被间接包含于程序 A 中。

1.3.2 文卷连接区

一个文卷连接区是包含有关一个文卷的信息，且用作文卷名与物理文卷之间以及文卷名与它的相关的记录区之间的联结的存储区域。

1.3.3 全程名和局部名

一个数据名命名一个数据项。一个文卷名命名一个文卷连接区。这些名分为全程的或局部的。

一个全程名可以用来引用与之相关的目标，要么从说明该全程名的程序中引用，要么从包含于该程序且该全程名的程序中的任何其他程序中引用。

而一个局部名只可从说明该局部名的程序中引用与之相关的目标。某些名总是全程的，某些名总是局部的，而某些名既可为全程的又可为局部的，这依赖于这些名在程序中的说明。

对记录名,如果在说明记录名的记录描述款中指出了 **GLOBAL** 子句,或者,在文卷节的记录描述款的情况下,如果在与记录描述款有关的文卷名的文卷描述款中指出了 **GLOBAL** 子句,则该记录名是全程的。对数据名,如果在说明该数据名的数据描述款中指出了 **GLOBAL** 子句,或者在该数据描述款下属的另一个描述款中指出了 **GLOBAL** 子句,则该数据名是全程的。对在一个数据描述款中说明的条件名,如果该描述款下属于一个指出了 **GLOBAL** 子句的另一个描述款时,则它是全程的。但是某些特定规则有时禁止对一些特定的数据描述款、文卷描述款和记录描述款指定 **GLOBAL** 子句。

对文卷名,如果在该文卷名的文卷描述款中指出了 **GLOBAL** 子句,则它是全程的。

如果在一个数据描述款中说明的数据名、文卷名或条件名不是全程的,则该名是局部的。

1.3.4 外部目标和内部目标

可存取的数据项通常要求存储数据以特定的表示,文卷连接区通常要求存储某些与文卷有关的特定信息。与数据项或文卷连接区有关的存储对说明目标的程序来说可以是 **EXTERNAL** 内部的。

对数据项或文卷连接区,如果与那个目标有关的存储是与运行单位相关而不是与运行单位中的某个特定程序相关,则它是外部的。一个外部目标可被描述该目标的运行单位中的任何程序所引用。对外部目标的引用总是对同一个目标的引用,在不同程序中使用各自的目标描述。在一个运行单位中,外部目标只有一种表示。

一个目标,如果与它有关的存储只与描述它的程序相关联,则它是内部的。 **INTERNAL** 内部目标

也可有局部名。

工作存储节中描述的数据记录通过它的数据描述款的 **EXTERNAL** 子句给定外部属性。任何由从属于描述外部记录的数据描述款描述的数据项也获得外部属性。如果一个记录或数据项不具有外部属性,则它是描述它的程序的内部数据的一部分。

如果一个文卷连接区不具有外部属性,则它对描述相关文卷名的程序来说是内部的。

描述为隶属于不包含 **EXTERNAL** 子句的文卷描述款或排序-合并文卷描述款的数据记录,以及描述为隶属于这些记录的数据描述款的任何数据项,对描述该文卷名的程序而言总是内部的。

在一个程序的连接节、通信节和报表节中描述的数据记录,下属数据项以及各种有关的控制信息被认为对描述该数据的程序来说总是内部的。对连接节中描述的数据有特殊的考虑,在那里建立了所描述的数据与对其他程序可存取的其他数据项之间的联系。

1.3.5 公用程序和初始程序

组成一个运行单位的所有程序可以不具有或具有一个或多个下列属性:公用的和初始的。

公用程序是这样的程序,尽管它直接包含在另一个程序中,但它可被直接或间接包含在另一个程序中的任何程序所调用。公用属性通过程序的标识部中指定 **COMMON** 子句获得。**COMMON** 子句简化了将被包含在同一程序中的所有程序使用的子程序的书写。

初始程序是这样的程序,当它被调用时,它的程序状态被初始化。于是,一个初始程序一旦被调用则它的程序状态与在那个运行单位中第一次被调用时的状态一样。在初始化一个初始程序的处理中,那个程序的内部数据被初始化;因此描述中含有 **VALUE** 子句的程序的内部数据项初始化为所定义的值,但描述中不含 **VALUE** 子句的数据项初始化为一个无定义的值。与程序有关的内部文卷连接区的文卷不处于打开方式。程序中包含的所有 **PERFORM** 语句的控制机制被置为它们的初始状态。初始属性通过在程序的标识部中指定 **INITIAL** 子句获得。

1.3.6 共享数据

一个运行单位中的两个程序在下列情况下可以引用公用数据：

(1) 如果程序中描述了一外部数据记录，则可从该程序中引用该外部数据记录的数据内容。

(2) 如果一个程序包含在另一个程序中，那么这两个程序都可引用这样的数据，该数据在包含程序中或在直接或间接地包含该被包含程序的任何程序中具有全程属性。

(3) 从调用程序到被调用程序通过引用传递参数的机制建立公用数据项；被调用程序可以使用不同的标识符，去引用调用程序中的数据项。

1.3.7 共享文卷

在一个运行单位中的两个程序在下列情况下可以引用公用文卷连接区：

(1) 外部文卷连接区可由描述该文卷连接区的任何程序引用。

(2) 如果一个程序包含在另一个程序中，那么这两个程序都可通过在包含程序中或在任何直接或间接地包含该包含程序中引用有关的全程文卷名来引用公用文卷连接区。

1.3.8 名的作用域

当程序直接或间接包含在别的程序中时，各个程序可以使用相同的用户自定义字去命名对象，而与其他程序使用这些用户自定义字无关（见 GB/T 4092.1 中 6.4.2.2.1.1 用户自定义字）。当存在相同命名的对象时，程序对这种名字的引用，即使它是不同的用户自定义字类型，也是对那个程序描述的对象引用，而不是对在另一个程序中描述的具有相同名字的对象引用。

下列类型的用户自定义字只可由说明该用户自定义字的程序中的语句和描述款引用：

cd 名

段名

节名

下列类型的用户自定义字可由任何 COBOL 程序引用，假设编译系统支撑相应的库或其他系统并且所引用的实体对那个系统是已知的。

库名

正文名

当在通信节中定义了下列类型的用户自定义字时，它们只可由包含该通信节的程序中的语句和描述款引用：

条件名

数据名

记录名

当在配置节中定义了下列类型的名时，它们只可由包含配置节的那个程序或由该程序所包含的程序中的语句或描述款引用：

字母名

类别名

条件名

助忆名

符号字符

当上列条件不适用时，对说明和引用，特定的约定将应用于下列类型的用户自定义字：

条件名

数据名

文卷名

位标名

程序名

记录名

报表名

1.3.8.1 程序名的约定

一个程序的程序名在该程序的标识部的程序标识(**PROGRAM-ID**)段中说明。程序名只可由 **CALL** 语句和程序末端标题引用。分配给组成一个运行单位的程序的程序名不必唯一,但是,当一个运行单位的两个程序同名时,这两个程序中至少有一个直接或间接包含在另一个分别编译的程序中,而该分别编译的程序不包含这两个程序中的另一个。

下列规则规定了程序名的作用域:

- (1) 如果该程序名是不具备公用属性而又被直接包含在另一个程序中的程序的名字,则该程序名只可由那个包含该程序的程序中的语句所引用。
- (2) 如果该程序名是具备公用属性而又被直接包含在另一个程序中的程序的名字,则该程序名只可由那个包含该程序以及任何被直接或间接包含于其中的程序(但具有公用属性的程序及包含于其中的程序除外)中的语句所引用。
- (3) 如果一个程序名是一个分别编译的程序的名字,则该程序名可由运行单位中的任何其他程序(除去它直接或间接包含的程序)中的语句引用。

1.3.8.2 条件名、数据名、文卷名、记录名和报表名的约定

当条件名、数据名、文卷名、记录名和报表名在一个源程序中被说明时,这些名只可由该源程序引用。

由单个程序分配的作为条件名、数据名、文卷名、记录名和报表名的唯一性的管理要求在这说明的其他部分解释(见 GB/T 4092.1 中 6.4.2.2.1.1 用户自定义字)。

- 一个程序不能引用它所包含的任一程序中说明的条件名、数据名、文卷名、记录名和报表名。
- 一个全程名可在说明它的程序中或在直接或间接被包含于该程序的任何程序中引用。
- 当一个程序 **B** 被直接包含在另一个程序 **A** 中时,两个程序可使用同一个用户自定义字定义的一个条件名、数据名、文卷名、记录名或报表名。当在程序 **B** 中引用这样的重名时,使用下列规则决定所引用的对象:
 - (1) 用于决定所引用的对象的名集合由程序 **B** 中定义的所有名,在程序 **A** 及任何直接或间接包含 **A** 的程序中定义的全程名组成。使用这个名集合,采用通常的限定规则及其他使引用唯一的规则直至标识出一个或多个对象为止。
 - (2) 如果只标识出一个对象,则它就是引用的对象。
 - (3) 如果标识出的对象多于 1 个,则具有局部于程序 **B** 的名的对象不会超过 1 个。如果零个或一个对象有局部于程序 **B** 的名,则下列规则适用:
 - a. 如果名在程序 **B** 中说明,则程序 **B** 中的对象是引用的对象。
 - b. 否则,如果程序 **A** 包含在另一个程序中,则引用的对象是:
 - 1) 如果该名在程序 **A** 中说明,则为程序 **A** 中的对象。
 - 2) 如果该名在程序 **A** 中未说明而在包含程序 **A** 的程序中说明,则为包含程序中的对象。这个规则适用于更上一层的包含程序,直至找到一个唯一的有效的名为止。

1.3.8.3 位标名的约定

如果一个具有外部或全程属性之一或二者兼有的数据项包含一个使用位标的表,则该位标也相应地具有属性之一或二者。因此,该位标名的作用域与命名该表(表的位标由该位标名命名)的数据名的作用域相同,并且该数据名的名的作用域规则适用于该位标名。位标名不可以被限定。

2 嵌套源程序

2.1 一般描述

一个 **COBOL** 源程序是语法正确的 **COBOL** 语句的集合。一个 **COBOL** 源程序可以包含其他 **COBOL** 源程序,并且这些被包含的程序可以引用包含它们的程序的某些资源。

2.2 组织

除 **COPY** 和 **REPLACE** 语句及程序末端标题外,一个 **COBOL** 源程序的语句、描述款、段和节分为四个部,以下列次序排列:

标识部
环境部
数据部
过程部

一个 **COBOL** 源程序的末尾由程序末端标题(如果指定的话),或以无附加源程序行来标志。

2.3 结构

下面给出组成一个 **COBOL** 源程序的描述款和语句出现的一般格式和次序。通用术语标识部、环境部、数据部、过程部、源程序和程序末端标题分别表示 **COBOL** 程序的标识部、**COBOL** 程序的环境部、**COBOL** 程序的数据部、**COBOL** 程序的过程部、**COBOL** 源程序和 **COBOL** 程序的程序末端标题。

2.3.1 一般格式

标识部
[环境部]
[数据部]
[过程部]
[源程序]
[程序末端标题]

2.3.2 语法规则

(1) 程序末端标题在下列情况下必须存在:

- a. 该 **COBOL** 源程序包含一个或多个其他 **COBOL** 源程序,或
- b. 该 **COBOL** 源程序包含在另一个 **COBOL** 源程序中。

2.3.3 一般规则

(1) 程序中一个部的开始由适当的部首指出。一个部的结束由下列之一指出:

- a. 程序中后继部的部首。
- b. 指出另一个源程序开始的标识部首。
- c. 程序末端标题。
- d. 在其后无其他源程序行出现的那个物理位置。

(2) 被直接或间接包含在另一个程序中的 **COBOL** 源程序(被认为是一个单独的程序)的规格说明可以额外地引用在包含程序中定义的某些资源。

(3) 由编译包含在另一个程序中的一个程序得到的目标代码,在规格说明方面被认为是与由编译包含程序产生的目标代码不可分割的。

2.4 程序的初始状态

程序的初始状态是该程序在运行单位中第一次被调用时的状态。

2.4.1 程序的特性

(1) 程序中包含在工作存储节和通信节中的内部数据被初始化。如果在数据项描述中使用了

VALUE 子句,则该数据项初始化为所定义值。如果一个数据项无 VALUE 子句,则该数据项的初始值无定义。

- (2) 与该程序有关的具有内部文卷连接区的文卷不处于打开方式。
- (3) 该程序中所有 PERFORM 语句的控制机制被置为它们的初始状态。
- (4) 包含在该同一程序中由 ALTER 语句引用的 GO TO 语句被置为它的初始状态。

2.4.2 初始状态程序

一个程序在下列情况下处于初始状态:

- (1) 在一个运行单位中该程序第一次被调用。

(2) 在引用该程序的 CANCEL 语句或引用直接或间接包含该程序的程序的 CANCEL 语句执行之后第一次调用该程序。

- (3) 每次调用具有初始属性的程序。

(4) 在引用具有初始属性的程序或引用直接或间接包含该程序的 CALL 语句执行后,每一次调用该程序。

2.5 程序末端标题

2.5.1 功能

程序末端标题指出有名的 COBOL 源程序的结束。

2.5.2 一般格式

END PROGRAM 程序名。

2.5.3 语法规则

- (1) 程序名必须符合形成用户自定义字的规则。
- (2) 程序名必须与前面 PROGRAM-ID 段中说明的某一程序名相同。

(3) 如果说明一个特定的程序名的 PROGRAM-ID 段在说明和引用另一个程序名的 PROGRAM-ID 段和程序末端标题之间陈述,则引用较前程序名的程序末端标题必须先于引用较后程序名的程序末端标题出现。

2.5.4 一般规则

(1) 程序末端标题必须出现在每一个程序中,不管它包含另一个程序还是被包含在另一个程序中。

- (2) 程序末端标题指明所指定的 COBOL 源程序的结束。

(3) 如果一个以程序末端标题终结的程序包含在另一个程序中,则下一个语句必须要么是标识符,要么是终结外包程序的程序末端标题。

(4) 如果以程序末端标题终结的程序包含在另一个程序中,则下一个源语句是一条 COBOL 语句,则该语句必须是和以程序末端标题终结的程序进行分别编译的程序的标识部首。

3 程序间通信模块的标识部

3.1 PROGRAM-ID 段和嵌套源程序

3.1.1 功能

PROGRAM-ID 段指定标识一个程序的名并把所选取的程序属性指派给该程序。

3.1.2 一般格式

PROGRAM-ID. 程序名 $\left[\text{IS} \left\{ \begin{array}{l} \text{COMMON} \\ \text{INITIAL} \end{array} \right\} \text{PROGRAM} \right].$

3.1.3 语法规则

- (1) 程序名必须符合形成用户自定义字的规则。

(2) 一个包含在另一个程序中的程序不可指派与包含它的分别编译的程序所包含的任何其他程序名相同的名。

(3) 任选的 **COMMON** 子句仅当该程序包含在另一个程序中时使用。

3.1.4 一般规则

(1) 程序名标识与某一特定程序有关的源程序、目标程序和所有列表。

(2) **COMMON** 子句指出该程序是公用的。一个公用程序包含在另一个程序中但可以从不是包含它的程序中调用它(见 1.3.8 名的作用域)。

(3) **INITIAL** 子句指出该程序是初始的。当一个初始程序被调用时,它及任何被它包含的程序被视为它们的初始状态(见 2.4 程序的初始状态)。

4 程序间通信模块的数据部

4.1 连接节

连接节位于源程序的数据部。连接节出现在被调用程序中,描述由调用程序和被调用程序引用的数据项。

程序中的连接节当且仅当在下列情况下有意义:对象程序在 **CALL** 语句的控制下起作用,调用程序中的 **CALL** 语句包含 **USING** 短语。

连接节用来描述通过调用程序使之成为可用,但在调用和被调用程序中都将引用的数据。在被调用程序的连接节中描述的数据项和调用程序中描述的数据项之间建立对应关系的机制在其他地方描述(见 5.1 过程部首和 5.2 **CALL** 语句)。在位标名情况下,不建立这样的对应关系,而且被调用程序和调用程序中的位标名总是指向不同的位标。

连接节的结构与前面描述的工作存储节相同,以节首开始,后跟独立数据项和/或记录描述款。

连接节的一般格式如下所示。

LINKAGE SECTION.

[77 层描述款]
[记录描述款]...

如果连接节中的数据项被一个非被调用程序的程序存取,结果无定义。

4.1.1 独立连接存储

连接节中彼此无层次关系的数据项不必组成为记录,它们被划分和定义为独立初等项。每个这样的项在以特定层号 77 开始的单独的数据描述款中定义。

在每个数据描述款中,下列数据子句是必需的:

层号 77
数据名

PRCTURE 子句或 **USAGE IS INDEX** 子句。

其他数据描述子句是任选的,在需要时可用来完备数据项的描述。

4.1.2 连接记录

连接节中彼此具有一定层次关系的数据元素必须按照组成记录描述的规则组成记录。连接节中与其他数据项无层次关系的数据元素也可描述为单个初等项的记录。

4.1.3 初值

连接节中除条件名描述款(88 层)外不可指定 **VALUE** 子句。

4.2 程序间通信模块的文卷描述款

4.2.1 功能

在程序间通信模块中,文卷节中的文卷描述款确定文卷连接区、相关的数据记录及相关的数据项的内部或外部属性,文卷描述款也决定一个文卷名是局部名还是全程名。

4.2.2 一般格式

格式 1:

FD 文卷名 1

IS	EXTERNAL
IS	GLOBAL

$$\left[\text{BLOCK CONTAINS } [\text{整数 1 TO}] \text{ 整数 2 } \left\{ \begin{array}{l} \text{RECORDS} \\ \text{CHARACTERS} \end{array} \right\} \right]$$

$$\left[\text{RECORD } \left\{ \begin{array}{l} \text{CONTAINS 整数 3 CHARACTERS} \\ \text{IS VARYING IN SIZE } [[\text{FROM 整数 4} \\ \text{[TO 整数 5] CHARACTERS}][\text{DEPENDING ON 数据名 1}] \\ \text{CONTAINS 整数 6 TO 整数 7 CHARACTERS} \end{array} \right\} \right]$$

$$\left[\text{LABEL } \left\{ \begin{array}{l} \text{RECORD IS} \\ \text{RECORDS ARE} \end{array} \right\} \left\{ \begin{array}{l} \text{STANDARD} \\ \text{OMITTED} \end{array} \right\} \right]$$

$$\left[\text{VALUE OF } \left\{ \begin{array}{l} \text{实现名 1 IS} \\ \text{字值 1} \end{array} \right\} \dots \right]$$

$$\left[\text{DATA } \left\{ \begin{array}{l} \text{RECORD IS} \\ \text{RECORDS ARE} \end{array} \right\} \{ \text{数据名 3} \} \dots \right]$$

$$\left[\text{LINAGE IS } \left\{ \begin{array}{l} \text{数据名 4} \\ \text{整数 8} \end{array} \right\} \text{ LINES } \left[\text{WITH FOOTING AT } \left\{ \begin{array}{l} \text{数据名 5} \\ \text{整数 9} \end{array} \right\} \right] \right]$$

$$\left[\text{LINES AT TOP } \left\{ \begin{array}{l} \text{数据名 6} \\ \text{整数 10} \end{array} \right\} \right] \left[\text{LINES AT BOTTOM } \left\{ \begin{array}{l} \text{数据名 7} \\ \text{整数 11} \end{array} \right\} \right]$$

[CODE-SET IS 字母表名 1].

格式 2:

FD 文卷名 1

IS	EXTERNAL
IS	GLOBAL

$$\left[\text{BLOCK CONTAINS } [\text{整数 1 TO}] \text{ 整数 2 } \left\{ \begin{array}{l} \text{RECORDS} \\ \text{CHARACTERS} \end{array} \right\} \right]$$

$$\left[\text{RECORD } \left\{ \begin{array}{l} \text{CONTAINS 整数 3 CHARACTERS} \\ \text{IS VARYING IN SIZE } [[\text{FROM 整数 4} \quad \text{[TO 整数 5]} \\ \text{CHARACTERS}][\text{DEPENDING ON 数据名 1}] \\ \text{CONTAINS 整数 6 TO 整数 7 CHARACTERS} \end{array} \right\} \right]$$

$$\left[\text{LABEL } \left\{ \begin{array}{l} \text{RECORD IS} \\ \text{RECORDS ARE} \end{array} \right\} \left\{ \begin{array}{l} \text{STANDARD} \\ \text{OMITTED} \end{array} \right\} \right]$$

$$\left[\underline{\text{VALUE OF}} \left\{ \text{实现名 1 IS} \left\{ \begin{array}{l} \text{数据名 2} \\ \text{字值 1} \end{array} \right\} \right\} \dots \right]$$

$$\left[\underline{\text{DATA}} \left\{ \begin{array}{l} \underline{\text{RECORD IS}} \\ \underline{\text{RECORDS ARE}} \end{array} \right\} \{ \text{数据名 3} \} \dots \right].$$

格式 3:

FD 文卷名 1

IS EXTERNAL
IS GLOBAL

$$\left[\underline{\text{BLOCK CONTAINS}} [\text{整数 1 TO}] \text{整数 2} \left\{ \begin{array}{l} \underline{\text{RECORDS}} \\ \underline{\text{CHARACTERS}} \end{array} \right\} \right]$$

$$\left[\underline{\text{RECORD}} \left\{ \begin{array}{l} \text{CONTAINS 整数 3 CHARACTERS} \\ \text{IS VARYING IN SIZE} [[\text{FROM 整数 4}] [\text{TO 整数 5}]] \\ \text{CHARACTERS}] [\underline{\text{DEPENDS ON 数据名 1}}] \\ \text{CONTAINS 整数 6 TO 整数 7 CHARACTERS} \end{array} \right\} \right]$$

$$\left[\underline{\text{LABEL}} \left\{ \begin{array}{l} \underline{\text{RECORD IS}} \\ \underline{\text{RECORDS ARE}} \end{array} \right\} \left\{ \begin{array}{l} \underline{\text{STANDARD}} \\ \underline{\text{OMITTED}} \end{array} \right\} \right]$$

$$\left[\underline{\text{VALUE OF}} \left\{ \text{实现名 1 IS} \left\{ \begin{array}{l} \text{数据名 2} \\ \text{字值 1} \end{array} \right\} \right\} \dots \right]$$

$$[\underline{\text{CODE-SET IS 字母表名 1}}]$$

$$\left\{ \begin{array}{l} \underline{\text{REPORT IS}} \\ \underline{\text{REPORTS ARE}} \end{array} \right\} \{ \text{报表名 1} \} \dots$$

4.2.3 语法规则

(1) 格式 1 是顺序文卷的文卷描述款。在这个文卷描述款中特定子句的可用性依赖于实现所支撑的顺序 I-O 模块的级别(见 GB/T 4092.4)。

(2) 格式 2 是相对文卷或索引文卷的文卷描述款。在这个文卷描述款中特定子句的可用性依赖于实现所支撑的相对 I-O 模块或索引 I-O 模块的级别(见 GB/T 4092.5 和 GB/T 4092.6)。

(3) 格式 3 是报表文卷的文卷描述款。报表文卷的文卷描述款的可用性取决于实现是否支撑报表编制模块(见 GB/T 4092.8)。

4.2.4 一般规则

(1) 如果顺序文卷的文卷描述款包含 **LINAGE** 子句和 **EXTERNAL** 子句,则 **LINAGECOUNTER** 数据项是一个外部数据项。如果顺序文卷的文卷描述款包含 **LINAGE** 子句和 **GLOBAL** 子句,则特殊寄存器 **LINAGE-COUNTER** 是一全程名。

(2) **EXTERNAL** 子句在 4.5 给出。**GLOBAL** 子句在 4.6 给出。文卷描述款中的所有其他子句在适当模块的说明中给出。

4.3 程序间通信模块的数据描述款

4.3.1 功能

程序间通信模块中,工作存储节或文卷节中的 01 层数据描述款决定数据记录及其下属数据项具有局部名还是全程名。

程序间通信模块中,工作存储节中的 01 层数据描述款决定数据记录及其下属数据项具有内容属性还是外部属性。

4.3.2 一般格式

01 [数据名 1]
[FILLER]

[REDEFINES 数据名 2]

[IS EXTERNAL]

[IS GLOBAL]

[{ PICTURE } IS 字符串]
[PIC][[USAGE IS] {
BINARY
COMPUTATIONAL
COMP
DISPLAY
INDEX
PACKED-DECIMAL
}]][[SIGN IS] { LEADING
TRAILING } [SEPARATE CHARACTER]][OCCURS 整数 2 TIMES
[{ ASCENDING
DESCENDING } KEY IS {数据名 3}...]...
[INDEXED BY {位标名 1}...]
OCCURS 整数-1 TO 整数 2 TIMES
DEPENDENT ON 数据名 4
[{ ASCENDING
DESCENDING } KEY IS {数据名 3}...]...
[INDEXED BY {位标名 1}...]
[{ SYNCHRONIZED } [LEFT
RIGHT]
[SYNC]
[{ JUSTIFIED } RIGHT]
[JUST]
[BLANK WHEN ZERO]
[VALUE IS 字值 1].]

4.3.3 语法规则

(1) 数据描述款中特定子句的可用性依赖于实现所支撑的核心模块的级别(见核心模块)。

(2) EXTERNAL 子句只可在工作存储节中层号为 01 的数据描述款中指出。

(3) EXTERNAL 子句和 REDEFINES 子句在一个数据描述款中不可同时指出。

(4) GLOBAL 子句只可在层号为 01 的数据描述款中指出。

(5) 对任何包含 GLOBAL 子句或 EXTERNAL 子句的描述款,以及与包含 EXTERNAL 子句或 GLOBAL 子句的父类描述款相关的记录描述,都必须指出数据名 1。

4.3.4 一般规则

[EXTERNAL] [GLOBAL] [REDEFINES] [VALUE IS 字值 1] 数据描述款的所有其他子

句在核心模块中给出。

4.4 程序间通信模块的报表描述款

4.4.1 功能

程序间通信模块中,报表节中的报表描述款决定一个报表名是局部名还是全程名。

4.4.2 一般格式

RD 报表名 1

IS GLOBAL

[**CODE** 字值 1]

[**{CONTROL IS** { {数据名 1}...
CONTROLS ARE } **FINAL** [数据名 1]... }]

[**PAGE** [**LIMIT IS**
LIMITS ARE] 整数 1 [**LINE**
LINES] [**HEADING** 整数 2]

[**FIRST DETAIL** 整数 3][**LAST DETAIL** 整数 4]

[**FOOTING** 整数 5]].

4.4.3 语法规则

(1) 报表描述款的可用性依赖于实现是否支撑报表编制模块。

4.4.4 一般规则

(1) 如果报表描述款包含 **GLOBAL** 子句,则特殊寄存器 **LINE-COUNTER** 和 **PAGECOUNTER** 是全程名。

(2) 报表描述款中的所有其他子句在报表编制模块中给出。

4.5 **EXTERNAL** 子句

4.5.1 功能

EXTERNAL 子句指明一个数据项或一文卷连接区是外部的。一个外部数据记录的成分数据项和组合数据项对运行单位中描述那个记录的所有程序是可用的。

4.5.2 一般格式

IS EXTERNAL

4.5.3 语法规则

(1) **EXTERNAL** 子句只可在文卷描述款或在工作存储节的记录描述款中指定。

(2) 在同一个程序中,作为层号为 01 且包含 **EXTERNAL** 子句的描述款的主体指定的数据名不可与任何其他包含 **EXTERNAL** 子句的数据描述款指定的数据名相同。

(3) 在包含 **EXTERNAL** 子句的数据描述款及其下属的数据描述款中不可使用 **VALUE** 子句。但对与这些数据描述款相关的条件名可使用 **VALUE** 子句。

4.5.4 一般规则

(1) 由数据名字句命名的记录中所包含的数据是外部的,它们可被运行单位中描述它或服从下面的一般规则地重定义它的任何程序存取和处理。

(2) 在一个运行单位中,如果两个或多个程序描述了同一个外部数据记录,则相关的记录描述款的各个记录名必须相同,且记录必须定义相同个数的标准数据格式字符。但是,描述外部记录的程序可以包含一个包括 **REDEFINES** 子句的数据描述款,它重定义该完整的外部记录,而这完整的重定义不必在运行单位中的其他程序中同样出现(见 GB/T 4092.2 中 5.10 **REDEFINES** 子句)。

(3) **EXTERNAL** 子句的使用并不意味着相关的文卷名或数据名是一个全程名(见 4.6 **GLOBAL** 子句)。

(4) 与这个描述款相关的文卷连接区是外部文卷连接区。

4.6 GLOBAL 子句

4.6.1 功能

GLOBAL 子句指明数据名、文卷名或报表名是全程名。一个全程名对包含于描述它的程序中的每个程序均是可用的。

4.6.2 一般格式

IS GLOBAL

4.6.3 语法规则

(1) **GLOBAL** 子句只可在文卷节或工作存储节的层号为 01 的数据描述款中指出,或在文卷描述款中或报表描述款中指出。

(2) 在同一个数据部中,在指定了相同数据名的两个数据项的数据描述款中不可包括 **GLOBAL** 子句。

(3) 如果为若干文卷指定了 **SAME RECORD AREA** 子句,则这些文卷的记录描述款或文卷描述款中不可包括 **GLOBAL** 子句。

4.6.4 一般规则

(1) 使用 **GLOBAL** 子句描述的数据名、文卷名或报表名是一个全程名。下属于一个全程名的所有数据名是全程名。与全程名相关的所有条件名是全程名。

(2) 直接或间接包含在描述全程名的程序中的程序的语句可不再描述而直接引用该名(见 1.3.1 名的作用域)。

(3) 如果在包含 **REDEFINES** 子句或 **RENAMES** 子句的数据描述款中使用了 **GLOBAL** 子句,只有 **REDEFINES** 或 **RENAMES** 子句的主体具有全程属性。

5 程序间通信模块的过程部

5.1 过程部首

过程部由下列部首来标识,并且必须以它开始:

PROCEDURE DIVISION[**USING** {数据名 1}...].

USING 短语只在该程序将由 **CALL** 语句调用且 **CALL** 语句中包含 **USING** 短语才是必须的。

过程部首的 **USING** 短语标识出程序将使用的由调用程序传递来的参数的名字。传递给被调用程序的参数在调用程序的 **CALL** 语句的 **USING** 短语中标识出来,两个名字表之间基于位置关系建立对应。

数据名 1 必须在连接节中以 01 层描述款或 77 层描述款定义。某一特定的用户自定义字不可以数据名 1 的形式出现多次。数据名 1 的数据描述款不可含有 **REDEFINES** 子句,但是,数据名 1 可以在连接节中的其他位置作为 **REDEFINES** 子句的客体出现。

有下列附加的规则要应用:

(1) 如果 **CALL** 语句中对于对应数据项的引用把参数说明为按内容传递,则在 **CALL** 语句执行时把数据项的值传递过去并且送入连接节中为数据名 1 说明的具有该属性的系统定义的存储项中。**CALL** 语句的 **BY CONTENT** 短语中的各个参数的数据描述必须与过程部首的 **USING** 短语中对应参数的数据描述完全相同,即不必转换、扩展或截断(见 5.2 **CALL** 语句)。

(2) 如果 **CALL** 语句中对于对应数据项的引用把参数说明为按引用传递,则目标程序执行时就好像被调用程序中的数据项与调用程序中的数据项占据相同的存储区域。被调用程序中的数据项描述与调用程序中对应的数据项描述必须描述相同个数的字符位置。

(3) 被调用程序执行的任何时刻,对数据名 1 的引用一定按照被调用程序连接节中给出的数据项描述来进行。

(4) 被调用程序连接节中定义的数据项,当且仅当满足下列条件之一时,才可在那个程序的过程部中引用:

- a. 它们是过程部首的 **USING** 短语中的操作分量。
- b. 它们是过程部首的 **USING** 短语中的操作分量的下属量。
- c. 它们是用 **REDEFINES** 子句,或 **RENAME** 子句来定义的满足上述条件的对象。
- d. 它们是满足规则 4C 中条件的任何数据项的下属项。
- e. 它们是与满足上述四个条件的数据项相关的条件名或位标名。

在 1 级,在过程部首的 **USING** 短语中及 **CALL** 语句的 **USING** 短语中至少必须允许五个数据名。

5.2 CALL 语句

5.2.1 功能

CALL 语句导致在一个运行单位中将控制从一个目标程序转到另一个目标程序。

5.2.2 一般格式

格式 1:

```
CALL { 字值 1 } [ USING { [ BY REFERENCE ] {标识符 2}... } ... ]
                      [ BY CONTENT {标识符 2}... } ... ]
```

```
ON OVERFLOW 命令语句 1
END CALL
```

格式 2:

```
CALL { 字值 1 } [ USING { [ BY REFERENCE ] {标识符 2}... } ... ]
                      [ BY CONTENT {标识符 2}... } ... ]
```

```
ON EXCEPTION 命令语句 1
NOT ON EXCEPTION 命令语句 2
END CALL
```

5.2.3 语法规则

(1) 字值 1 必须是一个非数值字值。

```
字值 1 必须是一个非数值字值,如:
CALL 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363, 364, 365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 377, 378, 379, 380, 381, 382, 383, 384, 385, 386, 387, 388, 389, 390, 391, 392, 393, 394, 395, 396, 397, 398, 399, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 418, 419, 420, 421, 422, 423, 424, 425, 426, 427, 428, 429, 430, 431, 432, 433, 434, 435, 436, 437, 438, 439, 440, 441, 442, 443, 444, 445, 446, 447, 448, 449, 450, 451, 452, 453, 454, 455, 456, 457, 458, 459, 460, 461, 462, 463, 464, 465, 466, 467, 468, 469, 470, 471, 472, 473, 474, 475, 476, 477, 478, 479, 480, 481, 482, 483, 484, 485, 486, 487, 488, 489, 490, 491, 492, 493, 494, 495, 496, 497, 498, 499, 500, 501, 502, 503, 504, 505, 506, 507, 508, 509, 510, 511, 512, 513, 514, 515, 516, 517, 518, 519, 520, 521, 522, 523, 524, 525, 526, 527, 528, 529, 530, 531, 532, 533, 534, 535, 536, 537, 538, 539, 540, 541, 542, 543, 544, 545, 546, 547, 548, 549, 550, 551, 552, 553, 554, 555, 556, 557, 558, 559, 560, 561, 562, 563, 564, 565, 566, 567, 568, 569, 570, 571, 572, 573, 574, 575, 576, 577, 578, 579, 580, 581, 582, 583, 584, 585, 586, 587, 588, 589, 590, 591, 592, 593, 594, 595, 596, 597, 598, 599, 600, 601, 602, 603, 604, 605, 606, 607, 608, 609, 610, 611, 612, 613, 614, 615, 616, 617, 618, 619, 620, 621, 622, 623, 624, 625, 626, 627, 628, 629, 630, 631, 632, 633, 634, 635, 636, 637, 638, 639, 640, 641, 642, 643, 644, 645, 646, 647, 648, 649, 650, 651, 652, 653, 654, 655, 656, 657, 658, 659, 660, 661, 662, 663, 664, 665, 666, 667, 668, 669, 670, 671, 672, 673, 674, 675, 676, 677, 678, 679, 680, 681, 682, 683, 684, 685, 686, 687, 688, 689, 690, 691, 692, 693, 694, 695, 696, 697, 698, 699, 700, 701, 702, 703, 704, 705, 706, 707, 708, 709, 710, 711, 712, 713, 714, 715, 716, 717, 718, 719, 720, 721, 722, 723, 724, 725, 726, 727, 728, 729, 730, 731, 732, 733, 734, 735, 736, 737, 738, 739, 740, 741, 742, 743, 744, 745, 746, 747, 748, 749, 750, 751, 752, 753, 754, 755, 756, 757, 758, 759, 760, 761, 762, 763, 764, 765, 766, 767, 768, 769, 770, 771, 772, 773, 774, 775, 776, 777, 778, 779, 780, 781, 782, 783, 784, 785, 786, 787, 788, 789, 790, 791, 792, 793, 794, 795, 796, 797, 798, 799, 800, 801, 802, 803, 804, 805, 806, 807, 808, 809, 810, 811, 812, 813, 814, 815, 816, 817, 818, 819, 820, 821, 822, 823, 824, 825, 826, 827, 828, 829, 830, 831, 832, 833, 834, 835, 836, 837, 838, 839, 840, 841, 842, 843, 844, 845, 846, 847, 848, 849, 850, 851, 852, 853, 854, 855, 856, 857, 858, 859, 860, 861, 862, 863, 864, 865, 866, 867, 868, 869, 870, 871, 872, 873, 874, 875, 876, 877, 878, 879, 880, 881, 882, 883, 884, 885, 886, 887, 888, 889, 890, 891, 892, 893, 894, 895, 896, 897, 898, 899, 900, 901, 902, 903, 904, 905, 906, 907, 908, 909, 910, 911, 912, 913, 914, 915, 916, 917, 918, 919, 920, 921, 922, 923, 924, 925, 926, 927, 928, 929, 930, 931, 932, 933, 934, 935, 936, 937, 938, 939, 940, 941, 942, 943, 944, 945, 946, 947, 948, 949, 950, 951, 952, 953, 954, 955, 956, 957, 958, 959, 960, 961, 962, 963, 964, 965, 966, 967, 968, 969, 970, 971, 972, 973, 974, 975, 976, 977, 978, 979, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000
```

(3) **USING** 短语中的各个数据项必须定义为文卷节、工作存储节、通信节或连接节中的数据项,并且必须是一个 01 层数据项、77 层数据项或一个初等数据项。

5.2.4 一般规则

(1) 字值 1 是被调用程序的名字。出现 **CALL** 语句的那个程序为调用程序。如果被调用程序是一个 **COBOL** 程序,则字值 1 必须是被调用程序的 **PROGRAM-ID** 段所包含的程序名。如果被调用程序不是 **COBOL** 程序,则由实现者定义形成程序名的规则。

(2) 当 **CALL** 语句执行时,如果由 **CALL** 语句指定的程序可以执行,则控制转到被调用程序。当控制从被调用程序返回时,控制转到 **CALL** 语句的结束处,或者,如果指定了 **NOT ON EXCEPTION** 短语,则转到命令语句 2。如果控制转到命令语句 2,则按照命令语句 2 中指定的各语句的规则继续执行。如果导致显式控制转移的过程分支或条件语句执行,则控制按那条语句的规则转移。否则,命令语句 2 执行完成后,控制转到 **CALL** 语句的结束处。

(3) 当 **CALL** 语句执行时,如果判别出在那时由 **CALL** 语句指定的那个程序不可用于执行,则下列两个动作之一将出现。用于判别被调用程序可用于执行与否而必须检查的目标时刻资源由实现者确定。

a. 如果在 **CALL** 语句中指出了 **ON OVERFLOW** 或 **ON EXCEPTION** 短语,则控制转到命令语句 1,按照命令语句 1 指定的各语句的规则继续运行。如果导致显式的控制转移的过程分支或条件语句执行,则控制按那条语句的规则转移,否则,命令语句 1 执行完成后,控制转到 **CALL** 语句的结束。并且,如果指出 **NOT ON EXCEPTION** 的话,将被忽略。

b. 如果在 **CALL** 语句中未指出 **ON OVERFLOW** 或 **ON EXCEPTION** 短语而又指出 **NOT ON EXCEPTION** 短语,则后者被忽略。**CALL** 语句的所有其他效果由实现者定义。

(4) 运行单位中的两个或多个程序可以具有相同的程序名,**CALL** 语句中对这一程序名的引用是通过使用程序名的作用域规则而决定(见 1.3.8.1 程序名的约定)。

例如,如果运行单位中在 **CALL** 语句中指明同程序名的程序只有两个:

a. 这两个程序中的一个必然直接或间接地包含于含有 **CALL** 语句的分别编译的程序,或者直接或间接地包含于本身直接或间接包含含有该 **CALL** 语句的程序的分别编译的程序。

b. 这两个程序中的另一个必是别个不同的分别编译的程序。

这个例子使用的机制如下:

a. 如果具有 **CALL** 语句中指定的程序名的两个程序之一直接或间接包含于含有 **CALL** 语句的程序中,则该程序为被调程序。

b. 如果具有 **CALL** 语句中指定的程序名的两个程序之一具有公用属性并且直接或间接包含于另一个直接或间接包含含有该 **CALL** 语句的程序的程序,则这个公用程序就被调用,除非调用程序包含于那个公用程序之中。

c. 否则,分别编译的那个程序被调用。

(5) 如果被调程序不具初始属性,则它及直接或间接包含于它的各个程序,在运行单位中第一次被调用时,或在对该被调程序执行 **CANCEL** 语句后第一次被调用时,处于其初始状态。

在其他所有情况下进入该被调程序时,该程序保持上次退出它时的状态不变。

(6) 如果被调程序具有初始属性,则它及直接或间接包含于它的程序在运行单位中每次对它调用时被置于其初始状态。

(7) 与被调程序的内部文卷连接区相关的文卷在程序处于初始状态时不处于打开方式(见 2.4 程序的初始状态)。

在其他所有情况下进入被调程序时,这些文卷的状态和位置与上次退出它时的状态和位置一样。

(8) 调用一个程序或从被调程序返回的处理并不改变与外部文卷连接区相关的文卷的状态和位置。

(9) 如果将被调用的是一 **COBOL** 程序,只在被调程序的过程部首中含有 **USING** 短语时 **CALL** 语句中才需包含 **USING** 短语,这时,**USING** 短语中操作分量的个数必须相同。如果将被调用的程序不是 **COBOL** 程序,则 **USING** 短语的使用由实现者定义。

(10) **CALL** 语句的 **USING** 短语中数据名出现的次序与被调程序的过程部首中相应的 **USING** 短语中数据名出现的次序决定了由调用程序和被调程序使用的数据名之间的对应关系。这种对应是按位置相同而不是按名字相同的。**USING** 短语中的第一个数据名与另一短语中的第一个数据名对应,第二个对应第二个,依此类推。

(11) **CALL** 语句中 **USING** 短语指出的参数的值在执行 **CALL** 语句时,对被调程序是可用的。

(12) **BY CONTENT** 和 **BY REFERENCE** 短语对跟在它们后面的参数是可传递的,直至遇到下一个 **BY CONTENT** 或 **BY REFERENCE** 短语为止。如果在第一个参数前既无 **BY CONTENT** 也无 **BY REFERENCE** 短语,则认为是有 **BY REFERENCE** 短语。

(13) 如果对一个参数 **BY REFERENCE** 短语, 则目标程序处理时就好像被调用程序中的数据项与调用程序中对应的数据项占有相同的存储区域。被调用程序中的数据项必须与调用程序的对应数据项描述有相同个数的字符位置。

(14) 如果对一个参数指定了或隐含有 **BY CONTENT** 短语, 则被调用程序不能改变 **CALL** 语句的 **USING** 短语中指出的这个参数的值, 但它能改变被调用程序的过程部首中对应的数据名所代表的参数数据项的值。 **CALL** 语句的 **BY CONTENT** 短语中的各个参数的数据描述必须与过程部首的 **USING** 短语中对应的参数的数据描述相同, 即不须转换、扩展或截断(见 5.1 过程部首)。

(15) 被调用程序可包含 **CALL** 语句。但是, 被调用程序不可执行直接或间接再调用程序的 **CALL** 语句。如果在上述的范围内执行 **CALL** 语句, 则这条 **CALL** 语句不可直接或间接引用任何控制已转到且执行未结束的被调用程序。

(16) **END-CALL** 短语限定 **CALL** 语句的作用域(见 GB/T 4092.1 中 6.6.4.3 语句的作用域)。

3.3 CANCEL 语句

3.3.1 功能

CANCEL 语句保证所引用的程序在下次被调用时将处于它的初始状态。

3.3.2 一般格式

CANCEL { 标识符 1 } ...
 { 字值 1 } ...

3.3.3 语法规则

- (1) 字值 1 必须是一非数值字值。
- (2) 标识符 1 必须引用一字符数据项。

3.3.4 一般规则

- (1) 字值 1 或由标识符 1 所引用的数据项的内容标识了将被撤销的程序。
- (2) 在显式或隐式的 **CANCEL** 语句执行之后, 语句中指出的程序停止与 **CANCEL** 语句出现的运行单位有任何逻辑联系。如果运行单位中显式或隐式的 **CANCEL** 语句执行后, 语句中引用的程序在该运行单位中又再次被调用, 则该程序处于它的初始状态(见 2.4 程序的初始状态)。
- (3) 在另一程序的 **CANCEL** 语句中列出的程序, 必须对该另一程序是可调用的(见 1.3.8 名的作用域和 5.2 **CALL** 语句)。
- (4) 当显式的或隐式的 **CANCEL** 语句执行后, 包含在由 **CANCEL** 语句指出的程序中的所有程序也被撤销。结果与以各个被包含程序在分别编译程序组中出现的相反次序对每个所包含的程序执行有效的 **CANCEL** 语句一样。
- (5) **CANCEL** 语句中列出的程序不可直接或间接地引用任何已被调用且未执行 **EXIT PROGRAM** 语句的程序。
- (6) 只在后继的含有被撤销程序名字的 **CALL** 语句执行后, 与该程序的逻辑关系才被重新建立。
- (7) 被调用程序被撤销, 要么由于作为 **CANCEL** 语句的一个操作分量而引用, 要么由于该程序为其成员的运行单位终止, 要么由于具有初始属性的被调用程序执行了 **EXIT PROGRAM** 语句。
- (8) 在显式的和隐式的 **CANCEL** 语句执行时, 如果所命名的程序在运行单位中未被调用, 或已被调用但现已撤销, 则为空操作。控制转到跟在显式的 **CANCEL** 语句之后下一条可执行的语句。
- (9) 由程序描述的外部数据记录中的数据项的内容在程序被撤销时不改变。

(10) 在显式的或隐式的 **CANCEL** 语句执行过程中, 对在 **CANCEL** 语句中显式列出的程序中的内部文卷连接区相关的处于打开方式的各个文卷执行一不带任何任选项的隐式的 **CLOSE** 语句。与这些文卷相关的任何 **USE** 过程都不执行。

5.4 EXIT PROGRAM 语句

5.4.1 功能

EXIT PROGRAM 语句标志被调用程序的逻辑结束。

5.4.2 一般格式

EXIT PROGRAM

5.4.3 语法规则

(1) 如果 **EXIT PROGRAM** 语句出现在句子的连续命令语句序列中,它必须作为那句的最后一条语句出现。

(2) **EXIT PROGRAM** 语句不可出现在指定 **GLOBAL** 短语的申述过程中。

5.4.4 一般规则

(1) 如果 **EXIT PROGRAM** 语句不在调用程序控制下的程序中执行,则 **EXIT PROGRAM** 语句使程序从下一条可执行语句开始继续执行。

(2) 不具有初始属性的被调用程序中的 **EXIT PROGRAM** 语句的执行,使得调用程序中跟在 **CALL** 语句后的下一条可执行语句继续执行。调用程序的程序状态并不改变,与它执行 **CALL** 语句时存在的状态相同,只是调用程序和被调用程序共享的数据项的内容和数据文卷的内容可能已经改变。被调用程序的程序状态都不改变,除由被调用程序执行的所有 **PERFORM** 语句的范围结束处被达到外。

(3) 除一般规则 2 中指出的动作外,具有初始属性的被调用程序中的 **EXIT PROGRAM** 执行相当于执行引用该程序的 **CANCEL** 语句(见 5.3 **CANCEL** 语句)。

5.5 USE 语句

5.5.1 功能

在程序间通信模块中,**USE** 语句决定在包含有 **USE** 语句的程序中的任一程序执行时,相应的申述过程是否执行。

5.5.2 一般格式

<u>USE</u>	[<u>GLOBAL</u>]	<u>AFTER</u>	<u>STANDARD</u>	{ <u>EXCEPTION</u> <u>ERROR</u> }
				(文卷名 1)...
		<u>PROCEDURE</u>	ON	{ <u>INPUT</u> <u>OUTPUT</u> <u>I-O</u> <u>EXTEND</u> }

5.5.3 语法规则

(1) 多文卷名和 **EXTEND** 短语的可用性依赖于实现所支撑的顺序 **I-O** 模块、相对 **I-O** 模块或索引 **I-O** 模块的级别(见 GB/T 4092.4;GB/T 4092.5 和 GB/T 4092.6)。

5.5.4 一般规则

(1) 当若干程序包含在其他程序中时,须遵循指定的优先规则。使用这些规则时,只有第一个合格申述被选出执行。选出执行的申述必须满足执行那个申述的规则。选择申述的优先次序是:

- a. 该申述包含在导致合格条件成立的语句的程序中。

b. 该申述指出了 **GLOBAL** 短语,且在直接包含上一次检查合格申述的程序的程序中。

c. 对各个更外层的包含程序使用规则 1b,直至规则 1b 用于最外层程序而选择的申述,如果未找到合格的申述。则不执行任何操作。

5.6 USE BEFORE REPORTING 语句

5.6.1 功能

在程序间通信模块中,USE BEFORE REPORTING 语句决定在包含有 USE BEFORE REPORTING 语句的程序执行时,相应的申述过程是否执行。

5.6.2 一般格式

USE [GLOBAL] BEFORE REPORTING 标识符 1

5.6.3 语法规则

(1) USE BEFORE REPORTING 语句的可用性取决于实现是否支撑报表编制模块。

5.6.4 一般规则

(1) 当若干程序包含在其他程序中时,须遵循特定的优先规则。在使用这些规则时,只有第一个合格的申述被选出执行。选出执行的申述必须满足执行那个申述的规则。选择申述的优先次序是:

a. 该申述包含在导致合格条件成立的语句的程序中。

b. 该申述指出了 **GLOBAL** 短语,且在直接包含上一次检查合格申述的程序的程序中。

c. 对各个更外层的包含程序使用规则 1b,直至规则 1b 用于最外层程序而选择的申述。如果未找到合格的申述,则不执行任何操作。

附加说明:

本标准由中华人民共和国机械电子工业部提出。

本标准由南京大学负责起草。

本标准主要起草人钱树人、王静英、冯惠、段祥。

本标准由 1983 年 12 月首次发布,1992 年 8 月第一次修订。