



中华人民共和国国家标准

GB/T 7027—2002
代替 GB/T 7027—1986

信息分类和编码的基本原则与方法

Basic principles and methods for information
classifying and coding

2002-07-18 发布

2002-12-01 实施

中华人民共和国
国家质量监督检验检疫总局 发布

目 次

前言	Ⅲ
引言	Ⅳ
1 范围	1
2 规范性引用文件	1
3 术语和定义	1
4 信息的分类与编码	1
4.1 信息分类	1
4.2 信息编码	1
5 信息分类的基本原则	2
5.1 科学性	2
5.2 系统性	2
5.3 可扩展性	2
5.4 兼容性	2
5.5 综合实用性	2
6 信息分类的基本方法	2
6.1 概述	2
6.2 线分类法	2
6.3 面分类法	3
6.4 混合分类法	3
7 信息编码的基本原则	3
7.1 唯一性	3
7.2 合理性	3
7.3 可扩充性	3
7.4 简明性	3
7.5 适用性	3
7.6 规范性	3
8 信息编码的基本方法	4
8.1 通则	4
8.2 代码类型	4
8.3 代码特征	7
8.4 代码表现形式	8
8.5 代码设计	10
8.6 代码赋值约定	11
附录 A(资料性附录) 各种信息分类编码方法的优缺点	12
A.1 信息分类方法优缺点	12
A.2 各种类型代码编码方法优缺点	12

前 言

本标准是对 GB/T 7027—1986《标准化工作导则 信息分类编码的基本原则和方法》的修订。在信息编码部分内容上,本标准参考了国际技术报告 ISO/IEC TR 9789:1994(E)《信息技术——数据交换用数据元素组织与表示指南——编码方法与原理》,采纳了其中比较成熟的相关技术内容。

本标准代替 GB/T 7027—1986《标准化工作导则 信息分类编码的基本原则和方法》。同 GB/T 7027—1986相比,本次修订所作的主要修改是:

- 修改了标准的名称。标准名称改为《信息分类和编码的基本原则与方法》。
- 标准的总体编排和结构按 GB/T 1.1—2000 进行了修改,增加了目次、前言、引言和附录 A。
- 对原标准的内容进行了相应的增删。增补的内容包括:第 2 章“规范性引用文件”、第 3 章“术语和定义”和第 4 章“信息的分类与编码”概述。删除的内容是:原标准的第 2.4 条“代码的校验”的有关算法。
- 对原标准的结构进行了调整:原标准的第 1.1 条“信息分类的基本原则”调整为第 5 章,原标准的第 1.2 条“信息分类的基本方法”调整为第 6 章,原标准的第 2.2 条“编码的基本原则”调整为第 7 章,原标准的第 2.3 条“代码的种类”和第 2.5 条“代码的类型”与 ISO/IEC TR 9789 的相关技术内容经过整理共同构成第 8 章“信息编码的基本方法”,原标准中分散叙述的各个信息分类和编码方法的优缺点集中汇总调整为“附录 A 各种信息分类编码方法的优缺点”。
- 对原标准中的代码名称进行了若干项调整:原标准中的“特征组合码”对应于本标准的“并置码”,原标准中的“复合码”对应于本标准的“组合码”,原标准中的“数值化字母顺序码”被本标准的“约定顺序码”所涵盖。

在信息分类编码标准化领域,本标准应与 GB/T 20001.3—2001《标准编写规则 第 3 部分:信息分类编码》和 GB/T 10113《分类编码通用术语》两项标准配套应用。

本标准的附录 A 是资料性附录。

本标准由中国标准研究中心提出并归口。

本标准主要起草单位:中国标准研究中心。

本标准主要起草人:李小林、冯卫、胡嘉璋。

GB/T 7027 于 1986 年 11 月首次发布,本次修订为第一次修订。

引 言

在通常情况下,人们对信息的理解是:一切有含义的具体或抽象事物或概念的真相及相关陈述,通过数据、消息及其进一步细节表达出来。

在信息分类编码领域,信息的表现形式是数据。

客观、明确的信息是计算机建立信息系统以及数据在其中进行交换的先决条件。

在信息系统中,数据是用字符(通常为数字或字母)、算术符号以及描述来表示,这些表示形式应该对其所涉及的每一个数据都有一个明确稳定的含义,从而达到处理与交流的目的。

信息要被不同用户组或应用系统所共享,就必须有一致认可的定义,举例来说,要有概念的语义含义(内涵)、概念的全部实例(外延)以及一致认可的表示法。

对各类信息概念的正确理解需要依赖于信息分类;对各类信息作出一致认可的表示需要依赖于信息编码。

信息分类和编码的基本原则与方法

1 范围

本标准规定了信息分类编码的基本原则和方法,适用于各类信息分类编码标准的编制。

2 规范性引用文件

下列文件中的条款通过本标准的引用而成为本标准的条款。凡是注日期的引用文件,其随后所有的修改单(不包括勘误的内容)或修订版均不适用于本标准,然而,鼓励根据本标准达成协议的各方研究是否可使用这些文件的最新版本。凡是不注日期的引用文件,其最新版本适用于本标准。

GB/T 1988—1998 信息技术 信息交换用七位编码字符集(eqv ISO/IEC 646:1991)

GB 2312—1980 信息交换用汉字编码字符集 基本集

GB/T 2260—2002 中华人民共和国行政区划代码

GB/T 2659—2000 世界各国和地区名称代码(eqv ISO 3166-1:1997)

GB/T 4657—2002 中央党政机关、人民团体及其他机构代码

GB/T 7408—1994 数据元和交换格式 信息交换 日期和时间表示法(eqv ISO 8601:1988)

GB/T 10113 分类编码通用术语

GB 11643—1999 公民身份号码

GB/T 13745—1992 学科分类与代码

GB/T 14721.1—1993 林业资源分类与代码 森林类型

GB/T 14805—1993 用于行政、商业和运输业电子数据交换的应用级语法规则(idt ISO 9735:1988)

GB/T 17710—1999 数据处理 校验码系统(idt ISO 7064:1983)

3 术语和定义

GB/T 10113中确立的术语适用于本标准。

4 信息的分类与编码

4.1 信息分类

信息分类是根据信息内容的属性或特征,将信息按一定的原则和方法进行区分和归类,并建立起一定的分类体系和排列顺序。

信息分类有两个要素:一是分类对象,二是分类的依据。分类对象由若干个被分类的实体组成。分类依据取决于分类对象的属性或特征。

信息内容属性的相同或相异,形成了各种不同的类。在信息分类体系中,类可称为类目。

4.2 信息编码

信息编码是将事物或概念(编码对象)赋予具有一定规律、易于计算机和人识别处理的符号,形成代码元素集合。代码元素集合中的代码元素就是赋予编码对象的符号,即编码对象的代码值。

所有类型的信息都能够进行编码:如关于产品、人、国家、货币、程序、文件、部件等各种各样的信息。

信息编码包含的内容有:数据表达成代码的方法、数据的代码表示形式、代码元素集合的赋值。

信息编码的主要作用有:标识、分类、参照。

标识的目的是要把编码对象彼此区分开,在编码对象的集合范围内,编码对象的代码值是其唯一性标志;信息编码的分类作用实质上是对类进行标识;信息编码的参照作用体现在编码对象的代码值可作为不同应用系统或应用领域之间发生关联的关键字。

5 信息分类的基本原则

5.1 科学性

宜选择事物或概念(即分类对象)最稳定的本质属性或特征作为分类的基础和依据。

5.2 系统性

将选定的事物、概念的属性或特征按一定排列顺序予以系统化,并形成一科学合理的分类体系。

5.3 可扩延性

通常要设置收容类目,以保证增加新的事物或概念时,不打乱已建立的分类体系,同时,还应为下级信息管理系统在本分类体系的基础上进行延拓细化创造条件。

5.4 兼容性

应与相关标准(包括国际标准)协调一致。

5.5 综合实用性

分类要从系统工程角度出发,把局部问题放在系统整体中处理,达到系统最优。即在满足系统总任务、总要求的前提下,尽量满足系统内各相关单位的实际需要。

6 信息分类的基本方法

6.1 概述

信息分类的基本方法有三种:线分类法、面分类法、混合分类法。其中线分类法又称层级分类法、体系分类法;面分类法又称组配分类法。

6.2 线分类法

6.2.1 方法

线分类法是将分类对象(即被划分的事物或概念)按所选定的若干个属性或特征逐次地分成相应的若干个层级的类目,并排成一个有层次的,逐渐展开的分类体系。在这个分类体系中,被划分的类目称为上位类,划分出的类目称为下位类,由一个类目直接划分出来的下一级各类目,彼此称为同位类。同位类类目之间存在着并列关系,下位类与上位类类目之间存在着隶属关系。

6.2.2 示例

GB/T 14721.1—1993《林业资源分类与代码 森林类型》是采用线分类法,并用五位数字代码进行表示的,该标准将森林类型分成三个层级,第一层级用第一、二位数字码表示森林植被型,第二层级用第三位数字表示森林类型组,第三层级用第四、五位数字表示森林类型。部分代码见表 1。

表 1

代 码	类 型 名 称
30000	经济林
31600	饮料林
31611	茶叶林
31612	咖啡林
31613	可可林
31800	鲜果林
31811	苹果林
31812	梨树林
31813	桃树林
.....

在表 1 中,经济林相对于饮料林、鲜果林为上位类类目,饮料林、鲜果林相对于经济林为下位类类

目,饮料林、鲜果林是同位类类目;同理,饮料林相对于茶叶林、咖啡林、可可林是上位类类目,茶叶林、咖啡林、可可林是饮料林的下位类类目,茶叶林、咖啡林、可可林是同位类类目。

6.2.3 要求

- 由某一上位类划分出的下位类类目的总范围应与该上位类类目范围相等;
- 当某一个上位类类目划分成若干个下位类类目时,应选择同一种划分基准;
- 同位类类目之间不交叉、不重复,并只对应于一个上位类;
- 分类要依次进行,不应有空层或加层。

6.3 面分类法

6.3.1 方法

面分类法是将所选定的分类对象的若干属性或特征视为若干个“面”,每个“面”中又可分成彼此独立的若干个类目。使用时,可根据需要将这些“面”中的类目组合在一起,形成一个复合类目。

6.3.2 示例

服装的分类可采用面分类法,选服装所用材料、男女式样、服装款式作为三个“面”,每个“面”又可分成若干个类目,见表2。

表 2

材 料	男 女 式 样	服 装 款 式
纯棉	男式	中山装
纯毛	女式	西服
中长纤维		猎装
.....	连衣裙
	

使用时,将有关类目组配起来。如纯毛男式中山装,中长纤维女式西服……等。

6.3.3 要求

- 根据需要选择分类对象本质的属性或特征作为分类对象的各个“面”;
- 不同“面”内的类目不应相互交叉,也不能重复出现;
- 每个“面”有严格的固定位置;
- “面”的选择以及位置的确定,根据实际需要而定。

6.4 混合分类法

混合分类法是将线分类法和面分类法组合使用,以其中一种分类法为主,另一种做补充的信息分类方法。

7 信息编码的基本原则

7.1 唯一性

在一个分类编码标准中,每一个编码对象仅应有一个代码,一个代码只唯一表示一个编码对象。

7.2 合理性

代码结构应与分类体系相适应。

7.3 可扩充性

代码应留有适当的后备容量,以便适应不断扩充的需要。

7.4 简明性

代码结构应尽量简单,长度尽量短,以便节省机器存储空间和减少代码的差错率。

7.5 适用性

代码应尽可能反映编码对象的特点,适用于不同的相关应用领域,支持系统集成。

7.6 规范性

在一个信息分类编码标准中,代码的类型,代码的结构以及代码的编写格式应当统一。

8 信息编码的基本方法

8.1 通则

编码方法应以预定的应用需求和编码对象的性质为基础,选择适当的代码结构。在决定代码结构的过程中,既要考虑各种代码的编码规则,又要考虑各种代码的优缺点(参见附录 A),还要分析代码的一般性特征,选取合适的代码表现形式,研究代码设计所涉及的各种因素,避免潜在的不良后果。

8.2 代码类型

图 1 根据代码的含义性(参见 8.3.2 条)给出了各种常用代码的类型。

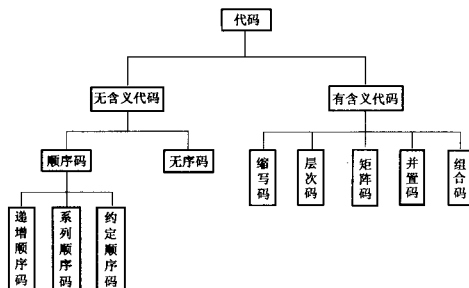


图 1

8.2.1 顺序码

8.2.1.1 规则

从一个有序的字符集中顺序地取出字符分配给各个编码对象。这些字符通常是自然数的整数,如:以“1”打头;也可以是字母字符,如:AAA、AAB、AAC,……。

8.2.1.2 应用

顺序码一般作为以标识或参照为目的的独立代码来使用,或者作为复合代码的一部分来使用,后一种情况经常附加着分类代码。

在码位固定的数字字段中,应使用零填满字段的位数直到满足码位的要求。

示例:在 3 位数字字段中,数字 1 编码为 001,而数字 15 编码为 015。

8.2.1.3 类型

顺序码有三种类型:递增顺序码、分组顺序码、约定顺序码。

8.2.1.3.1 递增顺序码

编码对象被赋予的代码值,可由预定数字递增决定。例如,预定数字可以是 1(纯递增型),或者是 10(只有 10 的倍数可以赋值),或者是其他数字(如:偶数情况下的 2),等等。

用这种方法,代码值不带有任何含义。相类似的编码对象的代码值不作分组。

为了以后原始代码集的修改,可能需要使用中间的代码值,这些中间代码值的赋值根据不必按 1 递增。

示例:GB/T 2659—2000《世界各国和地区名称代码》中,部分国家和地区的数字代码(见表 3)。

表 3

国家和地区名称	代 码
阿富汗 AFGHANISTAN	004
阿尔巴尼亚 ALBANIA	008
阿尔及利亚 ALGERIA	012
美属萨摩亚 AMERICAN SAMOA	016
安道尔 ANDORRA	020
安哥拉 ANGOLA	024

该标准中,后来增加的地区名称南极洲(ANTARCTICA)使用了中间代码值 010,属于对原始代码集的增补。

8.2.1.3.2 系列顺序码

这种代码首先要确定编码对象的类别,按各个类别确定它们的代码取值范围,然后在各类别代码取值范围内对编码对象顺序地赋予代码值。

示例:GB/T 4657—2002《中央党政机关、人民团体及其他机构代码》,就采用了三位数字的系列顺序码。

100~199 表示全国人大、全国政协、高检、高法机构

200~299 表示中央直属机关及直属事业单位

300~399 表示国务院各部委

.....

700~799 表示全国性人民团体、民主党派机关

系列顺序码只有在类别稳定并且每一具体编码对象在目前或可预见的将来不可能属于不同类别的条件下才能使用。

8.2.1.3.3 约定顺序码

约定顺序码不是一种纯顺序码。这种代码只能在全码编码对象都预先知道并且编码对象集合将不会扩展的条件下才能顺利使用。

在赋予代码值之前,编码对象应按某些特性进行排列,例如:依名称的字母顺序排序,按(事件、活动的)年代顺序排序等。这样得到的顺序再用代码值表达,而这些代码值本身也应是从有序的列表中顺序选出的。

示例:按英文字母顺序排列的数值化字母顺序码(见表 4)。

表 4

代 码	名 称
01	Apples(苹果)
02	Bananas(香蕉)
03	Cherries(樱桃)
04	Dates(枣)
.....

8.2.2 无序码

8.2.2.1 规则

无序码是将无序的自然数或字母赋予编码对象。此种代码无任何编写规律,是靠机器的随机程序编写的。

8.2.2.2 应用

无序码既可用作编码对象的自身标识,又可作为复合代码的组成部分(复合代码的其他部分则以其他编码规则为基础)。

8.2.3 缩写码

8.2.3.1 规则

这种代码的本质特性是依据统一的方法缩写编码对象的名称,由取自编码对象名称中的一个或多个字符赋值成编码表示。

8.2.3.2 应用

缩写码能有效用于那些相当稳定的、并且编码对象的名称在用户环境中已是人所共知的有限标识代码集。

示例:GB/T 2659—2000《世界各国和地区名称代码》中,部分国家的字母代码见表 5。

表 5

国家名称	代 码
奥地利 AUSTRIA	AT
加拿大 CANADA	CA
中国 CHINA	CN
法国 FRANCE	FR
美国 UNITED STATES	US

8.2.4 层次码

8.2.4.1 规则

层次码以编码对象集中的层级分类为基础,将编码对象编码成为连续且递增的组(类)。

位于较高层级上的每一个组(类)都包含并且只能包含它下面较低层级全部的组(类)。这种代码类型以每个层级上编码对象特性之间的差异为编码基础。每个层级上特性必须互不相容。

细分至较低层级的层次码实际上是较高层级代码段和较低层级代码段的复合代码。

层次码的一般结构如图 2 所示:

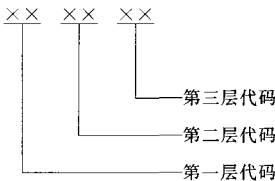


图 2

8.2.4.2 应用

层次码通常用于分类的目的。层级数目的建立依赖于信息管理的需求。层次码较少用于标识和参照的目的。

层次码非常适合于诸如统计目的、报告货物运转、基于学科的出版分类等情况。在实践中既有固定格式,也有可变格式。固定格式比可变格式更容易处理一些。

示例 1:固定递增格式。GB/T 13745—1992《学科分类与代码》中学科代码格式由 7 个数字位组成,下一级学科相对于上一级学科按固定的 2 位代码段递增,其部分代码见表 6。

表 6

代 码	学 科 名 称
110	数学
110 · 14	数理逻辑与数学基础
110 · 1410	演绎逻辑学

示例 2:可变递增格式。在通用十进制分类法(UDC)中,字符的数目和编码表达式的分段是可变的,其细节描述的程度能被延伸到想要达到的层级。“建筑学的屋顶坡度”这样一个概念可被编码表达式达成 624.024.13。

624	土木工程
624.02	建筑物成分
624.024	屋顶,屋顶用材料
624.024.13	屋顶坡度

8.2.5 矩阵码

8.2.5.1 规则

矩阵码以复式记录表的实体为基础。赋予这个表中行和列的值用于构成表内相关坐标上编码对象的代码表示。

这种方法的目的是对矩阵表中的编码对象赋予有含义的代码值,这些编码对象在不同的组合中具有若干共同特性。

8.2.5.2 应用

矩阵码可有效地用于标识那些具有良好结构和稳定特性的编码对象。

示例:GB 2312—1980《信息交换用汉字编码字符集 基本集》根据矩阵码编码方法对汉字信息交换用的基本图形字符编制了区位码,其中区号为矩阵表中的行号,位号为矩阵表中的列号。汉字字符“啊”用区位码 16—01 编码表示,在这里,16 为区号,01 为位号;同理,拉丁字符“A”用区位码 03—13 编码表示,图形字符“…”用区位码 01—13 编码表示。

8.2.6 并置码

8.2.6.1 规则

并置码是由一些代码段组成的复合代码,这些代码段提供了描绘编码对象的特性。这些特性是相互独立的。这种方法的编码表达式可以是任意类型(顺序码、缩写码、无序码)的组合。

8.2.6.2 应用

并置码非常适用于那些具有若干共同特性的商品分类。

应用代码段是要作出描绘性编码(何种产品、何时何地生产)或者是用作开发制造业方面的成组技术方法。

示例:轨道编码。

××××	××	××
等级	形状	尺寸

等级、形状和尺寸这三个特性在很大程度上是相互独立的。

8.2.7 组合码

8.2.7.1 规则

组合码也是由一些代码段组成的复合代码,这些代码段提供了编码对象的不同特性。与并置码不同的是,这些特性相互依赖并且通常具有层次关联。

8.2.7.2 应用

组合码经常被用于标识目的,以覆盖宽泛的应用领域。

示例:GB 11643—1999《公民身份号码》。

公民身份号码	含 义
×××××× ×××××××× ××××	公民身份号码的 18 位组合码结构
××××××	行政区划代码
××××××××	出生日期
×××	顺序号,其中奇数表示男性,偶数表示女性
×	校验码

整个 18 位组合码共分 4 段,前两个代码段标识了编码对象(公民)的空间和时间特性,第三个代码段则依赖于前两个代码段所限定的范围,第四个代码段依赖于前三个代码段赋值后的校验计算结果。

8.3 代码特征

8.3.1 概述

代码的一般性特征除第7章“信息编码的基本原则”所论述的唯一性、合理性、可扩充性、简明性、适用性以外,还包括:稳定性、含义性、代码长度、结构与格式、容量等特征。

8.3.2 稳定性

当代码为设计的变化留有余地而不必修改其结构时,代码就是稳定的。用户需要稳定的代码。代码值的赋值必须考虑相对于代码值自身以及代码结构作偶然修改的最小可能性。

当某个代码元素从代码元素集合中撤销时,原编码表示不应再为其他编码对象所用。

8.3.3 含义性

如果代码的编码表达式直接(例如:缩写码)表达或间接根据一个或多个表(例如:层次码、矩阵码、并置码)来表达它们的含意,则代码就被认为是有含义的。

在使用编码表达式时,有含义也与根据编码对象特性进行的归类和分组(类)有关。

在以分类为目的情况下,有含义是尤其重要的。对于以标识和参照为目的者,宜用无含义代码。

8.3.4 代码长度

代码长度是指编码表达式位置的数目。代码长度可被规定成固定的或可变的字符数目。

注:可变的代码长度有两条主要缺欠:其一是当存储代码值的数据字段所容纳的字符数比使用的代码值字符数多时,字符数目的不可预知会产生排列对齐问题。其二是由于字符冗余或增加引起的错误不能被人工或机器容易地检测出来。因此,代码长度宜使用固定的字符数目。

8.3.5 结构与格式

代码结构定义包括:构成编码表达式的位置或位置组的数目,以及每一位置上有效字符的集合。其中空格可以作为结构的组成部分。

检查语法错误的输入确认主要与结构相关。就各个位置组来说,编码表达式的每个位置都可以这样定义其格式:字母的、数字的、字母数字的、特殊字符的。

8.3.6 容量

容量是指编码表达式的数量,它是在选定的基数范围内,由每个位置上全部可用的字符组合构成的。

示例:(C表示容量)

- a) 对于位置数目是1,基数是2,使用二进制字符: $C=2$
- b) 对于位置数目是3,基数是10,使用十进制数字字符: $C=1\ 000$
- c) 对于位置数目是2,基数是26,使用字母字符: $C=676$

理论容量以全部字符的所有组合都得到使用为前提。由于实践或理论原因造成的初始限制,减少了这些理论容量。实际上,容量的抉择是在以下各因素之间折衷的结果:

- a) 对扩充系统的预见;
- b) 组成代码表达式的字符数目的限制;
- c) 书写和使用代码表达式的容易程度;
- d) 系统的期望使用寿命;
- e) 操作代价,等等。

8.4 代码表现形式

8.4.1 数字格式代码

数字格式代码是用一个或若干个阿拉伯数字表示编码对象的代码,简称为数字码。

数字码的特点是结构简单,使用方便,排序容易并且易于国内、外推广。但是对编码对象特征描述不直观。

在数字格式代码值赋值时,不宜使用全部是0或全部是9的值,如“0000”和“9999”。这些值应当保留用于特殊情形。

8.4.2 字母格式代码

字母格式代码是用一个或多个拉丁字母表示编码对象的代码,简称为字母码。

字母码的特点。其一是容量大,如用二位拉丁字母代码最多可表示 $676(26^2)$ 个类目,而二位数字代码最多只可表示 $100(10^2)$ 个类目。其二是字母码有时可提供便于人们识别的信息。如在 GB/T 2260 中, BJ 表示北京; TJ 表示天津。

字母码便于人们记忆,但不便于机器处理信息,特别是当编码对象数目较多或添加、更改频繁以及编码对象名称较长时,常常会出现重复和冲突的现象。因此,字母码常用于编码对象较少的情况。

为字母格式代码赋值时,应注意:

- 无含义字母码应当避免采用那些在发音时可能引起混淆的字符(听觉上的相似性);例如:字母 B、D、G、P 和 T,或者字母 M 和 N。
- 在字母代码中,或者在代码的一部分有 3 个或更多的连续字母字符时,要避免使用元音字母(A、E、I、O 和 U),以免无意间形成易被误认的简单语言单词。
- 在同一编码方案中,字母代码宜使用单一形式的大写或小写字母,而不宜大小写字母混用。

8.4.3 混合格式代码

混合格式代码是由数字、字母组成的代码,或由数字、字母、特殊字符组成的代码。可以简称为字母数字码或数字字母码。

混合格式代码的特点是基本兼有了数字型代码、字母型代码的优点,结构严密,具有良好的直观性,同时又有使用上的习惯。但是,由于代码组成格式复杂也带来了一定的缺点,即计算机输入不方便,录入效率低,错误率增高,不便于机器处理。

8.4.4 特殊字符

特殊字符(如: &, @, ……)可以用于数字与字母混合格式代码中以补充字母系统的字符;用这种方法,容量得到增加,并且可以为特殊处理保留语种字符的有效字符。

在代码结构中应使用常用的字符,并且应避免那些非字母或数字的字符(例如:连字符、句号、间隔、星号,等等),只是在分隔代码段时,才可以使用连字符或空格。用于规定代码系统的词表应当只含有尽可能少的字符种类。

下列字符应避免使用:

- 不属于 GB/T 1988 七位编码字符集的字符。
- 可能引起曲解或不正确转录的字符。例如:应注意排除空格,“123 ABC”应写成“123ABC”,因为空格没有含义,并且空格在转录时可能被忽略。
- 对于数据交换来说,在语法结构中可被当作服务性字符使用的那些字符。例如:冒号(:)、加号(+)、问号(?)、星号(*)、撇号(')在 GB/T 14805 标准中是被当作服务字符使用的,应避免使用这类字符。

8.4.5 代码格式规则

代码值的格式(或字符结构)最好采用全数字或全字母格式。只有在特殊位置上(例如:首位或末位)始终要用字母或数字格式时,才能使用字母数字混合格式,而随机的字母数字格式则不宜使用。

在不存在助记特性的情况下,人工记录数字格式的代码值通常比记录字母格式或混合格式的代码值要更加可靠些。受控的混合格式代码值(例如:在确定的位置上永远采用字母格式或者永远采用数字格式)比随机的混合格式代码值更加可靠些。例如:AA999(前两位字符永远采用字母格式,后三位字符永远采用数字格式)就比字母或数字有可能出现在任意位置上的情形具有更加可靠的格式。

在混合格式中,同类的字符类型应当当作分组处理并且不要分散于代码表达式的各个位置上。例如:在三位字符代码中,“字母—字母—数字”的结构(如:HW5)就比“字母—数字—字母”这样的顺序(如:H5W)所发生错误的要少很多。

当需要使用字母数字混合代码结构时,应当避免那些容易理解成其他字符或者容易同其他字符相

混淆的字符。例如：字母 I 与数字 1、字母 O 与数字 0、字母 Z 与数字 2、字母 G 与数字 6、字母 B 和 S 与数字 8，以及字母 O 与 Q。

为了避免对照排序时互不相容，任何特定字符的位置上应当要么只用字母，要么只用数字。

8.4.6 编码表达式的显示

对于手工处理，宜优先采用人工易读的编码显示方式。在这种情况下，代码值将以拉丁字母和阿拉伯数字方式出现。这种表达方式也常用于计算机输出的纸质文件和表册当中。

当需要采用机械或电子方式进行处理时，应采用易于自动识别的编码显示方式。其中，以若干个条排列编码成符号表示的条码编码方法得到了广泛使用。此外，其他自动化标识方法，如光学字符识读(OCR)设备或磁条、集成电路的智能卡等在实践中也已得到了使用。

8.5 代码设计

8.5.1 概述

代码设计过程中，应注意那些常常可能造成彼此相互冲突的要求。例如：如果一种代码结构对于未来的需要有充足的扩充能力，那么它就会在某种程度上牺牲其简明性。因此，每个方面的问题都必须考虑周全，制定折衷办法，以达到相关应用领域获得最佳效率。

代码分组和分段应当根据用户对信息的需求作格式安排，要考虑在准确性和完备性方面进行查看的最大限度宽松性，以及数据内容的紧凑性。

8.5.2 现有代码的使用

宜使用现有的代码。如果不是绝对需要，就不必设计新的代码。

8.5.3 代码含义

在使用恰当时，有含义代码为附加信息提供了一个基础，并且在人工使用方面比无含义代码更加容易、更为可靠些。然而，在有含义代码的开发过程中应当谨慎，以确保有含义的部分与稳定的实体相关联。例如，当地点的改变将会引起代码的改变时，某个组织的有含义代码就不宜与地点相关联。

无含义代码宜用于大多数标识目的以及所有的参照目的。

8.5.4 代码字符数目的确定

代码值应当由最少的字符数目组成以节省空间并减少数据通信时间，但同时还应根据代码用户的能力进行优化。

固定长度代码(例如：只采用三位字符，而不是一位、二位和三位字符同时混用)在使用上比可变长度代码更加可靠且更加容易。

为了记录的可靠性，多于 4 位字母字符或 5 位数字字符的代码值宜分解成较小的代码段，例如：×××-×××-××××就比××××××××××更为可靠。

在不必对已有代码元素重新编码或者扩大编码表达式格式的前提下，代码结构应当能为代码集合增添新的代码元素提供支持。

8.5.5 代码段的分隔

如果位置或代码段是完全相互独立并且能够独自成立(即：对于它们的含义来说，不需要其他的代码)，代码段应能被连字符(当需要显示时)所隔离。

8.5.6 代码的位置顺序

如果一个编码方案把一个完整实体集分成比较小的分组，那么高阶位置应当是显著的、全面的分类；低阶位置应当最具选择性和差别性(包括后缀)。一个例子就是 GB/T 7408 规定的日期数字表达式(YYYYMMDD)。如果一个复合代码被设计成由两个或更多的独立代码段组成，则出现在高阶位置上特有的代码段应当是基于惯用要求和处理效率来考虑的。

8.5.7 代码命名

代码或其各个所有独立的代码段都必须有自己的标准化的、唯一的、与应用标志相适应的命名。

8.5.8 代码容量计算

在计算涵盖全部位置的给定代码容量并且要保持代码唯一性时,应使用下列公式(假定使用 24 个字母字符和 10 个数值数字,因为要避免使用字母 I 和 O 可能引起的混淆):

$$C = 24^A \cdot 10^N$$

式中:

C——全部可能的有效代码组合数,即容量;

A——代码中字母位置的数目;

N——代码中数字位置的数目。

(在组合的情况下, $A+N$ 等于代码的全部位置数目)。

注:上面的公式假定给定的位置要么是字母的,要么是数字的,但决不是二者都适用。如果特定的位置允许字母字符和数字字符二者都适用,则公式变成:

$$C = 26^A \cdot 10^N \cdot 36^M \text{ 或}$$

$$C = 24^A \cdot 10^N \cdot 34^M \text{ (当字母 I 和 O 被禁用时)}$$

式中: M 为代码中字母字符和数字字符二者都适用的位置数目, $A+N+M$ 等于代码的全部位置数目。

在计算容量时,不应考虑校验码所占的位置。

8.5.9 校验码

为了避免抄录和键入过程中的错误,当代码较长时,应考虑设置校验码。校验码由构成编码表达式的字符经过一定的算术运算而得到,它可以检测出以下类型的错误:

- 单替代错误:一个单一字符被另一个单一字符替换;
- 单一对换错误:单个字符的对换,相邻的($d=1$)两个字符或相隔一个字符的($d=2$)两个字符之间的互换错误;
- 双替代错误:在同一个编码表达式中,两个分隔的单一字符的替换错误;
- 位移错误:编码表达式整体向左或向右的位移;
- 其他错误。

参见 GB/T 17710。

8.6 代码赋值约定

8.6.1 赋码规则

赋码规则应叙述清晰并且具有一致的适用性。例如:一个助记缩写词可以通过从编码条目的名称中删除全部元音而形成,象“日期(date)”编码为 DT 或“绿色(green)”编码为 GRN;也可以用构成元素的各个单词的第一个字母编码而成,象“文件结束(End of File)”编码为 EOF。

8.6.2 定义数据

数量或货币数额不宜赋码。例如:当数量 1~99 能被编码为 A, 100~199 被编码为 B 时,就会失去统计价值,因为一旦数字被编码,就不能得到真实的数字了。分类可以放在数据处理靠后的阶段进行,而不是放在输入数据预先编码的过程中进行。

8.6.3 “自然”数据的使用

假如具体的数据以其自然形态(例如:具体的百分比数量)就已经是适当并且够用的话,那么就不宜再为其开发代码结构。

8.6.4 收容类的使用

应注意辨别代码的类别是“混杂”类或是“其他”类。不宜在这样的类别中放置那些实际上是属于另外一个具体类别的实体代码元素。

附 录 A
(资料性附录)
各种信息分类编码方法的优缺点

A.1 信息分类方法优缺点

信息分类方法优缺点见表 A.1。

表 A.1

分类方法	优 点	缺 点
线分类法	<ul style="list-style-type: none"> ——层次性好,能较好地反映类目之间的逻辑关系; ——实用方便,既符合手工处理信息的传统习惯,又便于电子计算机处理信息 	<ul style="list-style-type: none"> ——结构弹性较差,分类结构一经确定,不易改动; ——效率较低,当分类层次较多时,代码位数较长,影响数据处理的速度
面分类法	<ul style="list-style-type: none"> ——具有较大的弹性,一个“面”内类目的改变,不会影响其他的“面”; ——适应性强,可根据需要组成任何类目,同时也便于机器处理信息; ——易于添加和修改类目 	<ul style="list-style-type: none"> ——不能充分利用容量,可组配的类目很多,但有时实际应用的类目不多; ——难于手工处理信息

A.2 各种类型代码编码方法优缺点

各种类型代码编码方法优缺点见表 A.2。

表 A.2

代码类型	优 点	缺 点
递增顺序码	<ul style="list-style-type: none"> ——能快速赋予代码值; ——简明; ——编码表达式容易确认 	<ul style="list-style-type: none"> ——编码对象的分类或分组不能由编码表达式来决定; ——不能充分利用最大容量
系列顺序码	<ul style="list-style-type: none"> ——能快速赋予代码值; ——简明; ——编码表达式容易确认 	<ul style="list-style-type: none"> ——不能充分利用最大容量
约定顺序码	<ul style="list-style-type: none"> ——能快速赋予代码值; ——简明; ——编码表达式容易确认 	<ul style="list-style-type: none"> ——不能充分利用最大容量; ——不能适应于将来可能的进一步扩展
无序码	<ul style="list-style-type: none"> ——容易并且快速赋予代码值,或许还是自动化的; ——简明; ——可利用最大容量 	<ul style="list-style-type: none"> ——编码对象的分类或分组不能依据编码表达式显示出来; ——如果要排除号码的复制,需要用某种预先设定的表或运算法则产生随机数
缩写码	<ul style="list-style-type: none"> ——用户容易记忆代码值,从而避免频繁查阅代码表; ——可以压缩冗长的数据长度 	<ul style="list-style-type: none"> ——编码依赖编码对象的初始表达(语言、度量系统,等等)方法; ——在每次增加代码值之后,如果不重新检查全部的代码值,则缩写过程的结果就不能保证代码值的唯一性

表 A.2 (续)

代码类型	优 点	缺 点
层次码	<ul style="list-style-type: none"> ——易于编码对象的分类或分组； ——能在较高的合计层级上汇总； ——代码值可以解释 	<ul style="list-style-type: none"> ——限制了理论容量的利用； 因精密原则而缺乏弹性； ——需要随代码层级的顺序从最高层级向下赋予代码值或者解释代码值； ——复杂性，它取决于层级数目，并导致要重新介绍已经应用于较高级上的特性
矩阵码	<ul style="list-style-type: none"> ——代码值可以解释； ——代码值容易赋予 	<ul style="list-style-type: none"> ——需要预先建立表，覆盖编码对象的全部特性； ——难于适应新的要求，诸如新的或更改的特性、以及新的组合等等
并置码	<ul style="list-style-type: none"> ——以代码值中表现出一个或多个特性为基础，可以对编码对象容易地进行分组； ——容量与每个特性可能带有的值的数量相联系； ——代码值可以解释 	<ul style="list-style-type: none"> ——因需要含有大量的特性可导致每个代码值有许多字符； ——难于适应新特性的要求
组合码	<ul style="list-style-type: none"> ——代码值容易赋予； ——有助于配置和维护代码值； ——能够在相当程度上解释代码值； ——有助于确认代码值 	<ul style="list-style-type: none"> ——理论容量不能充分利用