# 词法结构

本章介绍几个基本Modelica构造块，如字符以及标识符、字面量等词法单元。毫无疑问，Modelica最小构造块是某个字符集的单个字符。字符组合成词法单元，也称作符号。这些符号由Modelica翻译器的词法分析部分进行检测。符号的例子有字符常量，标识符和运算符等。注释不是真正的词法单元，既然最终它们会被丢弃，换句话说，注释在丢弃之前由词法分析器检测。

这里展示的信息源于附录B中更正式的规范。

## 2.1 字符集

Modelica语言字符集是Unicode字符集，但有几个地方地方将Unicode字符限制为相应的7位ASCII字符。详情参见附录B.1。

## 2.2 注释

Modelica语言有两种注释方式，注释不是其词法单元，因此Modelica翻译器将其处理为空白字符。空白字符是指空格，制表符和行分隔符（回车换行符），空白字符不能出现在符号中，如 <= 必须写作中间没有空格和注释的两个字符。【*注释语法与C++是一样的*】。下面是可以使用的注释方式：

// 注释 从 // 开始到行末尾的字符都被忽略。

/\* 注释 \*/ /\* 和 \*/ 之间的字符被忽略，包括行终止符。

Modelica注释不能嵌套，也就是说，/\* \*/不能嵌入到/\* \*/之中。以下注释是*非法的*：

/\* 添加注释 – 错误的注释， 嵌套注释是非法的！

/\* 这是一个有趣的模型 \*/

model interesting

...

end interesting;

\*/

还有一种“文档注释”，实际上是一个*文档编制字符串*，它是Modelica语言的一个部分，因此不会被Modelica编译器忽略掉。这种“注释”可以出现在声明、方程或者语句的后面，或者类定义开始的地方。例如：

model TempResistor "Temperature dependent resistor"

...

parameter Real R "Resistance for reference temp.";

...

end TempResistor;

## 2.3 标识符，名称和关键字

*标识符*是一连串的字母、数字和其他字符如下划线，用于命名Modelica语言中的各种项目。一些特定的字母组合是*关键字*，表示是Modelica语法中的保留字且不能用于标识符中。

### 2.3.1 标识符

Modelica标识符用于命名类、变量、常量和其他的项。有两种形式的标识符，第一种形式始终以字母或者下划线（“\_”）开头，后面跟任意数量的字母、数字或者下划线。大小写很重要，也就是说，名称为Inductor和inductor是不同的项。第二种形式（Q-IDENT）以单引号开头，后面跟一连串可打印的ASCII字母，其中单引号前面必须是反斜杠，并且以一个单引号结束，例如'12H'，'13\'H'， '+foo'。标识符中带引号的的控制字符必须使用字符转义。下面是类BNF的Modelica标识符定义规则，其中花括号（“{}”）表明重复0次或者多次，竖线（“|”）表明可替代项。在附录B中有完整的Modelica语法和词法单元BNF定义。

IDENT = NONDIGIT { DIGIT | NONDIGIT } | Q-IDENT

Q-IDENT = "’" { Q-CHAR | S-ESCAPE } "’"

NONDIGIT = "\_" | letters "a" to "z" | letters "A" to "Z"

DIGIT = 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

Q-CHAR = NONDIGIT | DIGIT | "!" | "#" | "$" | "%" | "&" | "(" | ")" | "\*" | "+" | "，" |

"-" | "." | "/" | ":" | ";" | "<" | ">" | "=" | "?" | "@" | "[" | "]" | "^" |

"{" | "}" | "|" | "~" | " "\_

S-ESCAPE = "\’" | "\"" | "\?" | "\\" |

"\a" | "\b" | "\f" | "\n" | "\r" | "\t" | "\v"

### 2.3.2 名称

名称是有特定解释和含义的标识符。例如，一个名称指的可能是一个Integer类型变量，一个Real类型变量，一个函数，一种类型等等。在代码的不同部分，也就是说不同的范围，一个名称可能有不同的含义。标识符作为名称在第5章作更详细的解释。包名称的含义在第13章有更详细的介绍。

### 2.3.3 Modelica关键字

除了附录B.1列出的关键字之外，下表中的Modelica*关键字*是保留字且不能用做标识符：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| algorithm | discrete | False | loop | Pure |
| And | Each | Final | model | Record |
| annotation | Else | Flow | not | redeclare |
|  | Elseif | For | operator | replaceable |
| block | elsewhen | Function | or | Return |
| break | encapsulated | If | outer | Stream |
| class | End | Import | output | Then |
| connect | enumeration | Impure | package | True |
| connector | equation | In | parameter | Type |
| constant | expandable | Initial | partial | when |
| constrainedby | extends | Inner | protected | while |
| Der | external | Input | public | within |

## 2.4 字面常量

字面常量是未命名的常量，根据其类型的不同其形式也不一样。每一种预定义的Modelica类型都有其相应的未命名常量的表达方式，这在接下来的章节中会一一介绍。此外，还可以表达数组字面量和记录字面量。

### 2.4.1 浮点数

浮点数表示成可选的小数点或者指数后跟一连串小数位数的十进制数字形式。浮点数至少有一个数字。指数用E或e表示，后面跟一个可选的符号（+或者-）以及一位或者多位小数位数。浮点数最小推荐范围是IEEE规定的双精度浮点数，其可表达的最大正数是1.7976931348623157E+308，最小正数是2.2250738585072014E−308。例如，以下是浮点数字面常量：

22.5，3.141592653589793，1.2E-35

同一个浮点数可用不同的字面量表示，例如，下面所有的字面量表示相同的数字：

13.，13E0，1.3e1，0.13E2

### 2.4.2 整型字面量

整型字面量是一连串十进制数，例如整数33，0，100，30030044。【*负数由一元减号后跟一个整型字面量构成*】。推荐的最小数值推荐范围是从-2147483648到+2147483647，即32位整数二进制补码实现。

### 2.4.3 布尔型字面量

布尔型字面量有true和false两个值。

### 2.4.4 字符串

字符串常量出现在成对双引号之间。除了双引号（”）和反斜杠以及换行符，Modelica语言字符集中的任意字符，能够*直接*包含在一个字符串中而不需要使用转义码。字符串中某些字符可以使用转义码来表示，也就是说，在字符串中，该字符前面以一个反斜杠引导。这些字符是：

\' 单引号—在字符串常量中出现时可能没有反斜杠

\" 双引号

\? 问号—在字符串常量中出现时可能没有反斜杠

\\ 反斜杠

\a 警告（铃声，代码7，ctrl-G）

\b 退格（代码8，ctrl-H）

\f 换页（代码12，ctrl-L）

\n 换行（代码10，ctrl-J）

\r 回车（代码13， ctrl-M）

\t 水平制表（代码9，ctrl-I）

\v 垂直制表（代码11，ctrl-K）

例如，一个包含一个制表符、词语：this is、双引号、空格、词语：between、双引号、空格、词语：us，和一个换行符的字符字面量应表示为："\tThis is\" between\" us\n"

在某种情况下，字符字面量的连接（参见Modelica文法）用“+”运算符表示，例如"a" + "b" 变成 "ab"。这在需要分多行来表示长字符串字面量的时候很有用。

【*注意，如果一个文件的内容读入到Modelica字符串中，假定读取函数负责处理文件中的不同的换行符（例如在Linux系统中，一行的行尾有一个“换行”字符，而windows系统中则有一个“换行”字符和一个“回车”字符。与编程语言中一样，存储文件内容的Modelica字符串中只包含“换行”字符。*

*对于长字符串注释，例如存储模型文档的“信息”注释，如果文档的每一行都需要用一个字符串连接运算符，这将非常不方便。当浏览或者编辑一个字符串字面量的时候，这里假定一个Modelica工具支持非打印“换行”字符，例如，下面的声明定义了一个字符串，它包含（非打印）换行字符：*

*assert(noEvent(length > s\_small)， "The distance between the origin of frame\_a and the origin of frame\_b of a LineForceWithMass component became smaller as parameter s\_small (= a small number， defined in the \"Advanced\" menu). The distance is set to s\_small， although it is smaller， to avoid a division by zero when computing the direction of the line force."， level = AssertionLevel.warning);*

】

## 2.5 运算符

预定义运算符的规范定义在第255页，并汇总在第3.2节运算符表中。