

IMI 技術仕様書

IMI 語彙記法

バージョン 1.0.1

2019 年 2 月 28 日

1 はじめに

IMI 共通語彙基盤では RDF、XML 等、多様な形式・フォーマットで利用可能な語彙を提供することを目的としている。このため、語彙自体は特定の利用形式に依存しない中立的な形式で記述する必要がある。IMI 語彙記法は、これを実現するための記法である。IMI 語彙記法では語彙の中の用語の定義を行う。IMI 語彙記法で定義された語彙は RDF や XML など必要な形式に変換することができる。

本仕様書において、例及び参考は参考情報であり規定ではない。

2 概要

IMI 語彙記法は、IMI 語彙及びデータモデルを記述するための標準的な記法である。IMI 語彙記法では、構造化項目名記法をより厳密な記述ができるように拡張した「厳密構造化項目名」を用いている。「厳密構造化項目名」は、構造化項目名記法と多くの部分を共有しているため、構造化項目名記法を用いた表と語彙との関係性から容易に、IMI 語彙やデータモデルの作成することが可能となる。

3 基本的な概念

3.1 IMI 定義文書

IMI 語彙記法に従って記述された文書を「**IMI 定義文書**」と呼ぶ。IMI 定義文書には、複数の語彙及び一つのデータモデルを含むことができる。

3.2 語彙

語彙は、データを構造的に構成する際に用いられる用語の集合である。語彙は URI を用いた識別子によって識別される。用語には物事や事象を表現するクラス用語（3.3）とクラス用語が表現する物事や事象の特性を表すプロパティ用語（3.4）がある。また、語彙は用語の集合の定義に加えて、名称、説明、作成者、ライセンス等の語彙自体を説明するメタデータをもつ。

3.3 クラス用語

クラス用語は、「人」、「組織」及び「活動」といった物事や事象を表現するための用語である。クラス用語は URI によって識別される。クラス用語は、それが表現する物事や事象の特性を表すために用いられる一連のプロパティ用語（3.4）をもつ。これら一連のプロパティ用語全体を、そのクラス用語の「**プロパティ用語集合**」と呼ぶ。プロパティ用語集合に含まれるプロパティ用語に対して、それぞれが使用できる回数が指定される。例えば、「組織」の「設立日」は一つしかないが、「住所」は複数になることがある。

また、クラス用語は名称や説明等のクラス用語自体を説明するメタデータをもつ。

3.3.1 クラス用語の継承

定義しようとするクラス用語が他のクラス用語が表す物事や事象をより特化させたものである場合、他のクラス用語を継承することができる。例えば、「法人」は「組織」を特化させたものと考えられるので、「組織」というクラス用語を継承して「法人」というクラス用語を定義することができる。他のクラス用語を継承したクラス用語は、継承元のクラス用語がもつ全てのプロパティ用語を継承するとともに、独自のプロパティ用語を加えたものをプロパティ用語集合としてもつ。このように明示的に継承することを**直接継承**するという。

継承元のクラス用語がさらに他のクラス用語を継承している場合もある。継承元のクラス用語を辿ると最終的に継承元をもたないクラス用語に到達するが、この過程で継承元として出現するすべてのクラス用語を合わせて最初のクラス用語の**継承元クラス用語列**と呼ぶ。クラス用語は、そのクラス用語の継承元クラス用語列に含まれるすべてのクラス用語を継承していることになる。

例えば、3つのクラス用語 C1、C2、C3 があり、

- クラス用語 C1 は、他のクラスを継承しない
- クラス用語 C2 は、クラス用語 C1 を直接継承
- クラス用語 C3 は、クラス用語 C2 を直接継承

のような継承関係になっている場合、次のことがいえる。

- クラス用語 C3 は、クラス用語 C2 を直接継承
- クラス用語 C3 は、クラス用語 C1 を継承
- クラス用語 C2 は、クラス用語 C1 を直接継承
- クラス用語 C3 の継承元クラス用語列は、C2 及び C1 を含む
- クラス用語 C2 の継承元クラス用語列は、C1 のみを含む

3.4 プロパティ用語

プロパティ用語は、クラス用語が表現する物事や事象の特性を表すために使用される用語である。プロパティ用語は URI によって識別される。プロパティ用語が表現する特性は、単純な数値や文字列で表現される場合と、クラス用語で表現される場合がある。例えば、「施設」の「概要」のような特性は、単純な文字列によって表現されるが、「地理座標」のような特性は、緯度や経度など複数の値により表される。このような場合はクラス用語を用いて表現される。プロパティ用語がどのような値又はクラス用語で表現されるかをプロパティ用語の「**値型**」と呼ぶ。値型には、数値や文字列などの単純な値の型又はクラス用語が指定できる。あるプロパティの値型があるクラス用語である場合、そのクラス用語を継承したクラス用語も、そのプロパティ用語の値となることができる。

例えば、クラス用語 C2 がクラス用語 C1 を継承していて、プロパティ用語 P1 の値型が C1 のとき、C1 だけでなく C2 もプロパティ用語 P1 の値として使用できる。

また、プロパティ用語は名称や説明等のプロパティ用語自体を説明するメタデータをもつ。

3.5 データモデル

データモデルは、語彙を利用して特定のデータの構造を定義するものである。データモデルでは、実際に使用するクラス用語やプロパティ用語を選択することに加え、各用語に関して語彙の定義の範囲内において、とりうる値の限定や、プロパティ用語の使用可能回数の制限を行うことができる。

また、データモデルは名称や説明等のデータモデル自体を説明するメタデータをもつ。

4 語彙記法リーダーと解釈

IMI 語彙記法は機械可読であり IMI 語彙記法で記述された語彙の定義は、HTML に変換したり、RDF 等へのシリアル化を行う際に利用することができる。このように、IMI 語彙記法で記述されたファイルを読み込み、解釈するソフトウェア等を「語彙記法リーダー」と呼ぶ。本仕様書では、語彙記法リーダーの出力の形式については規定しない。

4.1 エラーと警告

語彙記法リーダーは、語彙記法の解析処理中に文法の誤りや意味的な不整合などの問題に遭遇した場合は、問題の程度に応じてエラー又は警告を通知する。文法の誤りや処理の継続が不可能な意味的問題に対しては、エラーを通知し解析処理を終了する。意味的な矛盾があるが、無視して処理を継続可能な場合には、警告を通知し、解析処理を続行する。語彙記法リーダーはエラーや警告を適切な方法でユーザーに通知する必要がある。

5 IMI 語彙記法の文法

5.1 表記法

IMI 語彙記法の文法は、字句規則及び構文規則により定義される。字句規則及び構文規則を合わせて「文法規則」と呼ぶ。字句規則は、言語の最小字句要素である字句を構成する文字の組み合わせを定義する。構文規則は、言語を構成する字句の組み合わせを定義する。

対象となる文字の並びが複数の規則に一致する場合は、最も長く一致する規則を選択する。

字句規則及び構文規則に用いる構文要素には、IMI 語彙定義内にそのまま記述される終端記号、字句規則にしたがった文字列として記述される非終端記号、及び、文字の 3 種類がある。

終端記号には、キーワードとキーワード以外の 2 種類がある。

構文規則において、キーワード（5.2.2）は”@”の後ろに二重引用符（”）で囲んで表記する。

次の例において、@”vocabulary”はキーワードである。

例 1:

```
« 語彙定義 » := @”vocabulary” « 文字列リテラル »
```

キーワード以外の終端記号は二重引用符（”）で囲んで表記する。非終端記号は« » で囲んで表記する。

例えば、次の例において、« 値開下限制約 » 及び« 整数リテラル » は非終端記号で、”>” は終端記号である。

例 2:

```
« 値開下限制約 » := ">" « 整数リテラル »
```

文字は、Unicode 文字表記（5.1.8）又は一重引用符（'）で囲んで表記する。

字句規則の定義では、”::=”の左辺に、定義する非終端記号を、右辺に左辺の非終端記号の定義となる非終端記号及び文字の列を配置する。

構文規則の定義では、":=" の左辺に、定義する非終端記号を、右辺にその定義となる非終端記号及び終端記号の列を配置する。

構文規則における右辺の字句と字句との間には任意の数の《空白文字》、《コメント》、及び、《行コメント》を置くことができるが、これらの構文要素は構文規則の表記上は省略する。例えば、上記の例は、次の例の構文規則と同じ意味となり、IMI 語彙記法では、">" と《整数リテラル》の間に任意の数の《空白文字》、《コメント》、又は、《行コメント》を置くことができる。

例 3:

```
《値開下限制約》 := ">" ( 《空白文字》 | 《コメント》 | 《行コメント》 ) * 《整数リテラル》
```

文法規則の表記には 5.1.1 から 5.1.8 で説明する表記法を用いる

5.1.1 0 回以上の繰り返し

"*" は、直前の文字、字句又はグループを 0 回以上繰り返すことを意味する。

次の例は、《識別子》が、《識別子先頭文字》の後ろに《識別子文字》が 0 個以上続く文字列であることを意味する。

例 4:

```
《識別子》 ::= 《識別子先頭文字》 《識別子文字》 *
```

5.1.2 1 回以上の繰り返し

"+" は、直前の文字、字句又はグループを 1 回以上繰り返すことを意味する。

次の例は、《整数リテラル》が、《数字》が 1 個以上続く文字列であることを意味する。

例 5:

```
《整数リテラル》 ::= 《数字》 +
```

5.1.3 省略可

"?" は、直前の文字、字句又はグループが省略可能であることを意味する。

次の例は、《クラス定義》において、@"deprecated" は省略可能であることを意味する。

例 6:

```
《クラス定義》 := @"deprecated"? @"class" 《クラス用語名》 ( "{" 《型制約》 "}" ) ?
```

5.1.4 選択

複数の文字、字句または字句の列を "|" によって区切り "(" 及び ")" で囲んだものを選択という。選択は、"|" で区切られた複数の文字、字句又は字句の列から一つを選択することを意味する。右辺全体が選択となる場合は、 "(" 及び ")" は省略する。

例 7:

```
《回数制約》 := 《整数リテラル》 ".." ( 《整数リテラル》 | "n" )
```

上の例は、《回数制約》が、次のいずれかであることを意味する。

```
《 整数リテラル 》 "... " 《 整数リテラル 》
《 整数リテラル 》 "... " "n"
```

5.1.5 グループ

字句又は字句の列を“(”と”)”で囲んだものをグループという。グループは、繰り返し等の表記において、グループ全体が一つの字句のように扱われることを意味する。“(”及び”)”で囲まれた選択（5.1.4）、文字範囲選択（5.1.7）、除外（5.1.6）もグループと同様に扱われる。

5.1.6 除外

字句又は文字の列の後ろに、“-”に続けて字句、文字、又は、選択（5.1.4）を記述し、“(”及び”)”でその全体を囲んだものを除外という。除外は、“-”の右側が字句、又は、文字の場合、“-”の左側の字句又は文字の列のうち、“-”の右側の字句、又は、文字を含まないものであることを意味する。“-”の右側が選択の場合は、“-”の左側の字句又は文字の列のうち、選択に含まれる全ての文字、又は、字句を含まないものであることを意味する。選択に字句の列を指定することはできない。

次の例は、《文字》が0回以上繰り返す文字の列であるが、“*/”を含まないことを意味する。

例 8:

```
( 《 文字 》 * - "*/" )
```

5.1.7 文字範囲選択

文字と文字との間を“-”でつなぎ、“[”及び”]”でその全体を囲んだものを文字範囲選択という。文字範囲選択は、Unicode コードポイントが、“-”の左の文字の Unicode コードポイント以上で、“-”の右の文字の Unicode コードポイント以下である文字の中から一つの文字を選択することを意味する。

次の例では、《数字》が、'0'、'1'、'2'、'3'、'4'、'5'、'6'、'7'、'8'、及び、'9' の中の一つであることを意味する。

例 9:

```
《 数字 》 ::= ['0'-'9']
```

5.1.8 Unicode 文字表記

“U+”に続けて4桁から6桁の16進数を記述したものをUnicode文字表記という。Unicode文字表記は、指定された16進数で表される値がUnicodeコードポイントである文字を意味する。

例えば、U+0041 は、'A'を表す。

5.2 字句

字句には、識別子、キーワード、名前空間プレフィックス指定子、リテラル、及び、記号列がある。

5.2.1 識別子

識別子は、基本的に XML(<http://www.w3.org/TR/xml/>) の nameChars の規定と同様な字句構造をもつ。

キーワード（5.2.2）は識別子から除外する。

■5.2.1.1 文法

```

« 識別子 » ::= « 識別子先頭文字 » « 識別子文字 » *

« 識別子先頭文字 » ::= ['A'-'Z'] | '_' | ['a'-'z'] | [U+00C0-U+00D6] | [U+00D8-U+00F6] | [U+00F8-U ...
↳ +02FF] | [U+0370-U+037D] | [U+037F-U+1FFF] | [U+200C-U+200D] | [U+2070-U+218F] | [U+2C00-U+2 ...
↳ FEF] | [U+3001-U+D7FF] | [U+F900-U+FDCE] | [U+FDFF-U+FFFF] | [U+10000-U+FFFFFF]

« 識別子文字 » ::= « 識別子先頭文字 » | '-' | '.' | ['0'-'9'] | U+00B7 | [U+0300-U+036F] | [U+203F ...
↳ -U+2040]

```

5.2.2 キーワード

キーワードとは、識別子と同じ字句構造をもつが、その文字列が次の表に一致するものである。

表 1: キーワード

キーワード

deprecated

class

property

set

use

vocabulary

datamodel

5.2.3 名前空間プレフィックス指定子

名前空間プレフィックス指定子は、識別子（5.2.1）と同様の字句構造をもつがコロン（:）で終了するものである。

■5.2.3.1 文法

```

« 名前空間プレフィックス指定子 » ::= « 識別子先頭文字 » « 識別子文字 » * U+003A

```

5.2.4 リテラル

IMI 語彙記法では、整数リテラル、文字列リテラル、グループ名リテラル及びパターンリテラルの4種類のリテラルがある。また、文字列リテラルには、二重引用符（"）で囲む二重引用符文字列リテラルと引用符（'）で囲む引用符文

字列リテラルがある。

■5.2.4.1 整数リテラル

整数リテラルは、整数値を表すリテラルで、数字の列は10進数の値として解釈される。

5.2.4.1.1 文法

```
« 整数リテラル » ::= « 数字 » +
```

■5.2.4.2 二重引用符文字列リテラル

二重引用符文字列リテラルは文字列を表すリテラルで、二重引用符（”）で囲まれた文字の列がそのまま値となる。ただし、「二重引用符エスケープ」が含まれている場合、U+005C U+0022 は、U+0022（”）一文字を表し、U+005C U+005C は、U+005C（\）一文字を表す。

5.2.4.2.1 文法

```
« 二重引用符文字列リテラル » ::= U+0022 ( ( « 文字 » - ( U+0022 | U+005C )) « 二重引用符エスケープ » ) * U+0022

« 二重引用符エスケープ » ::= ( U+005C U+0022 | U+005C U+005C )
```

■5.2.4.3 引用符文字列リテラル

引用符文字列リテラルは文字列を表すリテラルで、引用符（'）で囲まれた文字の列がそのまま値となる。ただし、「引用符エスケープ」が含まれている場合、U+005C U+0027 は、U+0027（'）一文字を表し、U+005C U+005C は、U+005C（\）一文字を表す。

5.2.4.3.1 文法

```
« 引用符文字列リテラル » ::= U+0027 ( ( « 文字 » - ( U+0027 | U+005C )) « 引用符エスケープ » ) * U+0027

« 引用符エスケープ » ::= ( U+005C U+0027 | U+005C U+005C )
```

■5.2.4.4 グループ名リテラル

グループ名リテラルはグループ名を表すリテラルで、「[」及び「]」で囲まれた文字の列がそのまま値となる。グループ名リテラルに「]」を含めることはできない。

5.2.4.4.1 文法

```
« グループ名リテラル » ::= U+005B ( « 文字 » - U+005D ) * U+005D
```

■5.2.4.5 パターンリテラル

パターンリテラルは正規表現を表すリテラルである。パターンリテラルによる正規表現の構文は、XML Schema Part 2: Datatypes Second Edition に従うものとする。パターンリテラルでは、U+002F (/) によって囲まれた文字列がそのまま値となる。ただし、「パターンエスケープ」が含まれる場合、「パターンエスケープ」は、U+002F (/) 一文字として扱われる。

5.2.4.5.1 文法

```
« パターンリテラル » ::= U+002F ( « 文字 » - U+002F | « パターンエスケープ » ) * U+002F
```

```
« パターンエスケープ » ::= U+005C U+002F
```

5.2.5 記号列

記号列には、「\$」、「<」、「>」、「<=」、「>=」、「@」などがある。これらは、制約の種類や後ろに続く字句の意味を指定するために用いられる。記号列は演算子ではない。

5.3 IMI 語彙記法の構造

5.3.1 IMI 定義文書

一つのIMI定義文書には、複数の語彙の定義（5.3.4）及び一つのデータモデルの定義（5.3.5）を含むことができる。通常は、一つのファイルの内容全体をIMI定義文書とするが、それは必須ではない。他のファイルへの組み込みやストリーム転送などを行ってもよい。そのような場合、IMI定義文書の範囲は組み込まれるファイルの形式や転送プロトコル等の規定によって明確に規定されている必要がある。

5.3.2 文

IMI定義文書は、文の並びとして記述される。文には、語彙定義文、データモデル定義文、クラス定義文、プロパティ定義文、プロパティ設定文、用語使用宣言文の6種類がある。すべての文は、複数のメタデータ宣言と、それに続く定義または宣言によって構成される。

文の順序は、解釈に影響しない。

■5.3.2.1 文法

```
« 文 » ::= « 語彙定義文 » | « データモデル定義文 » | « クラス定義文 » | « プロパティ定義文 » | « プロパティ設定文 » | « 用語使用宣言文 »
```

5.3.3 メタデータ宣言

メタデータ宣言は、「語彙定義文」、「データモデル定義文」、「クラス定義文」、「プロパティ定義文」、「プロパティ設定文」及び「用語使用宣言文」の各文の中で使用され、文によって定義または宣言される対象の説明や状態などを表すメタデータを宣言する。

■5.3.3.1 文法

```

« メタデータ宣言 » := « ディレクティブ » « グループ指定子 » ? « 言語指定子 » ? ( « 文字列リテラル » ...
    | « 名前空間プレフィックス宣言 » )

« ディレクティブ » := "#" « 識別子 »

« 名前空間プレフィックス宣言 » := « 名前空間プレフィックス指定子 » « 文字列リテラル »

```

■5.3.3.2 既定のディレクティブ

次に予め定義されているディレクティブを示す。

表 2: 既定のディレクティブ

ディレクティブ	意味	メタデータの 種類	対象の文
#description	定義又は宣言する対象の説明文	文字列リテラル	すべて
#	#description と同じ	文字列リテラル	すべて
#name	表示する際などに使用される項目名	文字列リテラル	すべて
#prefix	IMI 定義文書内で用いられる名前空間プレフィックスを宣言する。名前空間プレフィックスの適用範囲は、定義又は宣言する対象に限られず、IMI 定義文書全体となる。	名前空間プレフィックス宣言	語彙定義文 データモデル定義文
#name_reading	#name で指定された文字列の読み仮名	文字列リテラル	すべて
#publisher_name	発行者名	文字列リテラル	語彙定義文 データモデル定義文
#publisher_ref	発行者の参照 URL	文字列リテラル	語彙定義文 データモデル定義文
#published_date	発行日。ISO8061 の形式に従って記述	文字列リテラル	語彙定義文 データモデル定義文
#creator	作成者名	文字列リテラル	語彙定義文 データモデル定義文

ディレクティブ	意味	メタデータの	
		種類	対象の文
#creator_ref	作成者の参照 URL	文字列リテラル	語彙定義文 データモデル定義文
#created_date	作成日。ISO8061 の形式に従って記述	文字列リテラル	語彙定義文 データモデル定義文
#version	バージョン番号。メジャーバージョン、マイナーバージョン、リビジョンをピリオドで区切って指定	文字列リテラル	語彙定義文 データモデル定義文
#license	ライセンス表記	文字列リテラル	語彙定義文 データモデル定義文
#license_ref	ライセンスの参照 URL	文字列リテラル	語彙定義文 データモデル定義文
#example	プロパティに対して使用できる値の例	文字列リテラル	プロパティ定義文 プロパティ設定文 用語使用宣言文

■5.3.3.3 グループ指定子

ディレクティブには、必要に応じてグループ指定子及び言語指定子を付加することができる。グループ指定子は、例えば creator が複数の場合、creator と creator_ref の対応関係を表すために用いる。

グループ指定子は、次の文法に従った文字列で、同じグループ指定子をもつメタデータ宣言は同じ対象に対するメタデータであることを表す。

```

< グループ指定子 >  := < グループ名リテラル >
< グループ名リテラル > ::= U+005B ( < 文字 > - U+005D )* U+005D

```

例えば、次の例において #creator[1] と #creator_ref[1] は同じ作成者に対するメタデータであり、#creator[1] と #creator[2] は異なる作成者に対するメタデータである。

例 10:

```

#creator[1] "経済産業省"
#creator_ref[1] "http://www.meti.go.jp/"
#creator[2] "IPA 独立行政法人情報処理推進機構"
#creator_ref[2] "http://www.ipa.go.jp/"

```

■5.3.3.4 言語指定子

言語指定子は、#name@en のように@に続けて ISO 639-1 が定める言語コードを指定することで、特定の言語における情報を記述する際に指定する。

■5.3.3.5 ディレクティブの拡張

語彙記法リーダーの開発者は、特定の語彙記法リーダーが使用するディレクティブを追加してもよい。その場合は、"#_my_directive"のように、"#"に続く識別子の先頭が "_" となるようにする。

5.3.4 語彙定義文

■5.3.4.1 文法

```

< 語彙定義文 > := < メタデータ宣言 > * < 語彙定義 > ";"
< 語彙定義 > := @ "vocabulary" < 文字列リテラル >

```

< 文字列リテラル > は、URI として正しい書式の文字列でなければならない。

■5.3.4.2 意味

語彙定義文は、< 文字列リテラル > で指定された URI を名前空間とする語彙を定義する。また、定義する語彙のメタデータを宣言する。

次の例は、<http://ipa.go.jp/ns/core/2> を名前空間とする語彙を定義するとともに、その語彙の名称や作成者等のメタデータを宣言するものである。

例 11:

```

#name "共通語彙基盤コア語彙"
#name@en "IMI Core Vocabulary"
# "コア語彙は、共通語彙基盤の基礎をなすもので、氏名、住所、組織等、あらゆる社会活動で使用される中 ...
  ↳ 核的な用語の集合です。多くのシステム間で情報交換のための基礎となる語彙で、データ交換、オープ ...
  ↳ ンデータの二次利用等の効率化に役立つものです。"
#description@en "IMI Core Vocabulary"
#created_date "2017-11-10"
#creator[1] "経済産業省"
#creator_ref[1] "http://www.meti.go.jp/"
#creator[1]@en "Ministry of Economy, Trade and Industry"
#creator[2] "IPA 独立行政法人情報処理推進機構"
#creator_ref[2] "http://www.ipa.go.jp/"
#creator[2]@en "Information-technology Promotion Agency, Japan."
#publisher "IPA 独立行政法人情報処理推進機構"
#publisher@en "Information-technology Promotion Agency, Japan."
#published_date "2017-11-10"
#version "2.4.1"
#license "CC0"
#license_ref "http://creativecommons.org/publicdomain/zero/1.0/"
vocabulary "http://imi.go.jp/ns/core/2" ;

```

■5.3.4.3 エラー及び警告

- 一つの IMI 定義文書の中に同一の名前空間をもつ複数の語彙定義文がある場合には、エラーを通知する。
- < 文字列リテラル > が正しい URI の書式でない場合はエラーを通知する。

< 文字列リテラル > の参照先について検証を行うか行わないかは、語彙記法リーダーの実装に依存するものとする。

5.3.5 データモデル定義文

■5.3.5.1 文法

```

« データモデル定義文 » := « メタデータ宣言 » * « データモデル定義 » ";"

« データモデル定義 » := @"datamodel"

```

一つの IMI 定義文書には、二つ以上のデータモデル定義文を含めることはできない。

■5.3.5.2 意味

データモデル定義文は、IMI 定義文書においてデータモデルを定義することを宣言する。また、定義するデータモデルの説明文などのメタデータを定義する。

次の例は、「学校一覧」という名称のデータモデルを定義する。

例 12:

```

#name "学校一覧"
# "地域内の学校の一覧"
#created_date "2017-11-10"
#creator[1] "〇〇市"
#version "1.0.0"
#license "CC0"
#license_ref "http://creativecommons.org/publicdomain/zero/1.0/"

datamodel ;

```

■5.3.5.3 エラー及び警告

- 一つの IMI 定義文書の中に二つ以上のデータモデル定義文が含まれる場合は、エラーを通知する。

5.3.6 クラス定義文

■5.3.6.1 文法

```

« クラス定義文 » := « メタデータ宣言 » * « クラス定義 » ";"

« クラス定義 » := @"deprecated"? @"class" « クラス用語名 » ( "{" « 型制約 » "}" )?

```

■5.3.6.2 意味

クラス定義文は、語彙の定義に用いられ、IMI 定義文書内のいずれかの語彙に含まれるクラス用語を定義する。クラス定義文によって、新たに定義されたクラス用語は空のプロパティ用語集合をもつ。

定義するクラス用語がどの語彙に含まれるかは、名前空間プレフィックスにより表されるクラス用語の名前空間と語彙の名前空間を比較することで決定される。具体的には、定義するクラス用語の名前空間と語彙の名前空間を先頭から比較し、語彙の名前空間の終わりに至るまで完全に一致する場合に、そのクラス用語の名前空間とその語彙の名前空間が一致するという。クラス用語が含まれる語彙は、IMI 定義文書内に定義された、名前空間が一致する語彙のうち最も長い名前空間をもつ語彙とする。

クラス用語が他のクラス用語を継承する場合、《型制約》により継承元のクラス用語を指定する。

《クラス定義》に @”deprecated” がある場合、そのクラス用語は、将来的に削除されることが予定されているために使用を推奨しないクラス用語であることを示す。

次の例は、名前空間が `http://imi.go.jp/ns/core/2` である語彙に、その語彙の実体型を継承元として新たなクラス用語「人型」を定義する。

例 13:

```
#prefix ic: "http://imi.go.jp/ns/core/2#"
vocabulary "http://imi.go.jp/ns/core/2" ;

#name "人型"
# "人の情報を表現するためのクラス用語"
#name@en "Person"
#description@en "A class term to express information of a person."
class ic:人型 {@ic:実体型} ;
```

■5.3.6.3 エラー及び警告

- 《クラス用語名》が同一のクラス用語又はプロパティ用語がすでに定義されている場合、エラーを通知する。
- 定義するクラス用語が継承元クラス用語列に含まれる場合、エラーを通知する。
- 継承元クラス用語として指定された URI がクラス用語のものでない場合、エラーを通知する。
- その他、制約にエラーがある場合、エラーを通知する。

ただし、継承元クラス用語について、対象の IMI 定義文書の範囲を越えて検証を行うかどうかは、語彙記法リーダーの目的によって決定してよい。

- 使用が許されていないディレクティブが使用されている場合は、警告を通知する。

5.3.7 プロパティ定義文

■5.3.7.1 文法

```
《プロパティ定義文》 := 《メタデータ宣言》 * 《プロパティ定義》 ";"
《プロパティ定義》 := @"deprecated"? @"property" 《プロパティ用語名》 《制約指定》 *
```

プロパティ定義文では、制約として、型制約、パターン制約、値等価制約、値開下限制約、値下限制約、値開上限制約、値上限制約、使用可能文字制約が使用できる。これらのうち、型制約は必須とする。

■5.3.7.2 意味

プロパティ定義文は、IMI 定義文書内のいずれかの語彙に含まれるプロパティ用語を定義する。

定義するプロパティ用語がどの語彙に含まれるかは、名前空間プレフィックスにより表されるプロパティ用語の名前空間と語彙の名前空間を比較することで決定される。具体的には、定義するプロパティ用語の名前空間と語彙の名前空間を先頭から比較し、語彙の名前空間の終わりに至るまで完全に一致する場合に、そのプロパティ用語の名前空間とその語彙の名前空間が一致するという。プロパティ用語が含まれる語彙は、IMI 定義文書内に定義された、名前空間が一致する語彙のうち最も長い名前空間をもつ語彙とする。

《プロパティ定義》において《型制約》は値型の指定を行う。《型制約》に指定される型は、W3C XML Schema Definition Language (XSD) 1.1 Part 2: Datatypes (<http://www.w3.org/TR/xmlschema11-2/>) が定める Built-in Datatypes (以下、XML Datatypes と呼ぶ) 又は、正しいクラス用語でなくてはならない。

《プロパティ定義》に @”deprecated” がある場合、そのプロパティ用語は、将来的に削除されることが予定されているために使用を推奨しないプロパティ用語であることを示す。

次の例は、名前空間が <http://imi.go.jp/ns/core/2> である語彙に、値型が「氏名型」である新たなプロパティ用語「氏名」を定義する。その値型は「氏名型」である。

例 14:

```
#prefix ic: "http://imi.go.jp/ns/core/2#"
vocabulary "http://imi.go.jp/ns/core/2" ;

#name "氏名"
# "人の氏名を記述するためのプロパティ用語"
#name@en "Name of person"
#description@en "The name of a person."
property ic:氏名 {@ic:氏名型} ;
```

■5.3.7.3 エラー及び警告

- 《プロパティ用語名》が同一のクラス用語又はプロパティ用語がすでに定義されている場合、エラーを通知する。
- 型制約による値型の指定がない場合、エラーを通知する。
- 型制約が複数ある場合、エラーを通知する。
- 値型が XML Datatypes 又はクラス用語でない場合、エラーを通知する。
- その他、制約にエラーがある場合、エラーを通知する。

ただし、値型として指定されたクラス用語について、対象の IMI 定義文書の範囲を越えて検証を行うかどうかは、語彙記法リーダーの目的によって決定してよい。

- 使用が許されていないディレクティブが使用されている場合は、警告を通知する。
- 使用が許されていない制約が使用されている場合は、警告を通知する。

5.3.8 プロパティ設定文

■5.3.8.1 文法

```
《プロパティ設定文》 := 《メタデータ宣言》 * 《プロパティ設定》 ";"
```

```

« プロパティ 設定 » := @"deprecated"? @"set" « クラス用語名 » ">" « プロパティ用語名 » « 制約指定 » ...
↳ *

```

プロパティ設定文では、制約として、型制約、回数制約、及び、プロパティ順序制約が使用できる。これらの制約を複数回指定することはできない。

■5.3.8.2 意味

プロパティ設定文は、語彙の定義に用いられ、《クラス用語名》で指定されたクラス用語のプロパティ用語集合に《プロパティ用語名》で指定されたプロパティ用語を追加する。回数制約が省略された場合は、プロパティ用語が省略可能であり、かつ、複数指定可能であることを意味する{0..n}が指定されたものとする。プロパティ順序制約が省略された場合の動作については、5.3.20 プロパティ順序制約を参照する。また、型制約がある場合、プロパティ用語の値型を制限することを意味する。この場合の値型は、《プロパティ用語名》で指定されたプロパティ用語の値型か、それを継承したクラス用語でなくてはならない。

プロパティ設定文に @"deprecated" がある場合、対象となるクラス用語における対象プロパティの使用は、将来的に削除されることが予定されているために使用を推奨しないプロパティであることを示す。

次の例は、<http://imi.go.jp/ns/core/2#人型>というクラス用語のプロパティ用語集合に <http://imi.go.jp/ns/core/2#氏名>というプロパティ用語を追加する。追加されたプロパティ用語は省略可能でかつ複数回の使用ができる。

例 15:

```

#prefix ic: "http://imi.go.jp/ns/core/2#"
datamodel ;

#name "氏名"
# "氏名を記述するためのプロパティ用語"
#description@en "A name of the person."
set ic:人型>ic:氏名 {0..n} ;

```

■5.3.8.3 エラー及び警告

- 《クラス用語名》が IMI 定義文書内で正しく定義されたクラス用語でない場合、エラーを通知する。
- 《プロパティ用語名》が正しいプロパティ用語ではない場合、エラーを通知する。
- 型制約、回数制約又はプロパティ順序制約のいずれかが複数回指定されている場合、エラーを通知する。
- 値型が《プロパティ用語名》で指定されたプロパティ用語の値型と同じか、それを継承したクラス用語でない場合、エラーを通知する。
- 回数制約が不正な場合、エラーを通知する。
- その他、制約にエラーがある場合、エラーを通知する。

ただし、値型として指定されたクラス用語や追加されるプロパティ用語について、対象の IMI 定義文書の範囲を越えて検証を行うかどうかは、語彙記法リーダーの目的によって決定してよい。

- 使用が許されていないディレクティブが使用されている場合は、警告を通知する。
- 使用が許されていない制約が使用されている場合は、警告を通知する。

5.3.9 用語使用宣言文

■5.3.9.1 文法

```

« 用語使用宣言文 » := « メタデータ宣言 » * « 用語使用宣言 » ";"
« 用語使用宣言 » := @ "use" « 厳密構造要素名 » « 制約指定 » *

```

用語使用宣言文では、制約として、パターン制約、回数制約、型制約、値等価制約、値開下限制約、値下限制約、値開上限制約、値上限制約、使用可能文字制約が使用できる。これらのうち、型制約及び回数制約は複数回指定することはできない。

■5.3.9.2 意味

用語使用宣言文は、データモデルの定義に用いられ、データモデルにおいて使用するクラス用語及びそのプロパティ用語を宣言する。用語使用宣言文では、用語の使用を宣言することに加え、対象プロパティに対して、回数、値の制限、値型の制限などを設定することができる。

回数制約が省略された場合は、プロパティ設定文による指定と同じ回数制約が指定されたものとする。また、型制約がある場合、プロパティ用語及びプロパティ設定文の値型を制限することを意味する。この場合の値型は、対象となるプロパティ用語の値型と同じか、それを継承したクラス用語でなくてはならない。

次の例は、データモデルの定義において、クラス用語「人型」を使用することと、クラス用語「人型」のプロパティ用語として「氏名」を使用することを宣言している。また、氏名の使用可能回数を1回に制限している。

例 16:

```

#prefix ic: "http://imi.go.jp/ns/core/2#"
vocabulary "http://imi.go.jp/ns/core/2" ;

datamodel;

use ic:人型>ic:氏名 {0..1} ;

```

■5.3.9.3 エラー及び警告

- « 厳密構造要素名 » により指定されたクラス用語が正しく定義されたクラス用語でない場合、エラーを通知する。
- « 厳密構造要素名 » により指定されたプロパティ用語が正しいプロパティ用語ではない場合、エラーを通知する。
- 型制約又は回数制約のいずれかが複数回指定されている場合、エラーを通知する。
- 値型が« 厳密構造要素名 » により指定されたプロパティ用語の値型と同じか、それを継承したクラス用語でない場合、エラーを通知する。
- 回数制約が不正な場合、エラーを通知する。
- その他、制約にエラーがある場合、エラーを通知する。

ただし、値型として指定されたクラス用語について、対象の IMI 定義文書の範囲を越えて検証を行うかどうかは、語彙記法リーダーの目的によって決定してよい。

- 使用が許されていないディレクティブが使用されている場合は、警告を通知する。
- 使用が許されていない制約が使用されている場合は、警告を通知する。

5.3.10 制約

クラス定義文、プロパティ定義文、プロパティ設定文、用語使用宣言文では、制約を記述することができる。

現在規定されている制約には、パターン制約、回数制約、型制約、値等価制約、値開下限制約、値下限制約、値開上限制約、値上限制約、使用可能文字制約、及び、プロパティ順序制約がある。

■5.3.10.1 値範囲に関する制約の合成

一つのプロパティに対して設定される、値等価制約、値開下限制約、値下限制約、値開上限制約、及び、値上限制約には、次のように合成され、最終的な値の範囲が決定される。まず、プロパティ定義文の値等価制約、値開下限制約、値下限制約、値開上限制約、及び、値上限制約の論理積をとりこれを、プロパティ定義文による値の範囲とする。つぎに、用語使用宣言文の値開下限制約、値下限制約、値開上限制約、及び、値上限制約の論理積をとりこれを、用語使用宣言文による値の範囲とする。最後にプロパティ定義文の値の範囲と用語使用宣言文の値の範囲の論理積をとったものが最終的な値の範囲として適用される。最終的な値の範囲が設定可能な値をもたないものになる場合はエラーとなる。

現在規定されている制約の一覧を次に示す。

表 3: 制約一覧

制約の種類	意味	対象となる 文	参照
パターン制約	対象の値型が文字列の場合に、正規表現により値となる文字列表現を規定する。	プロパティ定義文 用語使用宣言文	5.3.11
回数制約	クラス用語のプロパティとしての使用可能回数を規定する。	プロパティ設定文 用語使用宣言文	5.3.12
型制約	クラス定義文においては、継承元のクラス用語を規定する。それ以外の文では値型を規定する。	クラス定義文 プロパティ定義文 用語使用宣言文	5.3.13
値等価制約	対象の値型が整数値又は文字列の場合に、値が指定された値と等価であることを規定する。	プロパティ定義文 用語使用宣言文	5.3.14
値開下限制約	対象の値型が数値の場合に、値が指定された値より大きいことを規定する。	プロパティ定義文 用語使用宣言文	5.3.15
値下限制約	対象の値型が数値の場合に、値が指定された値以上であることを規定する。	プロパティ定義文 用語使用宣言文	5.3.16
値開上限制約	対象の値型が数値の場合に、値が指定された値未満であることを規定する。	プロパティ定義文 用語使用宣言文	5.3.17
値上限制約	対象の値型が数値の場合に、値が指定された値以下であることを規定する。	プロパティ定義文 用語使用宣言文	5.3.18

制約の種類	意味	対象となる 文	参照
使用可能文字制約	対象の値型が文字列の場合に、使用できる文字コレクションを規定する。	プロパティ定義文 用語使用宣言文	5.3.19
プロパティ順序制約	クラス用語のプロパティの順序を規定する。	プロパティ設定文	5.3.20

■5.3.10.2 文法

制約を記述するための制約指定の文法を次に示す。

```
« 制約指定 » := "{" « 制約 » "}"
```

```
« 制約 » := « パターン制約 » | « 回数制約 » | « 型制約 » | « 値等価制約 » | « 値開下限制約 » | « ...  
↳ 値下限制約 » | « 値開上限制約 » | « 値上限制約 » | « 使用可能文字制約 » | « プロパティ順序制 ...  
↳ 約 »
```

5.3.11 パターン制約

■5.3.11.1 文法

```
« パターン制約 » := "/" « パターンリテラル » "/"
```

■5.3.11.2 意味

対象の値型が文字列の場合に、値となり得る文字列を正規表現により規定する。対称の値型が文字列以外の場合は無効であるが、エラーとはしない。パターン制約は、プロパティ定義文、用語使用宣言文に対して有効とする。

パターン制約がプロパティ定義文及び用語使用宣言文の両方で指定された場合、用語使用宣言文の制約が有効となる。

次の例では、当該データモデルにおいて、郵便番号の値を7桁の数字に制約する。

例 17:

```
datamodel;  
  
use ic:住所型>ic:郵便番号 {/[0-9]{7}/};
```

■5.3.11.3 エラー及び警告

- « パターンリテラル » が正しい正規表現でない場合は、エラーを通知する。
- 一つの文にパターン制約が複数回指定された場合、エラーを通知する。

5.3.12 回数制約

■5.3.12.1 文法

```
« 回数制約 » := « 整数リテラル » "." ( « 整数リテラル » | "n" )
```

■5.3.12.2 意味

クラス用語の中で対象プロパティ用語が使用可能な回数を規定する。下限は整数リテラルで表される。上限は整数リテラルまたは"n"で表され、"n"は上限を規定しないことを意味する。回数制約は、プロパティ設定文及び用語使用宣言文に対し有効とする。プロパティ設定文において、回数制約が指定されていない場合の既定値は{0..n}とする。回数制約がプロパティ設定文及び用語使用宣言文の両方で指定された場合、用語使用宣言文の制約が有効となる。

次の例では、生年月日の使用可能回数を 0 以上 1 以下（省略可能で、かつ、最大でも 1 回）に制限している。

例 18:

```
set ic:人型>ic:生年月日 {0..1} ;
```

■5.3.12.3 エラー及び警告

- 上限値が下限値未満の場合は、エラーを通知する。
- 一つの文に回数制約が複数回指定された場合、エラーを通知する。
- プロパティ設定文及び用語使用宣言文の両方で指定され、かつ、用語使用宣言文の指定する範囲がプロパティ設定文で指定する範囲に含まれない範囲を含む場合、エラーを通知する。

ただし、回数制約の範囲について、対象の IMI 定義文書の範囲を越えて検証を行うかどうかは、語彙記法リーダーの目的によって決定してよい。

5.3.13 型制約

■5.3.13.1 文法

```
« 型制約 » := "@" « クラス項目 »
```

■5.3.13.2 意味

クラス定義文においては、継承元のクラス用語を規定する。クラス定義文において、型制約が指定されていない場合は、他のクラス用語を継承しないことを意味する。プロパティ定義文及び用語使用宣言文においては、プロパティの値型を規定する。用語使用宣言においては、対象のプロパティの値として実際に使用されるクラス用語を規定する。プロパティ定義文では型制約は必須である。

次の例では、生年月日の値型を日付型に設定している。

例 19:

```
property ic:生年月日 {@ic:日付型} ;
```

■5.3.13.3 エラー及び警告

- « クラス項目 » により指定されたクラス用語が正しく定義されたクラス用語でない場合、エラーを通知する。

ただし、クラス用語について、対象の IMI 定義文書の範囲を越えて検証を行うかどうかは、語彙記法リーダーの目的によって決定してよい。

5.3.14 値等価制約

■5.3.14.1 文法

```
« 値等価制約 » := "=" ( « 整数リテラル » | « 文字列リテラル » )
```

■5.3.14.2 意味

値等価制約は、プロパティ定義文、及び、用語使用宣言文に対して有効とする。この制約は、対象の値型が整数値又は文字列の場合のみ有効とする。プロパティ定義文に適用された場合、指定された値のみが使用できることを規定する。用語使用宣言文に適用された場合、対象のプロパティ用語が存在し、そのプロパティ用語の値が指定された固定値が入ることを意味する。

次の例では、種別の値を”学校”という固定値に設定している。

例 20:

```
use ic:施設>ic:種別 {"学校"}
```

■5.3.14.3 エラー及び警告

- 用語使用宣言文の値等価制約の指定値が、プロパティ定義文の制約による値の範囲に含まれない場合は、エラーを通知する。
- 同じ文に、すでに値等価制約、値開下限制約、値下限制約、値開上限制約、及び、値上限制約のいずれかが適用されている場合、エラーを通知する。
-

ただし、値等価制約の指定する値について、対象の IMI 定義文書の範囲を越えて検証を行うかどうかは、語彙記法リーダーの目的によって決定してよい。

5.3.15 値開下限制約

■5.3.15.1 文法

```
« 値開下限制約 » := ">" « 整数リテラル »
```

■5.3.15.2 意味

対象の値型が数値の場合に、値が指定された値より大きいことを規定する。

値開下限制約は、プロパティ定義文、及び、用語使用宣言文に対して有効とする。この制約は、対象の値型が整数値の場合のみ有効とする。

次の例は、当該データモデルにおいて、プロパティ用語「年」の値に、2000 より大きいという制約を課す。

例 21:

```
datamodel;

use ic:人型>ic:生年月日>ic:年 {>2000};
```

複数の文に値開下限制約が指定された場合は、プロパティ定義文、用語使用宣言文の順序により適用され、有効な値の範囲は、プロパティ定義文の制約の範囲と用語使用宣言文の制約の範囲の論理積とする。

次の例では、プロパティ定義文で「人数」の値を 0 より大きい値に制約しているが、当該データモデルにおいては、さらに、10 より大きく 100 未満に制約している。

例 22:

```
property ic:人数 {@xsd:nonNegativeInteger} {>0}

datamodel;

use ex:グループ型>ic:人数 {>10} {<100};
```

■5.3.15.3 エラー及び警告

- プロパティ定義文及び用語使用宣言文の両方で指定され、かつ、用語使用宣言文の制約の範囲が、プロパティ定義文の制約による値の範囲に含まれない値を許容する場合に警告を通知する。
- 同じ文に値等価制約が指定されている場合はエラーを通知する。
- 適用しようとする値下開限制約により最終的な値の範囲に設定可能な値がなくなる場合、エラーを通知する。（5.3.10.1 を参照）

ただし、値の範囲について、対象の IMI 定義文書の範囲を越えて検証を行うかどうかは、語彙記法リーダーの目的によって決定してよい。

5.3.16 値下限制約

■5.3.16.1 文法

```
« 値下限制約 » := ">=" « 整数リテラル »
```

■5.3.16.2 意味

対象の値型が数値の場合に、値が指定された値以上であることを規定する。

値下限制約は、プロパティ定義文、及び、用語使用宣言文に対して有効とする。この制約は、対象の値型が整数値の場合のみ有効とする。

次の例は、当該データモデルにおいて、プロパティ用語「年」の値に、2000 以上という制約を課す。

例 23:

```
datamodel;

use ic:人型>ic:生年月日>ic:年 {>=2000};
```

複数の文に値下限制約が指定された場合は、プロパティ定義文、用語使用宣言文の順序により適用され、有効な値の範囲は、プロパティ定義文の制約の範囲と用語使用宣言文の制約の範囲の論理積とする。

次の例では、プロパティ定義文で「人数」の値を 1 以上に制約しているが、当該データモデルにおいては、さらに、10 以上かつ 100 未満に制約している。

例 24:

```
property ic:人数 {@xsd:nonNegativeInteger} {>=1}

datamodel;

use ex:グループ型>ic:人数 {>=10} {<100};
```

■5.3.16.3 エラー及び警告

- プロパティ定義文及び用語使用宣言文の両方で指定され、かつ、用語使用宣言文の制約の範囲が、プロパティ定義文の制約による値の範囲に含まれない値を許容する場合に警告を通知する。
- 同じ文に値等価制約が指定されている場合はエラーを通知する。
- 適用しようとする値下限制約により最終的な値の範囲に設定可能な値がなくなる場合、エラーを通知する。（5.3.10.1 を参照）

ただし、値の範囲について、対象の IMI 定義文書の範囲を越えて検証を行うかどうかは、語彙記法リーダーの目的によって決定してよい。

5.3.17 値開上限制約

■5.3.17.1 文法

```
« 値開上限制約 » := "<" « 整数リテラル »
```

■5.3.17.2 意味

対象の値型が数値の場合に、値が指定された値未満であることを規定する。

値開上限制約は、プロパティ定義文、及び、用語使用宣言文に対して有効とする。この制約は、対象の値型が整数値の場合のみ有効とする。

次の例は、当該データモデルにおいて、プロパティ用語「年」の値に、2000 未満という制約を課す。

例 25:

```
datamodel;

use ic:人型>ic:生年月日>ic:年 {<2000};
```

複数の文に値開上限制約が指定された場合は、プロパティ定義文、用語使用宣言文の順序により適用され、有効な値の範囲は、プロパティ定義文の制約の範囲と用語使用宣言文の制約の範囲の論理積とする。

次の例では、プロパティ定義文で「人数」の値を 100 未満に制約しているが、当該データモデルにおいては、さらに、10 より大きく 50 未満に制約している。

例 26:

```
property ic:人数 {@xsd:nonNegativeInteger} {<100}

datamodel;

use ex:グループ型>ic:人数 {>10} {<50};
```

■5.3.17.3 エラー及び警告

- プロパティ定義文及び用語使用宣言文の両方で指定され、かつ、用語使用宣言文の制約の範囲が、プロパティ定義文の制約による値の範囲に含まれない値を許容する場合に警告を通知する。
- 同じ文に値等価制約が指定されている場合はエラーを通知する。
- 適用しようとする値開上限制約により最終的な値の範囲に設定可能な値がなくなる場合、エラーを通知する。（5.3.10.1 を参照）

ただし、値の範囲について、対象の IMI 定義文書の範囲を越えて検証を行うかどうかは、語彙記法リーダーの目的によって決定してよい。

5.3.18 値上限制約

■5.3.18.1 文法

```
« 値上限制約 » := "<=" « 整数リテラル »
```

■5.3.18.2 意味

対象の値型が数値の場合に、値が指定された値以下であることを規定する。

値上限制約は、プロパティ定義文、及び、用語使用宣言文に対して有効とする。この制約は、対象の値型が整数値の場合のみ有効とする。

次の例は、当該データモデルにおいて、プロパティ用語「年」の値に、2000 以下という制約を課す。

例 27:

```
datamodel;

use ic:人型>ic:生年月日>ic:年 {<=2000};
```

複数の文に値上限制約が指定された場合は、プロパティ定義文、用語使用宣言文の順序により適用され、有効な値の範囲は、プロパティ定義文の制約の範囲と用語使用宣言文の制約の範囲の論理積とする。

次の例では、プロパティ定義文で「人数」の値を 100 以下に制約しているが、当該データモデルにおいては、さらに、10 より大きく 50 以下に制約している。

例 28:

```
property ic:人数 {@xsd:nonNegativeInteger} {<=100}

datamodel;

use ex:グループ型>ic:人数 {>10} {<=50};
```

■5.3.18.3 エラー及び警告

- プロパティ定義文及び用語使用宣言文の両方で指定され、かつ、用語使用宣言文の制約の範囲が、プロパティ定義文の制約による値の範囲に含まれない値を許容する場合に警告を通知する。
- 同じ文に値等価制約が指定されている場合はエラーを通知する。
- 適用しようとする値上限制約により最終的な値の範囲に設定可能な値がなくなる場合、エラーを通知する。（5.3.10.1 を参照）

ただし、値の範囲について、対象の IMI 定義文書の範囲を越えて検証を行うかどうかは、語彙記法リーダーの目的によって決定してよい。

5.3.19 使用可能文字制約

■5.3.19.1 文法

```
« 使用可能文字制約 » := "$" « 文字列リテラル »
```

■5.3.19.2 意味

対象の値型が文字列の場合に、使用できる文字コレクションを規定する。「文字列リテラル」の内容は、「使用文字規制の記法」に従った文字列とする。使用可能文字制約は、プロパティ定義文、及び、用語使用宣言文に対して有効とする。

参考

IMI では、使用可能な文字の一覧を取得するための仕組みを提供している。「使用文字規制の記法」とは、その仕組みを利用するにあたって、取得する文字の一覧を指定するための文字列である。

使用文字規制の記法は、文字セットを参照する URI と指定強度を表す文字列を組み合わせた文字列である。指定強度には“推奨”“非推奨”の2種類あり、それぞれ次のような意味をもつ。

“推奨”：入力を受け付ける文字（それを使用することが望ましい）“非推奨”：警告を出す文字（望ましくはないが入力拒否まではしない）

これら指定強度を表すには、“非推奨”の場合は文字セット参照の URI の前に “[NotRecommended]” というキーワードを置く。キーワード無しで文字セット参照 URI が記述された場合、それは“推奨”の文字セットと見なされる。

次の例は、使用文字規制の記法を用いた「使用可能文字制約」の例である。

例

```
{ "$" "https://imi.go.jp/CommonCharacterSets/ISOIEC10646annexA-compliant_BASIC_JAPANESE" |
  [NotRecommended] "https://imi.go.jp/CommonCharacterSets/ISOIEC10646annexA-compliant_JIS2004_IDEO-
  GRAPHICS_EXTENSION" }
```

使用文字規制の記法の詳細については、「使用文字規制の記法」を参照する。

■5.3.19.3 エラー及び警告

- « 文字列リテラル » が、「使用文字規制の記法」が定める形式でない場合、エラーを通知する。

5.3.20 プロパティ順序制約

■5.3.20.1 文法

```
« プロパティ順序制約 » := "#" « 整数リテラル »
```

■5.3.20.2 意味

クラス用語のプロパティ用語集合の中におけるプロパティ用語の順序を規定する。プロパティ順序制約は、プロパティ設定文に対してのみ有効とする。プロパティ用語の順序は、まず、プロパティ順序制約をもつプロパティ用語がプロパティ順序制約の値が小さい順に並ぶ。このとき、プロパティ順序制約の値が同一のプロパティ設定文があれば、それらは、IMI 定義文書内の出現順序に並ぶものものとする。続いて、プロパティ順序制約をもたないプロパティ用語が IMI 定義文書内の出現順序に並ぶものとする。

■5.3.20.3 エラー及び警告

通知するエラーはない。

5.3.21 コメント及び空白文字

コメントには、複数行に渡る記述が可能な« コメント » と単一行の« 行コメント » の2種類の形式がある。

コメントは、字句規則上は意味をもち字句と字句を区切る効果があるが、構文規則上は、明示意味をもたず無視される。

空白文字は字句と字句を区切るために用いることができる。

空白文字は、字句規則上は意味をもち字句と字句を区切る効果があるが、構文規則上は、終端記号と他の字句を区切る場合を除き意味をもたず無視される。

コメント及び空白文字は、構文規則における右辺の字句と字句との間、及び、構文規則によって定義される各構文の前後に自由に置くことができる。

■5.3.21.1 文法

```
« コメント » ::= "/"* ( « 文字 » * - "/" ) "*"
« 行コメント » ::= "/" ( « 文字 » - « 改行文字 » ) * « 改行文字 »
« 空白文字 » ::= U+0009 | U+000A | U+000D | U+0020
```

5.4 字句構造

次に IMI 定義文書全体の字句構造を示す。

```
« IMI 定義文書 » := « 文 » *
« 文 » := « 語彙定義文 » | « データモデル定義文 » | « クラス定義文 » | « プロパティ定義文 » | « プロパティ設定文 » | « 用語使用宣言文 »
```

```

« 語彙定義文 » := « メタデータ宣言 » * « 語彙定義 » ";"

« 語彙定義 » := @"vocabulary" « 文字列リテラル »

« クラス定義文 » := « メタデータ宣言 » * « クラス定義 » ";"

« クラス定義 » := @"deprecated"? @"class" « クラス用語名 » ( "{" « 型制約 » "}" )?

« プロパティ定義文 » := « メタデータ宣言 » * « プロパティ定義 » ";"

« プロパティ定義 » := @"deprecated"? @"property" « プロパティ用語名 » « 制約指定 » *

« プロパティ設定文 » := « メタデータ宣言 » * « プロパティ設定 » ";"

« プロパティ設定 » := @"deprecated"? @"set" « クラス用語名 » ">" « プロパティ用語名 » « 制約指定 » ...
↳ *

« データモデル定義文 » := « メタデータ宣言 » * « データモデル定義 » ";"

« データモデル定義 » := @"datamodel"

« 用語使用宣言文 » := « メタデータ宣言 » * « 用語使用宣言 » ";"

« 用語使用宣言 » := @"use" « 厳密構造要素名 » « 制約指定 » *

« メタデータ宣言 » := « ディレクティブ » « グループ指定子 » ? « 言語指定子 » ? ( « 文字列リテラル » ...
↳ | « 名前空間プレフィックス宣言 » )

« 名前空間プレフィックス宣言 » := « 名前空間プレフィックス指定子 » « 文字列リテラル »

« 制約指定 » := "{" « 制約 » "}"

« 制約 » := « パターン制約 » | « 回数制約 » | « 型制約 » | « 値等価制約 » | « 値開下限制約 » | « ...
↳ 値下限制約 » | « 値開上限制約 » | « 値上限制約 » | « 使用可能文字制約 » | « プロパティ順序制 ...
↳ 約 »

« パターン制約 » := « パターンリテラル »

« 回数制約 » := « 整数リテラル » ".." ( « 整数リテラル » | "n" )

« 型制約 » := "@" « クラス項目 »

« 値等価制約 » := "=" ( « 整数リテラル » | « 文字列リテラル » )

« 値開下限制約 » := ">" « 整数リテラル »

« 値下限制約 » := ">=" « 整数リテラル »

« 値開上限制約 » := "<" « 整数リテラル »

« 値上限制約 » := "<=" « 整数リテラル »

« 使用可能文字制約 » := "$" « 文字列リテラル »

« プロパティ順序制約 » := "#" « 整数リテラル »

« 厳密構造要素名 » := « クラス項目 » « プロパティ列 » « 言語指定子 » ?

« クラス項目 » := « クラス用語名 » « グループ指定子 » ?

« クラス用語名 » := « 名前空間プレフィックス指定子 » « 識別子 »

« グループ指定子 » := « グループ名リテラル »

« プロパティ列 » := ( ">" « プロパティ項目 » ) *

```

« プロパティ項目 » ::= « プロパティ用語名 » « グループ指定子 » ?
 « プロパティ用語名 » ::= « 名前空間プレフィックス指定子 » « 識別子 »
 « 言語指定子 » ::= "@" « 言語名 »
 « ディレクティブ » ::= "#" « 識別子 » ?
 « コメント » ::= "/"* (« 文字 » * - "/") "*" /
 « 行コメント » ::= "//" (« 文字 » - « 改行文字 »)* « 改行文字 »
 « 文字列リテラル » ::= « 二重引用符文字列リテラル » | « 引用符文字列リテラル »
 « 二重引用符文字列リテラル » ::= U+0022 ((« 文字 » - (U+0022 | U+005C)) « 二重引用符エスケープ »)* U+0022
 « 二重引用符エスケープ » ::= (U+005C U+0022 | U+005C U+005C)
 « 引用符文字列リテラル » ::= U+0027 ((« 文字 » - (U+0027 | U+005C)) « 引用符エスケープ »)* U+0027
 « 引用符エスケープ » ::= (U+005C U+0027 | U+005C U+005C)
 « グループ名リテラル » ::= U+005B (« 文字 » - U+005D)* U+005D
 « 整数リテラル » ::= « 数字 » +
 « パターンリテラル » ::= U+002F (« 文字 » - U+002F | « パターンエスケープ »)* U+002F
 « パターンエスケープ » ::= U+005C U+002F
 « 名前空間プレフィックス指定子 » ::= « 識別子先頭文字 » « 識別子文字 »* U+003A
 « 識別子 » ::= « 識別子先頭文字 » « 識別子文字 »*
 « 言語名 » ::= « 言語名文字 » « 言語名文字 »
 « 識別子先頭文字 » ::= ['A'-'Z'] | '_' | ['a'-'z'] | [U+00C0-U+00D6] | [U+00D8-U+00F6] | [U+00F8-U+02FF] | [U+0370-U+037D] | [U+037F-U+1FFF] | [U+200C-U+200D] | [U+2070-U+218F] | [U+2C00-U+2DFF] | [U+3001-U+D7FF] | [U+F900-U+FDFF] | [U+FDFF-U+FFFF]
 « 識別子文字 » ::= « 識別子先頭文字 » | '-' | '.' | ['0'-'9'] | U+00B7 | [U+0300-U+036F] | [U+203F-U+2040]
 « 数字 » ::= ['0'-'9']
 « 空白文字 » ::= U+0009 | U+000A | U+000D | U+0020
 « 改行文字 » ::= U+000A
 « 文字 » ::= U+0009 | U+000A | U+000D | [U+0020-U+D7FF] | [U+E000-U+FFFF] | [U+10000-U+10FFFF]
 « 言語名文字 » ::= ['a'-'z'] ['A'-'Z']

この文書について

この文書は、「IMI 共通語彙基盤」の技術的な要件をとりまとめた技術仕様書のひとつです。

表題	IMI 語彙記法
バージョン	1.0.1
公開日	2019 年 2 月 28 日
作成者	独立行政法人情報処理推進機構 (IPA) 社会基盤センター IMI 検討部会
発行者	独立行政法人情報処理推進機構 (IPA) (法人番号 5010005007126)

この文書のご利用にあたって

著作権

この文書は、IPA が著作権を持ち、CC0 1.0 全世界 (<https://creativecommons.org/publicdomain/zero/1.0/legalcode.ja>) で公開します。

免責事項

本書の内容を適用した結果生じたこと、また適用できなかった結果について、IPA 及び IMI 検討部会は、一切の責任を負いませんのでご了承ください。

ご意見を募集しています

広くみなさまのご意見を募集しています。以下ご意見投稿のページに進み、ご記入ください。

<https://imi.go.jp/783/>

この文書の改定履歴

- 2019 年 2 月 28 日 IMIVocabularyNotation_V101_20190228.pdf 発行
- 2018 年 3 月 23 日 IMIVocabularyNotation_V10_20180323.pdf 発行