

AutoLISP で異尺度対応オブジェクトかどうか判別する

AutoLISP のプログラム上で、オブジェクトが異尺度対応であるかどうか判別する方法について説明します。

1. 基本的な手順

オブジェクトが異尺度対応であるかどうかを判別する手順は以下の通りです。

- ① entsel から、寸法オブジェクト名（図形名）を取得する。
- ② entget を用いて DXFデータリストを取得する。
- ③ リスト内に 拡張データがあることを確認する。
- ④ 拡張データの中に、“AcadAnnotative” が存在することを確認する
- ⑤ “AcadAnnotative” の中に、(1070 . 0) が無いことを確認する
- ⑥ 上記をクリアしたオブジェクトが、異尺度対応オブジェクトである

2. 実施例

AutoCAD に既存の Annotative寸法オブジェクト(異尺度対応) について確認した例を見てみましょう

① entsel にて寸法オブジェクトを選択、その 図形名を取得する

```
コマンド: (setq dim (car (entsel)))  
オブジェクトを選択: <図形名: 224704b7de0>
```

② (entget 図形名 “(*”) にて拡張データを含む DXFデータリストを取得する

```
コマンド: (setq lst (entget dim '(*)))  
((-1 . <図形名: 224704b7de0>) (0 . "DIMENSION") (5 . "786") (102 . "{ACAD_XDICTIONARY}") (360 . <図形名: 224704b7df0>) (102 . ")") (330 . <図形名: 2249fab61f0>) (100 . "AcDbEntity") (67 . 0) (410 . "Model") (8 . "0") (100 . "AcDbDimension") (280 . 0) (2 . "D18") (10 100.0 30.0 0.0) (11 75.0 31.9014 0.0) (12 0.0 0.0 0.0) (70 . 32) (1 . "") (71 . 5) (72 . 1) (41 . 1.0) (42 . 50.0) (73 . 0) (74 . 0) (75 . 0) (52 . 0.0) (53 . 0.0) (54 . 0.0) (51 . 0.0) (210 0.0 0.0 1.0) (3 . "Annotative") (100 . "AcDbAlignedDimension") (13 50.0 20.0 0.0) (14 100.0 20.0 0.0) (15 0.0 0.0 0.0) (16 0.0 0.0 0.0) (40 . 0.0) (50 . 0.0) (100 . "AcDbRotatedDimension") (-3 ("AcadAnnotative" (1000 . "AnnotativeData") (1002 . "{(" (1070 . 1) (1070 . 1) (1002 . ")"}))))
```

2. 実施例（つづき）

③ リスト内に 拡張データがあることを確認する。

リストの後ろの方を見ると -3 から始まる拡張データのリストがあることがわかります。これを取り出します。

```
コマンド: (setq lst (assoc -3 lst))  
(-3 ("AcadAnnotative" (1000 . "AnnotativeData") (1002 . "{") (1070 . 1) (1070 . 1) (1002 . "))))
```

④ 拡張データの中に、“AcadAnnotative” が存在することを確認する

リストの後ろの方を見ると -3 から始まる拡張データのリストがあることがわかります。これを取り出します。[]
まず、-3 を消します。

```
コマンド: (setq lst (cdr lst))  
(("AcadAnnotative" (1000 . "AnnotativeData") (1002 . "{") (1070 . 1) (1070 . 1) (1002 . "))))
```

次に、先頭の要素が “AcadAnnotative” である要素(リスト)のみを残し、他の要素(リスト)を削除します。（本例では、“AcadAnnotative” 以外のリストが存在しないので、何も変化しません。）

```
コマンド: (setq lst (vl-remove-if-not '(lambda (x) (equal (car x) "AcadAnnotative")) lst))  
(("AcadAnnotative" (1000 . "AnnotativeData") (1002 . "{") (1070 . 1) (1070 . 1) (1002 . "))))
```

最後に、car を用いて “AcadAnnotative” リストのみを取り出します。

```
コマンド: (setq lst (car lst))  
("AcadAnnotative" (1000 . "AnnotativeData") (1002 . "{") (1070 . 1) (1070 . 1) (1002 . "))))
```

⑤ “AcadAnnotative” リストの中に、(1070 . 0) が無いことを確認する

「“AcadAnnotative” リストが、拡張データの中に存在する。イコール、異尺度対応オブジェクトである。」と思いがちですが、このリストに (1070 . 0) が存在すると「以前異尺度対応だったが、非対応に再設定（オーバーライド）されたオブジェクト」ということになるようです。このため、リスト中に (1070 . 0) が存在しないことを確認します。

```
コマンド: (setq ans (not (vl-some '(lambda (x) (equal x (cons 1070 0))) lst)))  
T
```

この T で判別が終了します。

3. 判別関数の例

参考までに、選択したオブジェクトが異尺度対応かどうかを判別するプログラムの例を示します。

```
(defun detAnnotative (ent / lst ans)
  ; entget にて 図形名 ent の拡張データを含めたDXFデータリストを取得
  (setq lst (entget ent '("")))
  (if
    ; 拡張データがあればさらに判定処理
    (setq lst (assoc -3 lst))
    (progn
      ; -3 より後ろのリストを取得
      (setq lst (cdr lst))
      (if
        ; リスト内の要素(リスト)の先頭が、"AcadAnnotative" である要素(リスト)を抽出
        (setq lst (vl-remove-if-not '(lambda (x) (equal (car x) "AcadAnnotative")) lst))
        (progn
          ; "AcadAnnotative"リストを取り出す
          (setq lst (car lst))
          ; リスト内に (1070 . 0) が 存在しなければ ans = T、存在すれば ans = nil
          (setq ans (not (vl-some '(lambda (x) (equal x (cons 1070 0))) lst)))
        );progn
        ; "AcadAnnotative"リストが無ければ ans = nil
        (setq ans nil)
      );if
    );if
  );progn
  ; 拡張データが無ければ ans = nil
  (setq ans nil)
);if
; 戻り値
ans
);defun
;
; 使い方
; 図形名(オブジェクト名)を引数として、関数 detAnnotative を実行すると 異尺度対応オブジェクトであれば T が、
; 異尺度対応オブジェクトでなければ nil が戻り値となる。
; (setq ent (car (entsel)))
; (detAnnotative ent)
```