

1. 動的に容量を変える表

2002 年の規格 (COBOL2002) でも、データ記述項の OCCURS 句に DEPENDING 指定することにより、表要素の繰り返し数を動的に変える仕組みはありますが、表が占める実際の記憶域の大きさが変わることは想定していません。次期規格 (2008 年予定) COBOL では、記憶域の大きさまで動的に変わる表が導入される予定です。

なお、他のプログラミング言語で「配列」に相当する言語要素を、COBOL では表 (table) と呼びます。

1.1 背景

入力するレコードが、通常は 10 レコードしかないのに、季節的に 100 万レコードになるような場合を考えます。当然最初から 100 万レコード分の表要素宣言しておくのはもったいなく、必要に応じて拡張したいとの要求があります。また、これを現在の仕様範囲である固定サイズの表を利用をあきらめて、ファイル入力として 1 レコードずつ逐次処理していく方式に変更すると、実行時間が増えて、処理も複雑になるという問題があります。さらに、現在は 100 万レコードが最大だが、将来増えるかもしれないので、プログラムに上限を記述したくないとの要求もあります。

動的に大きさの変わる表は、以前から要望されていました。2002 年規格が制定される前にも、動的な表を導入しようと検討されましたが、時間切れで実装が見送られています。規格委員会では、次期規格で容量を動的に変える表を復活させようとしています。

1.2 容量が可変な表

記憶域の大きさを動的に変えられる表は、容量可変表 (Dynamic-capacity table) と呼ぶ。この表は、反復するデータ項目が物理的に連続していなくてもよい。また、実際に使用されるまで、要素を用意しなくてもよい。ただし、論理的には連続して反復しているように見せることになっている。

例で示すと、容量可変表の現在の容量が 30 のときに、添字に 10,000 を指定してデータ項目を参照したら、表の容量は 10,000 に増える。このとき、30 番目の要素から 10,000 番目の要素まで、物理的に記憶域が連続している必要はない。また、この時点で、31 番目から 9,999 番目までのデータ項目を用意する必要もない。

1.3 表の種類

次期規格案では、新しい種類の表を導入することで、既存の表も含めて言語要素を次のように整理しています。容量が固定である表と反復要素数が可変である表は、名前が新しいだけで、従来からある表です。

- ・容量固定表 (Fixed-capacity table)
反復回数が固定の表
- ・反復可変表 (Occurs-depending table)
OCCURS DEPENDING ON 句が指定された項目によって構成される表
- ・容量可変表 (Dynamic-capacity table)
OCCURS DYNAMIC 句が指定された項目によって構成される表

1.4 相互対応集団項目 (Compatible group)

集団項目の単位で転記や比較する場合、従来の COBOL では、集団項目全体をひとつの英数字項目などとみなす仕様でしたが、容量可変表を含む集団項目では適用されません。

転記や比較の作用対象の集団項目で、一方が容量可変表を含むとき、もう一方が含む表の種類は問わないが、次のような相互対応性 (Compatibility) がなければ、転記も比較もできません。

- (1) 容量可変表と対応する表は、集団項目の中の相対バイト位置が同じであること。
- (2) 対応する表のデータ項目のバイト長が一致し、データ項目が適合すること。

相互対応集団項目の転記では、容量可変表が対応する部分は要素ごとに転記し、その他の部分は、英数字転記となります。

1.5 容量可変表の宣言と操作の例

容量可変表の宣言は、OCCURS 句の拡張になる。宣言の例を示して説明します。

```
01 concert-record.  
   03 concert-code      PIC X(10).  
   03 applicants.  
       05 applicant OCCURS DYNAMIC  
           CAPACITY IN applicant-count
```

```
INITIALIZED.  
07 last-name      PIC X(30).  
07 first-name     PIC X(30)  
07 postal-code    PIC X(10).  
03 concert-location PIC X(50).
```

データ項目 `applicant` の記述で `OCCURS` 句の `DYNAMIC` 指定が、容量可変表の宣言を示します。ここでは、最大容量は指定しておらず、例えば、

```
MOVE "Hanako" TO first-name (90).
```

と書くことができます。この文の実行前に `applicant` の 89 番目の要素までが生成されていた場合は、この文の実行によって 90 番目の要素が生成され、現在の容量を保持するデータ項目 `applicant-count` に 90 が設定されます。ここで、`applicant-count` は、`CAPACITY` 指定に指定することで定義される、数字データ項目です。

現在の容量が 100 だとすると、次の文の実行で容量を 10 に減らす場合は次の記述を行います。

```
SET applicant-count TO 10.
```

容量を増やすには、次のように現在の容量より大きい数を指定します。

```
SET applicant-count TO 200.
```

なお、実際に記憶域が開放されるかどうかは、COBOL コンパイラの実装方式によります。また、`Applicant` の要素のうち、まだ値を入れていない要素は、`INITIALIZE` 指定があることで、値を使う場面になれば初期化されていることとしています。

このように、`applicant` の容量を変化させても、隣接するデータ項目 `concert-code` や `concert-location` の記憶域中の位置は変わりません。集団項目 `concert-record` を、他の相互対応集団項目との間で転記するとき、容量可変表である `applicant` の部分は要素ごとに転記し、その他の部分は英数字項目の転記をすることになります。