บทที่ 4
กลุ่มข้อมูล
(Data aggregates)

data items ใน PL/I-80 อาจจะรวมเข้าด้วยกัน เพื่อฟอร์มเป็นอะเรย์ (arrays) หรือโครงสร้าง (structures) ตัวแปรที่ใช้แทนกลุ่มของ data elements อาจจะเป็นตัวแปรอะเรย์ (array variable) หรือ ตัวแปรโครงสร้าง (structure variable) ซึ่งหมายถึง กลุ่มข้อมูล

4.1 อะเรย์ (Array)

อะเรย์เป็น data items ที่รวมเข้าด้วยกันอย่างมีระเบียบ โดยที่อีลิเม้นท์ทุกตัวต้องมี attribute เหมือนกัน การอ้างถึง อีลิเม้นท์ทั้งหมดในอะเรย์ กำหนดโดยชื่ออะเรย์ และการอ้างถึง อีลิเม้นท์ตัวใดตัวหนึ่ง ให้ใช้ integer subscripts*, เมื่อมีการ define อะเรย์แต่ละชุด จะต้องกำหนดคุณลักษณะต่อไปนี้คือ ชนิดข้อมูลของอีลิเม้นท์เหล่านั้น, มิติของอะเรย์ (dimension of the array) และจำนวนอีลิเม้นท์ทั้งหมด หรือจำนวนอีลิเม้นท์ในแต่ละมิติ ผลรวมของ จำนวนอีลิเม้นท์ทั้งหมด ในแต่ละมิติของอะเรย์ คือ จำนวนอีลิเม้นท์ทั้งหมดในอะเรย์ มิติของอะเรย์ define โดยการใช้ชุดของ dimension attribute list ตามหลังชื่อตัวแปร ในคำสั่ง DECLARE

dimension attribute list อย่างง่าย จะประกอบด้วย list ของค่าคงที่จำนวนเต็มบวก (positive integer constant) หนึ่งตัวสำหรับแต่ละมิติ เป็นการบอก จำนวนข้อมูลทั้งหมด (extent) ของ subscript สำหรับมิตินั้น แต่ละตัวให้คั่นด้วยเครื่องหมาย comma 1 ตัว และ list นี้ต้องอยู่ภายในเครื่องหมายวงเล็บ

ตัวอย่าง

```
DECLARE  A(3,4) CHARACTER(2);
```

หมายถึง การ define อะเรย์ ที่มีอีลิเม้นท์เป็น character string ความยาวเท่ากับ 2, มีมิติเท่ากับ 2 นั่นคือ จำนวนข้อมูลทั้งหมด ของมิติแรก คือ 3

---

* subscript หมายถึงตำแหน่งของอีลิเม้นท์ในอะเรย์นั้น

และจำนวนทั้งหมด ของมิติที่สองคือ 4 ดังนั้น A จึงเป็นอะเรย์ มี 3 โรว์ และ 4 คอลัมน์ อีลิเม้นท์ทุกตัว เป็น character string ความยาวเท่ากับ 2

ช่วงของค่า (range of values) ซึ่งตัว subscript สมนัยกับ มิติใด มิติหนึ่ง ปกติกำหนดโดย จำนวนอีลิเม้นท์ทั้งหมดของ มิตินั้น จากตัวอย่างข้างต้น อะเรย์ A ช่วงของ row subscript จาก 1 ถึง 3 ในขณะที่ column subscript มีช่วง จาก 1 ถึง 4 ช่วงของค่าอาจจะ define อย่าง explcitly สำหรับ subscript ของมิติใดมิติหนึ่ง โดยการแทนที่ ค่าจำนวนอีลิเม้นท์ทั้งหมด ด้วย เลขจำนวนเต็มคู่หนึ่งใน รูปแบบ m:n เมื่อ m แทนขอบเขตล่าง (lower bound) และ n แทนขอบเขตบน (upper bound) สำหรับ subscript ของมิตินั้น ค่าของ m และ n อาจจะเป็นเลข จำนวนเต็ม อะไรก็ได้ แต่ทั้งนี้ค่า m ต้องไม่มากกว่าค่า n

ตัวอย่าง

```
DECLARE  B(-2:5,-5:5,5:10) FIXED BINARY;
```

หมายถึง B เป็นอะเรย์ 3 มิติ มีช่วงของ subscripts จาก -2 ถึง 5, -5 ถึง 5 และ 5 ถึง 10 ตามลำดับ ดังนั้นอีลิเม้นท์ทั้งหมด (extents) ซึ่งสมนัยกัน คือ 8, 11 และ 6 ตามลำดับ อะเรย์ B จึงมี data items ที่เป็นชนิด fixed binary ทั้งหมด 25 ตัว

หมายเหตุ

อีลิเม้นท์ ของอะเรย์ เก็บภายในหน่วยความจำ ในลักษณะ row major order (เก็บทีละโรว์)

อีลิเม้นท์แต่ละตัวของอะเรย์ชุดหนึ่ง ถูกอ้างถึง โดยชื่อของอะเรย์ตามด้วย list ของ subscript expressions แต่ละตัวให้คั่นด้วยเครื่องหมาย comma และ ทั้งหมดนี้ ต้องอยู่ในวงเล็บ ค่าเหล่านี้จะเป็นตัวบอกตำแหน่ง ของ อีลิเม้นท์ ในอะเรย์, subscript expression ต้องมีชนิดเป็น FIXED BINARY และ ในแต่ละมิติของ

อะเรย์นั้นจะมี expression ชุดเดียว ค่าของ subscript expression แต่ละตัว ต้องอยู่ในช่วงที่กำหนด สำหรับ subscript ของมิตินั้น

ตัวแปรอะเรย์ อาจจะกำหนดค่าให้กับ ตัวแปรอะเรย์อีก หนึ่งชุดได้โดยตรง โดยไม่ต้องใช้ subscript ถ้าตัวแปรอะเรย์ ทั้งสองชุดนั้น ถูก declare ด้วย ชนิดข้อมูล, มิติ และช่วง subscript เหมือนกัน

DIMENSION, HBOUND และ LBOUND เป็น บิลท์-อิน ฟังก์ชัน ซึ่งถูกจัดให้เข้าถึง (access) จำนวนอีลิเม้นท์ทั้งหมด, ขอบเขตบน และขอบเขตล่าง ตามลำดับของแต่ละมิติของอะเรย์ชุดนั้น สำหรับรายละเอียดของฟังก์ชันเหล่านี้ ให้ดูบทที่ 12

4.2 โครงสร้าง (Structures)

โครงสร้างเป็นกลุ่ม data items ที่มีการจัดลำดับตามความสำคัญ (hierachically ordered), data items ซึ่งประกอบกันอยู่ในโครงสร้าง เรียกว่า สมาชิกของมัน (its members) และไม่จำเป็นจะต้องเป็นชนิดเดียวกัน อาจจะเป็น อะเรย์ หรือ โครงสร้างอื่น (substructures)

โครงสร้างหลัก เรียกว่า major structure ส่วนโครงสร้างย่อยอื่น ๆ เรียกว่า minor structure

การกำหนด (define) โครงสร้าง ให้ตั้งชื่อ สำหรับ โครงสร้างหลัก ชื่อ และ data attributes สำหรับสมาชิกของมัน รวมทั้งต้องมีเลขบอกระดับ (level number) สำหรับชื่อแต่ละชื่อ เพื่อกำหนดระดับของมัน ในการจัดลำดับความสำคัญ เลขบอกระดับ จะอยู่ข้างหน้าชื่อ และต้องแยกจากกันด้วย เครื่องหมาย blank ตั้งแต่ 1 ตัวขึ้นไปขั่น สำหรับ โครงสร้างหลัก เลขบอกระดับต้องเป็นเลข 1 เสมอ

การนิยาม (definitions) ของสมาชิกแต่ละตัว (รวมทั้ง เลขบอกระดับของมัน, ชื่อ และ attributes) ต้องคั่นจากกันด้วยเครื่องหมาย comma

เลขบอกระดับของสมาชิกของ โครงสร้างย่อย ต้องมากกว่า เลขบอกระดับของ โครงสร้างย่อย ชื่อโครงสร้าง ไม่จำเป็น ต้องกำหนด data type

attributes แต่อาจจะกำหนด dimension attribute, และ EXTERNAL, STATIC หรือ INITIAL attributes ได้

ตัวอย่าง การนิยามของโครงสร้างชุดหนึ่ง สำหรับการออกบิลชำระเงิน (billing account)

```
DECLARE 1 BILL
          2 NAME,
             3 LAST-NM  CHARACTER(20),
             3 FIRST-NM CHARACTER(20),
             3 MID-NM   CHARACTER(1),
          2 ADDRESS,
             3 STREET   CHARACTER(20),
             3 CITY     CHARACTER(10),
             3 STATE    CHARACTER(3),
             3 ZIP      CHARACTER(5),
          2 CHARGES,
             3 SHOP     FIXED DECIMAL(10,2),
             3 SNKBAR   FIXED DECIMAL(10,2),
             3 MISC     FIXED DECIMAL(10,2),
             3 DUES     FIXED DECIMAL(10,2);
```

เนื่องจาก ชื่อของสมาชิกแต่ละตัว ใน โครงสร้างชุดหนึ่ง อาจจะเหมือนกับ ชื่อของสมาชิก ของ โครงสร้างชุดอื่น หรือ อาจจะเป็นชื่อ ของ data item ในโครงสร้างย่อยของโครงสร้างชุดเดียวกัน ดังนั้นเมื่อมีการอ้างถึง สมาชิกของโครงสร้างนั้น จะมีความหมาย ไม่ชัดเจน (ambiguous) ความหมายคลุมเครือ เช่นนี้ เกิดขึ้นเฉพาะในกรณีที่ชื่อสมาชิก อยู่ภายใน ขอบเขตของ การนิยาม การแก้ปัญหานี้ เราใช้ qualified name ในการอ้างถึงสมาชิกของโครงสร้าง, ใน qualified name แต่ละตัว ชื่อสมาชิก จะต้อง

อยู่ข้างหลัง รายชื่อของชื่อโครงสร้าง ในลักษณะเรียงจากน้อยไปมาก (ascending order) ของตัวเลขบอกระดับ ชื่อแต่ละตัวให้ตามด้วย period 1 ตัว และอาจจะมีเครื่องหมาย blank อีกก็ได้ ชื่อโครงสร้างที่กล่าวข้างต้นนี้เท่านั้น ที่จะให้ความหมายเดียว (unique) เมื่ออ้างถึง ชื่อสมาชิก

<u>ตัวอย่าง</u> ให้พิจารณาโครงสร้างข้างล่างนี้

```
DECLARE   1 A;
            2 B,
              3 c FIXED,
              3 D FIXED,
            2 BB,
              3 c FIXED,
              3 D FIXED;
```

การอ้างถึง item C และ item D จะมีความหมายคลุมเครือ qualified name B.C หรือ B.D หรือ BB.C หรือ BB.D จึงจะมีความหมายเดียวชัดเจน เมื่อต้องการอ้างถึง อีลิเม้นท์ในโครงสร้างนั้น

<u>หมายเหตุ</u> qualified names ที่สมบูรณ์เขียนดังนี้

A.B.C

A.B.D

A.BB.C

A.BB.D

4.3 <u>อะเรย์ของโครงสร้าง</u> (Arrays of structures)

โครงสร้าง อาจจะมีสมาชิกเป็นอะเรย์ได้ ดังนั้นจึงต้องมีการ define อะเรย์ของโครงสร้าง (arrays of structures), อะเรย์ ซึ่งมีอีลิเม้นท์ เป็นชนิดเดียวของ โครงสร้าง define โดยการตั้งชื่อโครงสร้างพร้อมกับ dimension attribute

เมื่อโครงสร้างถูก defined ในทำนองเดียวกัน ในโครงสร้างย่อย ก็อาจจะกำหนด dimension attribute ได้เช่นกัน

ตัวอย่าง

```
DECLARE  1  STDLIST(100),
            2  STDNAME,
               3  LASTNM    CHARACTER(10),
               3  FIRSTNM   CHARACTER(10),
               3  MID-IN    CHARACTER(1),
            2  SSN          CHARACTER(S),
            2  COUNTRY      CHARACTER(10),
            2  GRADES(5)    CHARACTER(2);
```

หมายถึง อะเรย์ของโครงสร้าง มี STDLIST เป็น ชื่อโครงสร้างย่อย อีลิเม้นท์โครงสร้างแต่ละตัวของ อะเรย์นี้มี อะเรย์ GRADE เป็นสมาชิกอยู่ ในการอ้างถึง entry ตัวใดตัวหนึ่ง ในอะเรย์ ต้องใช้ qualified name รวมกับ subscript สำหรับชื่อโครงสร้าง ซึ่งมี dimension attribute และชื่อสมาชิก ถ้ามันมี dimension attribute ตัว subscripts ไม่จำเป็นต้องอยู่ตรงตำแหน่งที่สัมพันธ์กับชื่อ แต่จะต้องอยู่ในวงเล็บ แต่ละตัวคั่นด้วยเครื่องหมาย comma และต้องปรากฏในลำดับที่ถูกต้อง

ตัวอย่าง การอ้างถึง entry GRADE ตัวที่ 3 สำหรับ entry ของอะเรย์ STDLIST ตัวที่ 61 จะเขียนในรูปแบบอย่างใดอย่างหนึ่งก็ได้ข้างล่างนี้

```
STDLIST(61).GRADE(3)
STDLIST.GRADE (61,3)
STDLIST (61,3).GRADE
```