

### การศึกษาโครงสร้างของภาษาโคบอล

การศึกษาโครงสร้างของภาษาโคบอลในการวิจัยครั้งนี้ ได้ศึกษาจากภาษาโคบอลของบริษัท ไมโครซอฟท์ เวอร์ชัน 2.2 ในที่นี้จะขอกล่าวถึงโดยสังเขป โดยเน้นในส่วนที่เกี่ยวข้องกับเพิ่มข้อมูล และส่วนที่เกี่ยวข้องกับหน้าจอ

โปรแกรมภาษาโคบอลโดยทั่วไป ประกอบด้วย 4 ส่วน ดังต่อไปนี้

#### 3.1 IDENTIFICATION DIVISION

เป็นส่วนที่ใช้ระบุชื่อ และเป็นเอกสารสำหรับโปรแกรม

#### 3.2 ENVIRONMENT DIVISION

เป็นส่วนที่ใช้ระบุอุปกรณ์คอมพิวเตอร์ที่ใช้ในโปรแกรม

#### 3.3 DATA DIVISION

เป็นส่วนที่ใช้กำหนดชื่อ และลักษณะของข้อมูลที่จะใช้ในโปรแกรม

#### 3.4 PROCEDURE DIVISION

เป็นส่วนที่ประกอบด้วยคำสั่งที่จะใช้ในการประมวลผลตามต้องการ เนื่องจากจากรายละเอียดของคำสั่งในภาษาโคบอลมีมาก ในที่นี้จะขอกล่าวถึงเพียงบางคำสั่งที่ใช้เกี่ยวกับหน้าจอเท่านั้น ซึ่งได้แก่

- คำสั่ง ACCEPT
- คำสั่ง DISPLAY

สำหรับรายละเอียดของคำสั่งอื่น ๆ สามารถศึกษาเพิ่มเติมได้จากหนังสือ

Microsoft COBOL Reference Manual Ver.2.2

### 3.1 IDENTIFICATION DIVISION

ส่วนนี้เป็นส่วนสำหรับเขียนแนะนำเกี่ยวกับโปรแกรมนี้ การเขียนส่วนนี้จะเริ่มด้วยการเขียนชื่อของ Division จากนั้นก็จะเป็นชื่อโปรแกรม ชื่อผู้เขียนโปรแกรม วันเดือนปี ตลอดจนรายละเอียดอื่น ๆ ดังต่อไปนี้

#### IDENTIFICATION DIVISION .

PROGRAM-ID . program-name .

[AUTHOR . [comment-entry] ... ]

[INSTALLATION . [comment-entry] ... ]

[DATE-WRITTEN . [comment-entry] ... ]

[DATE-COMPILED . [comment-entry] ... ]

[SECURITY . [comment-entry] ... ]

### 3.2 ENVIRONMENT DIVISION

สำหรับส่วนนี้เป็นส่วนที่ใช้แสดงรายละเอียดเกี่ยวกับเครื่องคอมพิวเตอร์ที่ใช้ในการเขียนโปรแกรมนี้ ตลอดจนอุปกรณ์รับและแสดงผลที่ใช้ในโปรแกรมและรายละเอียดการเชื่อมโยงโปรแกรมกับเครื่องคอมพิวเตอร์ ดังต่อไปนี้

#### ENVIRONMENT DIVISION .

[CONFIGURATION SECTION .

[SOURCE-COMPUTER . source-computer-entry]

[OBJECT-COMPUTER . object-computer-entry]

[SPECIAL-NAMES .. special-names-entry]

[INPUT-OUTPUT SECTION .

[FILE-CONTROL .

ถ้าเป็นแฟ้มข้อมูลแบบลำดับ (Sequential file) ใช้

SELECT file-name

ASSIGN TO { DISK  
PRINTER }

[;ORGANIZATION IS [LINE] SEQUENTIAL]

[;ACCESS MODE IS SEQUENTIAL]

[;FILE STATUS IS data-name-1] .]

ถ้าเป็นแฟ้มข้อมูลแบบสัมพันธ์ (Relative file) ใช้

SELECT file-name

ASSIGN TO DISK

[;ORGANIZATION IS RELATIVE

[;ACCESS MODE IS { SEQUENTIAL  
RANDOM  
DYNAMIC }

RELATIVE KEY IS data-name-2 ]

[;FILE STATUS IS data-name-1] .]

ถ้าเป็นแบบตามลำดับที่มีตัวชี้ (Indexed file) ใช้

SELECT file-name

ASSIGN TO DISK

[;ORGANIZATION IS INDEXED

[;ACCESS MODE IS { SEQUENTIAL  
RANDOM  
DYNAMIC }

RECORD KEY IS data-name-3

[;FILE STATUS IS data-name-1] .]

[I-O-CONTROL . input-output-control-entry]

### 3.3 DATA DIVISION

ส่วนนี้ใช้แสดงลักษณะของแฟ้มข้อมูลที่ใช้ และชนิดของข้อมูลในแฟ้มข้อมูล รายละเอียดของข้อมูลที่อ่านเข้ามา และข้อมูลที่ระงับ นอกจากนี้ยังมีรายละเอียดเกี่ยวกับการจองเนื้อที่ การเก็บข้อมูลเพื่อใช้ในการทำงานของโปรแกรม ดังต่อไปนี้

DATA DIVISION .

[FILE SECTION .

[FD file-name

[ ;BLOCK CONTAINS [integer-1 TO] integer-2 {RECORDS  
CHARACTERS} ]

;LABEL {RECORD IS  
RECORDS ARE} {STANDARD  
OMITTED}

[ ;VALUE OF FILE-ID IS {data-name-1  
literal-1} ]

[ ;DATA {RECORD IS  
RECORDS ARE} data-name-2 [,data-name-3] ... ]

{record-description-entry} ... ] ...

[WORKING-STORAGE SECTION .

77-level-description-entry

record-description-entry ... ]

[LINKAGE SECTION .

77-level-description-entry

record-description-entry ... ]

[SCREEN SECTION .

{level-number }screen-name]

[BLANK SCREEN] -

[LINE NUMBER IS [PLUS] integer-1]

[COLUMN NUMBER IS [PLUS] integer-2]



[BACKGROUND-COLOR integer-3]

[BACKGROUND-COLOR integer-4]

[BLANK LINE]

[BELL]

[UNDERLINE]

[REVERSE-VIDEO]

[HIGHLIGHT]

[BLINK]

{ [VALUE] IS literal-1 }

{ [PICTURE IS character-string { [FROM { literal-2 } identifier-1 ] } ] }  
{ [PIC { [TO identifier-2 ] } { [USING identifier-3 ] } ] }

[BLANK WHEN ZERO]

[ { [JUSTIFIED] RIGHT ]  
[ { JUST } ] ]

[AUTO]

[SECURE]

[REQUIRED]

[FULL] . } ...

### 3.4 PROCEDURE DIVISION

เป็นส่วนที่ใช้แสดงการทำงาน การวิเคราะห์ข้อมูล การแก้ปัญหาต่าง ๆ การเคลื่อนย้ายข้อมูล การคำนวณ การเปรียบเทียบ การอ่าน และการพิมพ์ข้อมูล เป็นต้น รูปแบบของ Procedure Division เป็นดังนี้

PROCEDURE DIVISION .[section-name-1 SECTION .]

paragraph-name-1 .

sentence-1 .

sentence-2 .

.

.

paragraph-name-2 .

sentence-1 .

.

.

[section-name-2 SECTION .]

paragraph-name-n .

.

.

ชนิดของคำสั่งที่ใช้เขียนใน Procedure Division แบ่งเป็น 3 ชนิดคือ

## 3.4.1 คำสั่งบอกเล่า (Imperative Statement)

เป็นคำสั่งให้คอมพิวเตอร์ทำงานตามที่สั่งโดยไม่มีเงื่อนไขใด ๆ

คำสั่งดังกล่าวได้แก่

3.4.1.1 คำสั่งในการคำนวณ (Arithmetic) เช่น ADD, MULTIPLY เป็นต้น

3.4.1.2 คำสั่งการจัดข้อมูล (Data Manipulation) เช่น MOVE, STRING เป็นต้น

3.4.1.3 คำสั่งแยกวิธีการ (Procedure Branching) เช่น GO TO, PERFORM เป็นต้น

3.4.1.4 คำสั่งรับข้อมูล/แสดงผล  
เป็นคำสั่งที่ใช้สำหรับรับข้อมูลจากอุปกรณ์ภายนอก และการแสดงข้อมูลออกไปยังอุปกรณ์ที่เหมาะสม คำสั่งพวกนี้แบ่งออกเป็น 2 กลุ่ม คือ

- คำสั่งที่ใช้กับแฟ้มข้อมูล (File-oriented input/output) ได้แก่ OPEN, CLOSE, READ, WRITE เป็นต้น

- คำสั่งที่ไม่ต้องใช้เพิ่มข้อมูล (Data-oriented input/output) ได้แก่ ACCEPT, DISPLAY ดังรูปแบบต่อไปนี้

```

ACCEPT identifier FROM {
    DATE
    DAY
    TIME
    LINE NUMBER
    ESCAPE KEY
}

ACCEPT screen-name [ON ESCAPE imperative-statement]

```

```

DISPLAY { identifier-1 } { identifier-2 } ...
        { literal-1   } { literal-2   }
                                     [UPON mnemonic-name]

```

```

DISPLAY ((position-spec) { identifier }
        { literal         } }
        { ERASE           }

```

```

DISPLAY screen-name

```

### 3.4.2 คำสั่งเงื่อนไข (Condition Statement)

เป็นคำสั่งการทำงานที่ต้องมีการตรวจสอบเงื่อนไขที่กำหนดไว้ว่าเป็นจริงหรือไม่ ผลจากการตรวจสอบจะทำให้การทำงานแยกเป็น 2 ส่วน คือ ส่วนการทำงานเมื่อเงื่อนไขเป็นจริง กับการทำงานสำหรับส่วนที่เงื่อนไขไม่เป็นจริง ดังนั้นลำดับและขั้นตอนการทำงานของโปรแกรมจะขึ้นอยู่กับเงื่อนไขต่าง ๆ ในคำสั่งเหล่านี้ เช่น คำสั่ง IF, คำสั่ง READ ที่มีเงื่อนไข (READ... AT END...) หรือคำสั่งคำนวณที่มีเงื่อนไข (ADD... ON SIZE ERROR...)

### 3.4.3 คำสั่งชี้แนะคอมไพเลอร์ (Compiler Directive)

เป็นคำสั่งที่ใช้สำหรับการกำหนดหรือชี้ให้คอมไพเลอร์ทำการอย่างใดอย่างหนึ่ง เป็นคำสั่งที่ไม่มีเงื่อนไข เช่น COPY, USE เป็นต้น