บทที่ 2

ภาษาระดับที่ 4 และ การออกแบบภาษาโปรแกรมโดยทั่ว ๆ ไป

ภาษาระดับที่ 4

ภาษาระดับนี้เป็นภาษาที่เหมาะกับผู้ใช้ทั่ว ๆ ไป ทำให้ผู้ใช้สามารถที่จะเรียนรู้ได้ง่ายและรวดเร็ว และสามารถสร้างระบบงานขึ้นมาเองอย่างมีประสิทธิภาพซึ่งภาษาระดับที่ 4 มีคุณสมบัติดังต่อไปนี้ [3]

- เป็นภาษาที่ช่วยลดงานและลดเวลาในการสร้างระบบงาน โดยทั่วไปในการสร้างระบบงานจะประกอบด้วยขั้นตอนสำคัญคือ การออกแบบระบบ การพัฒนาระบบ และ การทดสอบระบบ ซึ่งสำหรับขั้นตอนการพัฒนาระบบนั้นถ้าใช้โปรแกรมภาษาระดับที่ 4 ในการพัฒนา จะต้องใช้เวลาน้อยกว่าการใช้ภาษาระดับสูงทั่ว ๆ ไป
- เป็นภาษาที่เหมาะสมกับผู้ใช้ทั่ว ๆ ไป โดยที่ผู้ใช้จะต้องสามารถเรียนรู้ได้ง่ายรวดเร็ว และ สามารถใช้งานได้โดยสะดวก
- การพัฒนาโปรแกรมลักษณะเดียวกัน โปรแกรมที่เขียนขึ้นโดยภาษาระดับที่ 4 ควรมีจำนวนคำสั่งน้อยกว่าเมื่อเทียบกับภาษาระดับสูงทั่ว ๆ ไป เช่น ภาษาโคบอล ภาษาเบสิก
- ภาษาระดับที่ 4 จะต้องฉลาดพอที่จะกำหนดข้อมูลบางอย่างให้แก่งานของผู้ใช้โดยปริยายถ้าผู้ใช้ไม่ได้กำหนดเพื่อเพิ่มความสะดวกแก่ผู้ใช้ เช่น ถ้าผู้ใช้ไม่ได้กำหนดขนาดของจอภาพที่ต้องการ ภาษาระดับที่ 4 ก็ควรจะกำหนดรูปแบบมาตรฐานให้เช่นกำหนดให้เป็นขนาด 24 บรรทัด 80 คอลัมน์ โดยปริยาย
- เป็นภาษาที่ทำให้ได้ผลลัพธ์ตามที่ต้องการ ในการใช้ภาษาระดับที่ 4 สร้างระบบงานนั้นจะต้องได้ผลลัพธ์ตามที่ผู้ใช้ต้องการด้วย

1. <u>ประเภทของภาษาระดับที่ 4</u> ประเภทของภาษาระดับที่ 4 สามารถแบ่งได้เป็นหลายประเภทดังต่อไปนี้ [3]

1.1 ภาษาที่ใช้สอบถามข้อมูลอย่างง่าย (Simple Query Language) จะจัดเก็บระเบียนข้อมูลที่จะถูกเรียกใช้ให้อยู่ในรูปแบบที่เหมาะสม พร้อมที่จะอ่านออกมารายงานได้ทันทีที่ผู้ใช้ต้องการ

1.2 ภาษาที่ใช้สอบถามข้อมูลซับซ้อน (Complex Query and Update Language) จะจัดเก็บข้อมูลให้อยู่ในรูปแบบของฐานข้อมูล การอ่านข้อมูลออกมาทำรายงานนั้น ข้อมูลอาจจะจัดเก็บอยู่ต่างแฟ้มกันและสามารถทำการเปลี่ยนแปลงแก้ไขข้อมูลในแฟ้มข้อมูลต่าง ๆ ได้ด้วย

1.3 ภาษาที่ช่วยในการจัดทำรายงาน (Report Generator) เป็นภาษาที่ช่วยในการดึงข้อมูลออกมาทำรายงานตามรูปแบบต่าง ๆ ที่ผู้ใช้ต้องการ

1.4 ภาษาที่เกี่ยวกับกราฟ (Graphics Language) เป็นภาษาที่ช่วยในการนำข้อมูลออกมาทำรายงาน แต่รายงานจะอยู่ในลักษณะของกราฟต่าง ๆ

1.5 ภาษาที่ช่วยในการตัดสินใจ (Decision Support Tools) โดยทั่วไปแล้วภาษานี้จะใช้กับงานทางด้านวิเคราะห์ต่าง ๆ เช่น การวิเคราะห์ด้านการเงิน ด้านการลงทุน หรือการวางแผนต่าง ๆ

1.6 ภาษาที่ช่วยในการสร้างงาน (Application Generator) เป็นภาษาที่ให้ผู้ใช้ระบุผลลัพธ์ที่ต้องการ พร้อมทั้งกำหนดแฟ้มข้อมูลหรือฐานข้อมูลต่าง ๆ ว่าจะถูกเปลี่ยนแปลงแก้ไข ต้องทำงานทางด้านคำนวณหรือ ตรรกศาสตร์อะไรบ้าง จากนั้นภาษานี้ก็จะทำการสร้างงานให้ได้ผลลัพธ์ตามที่ผู้ใช้ต้องการได้

1.7 ภาษาระดับสูงมาก (Very High Level Programming Language) เป็นภาษาที่สามารถใช้งานได้ง่าย มีความผิดพลาดน้อย มีการกำหนดค่าบางอย่างให้โดยปริยาย และ ผู้ใช้สามารถเปลี่ยนแปลงแก้ไขได้ง่ายตามที่ผู้ใช้ต้องการ

1.8 ภาษาที่อยู่ในลักษณะชุดคำสั่งสำเร็จรูป(Parameterlized Application Packages) เป็นภาษาที่มีลักษณะคล้ายโปรแกรมสำเร็จรูป ซึ่งมักจะใช้กับงานเฉพาะอย่าง และ มีพารามิเตอร์ต่าง ๆ ให้ผู้ใช้เลือกใช้ เมื่อผู้ใช้

ต้องการเปลี่ยนแปลงแก้ไขระบบก็สามารถทำได้โดยง่าย

2. ภาษาที่ช่วยในการจัดทำรายงาน (Report Generator) เป็นภาษาระดับที่ 4 ประเภทหนึ่งเพื่อช่วยในการจัดทำรายงาน ซึ่งจะมีคุณสมบัติต่าง ๆ ดังต่อไปนี้ (3)

2.1 สามารถกำหนดข้อมูลบางอย่าง ให้แก่รายงานของผู้ใช้โดยปริยาย ถ้าผู้ใช้ไม่ได้กำหนดเพื่อเพิ่มความสะดวกแก่ผู้ใช้ เช่น การกำหนดหมายเลขหน้าเริ่มต้นของรายงาน การกำหนดรูปแบบของข้อมูล หรือ การกำหนดขนาดของรายงาน ซึ่งถ้าภาษาที่ช่วยในการจัดทำรายงานนี้สามารถกำหนดค่าต่าง ๆ เหล่านี้ให้กับรายงานได้ ก็จะทำให้เกิดความสะดวกสบายขึ้นมากในการสร้างรายงาน

2.2 รายงานนั้นควรสามารถแสดงให้เห็นบนจอภาพได้ก่อน เพื่อที่ผู้ใช้จะได้พิจารณาดูว่าตรงตามความต้องการของเขาหรือไม่ ถ้าไม่ตรงตามที่ต้องการก็จะได้ทำการแก้ไขเปลี่ยนแปลง ก่อนที่จะทำการพิมพ์ออกทางเครื่องพิมพ์ต่อไป

2.3 รูปแบบของข้อมูลที่จะใช้ในการจัดทำรายงานนั้นควรมีหลายรูปแบบเพื่อที่ผู้ใช้จะได้เลือกรูปแบบการพิมพ์ที่เหมาะสมกับรายงานที่ต้องการได้ เช่น การตัดศูนย์นำหน้า (Zero Suppression) การแทนที่ศูนย์ที่อยู่ข้างหน้าด้วยสัญญลักษณ์ต่าง ๆ เช่น เครื่องหมายดอลล่าร์ ($) การแทรกเครื่องหมายจุลภาค (Comma) และอื่น ๆ

2.4 ข้อมูลที่จะใช้ในการทำรายงานนั้น นอกจากจะอ่านข้อมูลมาจากแฟ้มข้อมูลต่าง ๆ แล้ว ควรจะสามารถรายงานข้อมูลที่เกิดจากการคำนวณได้ด้วย เช่นการคำนวณตามสูตรต่าง ๆ ที่ผู้ใช้ระบุลงไปในรูปแบบรายงานที่ต้องการ เช่นผู้ใช้ต้องการให้รายงานค่าภาษี ซึ่งภาษีนี้สามารถคำนวณได้จากภาษี เท่ากับ 1 เปอร์เซนต์ของเงินเดือน

2.5 สามารถนำข้อมูลมาสร้างรายงานตามเงื่อนไขต่าง ๆ ได้ เช่น เท่ากัน ไม่เท่ากัน มากกว่า มากกว่าหรือเท่ากับ น้อยกว่า น้อยกว่าหรือเท่ากับ ซึ่งจะทำให้ผู้ใช้สามารถนำข้อมูลที่มีคุณสมบัติตรงตามเงื่อนไขที่ต้องการมาสร้างรายงานได้ เช่น ผู้ใช้ต้องการสร้างรายงานเฉพาะบุคคลที่มีเงินเดือนมากกว่า 10000 แต่น้อยกว่า 20000 ได้

2.6 สามารถทำการคำนวณหาค่าต่าง ๆ ต่อไปนี้ได้ เนื่องจาก

ค่าต่าง ๆ เหล่านี้เป็นค่าที่มีการระบุไว้ในรูปแบบรายงานโดยทั่ว ๆ ไป

- ค่าสถิติต่าง ๆ
- ค่าผลรวมต่าง ๆ เช่นผลรวมย่อย (Subtotal) ผลรวมทั้งหมด (Grand total)
- ค่าสะสมต่าง ๆ (Accumulator)
- ค่าเปอร์เซนต์ (Percent)

2.7 ผู้ใช้สามารถกำหนดค่าต่าง ๆ ต่อไปนี้เพื่อใช้ในการสร้างรายงานได้

- การเปลี่ยนค่าในเขตข้อมูล (Control Break)
- จำนวนบรรทัดต่อหน้าของรายงาน และจำนวนตัวอักษรต่อบรรทัดของรายงาน
- ความกว้างของแต่ละบรรทัด หรือแต่ละคอลัมน์
- หมายเลขหน้า
- รูปแบบรายงานที่ผู้ใช้ต้องการ เช่น หัวรายงานต่าง ๆ รายการข้อมูลที่ผู้ใช้ต้องการพิมพ์ และการพิมพ์สรุปท้ายต่าง ๆ ของรายงาน

การออกแบบภาษาโปรแกรมโดยทั่ว ๆ ไป

ในการออกแบบภาษาคอมพิวเตอร์จะต้องมีการกำหนดกฎเกณฑ์ที่เรียกว่าหลักไวยากรณ์มาช่วยเพื่อตรวจวิเคราะห์และสร้างประโยค หลักไวยากรณ์ที่มีการใช้กันมากคือไวยากรณ์ไม่พึ่งบริบท (Context Free Grammar) และ ไวยากรณ์แบบ บีเอ็นเอฟ (BNF) เพราะเป็นหลักไวยากรณ์ที่สร้างเพื่ออธิบายโครงสร้างของภาษา และ ความสัมพันธ์ระหว่างโครงสร้างต่าง ๆ ที่ประกอบในประโยคนั้น ๆ โดยไม่พิจารณาในเชิงความหมายของประโยค

1. ไวยากรณ์แบบไม่พึ่งบริบท

ไวยากรณ์แบบไม่พึ่งบริบทนี้มีส่วนประกอบ 4 ส่วนคือ [4]

1.1 เทอร์มินอล (Terminals) คือกลุ่มอักขระที่แบ่งแยกต่อไปไม่ได้อีกแล้วเช่นคำสำคัญ (Keyword) และ ตัวดำเนินการ (Operator) เมื่อดูจากกฎในไวยากรณ์คือกลุ่มอักขระที่อยู่แต่ทางขวามือของกฎเพียงอย่างเดียว จะไม่อยู่ทางซ้ายมือของกฎเลย ตัวดำเนินการมีอยู่ด้วยกันหลายประเภทคือ

- ตัวดำเนินการคำนวณ (Arithmetic Operator) ประกอบด้วยบวก (+) ลบ (-) คูณ (*) หาร (/) ยกกำลัง (^ หรือ **)

- ตัวดำเนินการเปรียบเทียบ (Relational Operator) ประกอบด้วย มากกว่า (>) น้อยกว่า (<) มากกว่าหรือเท่ากับ (>=) น้อยกว่าหรือเท่ากับ (<=) เท่ากับ (=) ไม่เท่ากับ (<> หรือ ><)

- ตัวดำเนินการทางตรรก (Logical Operator) ประกอบด้วย และ (AND) หรือ (OR) ไม่ (NOT)

1.2 นอนเทอร์มินอล (Nonterminals) คือกลุ่มอักขระที่ยังแบ่งแยกต่อไปได้อีก ส่วนย่อยเหล่านั้นอาจเป็นเทอร์มินอลหรือนอนเทอร์มินอลก็ได้ ถ้าดูจากกฎก็คือหน่วยที่จะต้องอยู่ทางขวามือของกฎอย่างน้อยหนึ่งกฎ

1.3 สัญญลักษณ์เริ่มต้น (Start Symbol) คือนอนเทอร์มินอลที่ถูกเลือกเป็นตัวแรกสุดที่จะพิจารณา

1.4 กฎที่ใช้เปลี่ยนแปลง (Production Rules) คือกฎที่ใช้กำหนดว่านอนเทอร์มินอลจะประกอบด้วยนอนเทอร์มินอล หรือเทอร์มินอลอะไรบ้าง กฎทุกกฎจะประกอบด้วยนอนเทอร์มินอลอยู่ทางซ้ายมือ ตามด้วยลูกศร และทางขวามือประกอบด้วยเทอร์มินอล หรือ นอนเทอร์มินอล

ในการแยกกฎแต่ละกฎของสัญญลักษณ์เริ่มต้นตัวเดียวกันจะใช้เครื่องหมาย ¦ ในการแยกแต่ละกฎ

ตัวอย่าง กำหนดให้ S เป็น สัญญลักษณ์เริ่มต้น

A B เป็น นอนเทอร์มินอล

a b c เป็น เทอร์มินอล

S -> AB หมายความว่า S ประกอบไปด้วย A ตามด้วย B

A -> aA¦a หมายความว่า A ประกอบไปด้วย a ตามด้วย A หรือ ประกอบไปด้วย a เท่านั้น

B -> Bb¦b หมายความว่า B ประกอบไปด้วย B ตามด้วย b หรือประกอบไปด้วย b เท่านั้น

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

2. ไวยากรณ์แบบบีเอ็นเอฟ (Backus Normal Form)

ลักษณะของไวยากรณ์แบบบีเอนเอฟ [5] ประกอบไปด้วยกฎต่าง ๆ แต่ละกฎจะมีสัญญลักษณ์ที่ถูกแทนที่ได้หนึ่งตัวปรากฏทางซ้ายของเครื่องหมาย ::= ส่วนทางขวาของเครื่องหมายประกอบไปด้วยสัญญลักษณ์ที่ถูกแทนที่ได้ กับสัญญลักษณ์ที่ถูกแทนที่ไม่ได้ สัญญลักษณ์ที่ถูกแทนที่ได้จะแทนด้วยเครื่องหมาย < > นอกจากนั้นถือว่าเป็นสัญญลักษณ์ที่ถูกแทนที่ไม่ได้

ในการแยกกฎแต่ละกฎ โดยที่กฎเหล่านั้นมีสัญญลักษณ์ที่ถูกแทนที่ได้ทางซ้ายมือเหมือนกันจะใช้เครื่องหมาย ¦ ในการแยกแต่ละกฎ

ตัวอย่าง <S> ::= <A><B> หมายความว่า S ถูกแทนที่ด้วย A ตามด้วย B

<A> ::= a<A>¦a หมายความว่า A ถูกแทนที่ด้วย a ตามด้วย A หรือถูกแทนที่ได้ด้วย a

<B> ::= <B>b¦b หมายความว่า B ถูกแทนที่ด้วย B ตามด้วย b หรือถูกแทนที่ได้ด้วย b

กรณีที่เขียนกฎรวมกันแล้ว ปรากฏว่าทางขวามือของเครื่องหมาย ::= ที่เขียนแยกด้วยเครื่องหมาย ¦ มีมากเกินไป แต่เป็นที่เข้าใจว่าข้อความเหล่านั้นคืออะไร สามารถเขียนย่อได้ด้วยเครื่องหมายจุดไข่ปลา (...)

ตัวอย่าง <LETTER> ::= A¦B...¦Z

นอกจากนี้ยังมีไวยากรณ์แบบอีบีเอ็นเอฟ (Extended Backus Normal Form) ซึ่งมีคุณสมบัติเหมือนกับไวยากรณ์แบบบีเอ็นเอฟทุกอย่างแต่จะมีเครื่องหมายเพิ่มเติมจากสัญญลักษณ์บีเอ็นเอฟดังนี้

กำหนดให้ X เป็น สตริงใด ๆ

2.1 {X} หมายถึง ไม่มี X หรือ มี X ตั้งแต่ 1 ตัวขึ้นไป

ตัวอย่าง <t> ::= <f>¦<t>*<f>

สามารถเขียนได้เป็น

<t> ::= <f>¦{*<f>}

2.2 $\{X\}_m^n$ หมายถึงมี X อย่างน้อย m ตัวแต่ต้องไม่เกิน n ตัว

กรณีไม่เขียน m จะหมายถึง m=0 คือไม่มี X ก็ได้

ตัวอย่าง <name> ::=<letter>{<let_dig>}$_0^\infty$

<let-dig>::=<letter>|<digit>

<letter> ::= A|B...Z

<digit> ::= 0|1...9

2.3 [X] หมายถึงไม่มี X หรือมี X เพียง 1 ตัว

ตัวอย่าง <t> ::= <f>[g]

2.4 ( ) ใช้เมื่อต้องการเขียนย่อเพื่อเปลี่ยนลำดับการต่อของสตริง

ตัวอย่าง <b> ::= <a><c>|<a><d>

สามารถเขียนย่อเป็น

<b> ::= <a>(<c>|<d>)

2.5 กรณีสัญญลักษณ์ไวยากรณ์อีบีเอ็นเอฟตรงกันกับสัญญลักษณ์ที่ถูกแทนที่ไม่ได้ให้เขียนสัญญลักษณ์ที่ถูกแทนที่ไม่ได้ภายในเครื่องหมาย ' ' และ ถ้าสัญญลักษณ์ที่ถูกแทนที่ไม่ได้คือเครื่องหมาย ' ให้เขียนเครื่องหมาย ' 2 ตัวแทน 1 ตัว

ตัวอย่าง <sub-var>::=<id>(<sub-list>)

ในกรณีนี้ (และ) เป็นสัญญลักษณ์ที่ถูกแทนที่ไม่ได้เขียนแทนด้วย <sub-var>::=<id> '(' <sub-list> ')'

ตัวอย่าง <literal>::=' <any char> '

ในกรณีนี้ ' เป็นสัญญลักษณ์ที่ถูกแทนที่ไม่ได้เขียนแทนด้วย <literal>::='''' <any char> ''''