สาระการเรียนรู้

1. ความหมายของขั้นตอนวิธี *(Algorithm)*
2. ความหมายของโปรแกรมคอมพิวเตอร์
3. ความหมายของตัวแปร
4. ความหมายของค่าคงที่
5. ความหมายของผังงาน
6. ความสำคัญของผังงาน
7. ความหมายของลูปแบบต่าง ๆ
8. ความหมายของโปรแกรมซูโดโค้ด
9. การเขียนผังงานจากโจทย์ที่กำหนดให้
10. การเขียนโปรแกรมซูโดโค้ดของผังงาน

ผลการเรียนรู้ที่คาดหวัง

เมื่อเรียนจบบทนี้แล้ว ผู้เรียนสามารถ

1. อธิบายความหมายของขั้นตอนวิธี (Algorithm) ได้
2. อธิบายความหมายของโปรแกรมคอมพิวเตอร์ได้
3. อธิบายความหมายของตัวแปรได้
4. อธิบายความหมายของค่าคงที่ได้
5. อธิบายความหมายของผังงานได้
6. อธิบายความสำคัญของผังงานได้
7. อธิบายความหมายของลูปแบบต่าง ๆ ได้
8. อธิบายความหมายของโปรแกรมซูโดโค้ดได้
9. เขียนผังงานจากโจทย์ที่กำหนดให้ได้
10. เขียนโปรแกรมซูโดโค้ดของผังงานได้

สาระสำคัญ

วิธีการทางคอมพิวเตอร์ (Algorithm) ลำดับขั้นตอนของรายการคำสั่ง สำหรับการแก้ปัญหาใดปัญหาหนึ่งเพื่อกำหนดขั้นตอนให้คอมพิวเตอร์ได้ทำงานตามวัตถุประสงค์ที่ต้องการ ซึ่งมีเครื่องมือสำหรับสร้างหรือพัฒนาโปรแกรม ก็คือ ผังงาน (Flow Chart) หรือการเขียนด้วยภาษาซูโดโค้ด (Psuedo Code Language)

อัลกอริธึม (Algorithm)

หมายถึง ลำดับขั้นตอนของรายการคำสั่งสำหรับการแก้ปัญหาใดปัญหาหนึ่ง

ตัวอย่างที่ 1

ธนาคารชาติต้องการคำนวณและบันทึกดอกเบี้ย *(INTEREST)* ของการจำนองบ้านของลูกค้าบางคน สมมติให้ยอดเงิน *(BALANCE)* ของการจำนองและอัตราดอกเบี้ย *(RATE)* ปรากฏในระเบียนของลูกค้า

ลำดับขั้นตอนควรเป็นดังนี้

ขั้นที่ 1 กำหนดงานให้ *NAME, BALANCE, RATE* เป็นรายการในระเบียนของลูกค้า

ขั้นที่ 2 คำนวณ *INTEREST = BALANCE x RATE*

ขั้นที่ 3 บันทึก *NAME* และ *INTEREST* ของลูกค้าในแฟ้มข้อมูลดอกเบี้ย

การเสนอลำดับขั้นตอนการทำงานของรายการคำสั่ง

นิยมใช้กัน 2 วิธีคือ

1. การใช้ผังงาน *(Flow Chart)*

1. ช่วยให้ผู้เขียนโปรแกรมสะดวกขึ้น
2. การตรวจสอบข้อผิดพลาดในโปรแกรมทำได้ง่าย
3. เขียนโปรแกรมได้อย่างมีประสิทธิภาพ
4. ประหยัดเวลาและหน่วยความจำของคอมพิวเตอร์
5. เป็นเอกสารประกอบโปรแกรม
6. ง่ายต่อการปรับปรุงแก้ไข
7. ช่วยให้อ่านและเข้าใจโปรแกรมได้ง่ายขึ้น

การเสนอลำดับขั้นตอนการทำงานของรายการคำสั่ง

2. การใช้ภาษาซูโดโค้ด

(Pseudo Code Language)

เป็นการเสนอลำดับขั้นที่นิยมมากอีกวิธีหนึ่ง

การเขียนโปรแกรมคอมพิวเตอร์ อาจเขียนโปรแกรมภาษาระดับสูง เช่น ภาษา *BASIC, C, COBOL, PASCAL,* หรือ ภาษา *FORTRAN* ก็ได้

โปรแกรมคอมพิวเตอร์ (Computer Program)

คอมพิวเตอร์จะแก้ปัญหาใด ๆ ด้วยวิธีการของโปรแกรมคอมพิวเตอร์ ซึ่งต้องอาศัยขั้นตอนที่ละเอียดและชัดเจน โดยนำขั้นตอนเหล่านี้มาเขียนเป็นคำสั่งให้คอมพิวเตอร์ทำงาน คำสั่งแต่ละคำสั่งเขียนให้คอมพิวเตอร์ทำงานนั้นเรียกว่า ***S t a t e m e n t***

โปรแกรม (Program)

คือ ชุดของคำสั่งหลาย ๆ คำสั่งที่ให้คอมพิวเตอร์ทำงานโดยมีวัตถุประสงค์อย่างใดอย่างหนึ่ง

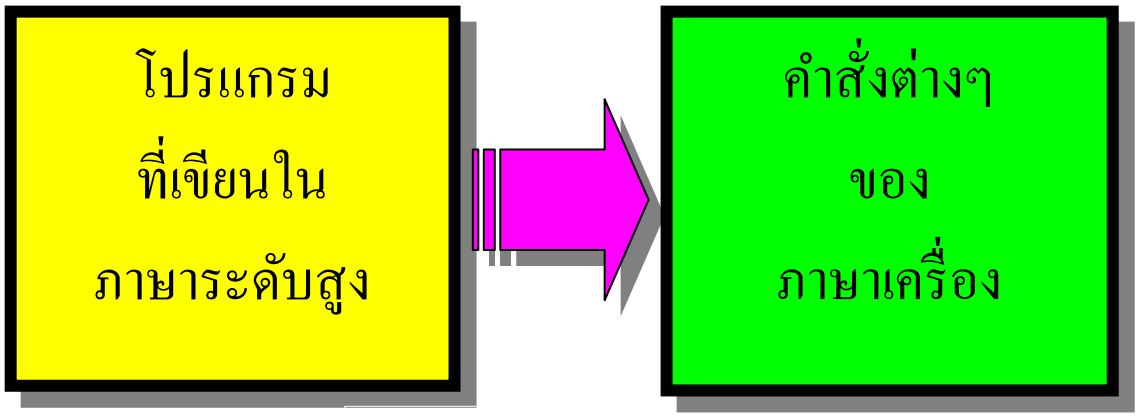
คอมพิวเตอร์จ ะทำงานตามคำสั่ง ทีละ คำสั่งตามลำดับขั้นตอนของคำสั่งที่เรียงกันลงมา

โปรแกรม (Program)

คำสั่งต่าง ๆ ในภาษาระดับสูงอาจเรียกว่า โปรแกรมต้นฉบับ *(Source Program)* ซึ่งโปรแกรมนี้คอมพิวเตอร์ไม่สามารถนำไปใช้งานได้ ต้องมีการเปลี่ยนแปลงหรือคอมไพล์ให้เป็นคำสั่งของภาษาเครื่องที่เรียกว่า ออบเจ็กต์โปรแกรม *(Object Program)* ก่อน

ภาพแสดงรูปแบบการประมวลผลของโปรแกรมคอมพิวเตอร์

Source Program *Object Program*



ตัวแปลภาษา

Compiler

ตัวแปร (Variable)

หมายถึง ชื่อหรือตัวแปรอักษรที่ใช้เรียกที่เก็บข้อมูลในหน่วยความจำ

- ทำให้เกิดความสะดวกแก่หน่วยควบคุมของเครื่องที่จะเรียกข้อมูลทำตามคำสั่งได้ง่าย
- โปรแกรมเมอร์จะเป็นผู้ตั้งขึ้นเอง
- ตั้งให้สอดคล้องกับประเภทของข้อมูล เช่น

ตัวแปร INTERST	*ใช้เก็บข้อมูลเกี่ยวกับ ดอกเบี้ย*
ตัวแปร RATE	*ใช้เก็บข้อมูลที่เกี่ยวกับ อัตราดอกเบี้ย เป็นต้น*

- ชื่อตัวแปรต้องไม่เป็นคำเฉพาะ *(Reserved Words)*

ค่าคงที่ (Constants)

หมายถึง ค่าที่โปรแกรมเมอร์กำหนดขึ้นเอง เพื่อประโยชน์ในการเขียนโปรแกรมและค่านี้จะคงที่ตลอดไม่มีการเปลี่ยนแปลง

ค่าคงที่แบ่งออกเป็น 2 ประเภทคือ

- ค่าคงที่ตัวเลข *(Numeric Constants)*

  หมายถึง คาเร็คเตอร์ที่เป็นตัวเลข สามารถนำไปใช้ในการคำนวณได้ อาทิ

999 +2058 -111 333.22 -0.2345 +67.89

สามจำนวนแรกไม่มีจุดทศนิยมเรียกว่าจำนวนเต็ม

1.23 E03 0.44E-2 33.77E1

○ ค่าคงที่สตริงก์

(String or Non-Numeric Constants)

หมายถึง คาเร็คเตอร์ตั้งแต่ 1 ตัวขึ้นไป นำมาเขียนเรียงกันอยู่ในเครื่องหมาย “ ” ***(เครื่องหมายคำพูด)*** อาทิ

“FIND THE AVERAGE”

“RATE”

“088-24-0997”

“TO BE NUMBER ONE”

สตริงก์ที่ไม่มีแคเร็คเตอร์อยู่เลย เรียกว่า *Null String “ ” (พิมพ์เครื่องหมายคำพูดติดกัน)* ค่าคงที่สตริงก์จะนำไปคำนวณไม่ได้

ผังงาน (Flowchart)

หมายถึง วิธีการที่ใช้แสดงขั้นตอนของการทำงานต่าง ๆ โดยใช้รูปแบบของสัญลักษณ์แทนความหมาย ให้เข้าใจและมองเห็นขั้นตอนการทำงานเหล่านั้น ได้โดยชัดเจนขึ้น

ความสำคัญของผังงาน

1. ทำให้เข้าใจและแยกแยะปัญหาต่าง ๆ ให้เข้าใจได้ง่ายขึ้น
2. ทำให้ผู้เขียนโปรแกรม มองเห็นถึงลำดับขั้นในการทำงาน
3. ช่วยในการหาข้อผิดพลาด ของโปรแกรมได้ง่าย
4. ช่วยทำให้ผู้อื่นสามารถเข้าใจงานที่เรา ทำขึ้นได้ง่าย
5. เป็นเอกสารประกอบ ในการเขียนโปรแกรม

สัญลักษณ์ที่ใช้ในผังงาน

ตัวอย่างของสัญลักษณ์ที่ใช้บ่อย ๆ

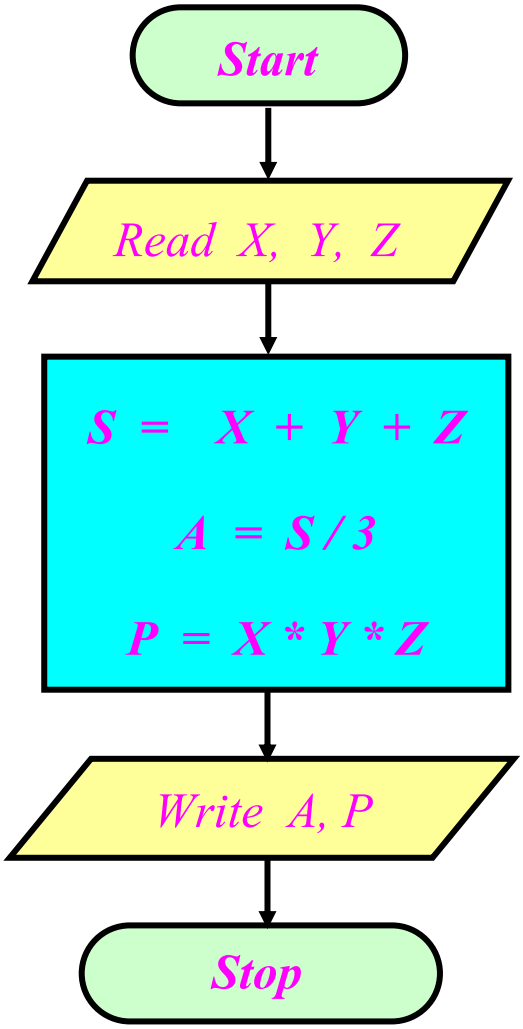
	การเริ่มต้นหรือสิ้นสุดการทำงาน
	การนำข้อมูลเข้าหรือออกจากเครื่องคอมพิวเตอร์ โดยไม่ได้ระบุอุปกรณ์
	การประมวลผลต่าง ๆ
	การเปรียบเทียบ เพื่อให้เกิดการตัดสินใจ
	จุดเชื่อมต่อในหน้าเดียวกัน
	การแสดงผลทางเครื่องพิมพ์
	การแสดงทิศทางของผังงาน

การจัดภาพและทิศทางของผังงาน

1. ทิศทางของผังงานเริ่มจากส่วนบนลงไปล่าง จากซ้ายไปขวา มีเครื่องหมายลูกศรกำกับทิศทาง
2. สัญลักษณ์หรือภาพที่ใช้ในการเขียนผังงาน มีขนาดต่างกันได้ แต่ต้องมีรูปมาตรฐาน ตามความหมายที่กำหนด
3. การเขียนทิศทางของผังงานควรเป็นไปอย่างมีระเบียบ และหลีกเลี่ยงการขีดโยงมาในทิศทางตัดกัน ถ้าจำเป็นต้องโยงถึงควรใช้เครื่องหมายจุดต่อเนื่องแทน และ ถ้าเป็นไปได้ควรเขียนผังงานให้จบในหน้าเดียวกัน
4. เขียนคำอธิบายในภาพเพียงสั้น ๆ และเข้าใจง่าย
5. ผังงานที่ดีควรมีความเป็นระเบียบเรียบร้อยและสะอาด อาจมีชื่อของพนักงาน ผู้เขียน วันที่ที่เขียน และเลขหน้าลำดับ เป็นต้น

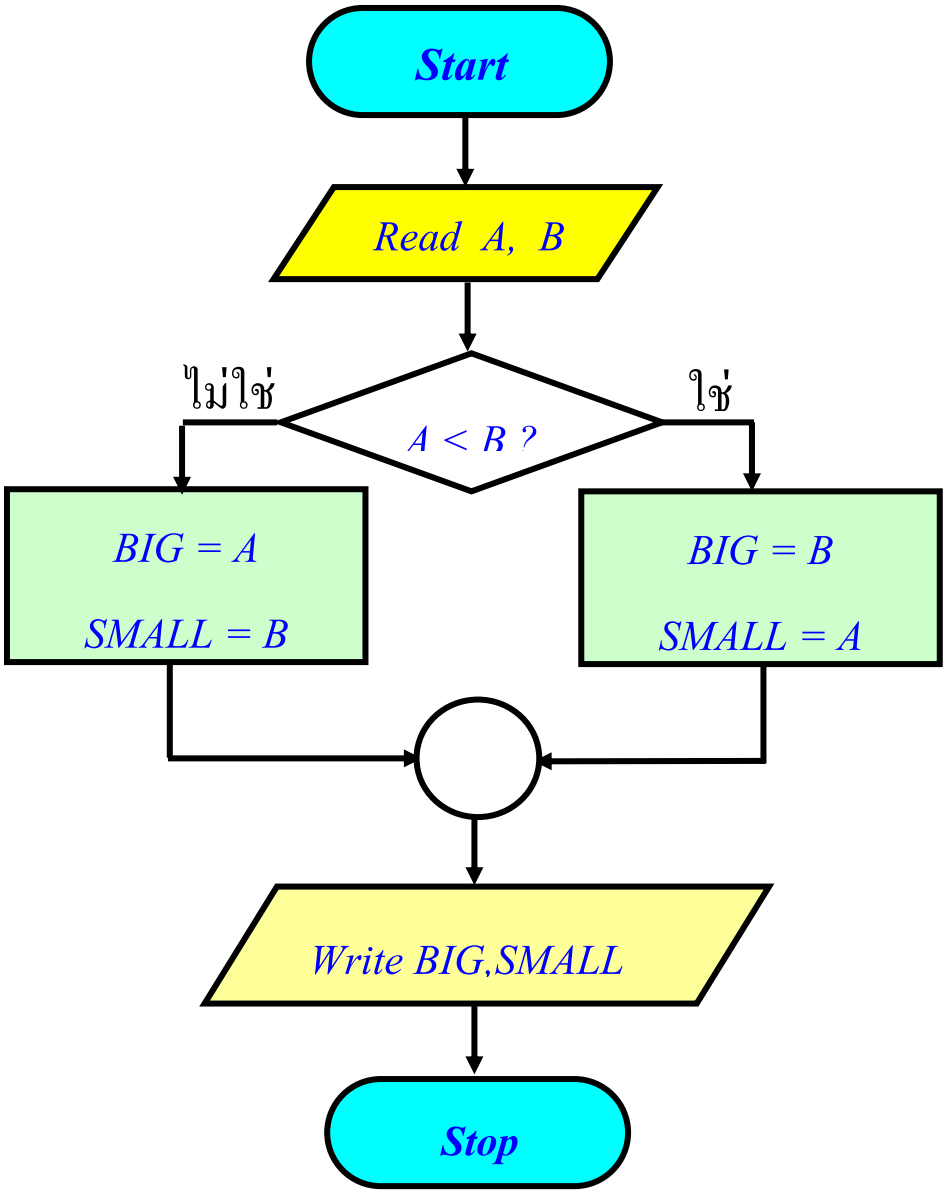
ตัวอย่าง

ฝังงานของการหาผลบวก *(S)* ค่าเฉลี่ย *(A)* และผลคูณ *(P)* ของจำนวน *3* จำนวน *(X, Y, และ Z)*



ตัวอย่าง

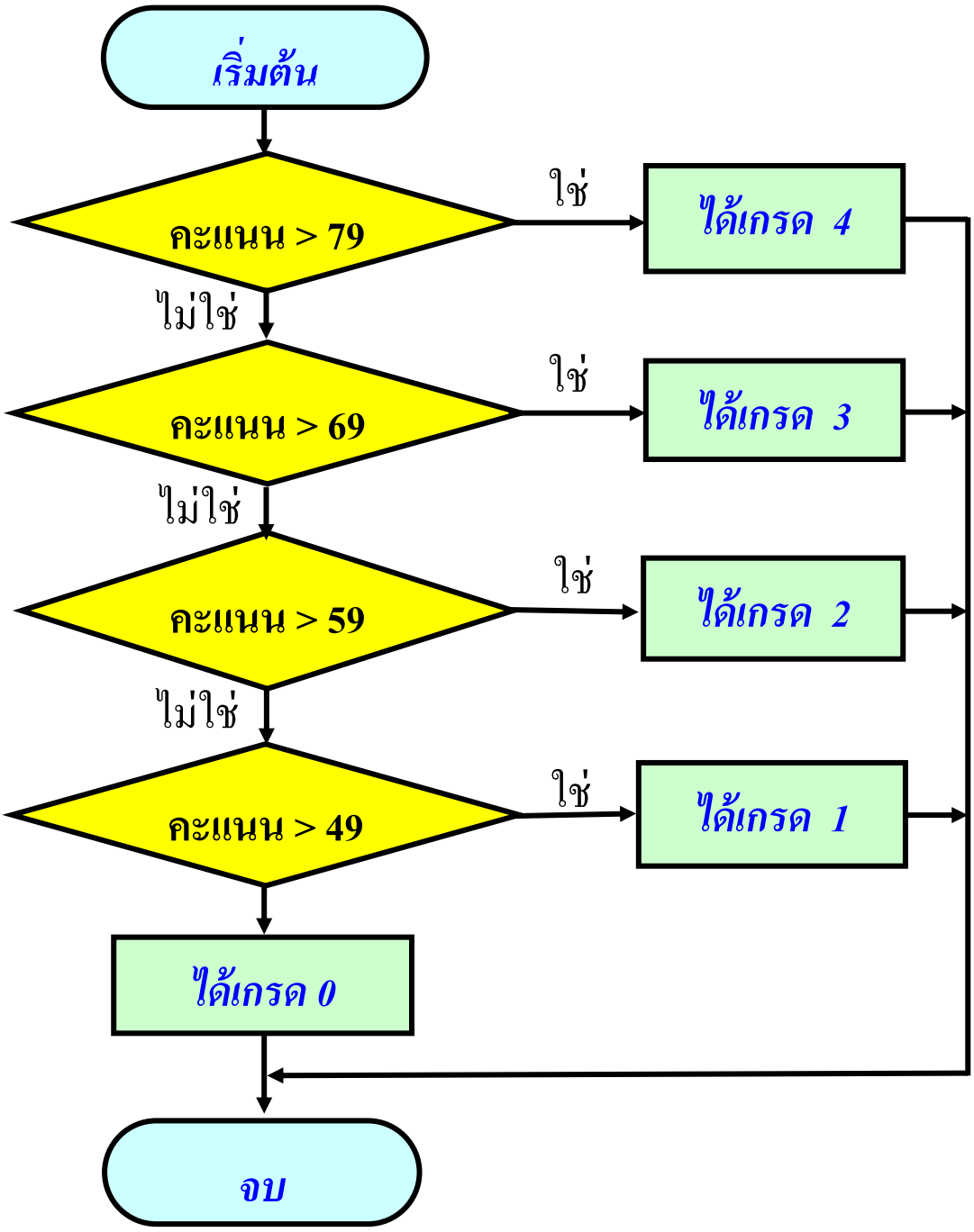
ผังงานที่แสดงการเปรียบเทียบข้อมูล 2 จำนวนเต็มคือ A และ B ที่อ่านเข้ามาและพิมพ์จำนวนทั้งสองจากมากไปหาน้อย ให้จำนวนมากอยู่ในตัวแปร BIG และให้จำนวนน้อยอยู่ในตัวแปร SMALL



ตัวอย่าง

จงผังงานแสดงการตัดเกรด ของวิชาคณิตศาสตร์คอมพิวเตอร์ ซึ่งมีเต็ม 100 คะแนน

ถ้าได้คะแนน 80 ขึ้นไป	จะได้เกรด 4
ถ้าได้คะแนน 70-79	จะได้เกรด 3
ถ้าได้คะแนน 60-69	จะได้เกรด 2
ถ้าได้คะแนน 50-59	จะได้เกรด 1
ถ้าได้คะแนน ต่ำกว่า 50	จะได้เกรด 0


เริ่มต้น
คะแนน > 79
ใช่
ได้เกรด 4
ไม่ใช่
คะแนน > 69
ใช่
ได้เกรด 3
ไม่ใช่
คะแนน > 59
ใช่
ได้เกรด 2
ไม่ใช่
คะแนน > 49
ใช่
ได้เกรด 1
ได้เกรด 0
จบ

ลูป (Loops)

หมายถึง ลำดับของคำสั่งในโปรแกรมที่ต้องการกระทำซ้ำแล้วซ้ำอีก จนกว่าจะถึงเงื่อนไขที่กำหนดไว้ โดยปกติการเขียนโปรแกรมคอมพิวเตอร์ นิยมเขียนโปร แกรมโครงสร้างจากบนลงล่าง

มีโครงสร้างหลักอยู่ 3 รูปแบบ คือ

1. โครงสร้างแบบเรียงลำดับ

   (Sequence Logic)

2. โครงสร้างแบบทางเลือก

   (Selection Logic)

   - ทางเลือกทางเดียว *(Single Alternative)*
   - ทางเลือกสองทาง *(Double Alternative)*
   - ทางเลือกหลายแบบ *(Multiple Alternatives)*

3. โครงสร้างแบบมีการทำงานซ้ำ ๆ กัน

   - โครงสร้างแบบ *DO WHILE*
   - โครงสร้างแบบ *DO UNTIL*

โปรแกรมซูโดโค้ด (Pseudocode Programs)

- ใช้สำหรับการออกแบบในรายละเอียดของโปรแกรม
- มีลักษณะเป็นการใช้ภาษาอังกฤษธรรมดาคล้ายกับการเขียนโปรแกรมภาษา PL/1, PASCAL หรือภาษา ALGOL
- ผู้ออกแบบโปรแกรมมีอิสระหรือคล่องตัวมากกว่า
- ซูโดโค้ดไม่ใช่ภาษาคอมพิวเตอร์
- ไม่ได้กำหนดกฎเกณฑ์ไวยากรณ์ที่ตายตัว

- มี *Key Worlds* เพื่อแสดงโครงสร้างรายละเอียดของคำสั่ง เช่น คำว่า *Read, Write, If-Then-Eles, Do While, Do Until* เป็นต้น
- มีตัวแสดงขอบเขตของโครงสร้างของคำสั่ง เช่น *EndIf, EndDo, EndCase* เป็นต้น
- ไม่มีกฎเกณฑ์ตามตัวสำหรับการตั้งชื่อตัวแปร จะใช้ชื่ออย่างไรก็ได้
- พยายามใช้คำสั่ง *Goto* ให้น้อยที่สุด
- พยายามจัดโครงสร้างให้เป็นกลุ่ม *(Block)* หรือจัดโปรแกรมให้ประกอบด้วยโมดุล *(Module)* ย่อย ๆ
- อาจใช้โครงสร้างคล้ายคลึงกับภาษาคอมพิวเตอร์ที่ต้องการใช้ในการพัฒนาโปรแกรม

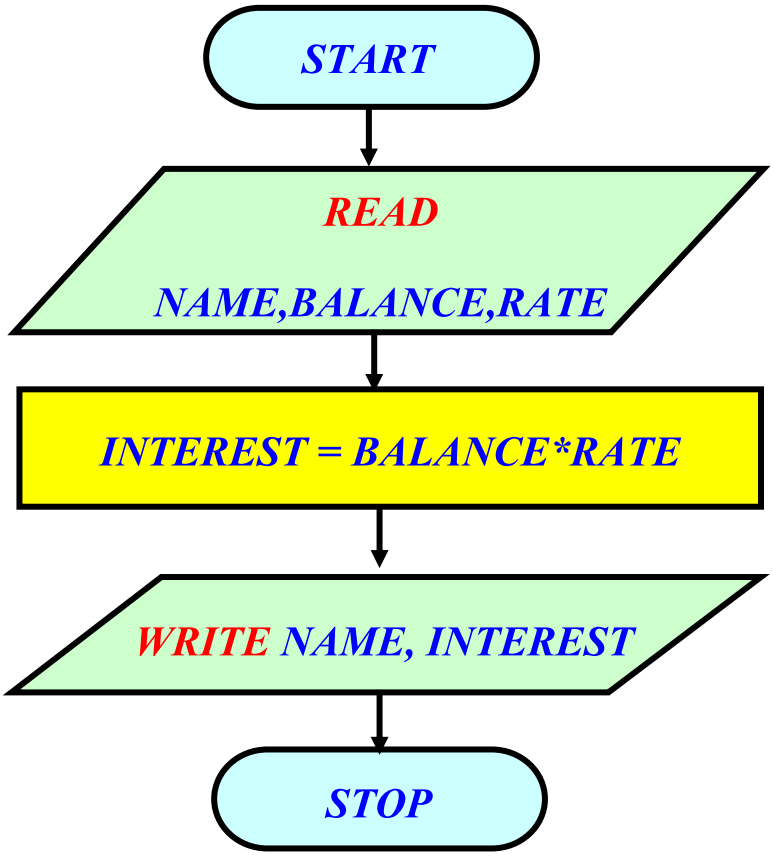
รูปแบบของโปรแกรมซูโดโค้ด

โปรแกรมซูโดโค้ด ที่ใช้กันมี 3 รูปแบบ คือ

1. แบบเรียงลำดับ *(Sequence Logic)*

○ คำสั่งในโปรแกรมชุดโค้ดนี้จะทำงานตามลำดับจากบนลงล่าง นิยมเขียนคำสั่งบอกความสิ้นสุดของโปรแกรมด้วยการเขียน END ไว้ตอนล่างของโปรแกรม

ตัวอย่างที่ 1 จงเขียนโปรแกรมซูโดโค้ดจากผังงาน ดังรูป



วิธีการ

```
Read  NAME, BALANCE, RATE
      INTEREST = BALANCE * RATE
Write NAME, INTEREST
END
```

2. แบบทางเลือก

(Selection Logic)

แบ่งออกเป็น ดังนี้

2.1 แบบทางเลือกเดียว*(Single Alternative)*

เรียกโครงสร้างนี้ว่า ***IF THEN Structure***

```
Psuedo Code Program
    IF condition
        [Procedure A]
    ENDIF
```

ตัวอย่างที่ 2

จงเขียนโปรแกรมซูโดโค้ด โดยใช้โครงสร้าง *IF THEN* คำนวณหาค่าจ้าง *(WAGES)* กำหนดจำนวนชั่วโมงการทำงาน *(HOURS)* และ อัตราค่าจ้างให้

วิธีการ

```
NAME,RATE,HOURS
WAGES = HOURS x RATE
IF  HOURS > 40
      WAGES = WAGES  + (HOURS – 40) x 0.5 x RATE
ENDIF
Write NAME, WAGES
END
```

2.2 แบบทางเลือก 2 ทาง

(Double Alternative)

เรียกโครงสร้างนี้ว่า *IF THEN ELSE Structure*

Psuedo Code Program

```
IF condition
        [Procedure A]
ELSE
        [Procedure B]
ENDIF
```

ตัวอย่างที่ 3

โปรแกรมซูโดโค้ดโดยใช้โครงสร้าง *IF THEN ELSE*

วิธีการ

```
Read A,  B
IF  A < B
      BIG = B
      SMALL = A
ELSE
      BIG = A
      SMALL = B
ENDIF
Write  BIG, SMALL
END
```

2.3 แบบทางเลือกหลายทาง

(Multiple Alternative)

หรือแบบมากกว่า 1 เงื่อนไข

2.3.1 โครงสร้าง *IF... ELSE*

Psuedo Code Program

```
IF condition(1)
        [Procedure A1]
ELSE
    IF condition A(2)
        [Procedure A2]
    ENDIF
ENDIF
```

ตัวอย่างที่ 4

โปรแกรมซูโดโค้ด โดยใช้โครงสร้าง *IF... ELSE*

วิธีการ

```
IF Mark > 79
    ? “Grade = 4”
ELSE
    IF Mark > 69
        ? “Grade = 3”
    ELSE
        IF Mark > 59
            ? “Grade = 2”
        ELSE
            IF Mark > 49
                ? “Grade = 1”
            ELSE
                ? “Grade = 0”
            ENDIF
        ENDIF
    ENDIF
ENDIF
```

2.3.2 โครงสร้าง DO CASE คล้ายคำสั่ง *IF....ELSE* แต่ใช้งานสะดวกกว่า

Psuedo Code Program

```
DO  CASE
      CASE   condition (1)
            [Procedure A1]
      CASE   condition (2)
            [Procedure A2]
      OTHERWISE
            [Procedure Am]
ENDCASE
```

ตัวอย่างที่ 5

โปรแกรมซูโดโค้ด โดยใช้โครงสร้าง DO CASE

วิธีการ

```
DO CASE

    CASE Mark > 79
            ? “Grade = 4”
    CASE Mark > 69
            ? “Grade = 3”
    CASE Mark > 59
            ? “Grade = 2”
    CASE Mark > 49
            ? “Grade = 1”
    OTHERWISE
            ? “Grade = 0”
ENDCASE
```

3. แบบทำงานซ้ำ ๆ

3.1 โครงสร้าง DOWHILE….ENDDO

```
Psuedo Code Program
        :
    DOWHILE condition
            [Procedure]
            [(body  of loop]
    ENDDO
        :
```

เมื่อพบคำสั่ง *DOWHILE* เครื่องจะทำการตรวจสอบเงื่อนไข ถ้าเงื่อนไขเป็นจริง จะทำคำสั่งที่อยู่ระหว่าง *DOWHILE…ENDDO* เมื่อพบ *ENDDO* จะวนกลับไปยัง *DOWHILE* อีกเพื่อตรวจสอบเงื่อนไข หากยังเป็นจริงก็จะทำซ้ำต่อไปเรื่อย ๆ แต่ถ้าเป็นเท็จจะออกจากการทำซ้ำนั้นไปทำคำสั่งที่อยู่ถัดจาก *ENDDO*

ตัวอย่างที่ 6

โปรแกรมซูโดโค้ดของการหาผลบวกของเลข 1 ถึง 15

วิธีการ

```
Sum = 0
N = 1
    DOWHILE N < = 15
        STORE Sum + N TO Sum
        N = N + 1
    ENDDO
        ? “Sum of 1+2+3+…+ 15  is” , Sum
```

3.2 โครงสร้าง *DO UNTIL … ENDDO*

```
Psuedo Code Program
        :
    DO UNTIL condition
        [Procedure]
        [body  of loop]
    ENDDO
        :
```

เมื่อพบคำสั่ง *DO UNTIL* เครื่องจะทำการตรวจสอบเงื่อนไข ถ้าเงื่อนไขเป็นเท็จจะทำหน้าที่อยู่ระหว่าง *DO UNTIL…ENDDO* เมื่อพบ *ENDDO* อีกเพื่อตรวจสอบเงื่อนไข หากยังคงเป็นเท็จ ก็จะทำหน้าซ้ำต่อไปเรื่อย ๆ แต่ถ้าเป็นจริงจะออกจากการทำซ้ำนั้น ไปทำคำสั่งที่อยู่ถัดจาก *ENDDO*

ตัวอย่างที่ 7

โปรแกรมซูโดโค้ดของการหาผลบวกของเลข 1 ถึง 5

วิธีการ

```
Sum = 0
N= 1
DO UNITIL  N > 15
      STORE Sum + N TO Sum
      N = N + 1
ENDDO
      ? “Sum of 1+2+3+…+15 is”,Sum
```

ตัวอย่างที่ 8

แฟ้มข้อมูลส่วนบุคคลของบริษัทแห่งหนึ่งประกอบด้วยอายุและรายชื่อของลูกค้า ถ้าบริษัทต้องการแสดงรายชื่อของลูกค้าทั้งหมดที่มีอายุต่ำกว่า 30 ปี

จงเขียนโปรแกรมซูโดโค้ดในการกระทำดังกล่าว

วิธีการ

```
Read    NAME, AGE  from  record
DO UNITIL  End  of  File
      IF  AGE < 30
            Write NAME
      ENDIF
            Read  NAME, AGE  from next record
ENDDO
END
```