

บทที่ 5

การเขียนโปรแกรมแบบมีการตัดสินใจ

หัวข้อ

- 5.1 Boolean Operator
- 5.2 การเปรียบเทียบข้อมูลชนิด String
- 5.3 ประโยคคำสั่งควบคุมแบบมีการตัดสินใจ
- 5.4 ประโยคคำสั่ง Switch

5.1 Boolean Operator

ผลของการเปรียบเทียบข้อมูล จะมี 2 ค่าคือ จริง(True) หรือ เท็จ(False)

ตัวดำเนินการแบบบูลีน แบ่งออกเป็น 2 กลุ่มย่อย ได้แก่ ตัวดำเนินการเปรียบเทียบ(Relational Operator) และตัวดำเนินการทางคณิตศาสตร์(Logical Operator)

ตัวดำเนินการเปรียบเทียบ(Relational Operator)

ตัวดำเนินการเปรียบเทียบ เป็นตัวดำเนินการแบบ Binary Operator กล่าวคือ ต้องมีตัวถูกกระทำ 2 ตัว หรือมีข้อมูล 2 ตัวมาเปรียบเทียบกัน

ตารางที่ 5.1 แสดงเครื่องหมายที่ใช้ในการเปรียบเทียบข้อมูล

เครื่องหมาย	การเปรียบเทียบ	ตัวอย่าง	ผลลัพธ์
==	เท่ากับ	5 == 7	เท็จ
!=	ไม่เท่ากับ	5 != 7	จริง
>	มากกว่า	5 > 7	เท็จ
<	น้อยกว่า	5 < 7	จริง
>=	มากกว่าหรือเท่ากับ	5 >= 7	เท็จ
<=	น้อยกว่าหรือเท่ากับ	5 <= 7	จริง

ตัวดำเนินการทางตรรกศาสตร์(Logical Operator)

ใช้เมื่อต้องการรวมนิพจน์ในการเปรียบเทียบข้อมูลมากกว่า 1 นิพจน์ขึ้นไปเข้าด้วยกัน โดยการรวมจะถูกระบุด้วยเครื่องหมายทางตรรกะ ซึ่งได้แก่ AND, OR และ NOT เพื่อเปรียบเทียบค่าข้อมูลกับนิพจน์ดังกล่าวทีละพจน์ แล้วหาคำตอบที่ต้องการ ซึ่งมีเพียงคำตอบเดียว คือ “จริง(True)” หรือ “เท็จ(False)” เท่านั้น

ตารางที่ 5.2 แสดงตัวดำเนินการทางตรรกศาสตร์

เครื่องหมาย	ความหมาย	ตัวอย่าง a = 10, b = 5	ผลลัพธ์
&&	AND	(a > b) && (a == 10)	จริง
	OR	(a > b) (a == 10)	จริง
!	NOT	!(a == b)	จริง
^	Exclusive or	!(a == b) ^ !(a == 10)	จริง

ตารางที่ 5.3 แสดงค่าความจริงทางตรรกศาสตร์(T = True, F = False, a = 10, b = 5)

a	b	a b	a && b	a ^ b	!a
F	F	F	F	F	T
T	F	T	F	T	F
F	T	T	F	T	T
T	T	T	T	F	F

(a > b) && (a == 10) → T && T → T

(a > b) || (a == 10) → F || T → T

!(a == b) → !F → T

!(a == b) ^ !(a == 10) → !F ^ !T → T ^ F → T

ตารางที่ 5.4 ลำดับความสำคัญของตัวดำเนินการทั้งหมด

ลำดับที่	เครื่องหมาย
1	()
2	!, ++, --
3	*, /, %
4	+, -

5	<, <=, >, >=
6	==, !=
7	&&
8	
9	=, +=, -=, *=, /=, %=

5.2 การเปรียบเทียบข้อมูลชนิด String

มี 2 Method คือ “equals()” และ “compareTo()”

Method “equals()”

“equals()” เป็น Method ที่ใช้เพื่อเปรียบเทียบข้อมูลชนิด String ว่าเป็นข้อความที่เหมือนกันหรือไม่ ข้อความจะเหมือนกันก็ต่อเมื่อเป็นข้อความเดียวกัน ผลที่ได้จากการเปรียบเทียบมีเพียงค่าเดียวคือ “จริง(True)” หรือ “เท็จ(False)” เท็จจริง

ตัวอย่างที่ 5.1 การเปรียบเทียบข้อความโดยใช้ Method “equals()”

- (1) // โปรแกรมเปรียบเทียบข้อความด้วยเมธอด equals()
- (2) public class equalString
- (3) {
- (4) public static void main (String[] args)
- (5) {
- (6) String word1 = “Somsak”;
- (7) String word2 = “Surachet”;
- (8)
- (9) System.out.println(“word1 เปรียบเทียบกับ word2 = “ + word1.equals(word2));
- (10) System.out.println(“word1 เปรียบเทียบกับคำว่า Somsak = “ + word1.equals(“Somsak”));
- (11) }
- (12) }

ผลลัพธ์

word1 เปรียบเทียบกับ word2 = false

word1 เปรียบเทียบกับคำว่า Somsak = true

Method “compareTo()”

เป็น Method ที่ใช้เพื่อเปรียบเทียบข้อมูลชนิด String ว่าเป็นข้อความที่เท่ากันหรือไม่

ผลลัพธ์ที่เป็นไปได้ มี 3 ค่า คือ

- 0 เมื่อข้อความนั้นมีค่าเท่ากัน
- น้อยกว่า 0 เมื่อข้อความที่เป็นตัวตั้งมีค่าน้อยกว่าข้อความที่นำมาเปรียบเทียบ
- มากกว่า 0 เมื่อข้อความที่เป็นตัวตั้งมีค่ามากกว่าข้อความที่นำมาเปรียบเทียบ

รูปแบบ

```
word1.compareTo(word2);
```

ตัวอย่างที่ 5.2 การเปรียบเทียบข้อความโดยใช้ Method “compareTo()”

- ```
(1) // โปรแกรมเปรียบเทียบข้อความด้วยเมธอด compareTo()
(2) public class compareString
(3) {
(4) public static void main (String[] args)
(5) {
(6) String word1 = “her”;
(7) String word2 = “his”;
(8)
(9) System.out.println(“his เมื่อนำไปเปรียบเทียบกับ her = “ + word1.compareTo(word2));
(10) System.out.println(“her เมื่อนำไปเปรียบเทียบกับ his = “ + word2.compareTo(word1));
(11) }
(12) }
```

### 5.3 ประโยคคำสั่งควบคุมแบบมีการตัดสินใจ

ประโยคคำสั่งที่ใช้สร้างเงื่อนไขเพื่อการตัดสินใจ นั่นคือ คำสั่ง "IF" มีหลายรูปแบบได้แก่

IF แบบมีเงื่อนไขเดียว (Simple IF)

IF แบบ 2 เงื่อนไข (IF...ELSE)

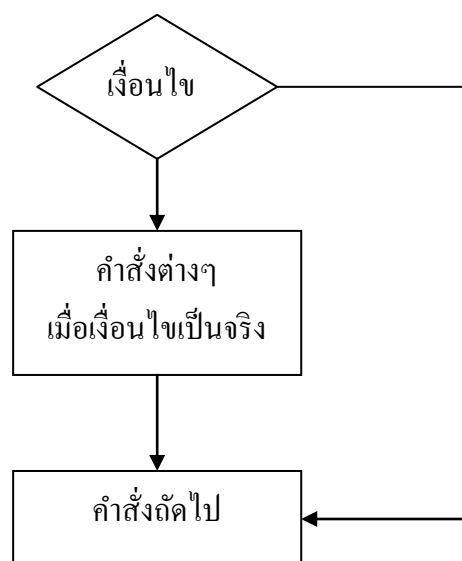
IF แบบซ้อน (Nested IF)

#### 5.3.1 IF แบบมีเงื่อนไขเดียว (Simple IF)

เป็นคำสั่งที่มีเงื่อนไขเพื่อการตัดสินใจเพียงเงื่อนไขเดียว ถ้าตรวจสอบเงื่อนไขแล้วมีค่าเป็นจริง จะทำงานตามคำสั่งของ IF แต่ถ้าเงื่อนไขเป็นเท็จ โปรแกรมจะทำงานนอกคำสั่ง IF ต่อไป

รูปแบบ

```
If (เงื่อนไข)
{
 ประโยคคำสั่งเมื่อเงื่อนไขเป็นจริง;
}
ประโยคคำสั่งอื่นๆ
```



รูปที่ 5.1 แสดงแผนภาพการทำงานของโปรแกรมเมื่อพบคำสั่ง IF แบบมีเงื่อนไขเดียว

**ตัวอย่างที่ 5.3** ตัวอย่างการเขียนประโยคคำสั่ง IF แบบมีเงื่อนไขเดียว ทดสอบข้อมูลชนิด int

```
int count = 5;
if (count <= 10)
{
 sum = sum + num;
}
System.out.println("ผลรวม = " + sum);
```

**ตัวอย่างที่ 5.4** ตัวอย่างการเขียนประโยคคำสั่ง IF แบบมีเงื่อนไขเดียว ทดสอบข้อมูลชนิด String

```
String cont = "Y";
if (cont.equals("Y"))
{
 netSalary = salary – socialFund + Ot;
}
```

**ตัวอย่างที่ 5.5** ตัวอย่างการเขียนประโยคคำสั่ง IF แบบมีเงื่อนไขเดียว ทดลองเขียนโปรแกรมเพื่อหาผลสอบของนักเรียนแต่ละคน โดยให้รับชื่อ และคะแนนสอบของนักเรียนทางแป้นพิมพ์ โดยมีเงื่อนไขคือ หากคะแนนสอบมีค่าน้อยกว่า 60 ถือว่าไม่ผ่านเกณฑ์ จากนั้นให้สั่งพิมพ์ทั้งชื่อ คะแนนสอบ และผลสอบออกทางจอภาพ

กำหนดตัวแปร

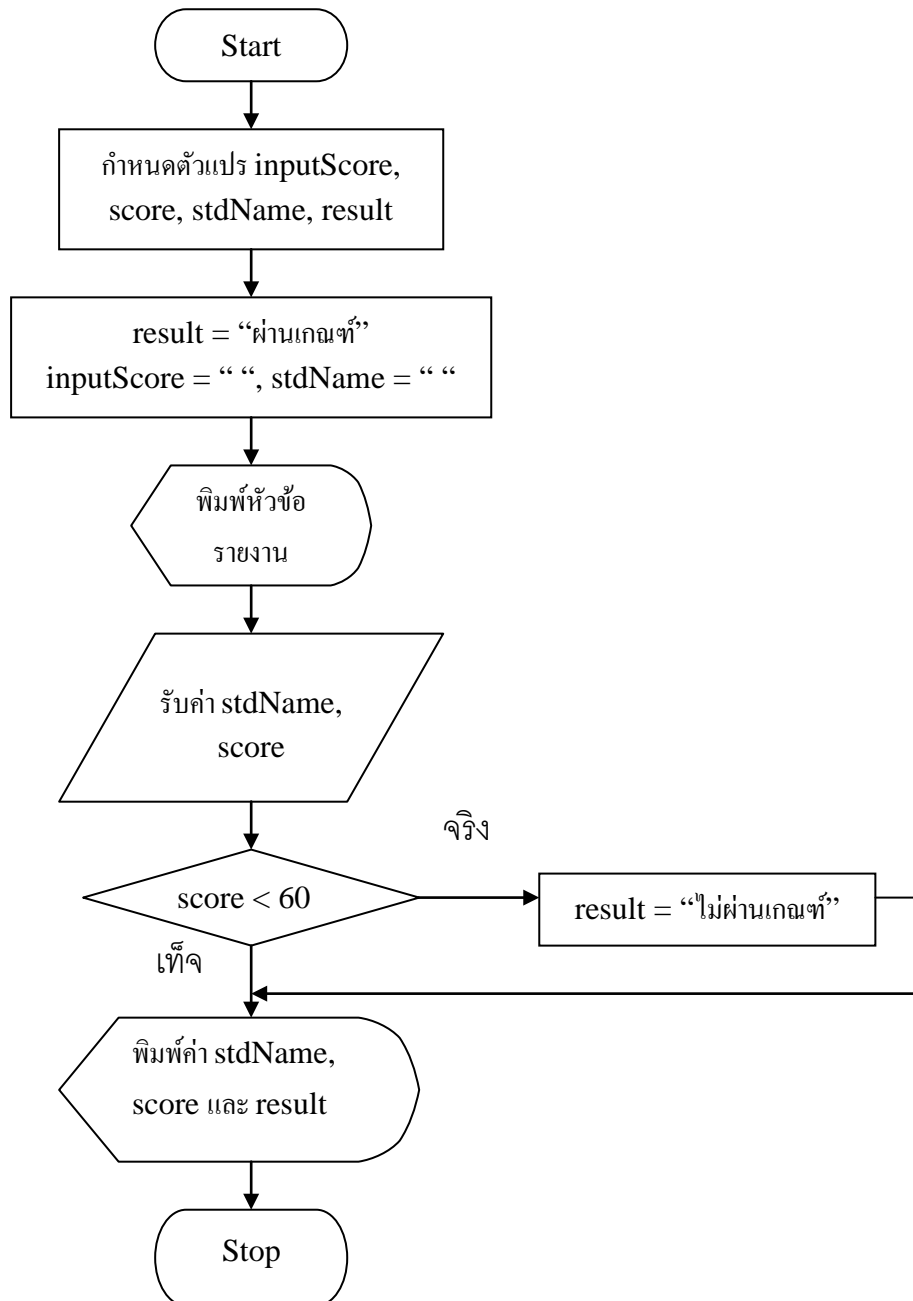
|         |     |              |
|---------|-----|--------------|
| stdName | แทน | ชื่อนักเรียน |
| score   | แทน | คะแนนสอบ     |
| result  | แทน | ผลสอบ        |

อัลกอริทึม

1. กำหนดค่าเริ่มต้นให้ผลสอบ (result) เท่ากับ “ผ่านเกณฑ์”
2. รับค่าชื่อนักเรียน(stdName) และคะแนนสอบ(score)

3. ทดสอบคะแนนสอบ (score) < 60 เป็นจริงหรือเท็จ
  - 3.1 ถ้าจริง ให้ผลสอบเท่ากับ “ไม่ผ่านเกณฑ์” แล้วไปทำข้อ 4
  - 3.2 ถ้าเท็จ ให้ไปทำข้อ 4

ผังงาน



รูปที่ 5.2 แสดงผังงานของตัวอย่างที่ 5.5

โค้ดโปรแกรม

(1) // โปรแกรมหาผลสอบจากคะแนนที่รับมาทางคีย์บอร์ด

```

(2) import java.io.*;
(3) public class simplelf
(4) {
(5) public static void main (String[] args) throws IOException
(6) {
(7) String inputScore = " "; // กำหนดตัวแปรมารับ score ให้เป็น String ก่อน
(8) String stdName = " ";
(9) String result = "ผ่านเกณฑ์";
(10) int score;
(11)
(12) // รับค่าชื่อและคะแนนสอบ
(13) BufferedReader stdin = new BufferedReader(new InputStreamReader(System.in));
(14) System.out.print("กรุณานำป้อนชื่อนักเรียน");
(15) stdName = stdin.readLine(); // รับค่าชื่อ
(16) System.out.print("กรุณานำป้อนคะแนนสอบ");
(17) inputScore = stdin.readLine(); // รับค่าคะแนนสอบเป็น String
(18) score = Integer.parseInt(inputScore); // แปลง score จาก String ให้เป็น integer
(19)
(20) // ทดสอบค่าคะแนน
(21) if (score < 60)
(22) {
(23) result = "ไม่ผ่านเกณฑ์";
(24) }
(25) System.out.println("ชื่อ " + stdName);
(26) System.out.println("คะแนนสอบ " + score);
(27) System.out.println("ผลสอบที่ได้ " + result);
(28) }
(29) }

```



## ผลลัพธ์

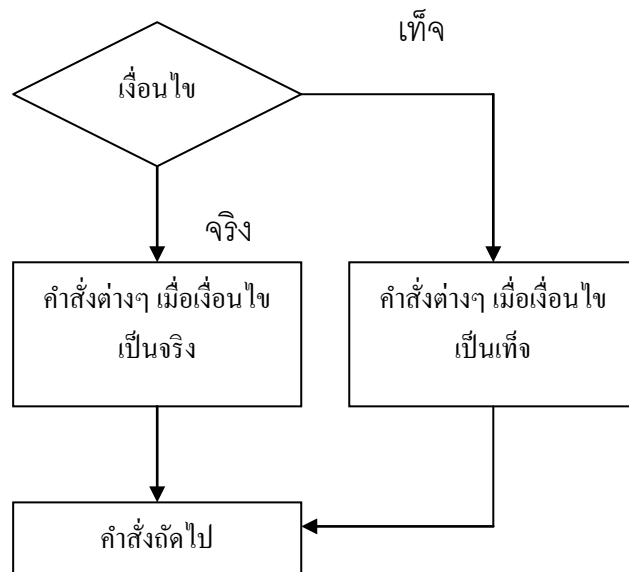
กรุณาป้อนชื่อนักเรียน      สุทธิดา  
กรุณาป้อนคะแนนสอบ      65  
ชื่อ      สุทธิดา  
คะแนนสอบ      65  
ผลสอบที่ได้      ผ่านเกณฑ์

### 5.3.2 IF แบบ 2 เงื่อนไข (IF...ELSE)

IF...ELSE เป็นคำสั่งที่มีเงื่อนไขเพื่อการตัดสินใจ 2 เงื่อนไข ถ้าทดสอบเงื่อนไขแรกแล้วพบว่าเป็นจริง จะทำงานตามคำสั่งที่เงื่อนไขแรกกำหนด หากทดสอบแล้วพบว่าเป็นเท็จ โปรแกรมจะทำงานตามคำสั่งหลัง ELSE ซึ่งนับว่าเป็นเงื่อนไขที่ 2

#### รูปแบบ

```
if (เงื่อนไข)
{
 ประโยคคำสั่งต่างๆ เมื่อเงื่อนไขเป็นจริง;
}
else
{
 ประโยคคำสั่งต่างๆ เมื่อเงื่อนไขเป็นเท็จ;
}
คำสั่งอื่นๆ
```



รูปที่ 5.3 แสดงการทำงานของ IF...ELSE

**ตัวอย่างที่ 5.6** ตัวอย่างการเขียนประโยคคำสั่ง IF...ELSE

```

int hourWorked = 45;
double grossPay;
double rate = 15.00;
if (hourWorked > 40)
{
 grossPay = 40 * rate + (hourWorked - 40) * 1.5 * rate;
}
else
{
 grossPay = hourWorked * rate;
}
System.out.println("ค่าแรงที่ได้รับทั้งหมด = " + grossPay);

```

**ตัวอย่างที่ 5.7** ทดลองการเขียนโปรแกรมเพื่อคำนวณหาราคาขายสินค้าสุทธิ โดยพิจารณาจากสถานะของลูกค้า ถ้าเป็นสมาชิกจะให้ส่วนลด 10% แต่ถ้าไม่ใช่สมาชิกจะไม่ให้ส่วนลด โดยให้มีการรับค่าสถานะสมาชิก ถ้าเป็นสมาชิกให้ป้อนข้อมูลเป็น “Y” แต่ถ้าไม่ใช่สมาชิกให้ป้อนข้อมูลเป็น “N” และรับราคาสินค้าทางคีย์บอร์ด

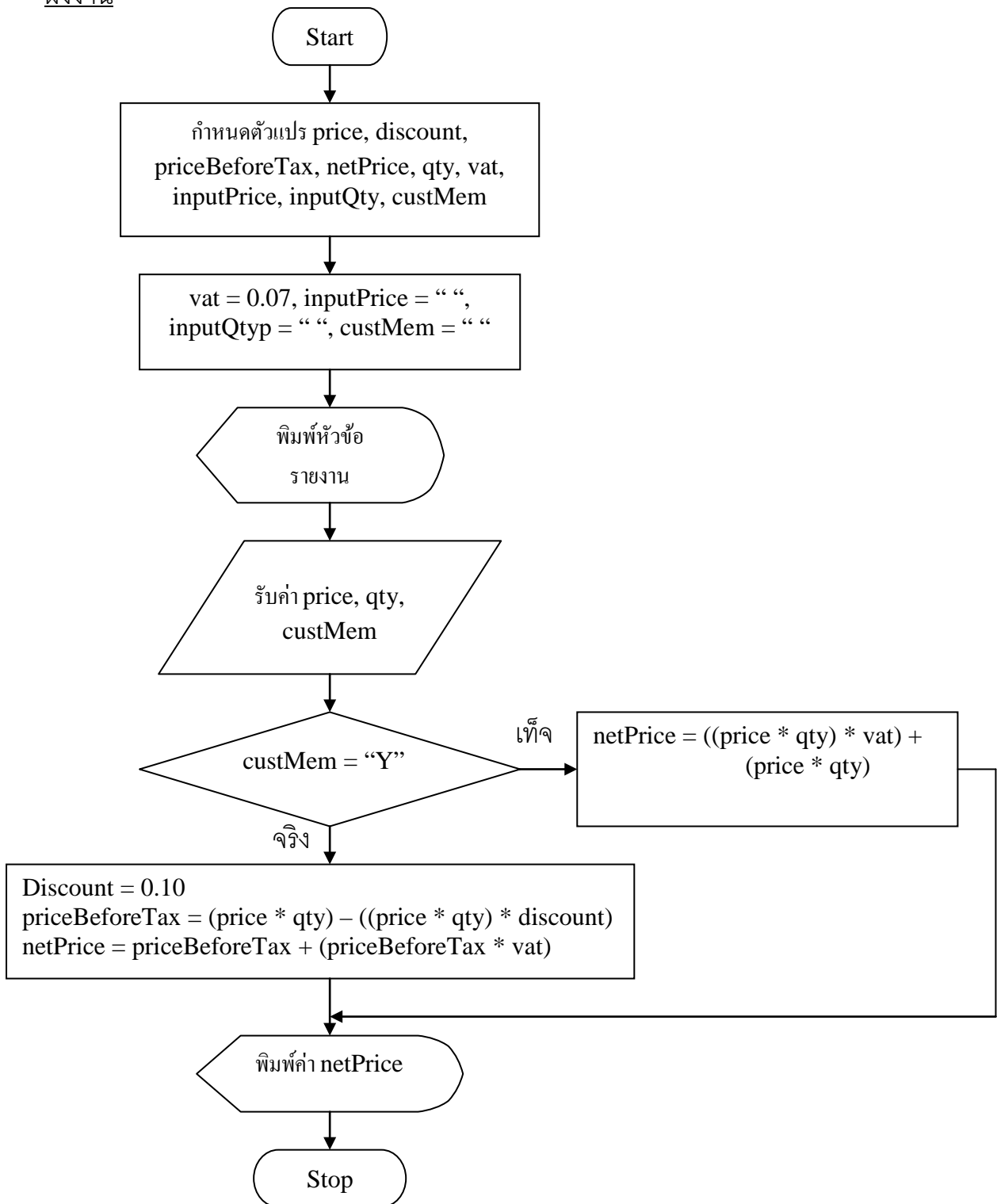
กำหนดตัวแปร

|                |     |                                 |
|----------------|-----|---------------------------------|
| custMen        | แทน | สถานะของลูกค้าเป็นสมาชิกหรือไม่ |
| price          | แทน | ราคาขายสินค้า                   |
| qty            | แทน | จำนวนสินค้า                     |
| discount       | แทน | ส่วนลด                          |
| vat            | แทน | ภาษีมูลค่าเพิ่ม                 |
| priceBeforeTax | แทน | ราคาขายก่อนภาษีมูลค่าเพิ่ม      |
| netPrice       | แทน | ราคาขายสินค้าสุทธิ              |

อัลกอริทึม

1. กำหนดตัวแปร
2. กำหนดค่าเริ่มต้นให้ vat = 0.07
3. รับค่าราคาขายสินค้า จำนวนสินค้า และสถานะสมาชิก
4. เป็นลูกค้าสมาชิกหรือไม่
  - 4.1 ใช่ ให้ส่วนลด 10% คำนวณราคาขายสินค้าสุทธิตามสูตร
 
$$\text{priceBeforeTax} = (\text{price} * \text{qty}) - ((\text{price} * \text{qty}) * \text{discount})$$
 และ
 
$$\text{netPrice} = \text{priceBeforeTax} + (\text{priceBeforeTax} * \text{vat})$$
 ตามลำดับ แล้วไปทำข้อ 5
  - 4.2 ไม่ใช่ ไม่ให้ส่วนลด และคำนวณราคาขายสินค้าสุทธิตามสูตร
 
$$\text{netPrice} = ((\text{price} * \text{qty}) * \text{vat}) + (\text{price} * \text{qty})$$
 แล้วไปทำข้อ 5
5. สั่งพิมพ์ราคาขายสุทธิ
6. จบการทำงาน

ผังงาน



รูปที่ 5.4 แสดงผังงานของตัวอย่างโปรแกรมที่ 5.7

โค้ดโปรแกรม

(1) // โปรแกรมหาราคาขายสุทธิ โดยใช้ IF...ELSE

```

(2) import java.io.*;
(3) public class simplelf
(4) {
(5) public static void main (String[] args)
(6) {
(7) double price, discount;
(8) double priceBeforeTax, NetPrice;
(9) int qty;
(10) double vat = 0.07;
(11) String custMem = " ";
(12) String inputPrice = " ", inputQty = " "; //กำหนดตัวแปรมารับ price และ qty ให้เป็น String
(13)
(14) // รับค่าราคาสินค้า จำนวนสินค้า และสถานะสมาชิก
(15) BufferedReader stdin = new BufferedReader(new InputStreamReader(System.in));
(16) System.out.print("กรุณาป้อนราคาสินค้า");
(17) inputPrice = stdin.readLine(); // รับค่าราคาสินค้า
(18) price = Double.parseDouble(inputPrice); // แปลง price จาก String ไปเป็น Double
(19) System.out.print("กรุณาป้อนจำนวนสินค้าที่ขาย");
(20) inputQty = stdin.readLine();
(21) qty = Integer.parseInt(inputQty); // แปลง qty จาก String ไปเป็น Integer
(22) System.out.print("เป็นลูกค้าสมาชิกหรือไม่[Y/N] : ");
(23) custMem = stdin.readLine();
(24)
(25) // ทดสอบเงื่อนไขตามสถานะลูกค้า
(26) if (custMem.equals("Y"))
(27) {
(28) discount = 1.10;
(29) priceBeforeTax = (price * qty) - ((price * qty) * discount);
(30) netPrice = priceBeforeTax + (priceBeforeTax * vat);

```

```

(31) }
(32) else
(33) {
(34) netPrice = ((price * qty) * vat) + (price * qty);
(35) }
(36) System.out.println("ราคาขายสินค้าสุทธิ = " + netPrice);
(37) }
(38) }

```

### ผลลัพธ์

```

กรุณป้อนราคาสินค้า : 550
กรุณป้อนจำนวนสินค้าที่ขาย : 2
เป็นลูกค้าสมาชิกใช่หรือไม่ [Y/N] : Y
ราคาขายสินค้าสุทธิ = 1059.3

```

### 5.3.3 IF แบบซ้อน (Nested IF)

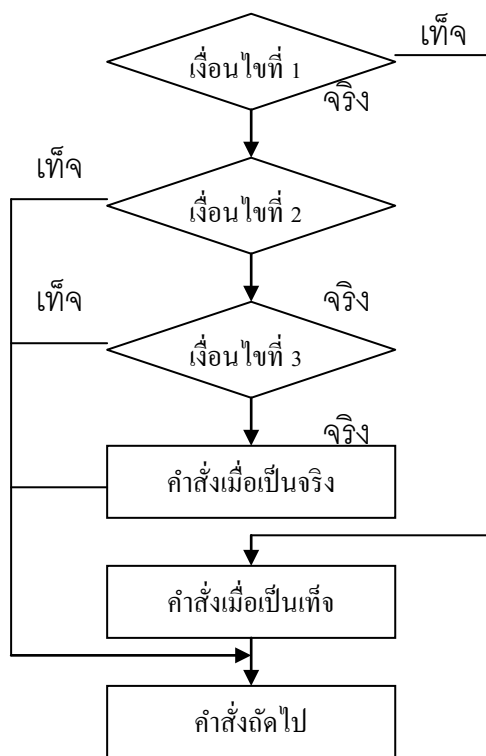
เป็นคำสั่งที่มีเงื่อนไขเพื่อการตัดสินใจมากกว่า 2 เงื่อนไขขึ้นไป หรือกล่าวโดยสรุปคือ การใช้คำสั่ง IF ซ้อน IF มี 2 รูปแบบ ดังนี้

#### รูปแบบที่ 1

```

if (เงื่อนไขที่ 1)
{
 if (เงื่อนไขที่ 2)
 {
 if (เงื่อนไขที่ 3)
 {
 ...
 }
 }
}
else
{
 คำสั่งต่างๆ เมื่อเงื่อนไขเป็นเท็จ ;
}
คำสั่งถัดไป

```



รูปที่ 5.6 แสดงการทำงานของ Nested IF รูปแบบที่ 1

ตัวอย่างที่ 5.8 การเขียนคำสั่ง IF แบบซ้อน ตามรูปแบบที่ 1

```

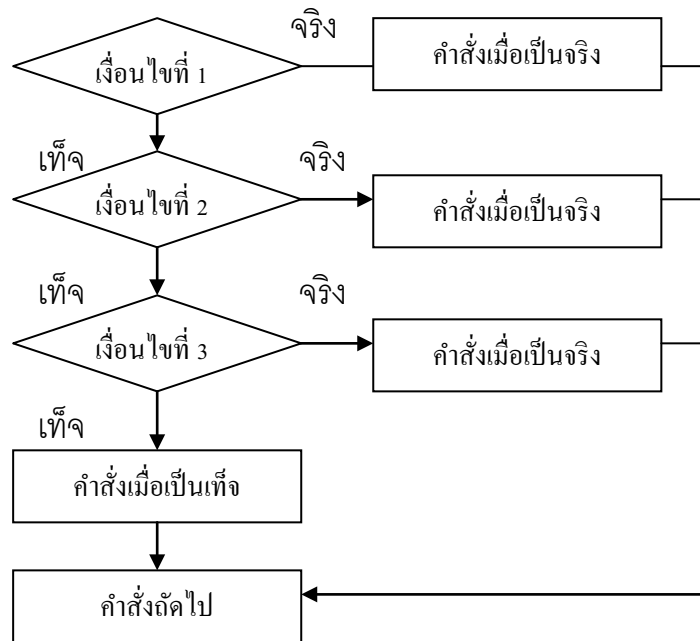
if (num1 > num2)
{
 if (num3 > num2)
 {
 System.out.println("num1 และ num3 มีค่ามากกว่า num2");
 }
}
else
 System.out.println("num1 มีค่าน้อยกว่าหรือเท่ากับ num2");

```

## รูปแบบที่ 2

```
if (เงื่อนไขที่ 1)
{
 คำสั่งเมื่อเงื่อนไขที่ 1 เป็นจริง ;
}
else if (เงื่อนไขที่ 2)
{
 คำสั่งเมื่อเงื่อนไขที่ 2 เป็นจริง ;
}
else if (เงื่อนไขที่ 3)
{
 คำสั่งเมื่อเงื่อนไขที่ 3 เป็นจริง ;
}
else
{
 คำสั่งต่างๆ เมื่อเงื่อนไขเป็นเท็จ ;
}
```





**ตัวอย่างที่ 5.9** การเขียนคำสั่ง IF แบบซ้อน ตามรูปแบบที่ 2

```

if (menu.equals("1")) {
 charge = 200; }
if (menu.equals("2")) {
 charge = 150; }
if (menu.equals("3")) {
 charge = 100; }
else {
 charge = 0; }

```

**ตัวอย่างที่ 5.10** ทดลองเขียนโปรแกรมคำนวณหาเกรดที่นักเรียนได้ จากคะแนนที่รับเข้ามาทางคีย์บอร์ด โดยมีเงื่อนไขการตัดเกรด ดังนี้

ถ้าคะแนนตั้งแต่ 0-49 ได้เกรด F  
ถ้าคะแนนตั้งแต่ 50-59 ได้เกรด D  
ถ้าคะแนนตั้งแต่ 60-69 ได้เกรด C  
ถ้าคะแนนตั้งแต่ 70-79 ได้เกรด B  
ถ้าคะแนนตั้งแต่ 80-100 ได้เกรด A

### กำหนดตัวแปร

score แทน คะแนนสอบ

grade แทน เกรด

### อัลกอริทึม

1. กำหนดตัวแปร score, grade
2. รับค่า score
3. ทดสอบค่า score
  - 3.1 ถ้า score  $\leq$  100 และ score  $\geq$  80 สั่งพิมพ์ เกรด = "A" แล้วไปทำข้อ 4
  - 3.2 ถ้า score  $\leq$  79 และ score  $\geq$  70 สั่งพิมพ์ เกรด = "B" แล้วไปทำข้อ 4
  - 3.3 ถ้า score  $\leq$  69 และ score  $\geq$  60 สั่งพิมพ์ เกรด = "C" แล้วไปทำข้อ 4
  - 3.4 ถ้า score  $\leq$  59 และ score  $\geq$  50 สั่งพิมพ์ เกรด = "D" แล้วไปทำข้อ 4
  - 3.5 ถ้า score  $\leq$  49 และ score  $\geq$  0 สั่งพิมพ์ เกรด = "F" แล้วไปทำข้อ 4
  - 3.6 นอกเหนือจากนั้นให้พิมพ์ข้อความว่า "คะแนนจะต้องอยู่ระหว่าง 0 - 100" แล้วไปทำข้อ 4
4. จบการทำงาน

### โค้ดโปรแกรม

```
(1) // โปรแกรมคำนวณหาเกรดจากคะแนนที่รับเข้ามาทางคีย์บอร์ด
(2) import java.io.*;
(3) public class nestedIF
(4) {
(5) public static void main (String[] args) throws IOException
(6) {
(7) int score;
(8) String grade = " "; // กำหนดค่าเริ่มต้นให้ตัวแปร grade
```

```

(9) String inputScore = " "; // ใช้เก็บค่าคะแนนไว้ในตัวแปรชนิด String ก่อน
(10)
(11) // รับค่าคะแนนสอบ
(12) BufferedReader stdin = new BufferedReader(new
 InputStreamReader(System.in));
(13) System.out.print("กรุณาป้อนคะแนน : ");
(14) inputScore = stdin.readLine(); // รับค่า score เก็บไว้ในตัวแปร String ก่อน
(15) score = Integer.parseInt(inputScore); // แปลงค่า score จาก String ไปเป็น Integer
(16)
(17) // ทดสอบคะแนนเพื่อหาเกรดที่ได้
(18) if ((score <= 100) && (score >= 80)) {
(19) grade = "A";
(20) System.out.println("เกรดที่ได้คือ : " + grade); }
(21) if ((score <= 79) && (score >= 70)) {
(22) grade = "B";
(23) System.out.println("เกรดที่ได้คือ : " + grade); }
(24) if ((score <= 69) && (score >= 60)) {
(25) grade = "C";
(26) System.out.println("เกรดที่ได้คือ : " + grade); }
(27) if ((score <= 59) && (score >= 50)) {
(28) grade = "D";
(29) System.out.println("เกรดที่ได้คือ : " + grade); }
(30) if ((score <= 49) && (score >= 0)) {
(31) grade = "F";
(32) System.out.println("เกรดที่ได้คือ : " + grade); }
(33) else {
(34) System.out.println("คะแนนจะต้องอยู่ระหว่าง 0 - 100"); }
(35) // จบคำสั่ง IF

```

(36) }

(37) }

### ผลลัพธ์

กรุณาป้อนคะแนน : 59

เกรดที่ได้คือ : D

## 5.4 ประโยคคำสั่ง switch

คำสั่ง switch จะตรวจสอบเงื่อนไขโดยดูจากค่าของตัวแปรที่รับมา หากตรงกับค่าคงที่ใด(case) ที่ได้กำหนดไว้ ก็จะดำเนินการตามคำสั่งของ case นั้นทันที

### รูปแบบ

```
switch (ตัวแปร)
{
 case ค่าที่ 1 : คำสั่งที่ 1;
 break;
 case ค่าที่ 2 : คำสั่งที่ 2;
 break;
 .
 .
 .
 case ค่าที่ N : คำสั่งที่ N;
 break;
 default : คำสั่งเมื่อไม่มีค่าที่ตรงกับที่ระบุใน case ;
}
```

**ตัวอย่างที่ 5.11** สมมติว่าธนาคารคิดดอกเบี้ยตามระยะเวลาการกู้เงิน โดยแบ่งเป็น 3

กลุ่ม คือ

|            |                |                 |
|------------|----------------|-----------------|
| กลุ่มที่ 1 | ระยะเวลา 5 ปี  | คิดดอกเบี้ย 12% |
| กลุ่มที่ 2 | ระยะเวลา 15 ปี | คิดดอกเบี้ย 18% |
| กลุ่มที่ 3 | ระยะเวลา 30 ปี | คิดดอกเบี้ย 24% |

จากโจทย์กำหนดให้ตัวแปรที่ใช้เพื่อตรวจสอบเงื่อนไข คือ ระยะเวลาการกู้เงิน ในที่นี้กำหนดชื่อตัวแปรเป็น "Year" มีชนิดเป็น Integer ดังนั้น ค่าที่จะต้องระบุใน case จะต้องเป็นตัวเลขที่สามารถตรวจสอบกับเงื่อนไขที่โจทย์ให้มาได้ นั่นคือ 5, 15 และ 30

สามารถใช้คำสั่ง switch เพื่อตรวจสอบเงื่อนไขข้างต้นได้ดังนี้

```
switch (year)
{
 case 5: iRate = 12;
 break;
 case 15: iRate = 18;
 break;
 case 30: iRate = 24;
 break;
 default: System.out.println("ตัวเลขผิด ต้องเป็น 5,15 หรือ 30 เท่านั้นโปรดป้อนใหม่");
}
```

\*\*\*\*\* ให้ดูตัวอย่างที่ 5.12 ในหนังสือหน้า 127 \*\*\*\*\*