



# 5

## คำสั่งควบคุมแบบวนซ้ำ

### ความคิดรวบยอด

 Video USSาย

Video 5 – คำสั่งควบคุมแบบวนซ้ำ

#### 1

##### คำสั่ง FOR

คำสั่ง for เป็นคำสั่งแบบวนซ้ำ (loop) ที่เหมาะสำหรับงานที่ต้องการ  
การกำหนดจำนวนครั้งของการวนซ้ำที่ตายตัว

#### 2

##### คำสั่ง WHILE

คำสั่ง while เป็นคำสั่งแบบวนซ้ำ (loop) ที่ตรวจสอบเงื่อนไขก่อนการ  
ทำงานงานใน loop

#### 3

##### คำสั่ง DO-WHILE

คำสั่ง do-while เป็นคำสั่งแบบวนซ้ำ (loop) ที่ตรวจสอบเงื่อนไขหลัง  
การทำงานงานใน loop เพราะฉะนั้นคำสั่งแบบ do-while จะ  
ทำงานใน loop อย่างน้อย 1 รอบเสมอ



[goo.gl/Q5ktXo](https://goo.gl/Q5ktXo)



การวนซ้ำ (Loop) หมายถึง กลุ่มของคำสั่งที่ต้องการประมวลผลซ้ำหลายครั้ง ทั้งนี้ขึ้นอยู่กับเงื่อนไขที่กำหนดไว้ ภาษาจาวามีคำสั่งวนซ้ำอยู่ 3 คำสั่ง ได้แก่ คำสั่ง for, คำสั่ง while และคำสั่ง do...while

## คำสั่ง for

คำสั่ง for เป็นคำสั่งวนลูปที่มีส่วนประกอบอยู่ 3 ส่วน ได้แก่  
1) ส่วนเริ่มต้น (initialization) 2) ส่วนเงื่อนไข (condition) และ  
3) ส่วนปรับค่า (update)

```
for (initialization; condition; update){
    statement1;
    statement2;
}
```

## คำสั่ง while

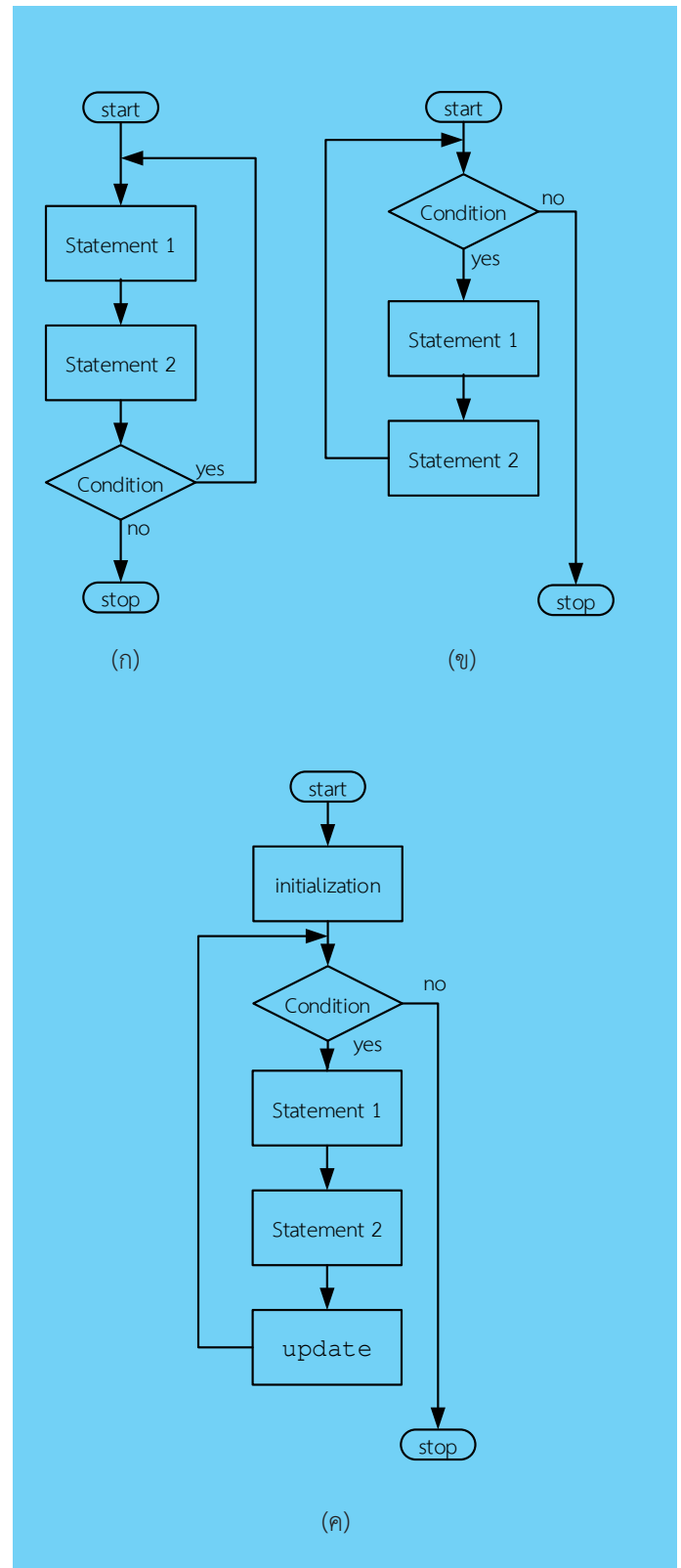
การทำงานของคำสั่ง while จะเริ่มต้นด้วยการตรวจสอบเงื่อนไข (condition) ถ้าผลการตรวจสอบเงื่อนไขเป็นจริง คำสั่ง (statement) ที่อยู่ภายใต้เครื่องหมาย { } ทุกคำสั่งจะได้รับการประมวลผลหนึ่งครั้ง แล้วโปรแกรมจะกลับไปตรวจสอบเงื่อนไข ถ้าผลการตรวจสอบยังคงเป็นจริงอยู่ โปรแกรมจะประมวลผลคำสั่งเดิมอีก 1 รอบ และจะทำซ้ำลักษณะนี้จนกระทั่งผลการตรวจสอบเงื่อนไขเป็นเท็จ จึงจะยุติการทำงานในลูป

```
while (condition){
    statement1;
    statement2;
}
```

## คำสั่ง do-while

การทำงานของคำสั่ง do...while จะเริ่มต้นด้วยการทำคำสั่ง (statement) ที่อยู่ภายใต้เครื่องหมาย { } แล้วจึงค่อยตรวจสอบเงื่อนไข (condition) ถ้าผลการตรวจสอบเป็นจริง ก็จะประมวลผลคำสั่งที่อยู่ใต้เครื่องหมาย { } อีก และทำซ้ำในลักษณะนี้จนกระทั่งการตรวจสอบเงื่อนไขเป็นเท็จ จึงจะยุติการทำงานในลูป นั่นคือจะมีการทำคำสั่งใน loop อย่างน้อย 1 ครั้งเสมอ

```
do{
    statement1;
    statement2;
}while (condition);
```



Flow Chart การทำงานของคำสั่งแบบวนซ้ำ (ก) คำสั่ง do-while  
(ข) คำสั่ง while และ (ค) คำสั่ง for



### ตัวอย่างที่ 5-1

```
1 public class ForTest1 {  
2     public static void main(String[] args) {  
3         int digit;  
4         for(digit=0; digit<=9; digit++)  
5             System.out.print(digit);  
6     }  
7 }
```

Output:  
0123456789

ตัวอย่างการใช้คำสั่ง *for*

### ตัวอย่างที่ 5-2

```
1 public class WhileTest1 {  
2     public static void main(String[] args) {  
3         int digit = 0;  
4         while(digit <= 9) {  
5             System.out.print(digit);  
6             digit++;  
7         }  
8     }  
9 }
```

Output:  
0123456789

ตัวอย่างการใช้คำสั่ง *while*

### ตัวอย่างที่ 5-3

```
1 public class DoWhileTest1 {  
2     public static void main(String[] args) {  
3         int digit = 0;  
4         do {  
5             System.out.print(digit);  
6             digit++;  
7         } while(digit <= 9);  
8     }  
9 }
```

Output:  
0123456789

ตัวอย่างการใช้คำสั่ง *do-while*

จากตัวอย่างที่ 5-1, 5-2 และ 5-3 เป็นตัวอย่างการใช้คำสั่งแบบวนซ้ำ นักเรียนจะสังเกตว่าผลลัพธ์ของทั้ง 3 โปรแกรมเหมือนกันคือพิมพ์ตัวตั้งแต่ 0 ถึง 9 ออกทางหน้าจอ แต่ใช้คำสั่งแตกต่างกันเป็นเพราะว่าทั้งคำสั่ง *for*, *while* และ *do-while* ต่างก็มีการทำงานคล้ายๆ กัน คือ ทำซ้ำคำสั่งจนกว่าเงื่อนไขการวนซ้ำจะเป็นเท็จ



## สถานการณ์ปัญหา 5-1

ที่บริษัทผลิตซอฟต์แวร์แห่งหนึ่ง วันหนึ่งมีลูกค้ามาปรึกษาเกี่ยวกับการสร้างโปรแกรมจำลองการเคลื่อนที่ในแนวเส้นตรง ลูกค้านี้ให้เงื่อนไขมาว่าระยะทางที่เคลื่อนที่ไปได้จะเท่ากับความเร็วคูณกับเวลา ความต้องการของลูกค้าจะป้อนความเร็ว กับเวลาเข้าไปในโปรแกรม จากนั้นโปรแกรมจะคำนวณระยะทางที่เคลื่อนที่ไปได้ของแต่ละวินาที ถ้านักเรียนเป็นคนพัฒนาโปรแกรม นักเรียนจะวิเคราะห์ปัญหา และเสนอแนวทางแก้ไขอย่างไร

## สถานการณ์ปัญหา 5-2

ที่บริษัทผลิตซอฟต์แวร์แห่งหนึ่ง วันหนึ่งมีลูกค้ามาปรึกษาเกี่ยวกับการสร้างโปรแกรมเรียนรู้ภาษาอาเซียน โดยโปรแกรมรับจะชื่อประเทศสมาชิก AEC (กำหนดให้ชื่อประเทศจำต้องขึ้นต้นด้วยตัวพิมพ์ใหญ่ เช่น Thailand หรือ Loa เป็นต้น) แล้วแสดงคำทักทายของประเทศนั้นๆ ออกทางจอภาพ (สกดด้วยภาษาไทยหรือภาษาอังกฤษก็ได้) หากไม่ใช่ประเทศสมาชิก AEC ให้ขึ้นข้อความว่า “ประเทศนี้ไม่ได้เป็นสมาชิกอาเซียน” โดยโปรแกรมจะวนซ้ำเพื่อรับคำสั่ง จนกว่าจะกด q เพื่อจบการทำงาน ถ้านักเรียนเป็นคนพัฒนาโปรแกรม นักเรียนจะวิเคราะห์ปัญหา และเสนอแนวทางแก้ไขอย่างไร

## สถานการณ์ปัญหา 5-3

บริษัทผลิตซอฟต์แวร์แห่งหนึ่ง วันหนึ่งมีลูกค้ามาปรึกษาเกี่ยวกับการสร้างโปรแกรมโปรแกรมเพื่อคำนวณค่าบริการโทรศัพท์รายเดือน โดยให้ผู้ใช้ป้อนระยะเวลาที่ใช้งานไป และจำนวนของอินเทอร์เน็ตที่ใช้ไป เพื่อคำนวณออกมาเป็นค่าโทรศัพท์ โดยรายละเอียดโปรแกรมเป็นดังนี้

ค่าบริการรายเดือน 349 บาท

โทรฟรีทุกเครือข่าย 200 นาที

และใช้อินเทอร์เน็ตได้ 500 MB

ค่าโทรและจำนวนอินเทอร์เน็ตส่วนเกินคิดเป็น

ค่าโทร 1.27 บาท/นาที

อินเทอร์เน็ต 1.27 บาท/MB

เมื่อคำนวณเสร็จแล้วโปรแกรมจะวนซ้ำกลับไปรับค่าจากผู้ใช้นี้ใหม่ จนกว่าผู้ใช้จะกดปุ่ม q เพื่อออกจากการทำงานถ้านักเรียนเป็นคนพัฒนาโปรแกรม นักเรียนจะวิเคราะห์ปัญหา และเสนอแนวทางแก้ไขอย่างไร

```
Output:
***** DISTANCES CALCULATION***
Formular: S=v*t
Enter velocity(m/s): 5
Enter time(s): 3
Distances:
t = 0: S = 0;
t = 1: S = 5;
t = 2: S = 10;
t = 3: S = 15;
```

ตัวอย่างผลลัพธ์ 4-1

```
Output:
*****ASIAN GREETING WORDS*****
Enter the fisrt charater: T
Greeting: SA-WAD-DEE

*****ASIAN GREETING WORDS*****
Enter the fisrt charater: _
```

ตัวอย่างผลลัพธ์ 5-2

```
Output:
***PHONE FEE CALCULATION***
Press 'q' to quite
Enter call duration (minutes): 100
Enter internet (MB): 400
TOTAL (THB): 349

***PHONE FEE CALCULATION***
Press 'q' to quite
Enter call duration (minutes):_
```

ตัวอย่างผลลัพธ์ 5-3

### การทักทายการเรียนรู้

1. นักเรียนวิเคราะห์ว่าจะใช้คำสั่งอะไรบ้างในการเขียนโปรแกรม
2. นักเรียนวิเคราะห์ว่าจะแก้ปัญหาดังกล่าวอย่างไร อธิบายเป็น Flow Chart อธิบายว่าปัญหาใหม่มีความเหมือนหรือแตกต่างกับปัญหาที่นักเรียนเคยแก้ได้อย่างไร
3. นักเรียนออกแบบและพัฒนาโปรแกรมเพื่อแก้ปัญหาดังกล่าว