4
คำสั่งควบคุมแบบมีทางเลือก

ความคิดรวบยอด

1 คำสั่ง IF

คำสั่ง if เป็นคำสั่งที่ใช้เป็นทางเลือกในการตัดสินใจทำงานอย่างใดอย่างหนึ่ง โดยมีการเขียนนิพจน์ที่เป็นเงื่อนไขทางตรรกศาสตร์ในคำสั่ง if และจะทำคำสั่งใน if ถ้าเงื่อนไขเป็นจริง

2 คำสั่ง SWITCH

การเขียนโปรแกรมที่ต้องการเลือกทำหลายทางเลือก เราสามารถนำคำสั่ง if มาซ้อนกันได้ แต่ถ้าเงื่อนไขที่ต้องตัดสินใจขึ้นกับตัวแปรเดียวหรือนิพจน์ใดๆ เราสามมารถใช้คำสั่ง switch แทนได้

Video บรรยาย

Video 4 – คำสั่งควบคุมแบบมีทางเลือก

goo.gl/Q5ktXo

คำสั่งควบคุมแบบมีทางเลือก 4

คำสั่งควบคุมแบบมีทางเลือก คือ คำสั่งที่ใช้ในการตัดสินใจหรือเลือกทำเหตุการณ์ใดเหตุการณ์หนึ่ง โดยที่มีการตรวจสอบเงื่อนไขของคำสั่งก่อนทำงาน เพื่อตัดสินใจเลือกทิศทางการทำงานของโปรแกรม ซึ่งคำสั่งควบคุมแบบมีทางเลือกนี้ มีอยู่ 2 คำสั่งหลักๆ ได้แก่ คำสั่ง if และ คำสั่ง switch

คำสั่ง if

คำสั่ง if เป็นคำสั่งสำหรับให้โปรแกรมเลือกทำ โดยคำสั่ง if มีอยู่ 3 รูปแบบ คือ

คำสั่ง if แบบทางเดียว เป็นคำสั่งที่ใช้เป็นทางเลือกในการตัดสินใจทำงานอย่างใดอย่างหนึ่ง โดยมีการเขียนนิพจน์ที่เป็นเงื่อนไขทางตรรกศาสตร์ในคำสั่ง if และจะทำคำสั่งใน if ถ้าเงื่อนไขเป็นจริง

```
if (condition)
   statement;
```

หรือ

```
if (condition){
   statements;
}
```

รูปแบบการเขียนคำสั่ง if ทางเดียว

คำสั่ง if สองทางเลือก โปรแกรมจะทำตามชุดคำสั่งใดคำสั่งหนึ่งจาก 2 ทางเลือก โดยตรวจสอบนิพจน์เงื่อนไขที่กำหนดว่าเป็นจริงหรือเท็จ ถ้านิพจน์เป็นจริง โปรแกรมจะทำงานตามชุดคำสั่งภายใต้ if แต่ถ้านิพจน์เป็นเท็จ โปรแกรมจะทำงานตามชุดคำสั่งที่อยู่ภายใต้ else

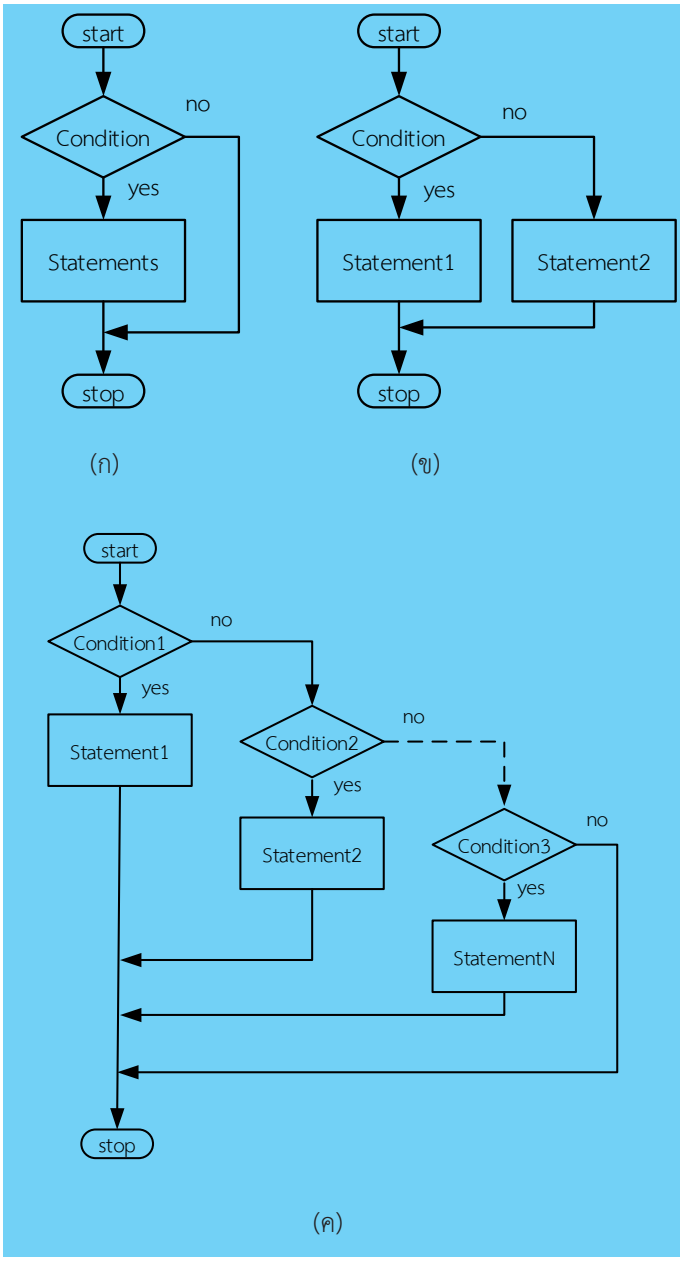
```
If(condition){
       statement1;
}
else{
       statement2;
}
```

รูปแบบการเขียนคำสั่ง if สองทางเลือก

Nested-if เป็นคำสั่งควบคุมให้โปรแกรมเลือกทำงานชุดคำสั่งใดชุดคำสั่งหนึ่งจากหลายทางเลือก แต่ละทางเลือกจะมีการกำหนดนิพจน์เงื่อนไขของแต่ละทางไว้ด้วย โดยโปรแกรมจะตรวจสอบนิพจน์เงื่อนไขของแต่ละทางเลือก หากพบว่าทางเลือกไหนมีนิพจน์เป็นจริง ก็จะทำงานในทางเลือกนั้นโดยไม่พิจารณาทางเลือกอื่นที่ยังไม่ตรวจสอบอีก

```
if(condition1){
       statements1;
}else if (condition2){
       statements2;
}
.
.
else{
       statementsN;
}
```

รูปแบบการเขียนคำสั่ง if แบบหลายทางเลือก



Flow Chart การทำงานของคำสั่งควบคุมแบบมีทางเลือก (ก) คำสั่ง if แบบทางเดียว (ข) คำสั่ง if แบบสองทาง และ (ค) คำสั่ง if แบบหลายทาง (nested if)

คำสั่ง switch

การเขียนโปรแกรมที่ต้องการเลือกทำหลายทางเลือก เราสามารถนำคำสั่ง if มาซ้อนกันได้ แต่ถ้าเงื่อนไขที่ต้องตัดสินใจขึ้นกับตัวแปรเดียวหรือนิพจน์ใดๆ เราสามมารถใช้คำสั่ง switch แทนได้

คำสั่ง switch เป็นคำสั่งที่จะตรวจสอบค่าของตัวแปรหรือนิพจน์ ถ้าตรงตาม case ใดก็จะทำตามคำสั่งนั้นๆ จนกว่าจะเจอคำสั่ง break ในกรณีที่ไม่ตรงกับ case ที่ระบุในโปรแกรม จะมาทำงานที่คำสั่ง default โดยอัตโนมัติ

```
switch(variable/expression){
  case constant1:   statements;
                    break;
  case constant2:   statements;
                    break;
       .
       .
       .
  case constantN:   statements;
                    break;
  default:          statements;
                    break;
}
```

รูปแบบการเขียนคำสั่ง switch

ตัวอย่างที่ 4-1

```
1  public static void main(String[] args) {
2     Scanner in = new Scanner(System.in);
3     System.out.print("Enter score: ")
4     float score = in.nextFloat();
5     if(score < 50) {
6       System.out.println("Failed");
7       System.out.println("You must take this
            course again.");
8     }
9  }
```

```
Output:
Enter score: 10
Failed
You must take this course again.
```

จากตัวอย่างที่ 4-1 เป็นการใช้คำสั่ง if แบบทางเดียวเพื่อตรวจสอบค่าคะแนนที่ป้อนเข้ามา โดยเงื่อนไขในคำสั่ง if (บรรทัดที่ 5) เขียนไว้เพื่อตรวจสอบค่า score หากมีค่าน้อยกว่า 50 ก็จะทำงานภายในคำสั่ง if จากกรณีตัวอย่างผู้ใช้ป้อนค่าคะแนนเป็น 10 ค่านี้จะถูกเก็บไว้ในตัวแปร score ทำให้นิพจน์ภายใน if จึงเป็นจริง โปรแกรมจึงพิมพ์ข้อความว่า Failed และ `You must take this course again.`

ตัวอย่างที่ 4-2

```
1  public class ScoreTest {
2    public static void main(String[] args) {
3      Scanner in = new Scanner(System.in);
4      float score = in.nextFloat();
5      if(score >= 50){
6        System.out.println("Passed");
7      }
8      else {
9        System.out.println("Failed");
10       System.out.println("You must take this
11         course again.");
12     }
13   }
14 }
```

```
Output:
Enter score: 10
Failed
You must take this course again.
```

จากตัวอย่างที่ 4-2 เป็นการใช้งานคำสั่ง if...else เพื่อตรวจสอบคะแนนสอบของนักเรียน โดยจะแสดงข้อความ Passed ออกทางจอภาพ ถ้าคะแนนสอบของนักเรียนมากกว่าหรือเท่ากับ 50 แต่ถ้าเงื่อนไขเป็นเท็จ (คะแนนสอบของนักเรียนน้อยกว่า 50 คะแนน) จะแสดงข้อความ Failed และ You must take this course again. จากตัวอย่างผู้ใช้ได้ป้อนคะแนนเข้ามา 10 คะแนน ทำให้เงื่อนไขภายในคำสั่ง if เป็นเท็จ โปรแกรมจึงทำงานภายในขอบเขตของคำสั่ง else

คำสั่งควบคุมแบบมีทางเลือก 4

ตัวอย่างที่ 4-3

```
1  import java.util.Scanner;
2  public class GradeTest {
3    public static void main(String[] args) {
4      Scanner in = new Scanner(System.in);
5      System.out.print("Enter Score: ");
6      float score = in.nextFloat();
7      char grade;
8      if(score >= 80)
9        grade = 'A';
10      else if(score >= 70)
11       grade = 'B';
12     else if(score >= 60)
13       grade = 'C';
14     else if(score >= 50)
15       grade = 'D';
16     else
17       grade = 'F';
18     System.out.println(grade);
19   }
20 }
```

```
Output:
Enter score: 80
A
```

ตัวอย่างการใช้งาน nested-if ได้แก่โปรแกรมคำนวณเกรด แล้วแสดงผลออกทางจอภาพ โดยโปแกรมจะรอรับค่าคะแนนจากผู้ใช้ในช่วง 0 – 100 แล้วแสดงเกรดในระบบ A – F ออกทางจอภาพ จากตัวอย่างผู้ใช้ได้ป้อนค่าคะแนนเข้ามา 80 คะแนน คำสั่งในเงื่อนไข if (บรรทัดที่ 8) จึงเป็นจริง ทำให้ตัวแปร grade ม่ค่าเป็นอักจะ 'A' จากนั้นโปรแกรมจะกระโดดไปทำงานที่บรรทัดที่ 18 คือพิมพ์อักขระ A ออกมานั่นเอง

ตัวอย่างที่ 4-4

```
1  public class SwitchDemo {
2    public static void main(String[] args) {
3      int day = 5;
4      String dayString;
5      switch (day) {
6        case 1: dayString = "Monday";
7          break;
8        case 2: dayString = "Tuesday";
9          break;
10       case 3: dayString = "Wednesday";
11         break;
12       case 4: dayString = "Thursday";
13         break;
14       case 5: dayString = "Friday";
15         break;
16       case 6: dayString = "Saturday";
17         break;
18       case 7: dayString = "Sunday";
19         break;
20       default: dayString = "Invalid day";
21     }
22     System.out.println(dayString);
23   }
24 }
```

```
Output:
Friday
```

จากตัวอย่างที่ 4-4 เป็นการใช้คำสั่ง switch เพื่อพิมพ์ชื่อวันออกทางหน้าจอ โดยใช้หมายเลขวันที่เก็บไว้ในตัวแปร day ในตัวอย่างค่าของ day คือ 5 เพราะฉะนั้นเมื่อโปรแกรมนำ day ไปตรวจสอบด้วยคำสั่ง switch จะตรงกับ case 5 โปรแกรมจึงเก็บข้อความ "Friday" ไว้ในตัวแปร dayString (ตัวแปรชนิด String คือตัวแปรใช้เก็บข้อความ) เมื่อโปรแกรมทำงานถึงคำสั่ง break; ก็จะกระโดดมาทำคำสั่ง System.out.print (บรรทัด 22) เพื่อพิมพ์ค่าของ dayString ออกทางหน้าจอ ผลลัพธ์ที่ได้คือโปรแกรมจะพิมพ์ค่า Friday ออกทางหน้าจอ

สถานการณ์ปัญหา 4-1

ที่บริษัทผลิตซอฟต์แวร์แห่งหนึ่ง วันหนึ่งมีลูกค้ามาปรึกษาเกี่ยวกับการสร้างโปรแกรมคิดเงินร้านค้า ปัญหาที่พบบ่อยๆ คือคิดเงินผิดจึงอยากจะได้โปรแกรมที่จะช่วยให้การคิดเงินถูกต้องและรวดเร็วขึ้น ถ้านักเรียนเป็นคนพัฒนาโปรแกรม นักเรียนจะวิเคราะห์ปัญหา และเสนอแนวทางแก้ไขอย่างไร

```
Output:
Product price : 100
Cash : 150
Change : 50
```

ตัวอย่างผลลัพธ์ 4-1

สถานการณ์ปัญหา 4-2

ที่บริษัทผลิตซอฟต์แวร์แห่งหนึ่ง วันหนึ่งมีลูกค้ามาปรึกษาเกี่ยวกับการสร้างโปรแกรมตรวจสอบโรคอ้วน โดยค่าดรรชนีมวลกาย ตั้งแต่ 30 ขึ้นไปเรียกว่าเป็นโรคอ้วน โดยความต้องการของผู้ใช้ต้องการให้มีการป้อนค่าน้ำหนักและส่วนสูงเข้าสู่โปรแกรม จากนั้นโปรแกรมจะทำการคำนวณและบอกว่าเป็นโรคอ้วนหรือไม่ ถ้านักเรียนเป็นคนพัฒนาโปรแกรม นักเรียนจะวิเคราะห์ปัญหา และเสนอแนวทางแก้ไขอย่างไร

```
Output:
Enter weight(kg): 55
Enter hight(cm): 165
BMI: 21.2 - Normal
```

ตัวอย่างผลลัพธ์ 4-2

สถานการณ์ปัญหา 4-3

ที่บริษัทผลิตซอฟต์แวร์แห่งหนึ่ง วันหนึ่งมีลูกค้ามาปรึกษาเกี่ยวกับการสร้างโปรแกรมตรวจสอบสุขภาพของนักเรียน ซึ่งปัญหาของนักเรียนในโรงเรียนส่วนใหญ่ก็คือโรคอ้วน หากนักเรียนในโรงเรียนได้รับข้อมูลการตรวจสุขภาพของตัวเองตั้งแต่เนิ่นๆ ก็จะเป็นผลดีต่อการกำหนดวิธีการดำเนินชีวิตประจำวัน เช่น การควบคุมอาหาร แต่เนื่องจากนักเรียนในโรเรียนมีมากทำให้การดำเนินการล่าช้า โดยความต้องการของผู้ใช้คือโปรแกรมสามารถรายงานผลออกมา 4 ระดับ คือ

BMI <18.5 หมายถึง น้ำหนักต่ำกว่ามาตรฐาน (Underweight)

BMI 18.5–24.9 หมายถึง ปกติ (Normal weight)

BMI 25–29.9 หมายถึง น้ำหนักเกิน (Overweight)

BMI >30 หมายถึง อ้วน (Obesity)

ถ้านักเรียนเป็นคนพัฒนาโปรแกรม นักเรียนจะวิเคราะห์ปัญหา และเสนอแนวทางแก้ไขอย่างไร

```
Output:
Enter weight(kg): 70
Enter hight(cm): 165
BMI: 27.0 - Overweight
```

ตัวอย่างผลลัพธ์ 4-3

ภารกิจการเรียนรู้

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