

Part



5

Java Programming Language
Mr.Rungrote Phonkam
rungrote@it.kmitl.ac.th



Contents

1. Expressions
2. Control Flow
 - 2.1 Condition
 - 2.2 Multiple Branch
 - 2.3 Loppss
 - 2.4 Jump



1. Expressions

เอ็กซ์เพรสชัน

- ประเภทการคำนวณ
 - หมายถึงการเขียนสแตตเมนต์ในซอร์สโค้ด เพื่อให้เกิดการประมวลผลที่ได้ผลลัพธ์
 - สร้างได้ทั้งจากข้อมูลค่าคงที่ ตัวแปร และตัวดำเนินการ

ตัวอย่าง

$5 / b \% 2 + 10;$

$(a < b) \&\& (c > d);$

$5 * 800;$

$a = 12.5;$

ประเภทตัวดำเนินการ	ตัวดำเนินการ
Postfix operators	[] . (params) expr++ expr--
Unary operators	++expr --xpr +expr -expr ~ !
Creation or cast	new (type)expr
Multiplicative	* / %
additive	+ -
shift	<< >> >>>
relational	< > <= >= instanceof
equality	== !=
bitwise AND	&
bitwise exclusive OR	^
bitwise inclusive OR	
logical AND	&&
Logical OR	
Conditional	? :
Assignment	= += -= *= /= %= &= ^= = <<= >>= >>>=



1. Expressions

สมการ $a + b - 100 * 2 / 5$

เมื่อให้ $a = 45$ และ $b = 20$

ขั้นตอนการดำเนินการคือ $((a + b) - ((100 * 2) / 5))$

1: $a + b - (200) / 5$

2: $a + b - (50)$

3: $(65) - 50$

4: 15

สมการ $--a \% b++ * 15$

เมื่อให้ $a = 3$ และ $b = 1$

ขั้นตอนการดำเนินการคือ $--a \% b++ * 15$

1: $(--3) \% (1++) * 15$

2: $2 \% 2 * 15$

3: $0 * 15$

4: 0



1. Expressions

เอ็กซ์เพรสชัน

- ประเภทการเรียกใช้
 - หมายถึงการเขียนสแตตเมนต์ในการเรียกใช้เมธอดของคลาส (กรณีไม่ได้สร้าง Instance) หรือของ Object

ตัวอย่าง

คำสั่ง 1

```
Integer.parseInt("12");
```

คำสั่ง 2

```
System.out.println("Hello Java");
```



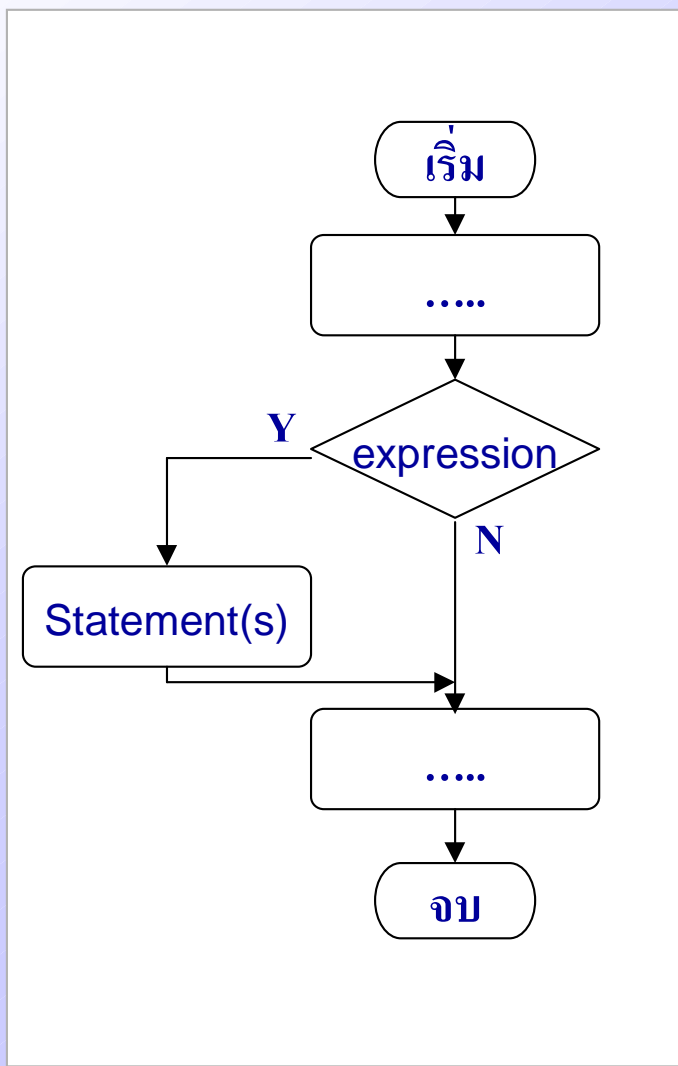
2. Control Flows

สำหรับควบคุมการทำงานของสแตจเมนต์

- เงื่อนไข (Conditions)
 - if ... else
- ตรวจสอบค่า (Multiple Branch)
 - switch .. case
- ลูป (Loop)
 - for ...
 - do ... while
 - while ...
- กระโดด (Jump)
 - break, continue, return



2.1. Condition



เงื่อนไขทางเดียว

Expression true false

Statement ทำงาน ไม่ทำงาน



2.1. Condition

รูปแบบ

```
if ( expression )  
    statements
```

เมื่อ

- if** คือคีย์เวิร์ดเพื่อตรวจสอบข้อมูลจากเอ็กเพรสชัน
- expression** คือเอ็กเพรสชันที่ให้ผลลัพธ์เป็นค่าตรรกะ โดยให้อยู่ภายในวงเล็บ
- statements** คือคำสั่งในการทำงานใดๆ ถูกทำงานเมื่อ expression เป็น true



2.1. Condition

```
class CheckPositiveNumber1
{
    public static void main(String args[])
    {
        if (Integer.parseInt(args[0])>0)
            System.out.println("Your Number [" +
                Integer.parseInt(args[0]) + "] is Positive");
    }
}
```

```
java CheckPositiveNumber1 10
Your Number [10] is Positive
```



2.1. Condition

```
class CheckPositiveNumber2
{   public static void main(String args[])
    {       if (Integer.parseInt(args[0])>0)
            {           System.out.print("Your Number [");
                        System.out.print(Integer.parseInt(args[0]));
                        System.out.println("] is Positive");
            }
    }
}
```

```
java CheckPositiveNumber2 15
Your Number [15] is Positive
```



2.1. Condition

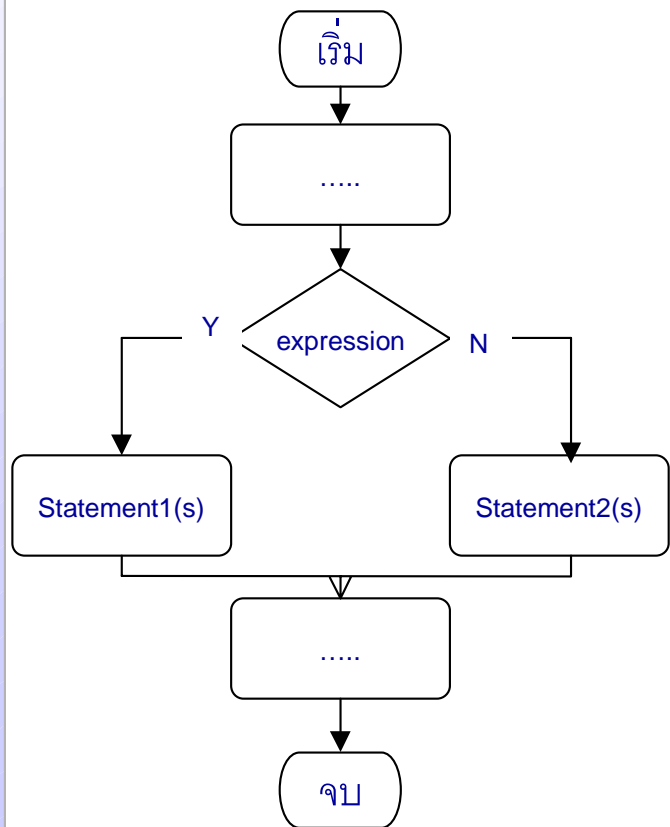
```
class CheckPositiveNumber3
{
    public static void main(String args[])
    {
        if (Integer.parseInt(args[0])>0)
            System.out.pring("Your Number [");
            System.out.pring(Integer.parseInt(args[0]));
            System.out.pringln("] is Positive");
    }
}
```

```
java CheckPositiveNumber3 20
Your Number [20] is Positive
```



2.1. Condition

เงื่อนไขสองทาง



Expression	true	false
Statement1	ทำงาน	ไม่ทำงาน
Statement2	ไม่ทำงาน	ทำงาน



2.1. Condition

รูปแบบ

```
if ( expression )  
    statement1  
else statement2
```

เมื่อ

if	คือคีย์เวิร์ดเพื่อตรวจสอบข้อมูลจากเอ็กเพรสชัน
expression	คือเอ็กเพรสชันที่ให้ผลลัพธ์เป็นค่าตรรกะ โดยให้อยู่ภายในวงเล็บ
else	คือคีย์เวิร์ดสำหรับควบคุมให้ทำงานตามคำสั่งเมื่อ expression ให้ผลลัพธ์เป็น false
statement1	คือสแตตเมนต์ที่ถูกทำงานเมื่อ expression ให้ผลลัพธ์เป็น true
statement2	คือสแตตเมนต์ที่ถูกทำงานเมื่อ expression ให้ผลลัพธ์เป็น false



2.1. Condition

```
class YourAge1
{
    public static void main(String args[])
    {
        if (Integer.parseInt(args[0])>30)
            System.out.println("You're Older");
        else
            System.out.println("You're Younger");
    }
}
```

```
java YourAge1 18
You're Younger
java YourAge1 35
You're Older
```





2.1. Condition

```
class YourAge2
{
    public static void main(String args[])
    {
        if (Integer.parseInt(args[0])>30)
            System.out.println("You're Older");
        if (Integer.parseInt(args[0])<30)
            System.out.println("You're Younger");
    }
}
```

```
java YourAge2 15
You're Younger
java YourAge2 59
You're Older
```

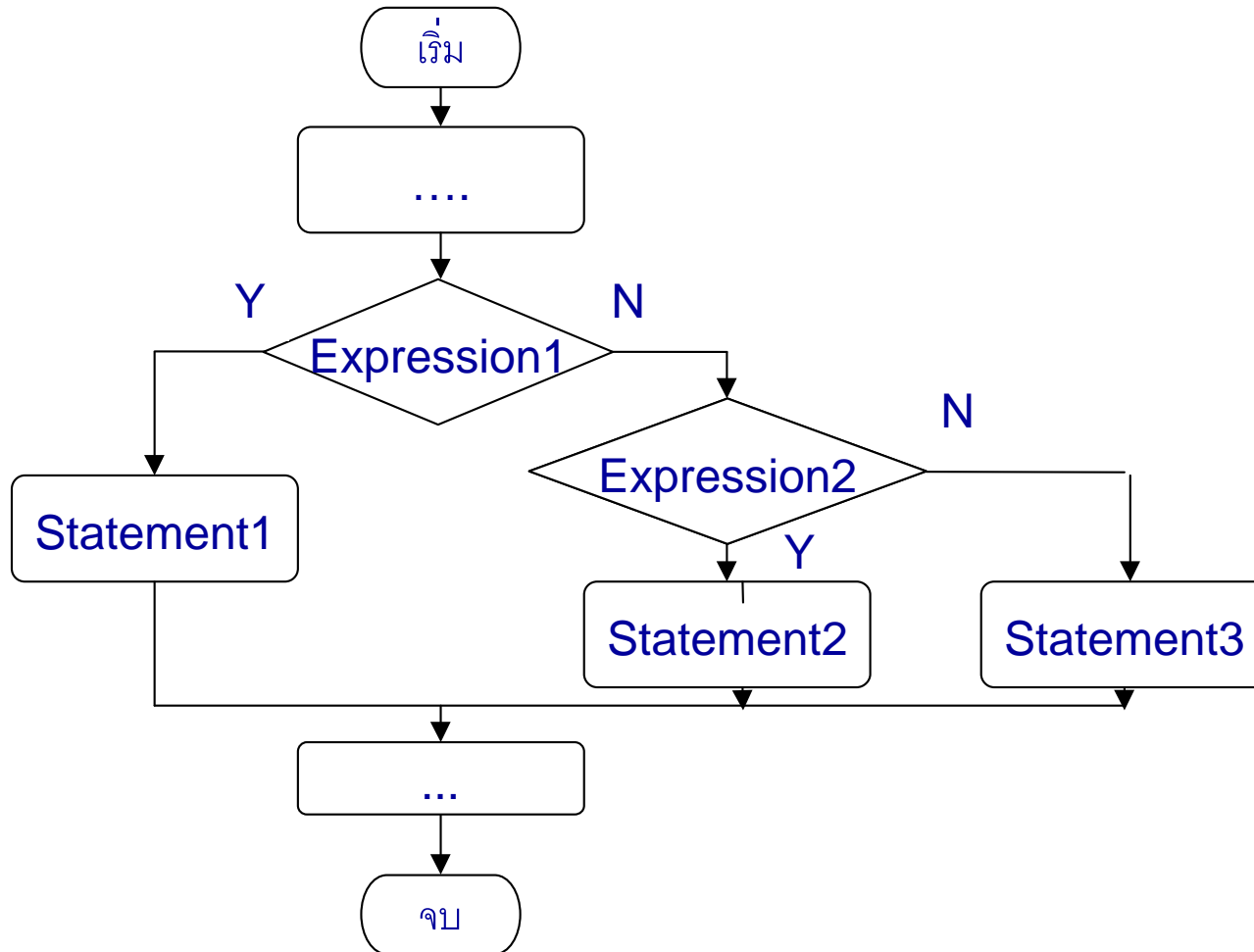



2.1. Condition

```
class YourAge3
{ public static void main(String args[])
  { if (Integer.parseInt(args[0])>30)
    { System.out.println("You're Older");
      System.out.println("Younger than 30 about " + 
        (30 - Integer.parseInt(args[0]]));
    }
  }
  else
  { System.out.println("You're Younger");
    System.out.println("Older than 30 about " + 
      (Integer.parseInt(args[0]) - 30));
  }
}
}
```



2.1. Condition





2.1. Condition

เงื่อนไขซ้อน

Expression1	true	true	false	false
Expression2	true	false	true	false
Statement1	ทำงาน	ทำงาน	ไม่ทำงาน	ไม่ทำงาน
Statement2	ไม่ทำงาน	ไม่ทำงาน	ทำงาน	ไม่ทำงาน
Statement3	ไม่ทำงาน	ไม่ทำงาน	ไม่ทำงาน	ทำงาน



2.1. Condition

```
class CheckUserPassword
{
    public static void main(String args[])
    {
        if (arg.length == 2)
            if (args[0].equals("JAVA"))
            {
                if (args[1].equals("best"))
                    System.out.println("Corrected USER and
                                        PASSWORD");
            }
            else System.out.println("Wrong USER and
                                    PASSWORD");
            else System.out.println("Please, Fill parameter USER
                                    and PASSWORD");
    }
}
```



2.1. Condition

```
java CheckUserPassword
```

```
Please, Fill parameter USER and PASSWORD
```

```
java CheckUserPassword Hi Java
```

```
Wrong USER and PASSWORD
```

```
java CheckUserPassword JAVA BEST
```

```
Corrected USER and PASSWORD
```



2.1. Condition

```
class YourAge4
{ public static void main(String args[])
  { int a = Integer.parseInt(args[0]);
    if (a <=0)
      System.out.println("Impossible! Check Your Age");
    else if (a <= 10)
      System.out.println("Hi Babe!");
    else if (a <= 20)
      System.out.println("Wow! Teeage.");
    else if (a <= 30)
      System.out.println("Good!
                          Younger.");
    else if (a >30)
      System.out.println("How about
                          your healthy?");
  }
}
```



2.1. Condition

```
java YourAge4 -3
Impossible! Check Your Age
java YourAge4 5
Hi Babe!
java YourAge4 18
Wow! Teeage.
java YourAge4 25
Good! Younger.
java YourAge4 31
java YourAge4
```



2.2. Multiple Branch

รูปแบบ

```
switch ( Variable )
{
    case Data_1: statement_1; break;
    case Data_2: statement_2; break;
    ...
    case Data_n: statement_n; break;
    default: statement;
}
```

switch คือคีย์เวิร์ดสำหรับควบคุมการทำงานเงื่อนไขหลายเส้นทาง
ต้องตามด้วยตัวแปรอยู่ในเครื่องหมายวงเล็บเสมอ

Variable คือตัวแปรใดๆ

break คือคีย์เวิร์ดประกอบการใช้งาน *switch...case* สำหรับหยุด

case คือคีย์เวิร์ดสำหรับกำหนดเส้นทางเพื่อให้ทำคำสั่งโดยขึ้นอยู่กับข้อมูล

default คือคีย์เวิร์ดสำหรับทำงานคำสั่ง เมื่อข้อมูลในตัวแปรไม่ตรงกับ *case* ใดๆ



2.2. Multiple Branch

```
class MonthName
{
    public static void main(String args[])
    {
        int a = Integer.parseInt(args[0]);
        switch (a)
        {
            case 1: System.out.println("Jan"); break;
            case 2: System.out.println("Feb"); break;
            case 3: System.out.println("Mar"); break;
            case 4: System.out.println("Apr"); break;
            case 5: System.out.println("May"); break;
            case 6: System.out.println("Jun"); break;
            case 7: System.out.println("Jul"); break;
            case 8: System.out.println("Aug"); break;
            case 9: System.out.println("Sep"); break;
            case 10: System.out.println("Oct"); break;
            case 11: System.out.println("Nov"); break;
            case 12: System.out.println("Dec"); break;
        }
    }
}
```



2.2. Multiple Branch

```
class DayName1
{   public static void main(String args[])
    {       int a = Integer.parseInt(args[0]);
            switch (a)
            {       case 1: System.out.println("Sun.");
                    case 2: System.out.println("Mon.");
                    case 3: System.out.println("Tue.");
                    case 4: System.out.println("Wed.");
                    case 5: System.out.println("Thu.");
                    case 6: System.out.println("Fri.");
                    case 7: System.out.println("Sat.");
            }
    }
}
```



2.2. Multiple Branch

DayName1 5

Thu

Fri

Sat



2.2. Multiple Branch

```
class DayInMonth
{   public static void main(String args[])
    {   int month = Integer.parseInt(args[0]);
        int year = Integer.parseInt(args[1]);
        int numDays = 0;
        switch (month)
        {
            case 1:   case 3:   case 5:   case 7:
            case 8:   case 10:  case 12:
                        numDays = 31;   break;
            case 4:   case 6:   case 9:   case 11:
                        numDays = 30;   break;
            case 2:
                if ( ((year % 4 == 0) && !(year % 100 == 0)) ||
                    (year % 400 == 0) )
                    numDays = 29;
                else    numDays = 28;    break;
        }
        System.out.println(numDays);
    }
}
```



2.2. Multiple Branch

```
class DayName2
{ public static void main(String args[])
  { int a = Integer.parseInt(args[0]);
    switch (a)
    { case 1: System.out.println("Sun."); break;
      case 2: System.out.println("Mon."); break;
      case 3: System.out.println("Tue."); break;
      case 4: System.out.println("Wed.");break;
      case 5: System.out.println("Thu.");break;
      case 6: System.out.println("Fri."); break;
      case 7: System.out.println("Sat."); break;
      default: System.out.println("Invalid Number");
    }
  }
}
```



2.3. Loops

รูปแบบ

```
for ( Initial_Counter ; Condition ; Increase_Counter หรือ  
      Decrease_Counter )  
    Statements
```

เมื่อ

for	คือคีย์เวิร์ดสำหรับกำหนดการทำงานแบบลูป
(; ;)	คือเอ็กเพรสชันที่ใช้ควบคุมการทำงานลูป มีทั้งหมดสามกลุ่มที่ถูกแยกด้วยสัญลักษณ์ ;
<i>Initial_Counter</i>	คือการทำค่ากับตัวแปรตัวนับ
<i>Condition</i>	คือเงื่อนไขสำหรับให้ลูปเกิดการ ทำงาน
<i>Increase_Counter</i> หรือ <i>Decrease_Counter</i>	คือ การเปลี่ยนแปลงค่าตัวนับ มีทั้งการเพิ่มค่าและลดค่า เช่น $a++$ หรือ $a--$ เป็นต้น



2.3. Loops

```
class PrintWord1
{ public static void main(String args[])
  {   int a;
      for(a = 0; a < 10; a++)
          System.out.println("Sawadee");
  }
}
```

```
java PrintWord1
Sawadee
...
Sawadee
```



2.3. Loops

```
class PrintWord2
{
    public static void main(String args[])
    {
        for(int a = 0; a < 10; a++)
        {
            System.out.print(a);
            System.out.println(" Sawadee");
        }
    }
}
```

```
java PrintWord2
0 Sawadee
...
9 Sawadee
```




2.3. Loops

```
class PrintWord4
{
    public static void main(String args[])
    {
        for(int a = 0; a < args[0].length(); a++)
            System.out.println(args[0]);
    }
}
```



2.3. Loops

รูปแบบ

```
while ( Condition )  
    Statement
```

เมื่อ

Condition คือเงื่อนไขในการทำงานของลูป เป็นเงื่อนไขที่ให้ผลลัพธ์เป็นค่าตรรกะ เมื่อเป็น true สเตจเมนต์ถูกทำงาน และสเตจเมนต์



2.3. Loops

รูปแบบ

```
Initial_Counter  
while ( Condition )  
{  
    Statement  
  
    Increase_Counter หรือ Decrease_Counter  
}
```

เมื่อ

Initial_Counter

คือการให้ค่ากับตัวแปรตัวนับเช่นเดียวกับ
การใช้ใน for



2.3. Loops

```
class FormularMultiple1
{ public static void main(String args[])
  { int a = 1; // Initial Counter
    while (a <= 12) // Counter Check
    {      System.out.print( a + " x " + args[0] + " = ");
          System.out.println(Integer.parseInt(args[0]) * a);
          a++;      // Increase Counter
    }
  }
}
```

```
java FormularMultiple1 5
```

```
1 x 5 = 5
```

```
2 x 5 = 10
```

```
3 x 5 = 15
```

```
...
```

```
11 x 5 = 55
```

```
12 x 5 = 60
```



2.3. Loops

```
class FormularMultiple2
{ public static void main(String args[])
  { int a = 1; // Initial Counter
    while (a <= 12)
    {      System.out.print( a + " x " + args[0] + " = ");
          System.out.println(Integer.parseInt(args[0]) * a++);
    }
  }
}
```

```
java FormularMultiple2 3
```

```
1 x 3 = 3
```

```
2 x 3 = 6
```

```
3 x 3 = 9
```

```
...
```

```
11 x 3 = 33
```

```
12 x 3 = 36
```



2.3. Loops

```
class Factorial1
{
    public static void main(String args[])
    {
        int fac = 1, val = Integer.parseInt(args[0]);
        while (val >0)
            fac *= val--;
        System.out.println(fac);
    }
}
```

```
java Factorial1 5
120
```



2.3. Loops

รูปแบบ

```
do  
    Statements  
while ( Condition );
```

เมื่อ

Condition คือเงื่อนไขในการทำงานของลูป เป็นเงื่อนไขที่ให้ผลลัพธ์เป็นค่าตรรกะ เมื่อเป็น true ลูปถูกทำงาน และเป็นลูปหยุดทำงานเมื่อมีค่าเป็น false



2.3. Loops

ข้อเปรียบเทียบระหว่างลูป while ... และ do ... while

ลูป	ตำแหน่งเงื่อนไข	เงื่อนไขก่อนเข้าลูป	จำนวนครั้งต่ำสุด
		True	False
while	ส่วนบน	ทำงาน	ไม่ทำงาน 0
do...while	ส่วนล่าง	ทำงาน	ทำงาน 1



2.3. Loops

```
class DoWhile
{ public static void main(String args[])
  { do
    System.out.println("Don't forget me");
    while (5>10);
  }
}
```

```
java DoWhile
Don't forget me
```



2.3. Loops

```
class BarChart
{   public static void main(String args[])
    {       for (int a = 0; a < arg.length; a++)
            {           for(int b=0; b<Integer.parseInt(args[a]); b++)
                    System.out.print("*");
                    System.out.print("\n");
            }
    }
}
```

```
java BarChart 3 7 5
```

```
***
```

```
*****
```

```
*****
```



2.3. Loops

```
class InfinityLoop
{
    public static void main(String args[])
    {
        for (int a = 1; a > 0; a++)
            System.out.print("I'll be Back\t");
    }
}
```

java InfinityLoop

I'll be Back I'll be Back I'll be Back I'll be Back ...



2.3. Loops

Infinity Loop

แบบที่ 1

```
for ( ; ; ) ;
```

แบบที่ 2

```
while (true) ;
```

แบบที่ 3

```
do { } while (true);
```



2.4. Jumps

รูปแบบ

```
break Label_Name
```

เมื่อ

Label_Name คือชื่อลาเบลที่ถูกตั้งไว้เพื่อให้การทำงานหลุดออกจากสโคปไปยังตำแหน่งลาเบล แต่ถ้าไม่ระบุหมายถึงเป็นการหลุดจากสโคป และไปทำงานคำสั่งที่ต่อท้ายจากสโคป



2.4. Jumps

```
class BreakScope1
{ public static void main(String args[])
  {   for(int a=0; a<10; a++)
      {   System.out.println("Line 1");
          break;
          System.out.println("Line 2");
      }
      System.out.println("Line 3");
  }
}
```



2.4. Jumps

```
class BreakScope2
{   public static void main(String args[])
    {       Comehere: for(int a=0; a<10; a++)
        {           System.out.println("Line 1");
            for(int a=0; a<10; a++)
                {           System.out.println("Line 2");
                    break Comehere;
                }
            }
        System.out.println("Line 3");
    }
}
```



2.4. Jumps

รูปแบบ

continue *label*

เมื่อ

label คือชื่อลาเบลที่ถูกตั้งไว้เพื่อให้การทำงานหลุดออกจะสโคปไปยังตำแหน่งลาเบล แต่ถ้าไม่ระบุหมายถึงเป็นการหลุดจากสโคป และไปทำงานคำสั่งที่เริ่มต้นจากสโคป แต่ลาเบลต้องอยู่ในสโคปเดียวกับคำสั่ง `continue` เสมอ



2.4. Jumps

```
class OddNumber
{
    public static void main(String args[])
    {
        for (int a=1; a<=100; a+=1)
        {
            if ((a%2)==0)
                continue;
            System.out.println(a + " is Odd Number");
        }
    }
}
```

```
java OddNumber
1 is Odd Number
...
99 is Odd Number
```



2.4. Jumps

```
class BreakScope3
{ public static void main(String args[])
  {   Comehere: for(int a=0; a<10; a++)
      {   System.out.println("Line 1");
          for(int a=0; a<10; a++)
              {   System.out.println("Line 2");
                  continue Comehere;
              }
          }
      System.out.println("Line 3");
  }
}
```



2.4. Jumps

รูปแบบ

```
return Expression
```

เมื่อ

Expression คือข้อมูลที่ต้องการคืนกลับหลังจากการทำงานของเมธอด

- ใช้สำหรับการหยุดการทำงานของเมธอด
- เมื่อเมธอดเป็นแบบคืนค่าได้ คีย์เวิร์ด `return` ตามด้วยค่าที่ต้องการส่งกลับ