

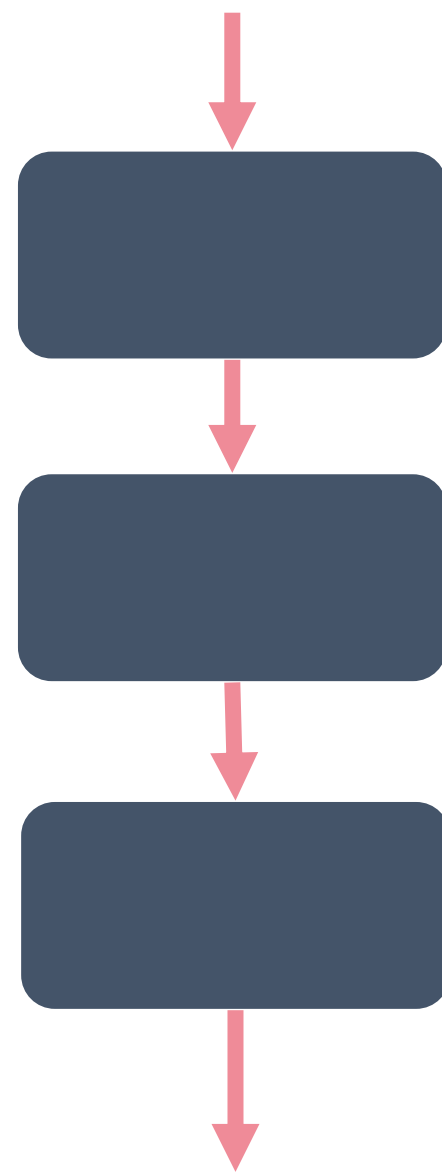
# บทที่ 2 คำสั่งควบคุม และโครงสร้างแบบซ้อน

บรรยายโดย ผศ.ดร.ธราวิเชษฐ์ ธิติจรูญโรจน์

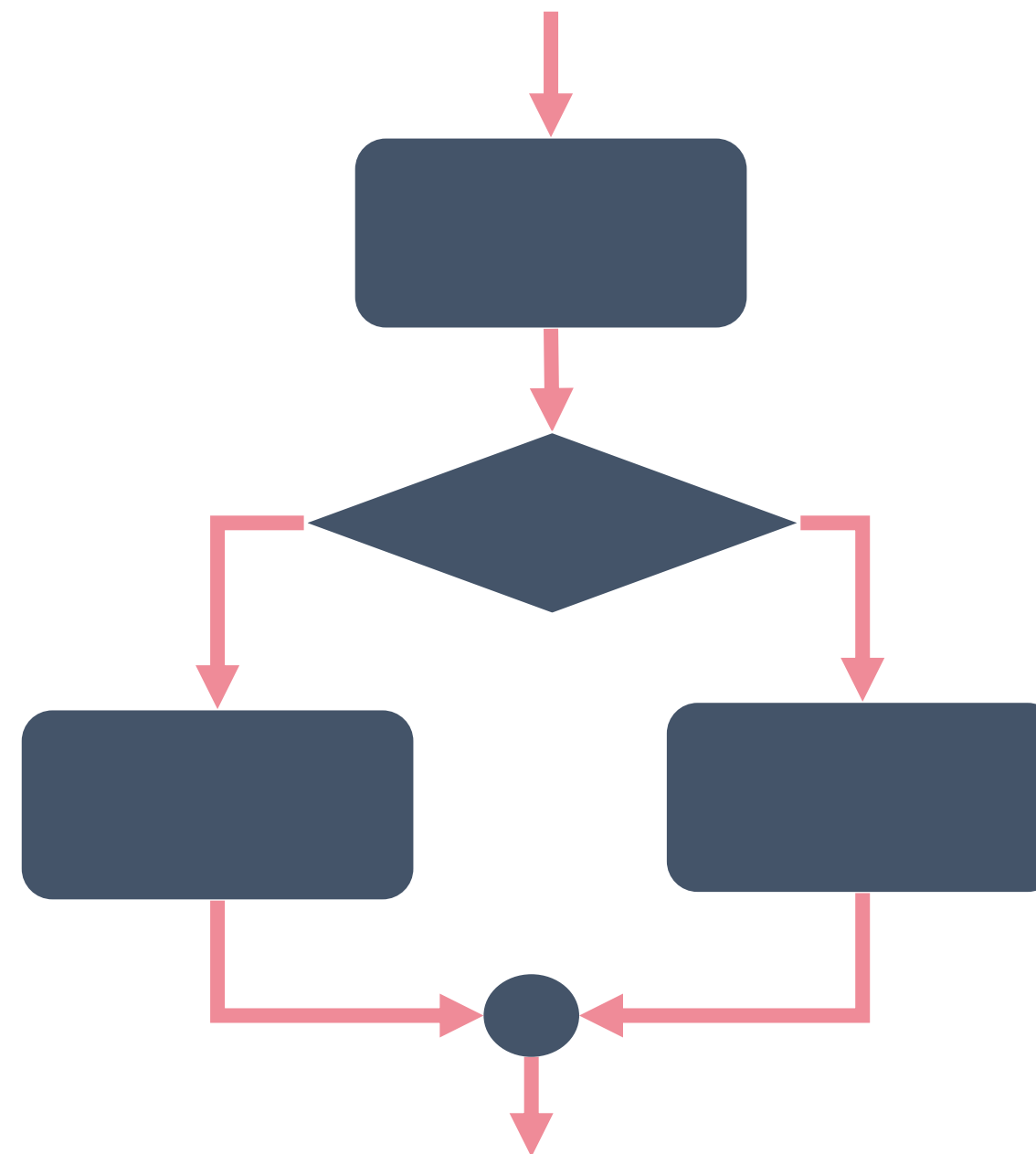
คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

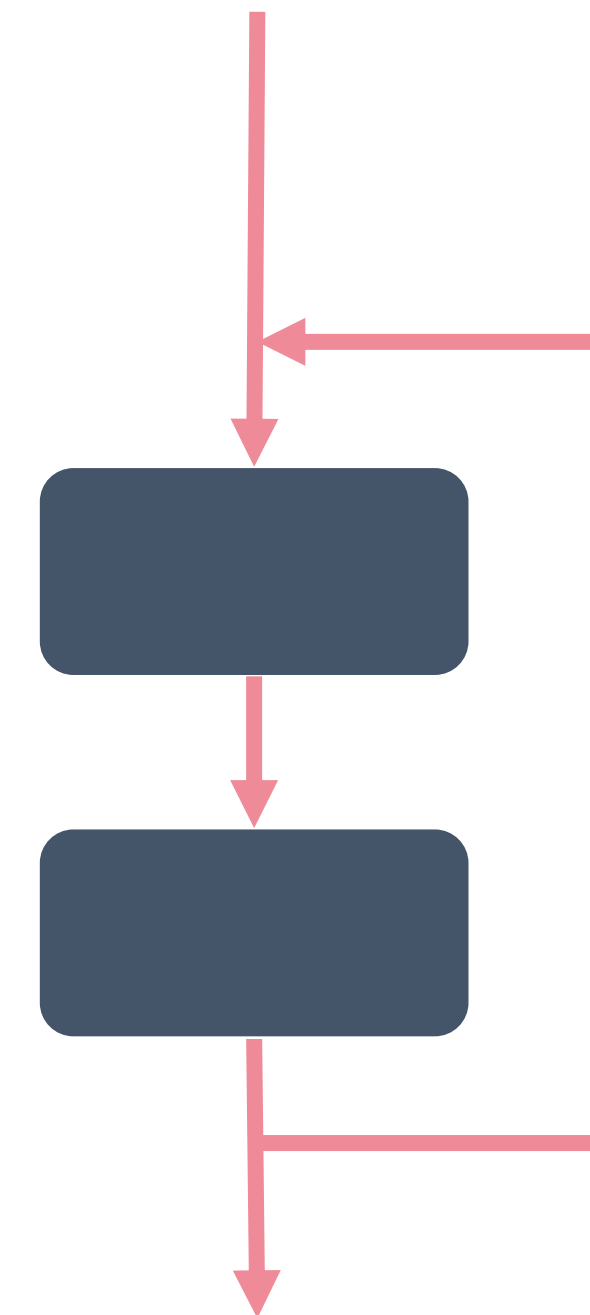
# โครงสร้างการเขียนโปรแกรม



การทำงานแบบเรียงลำดับ  
(Sequence)

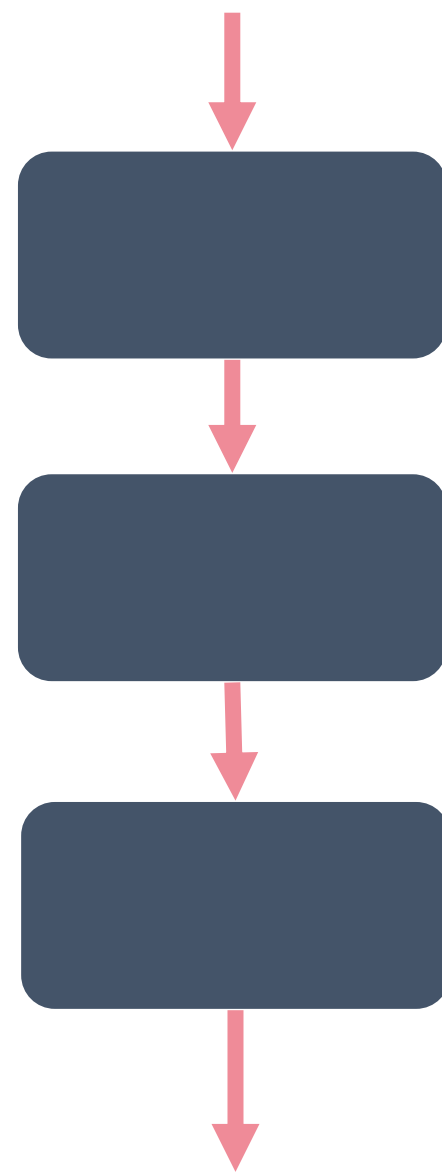


การทำงานแบบมีเงื่อนไข  
(Selection)

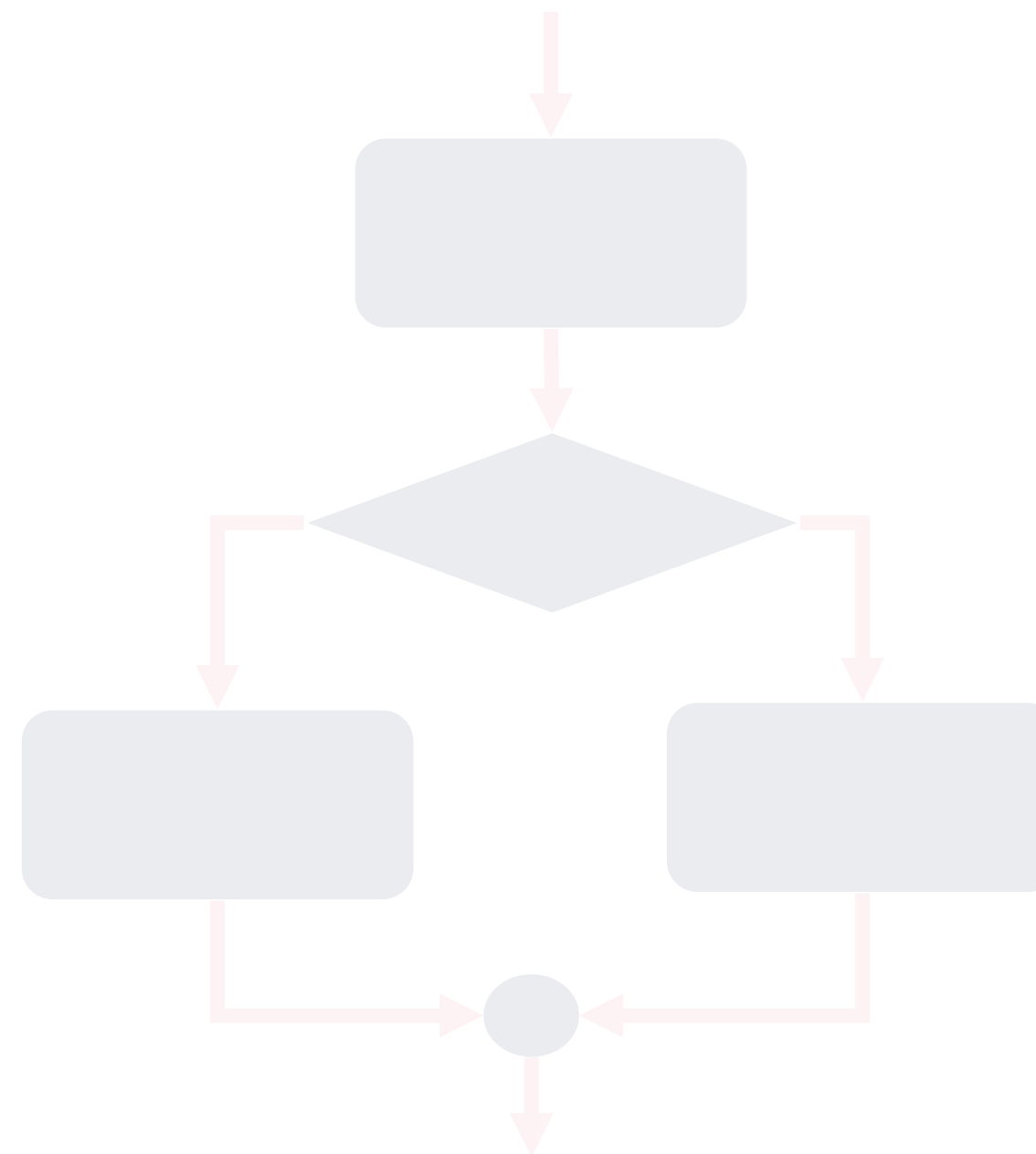


การทำงานแบบทำซ้ำ  
(Iteration)

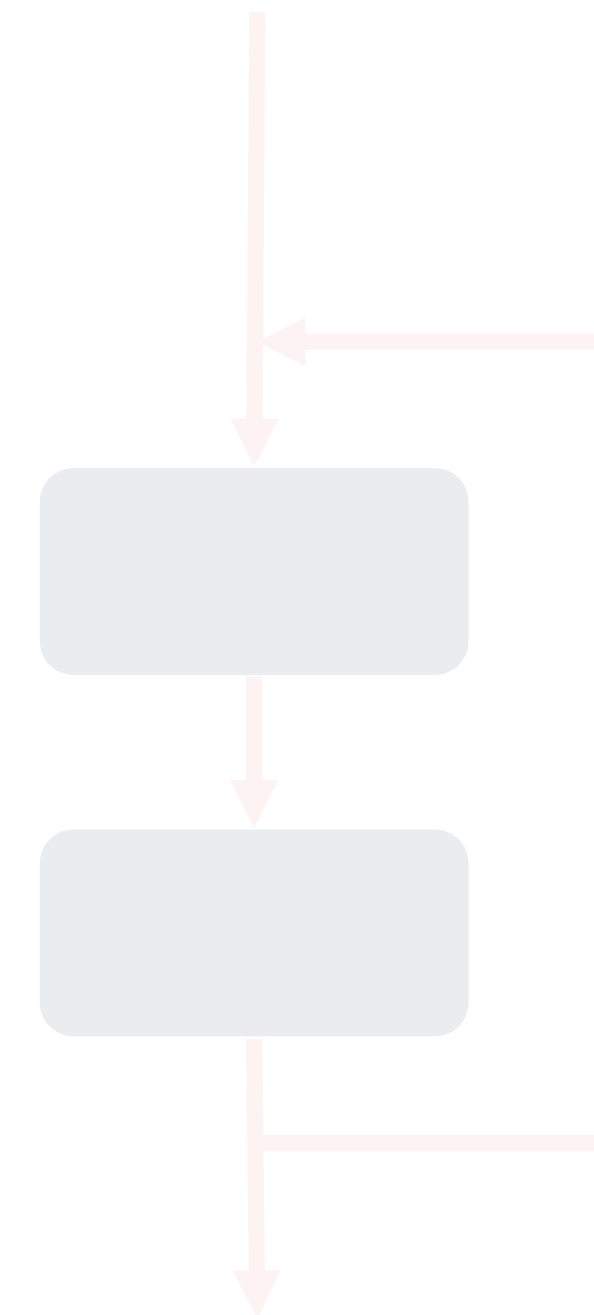
# โครงสร้างการเขียนโปรแกรม



การทำงานแบบเรียงลำดับ  
(Sequence)

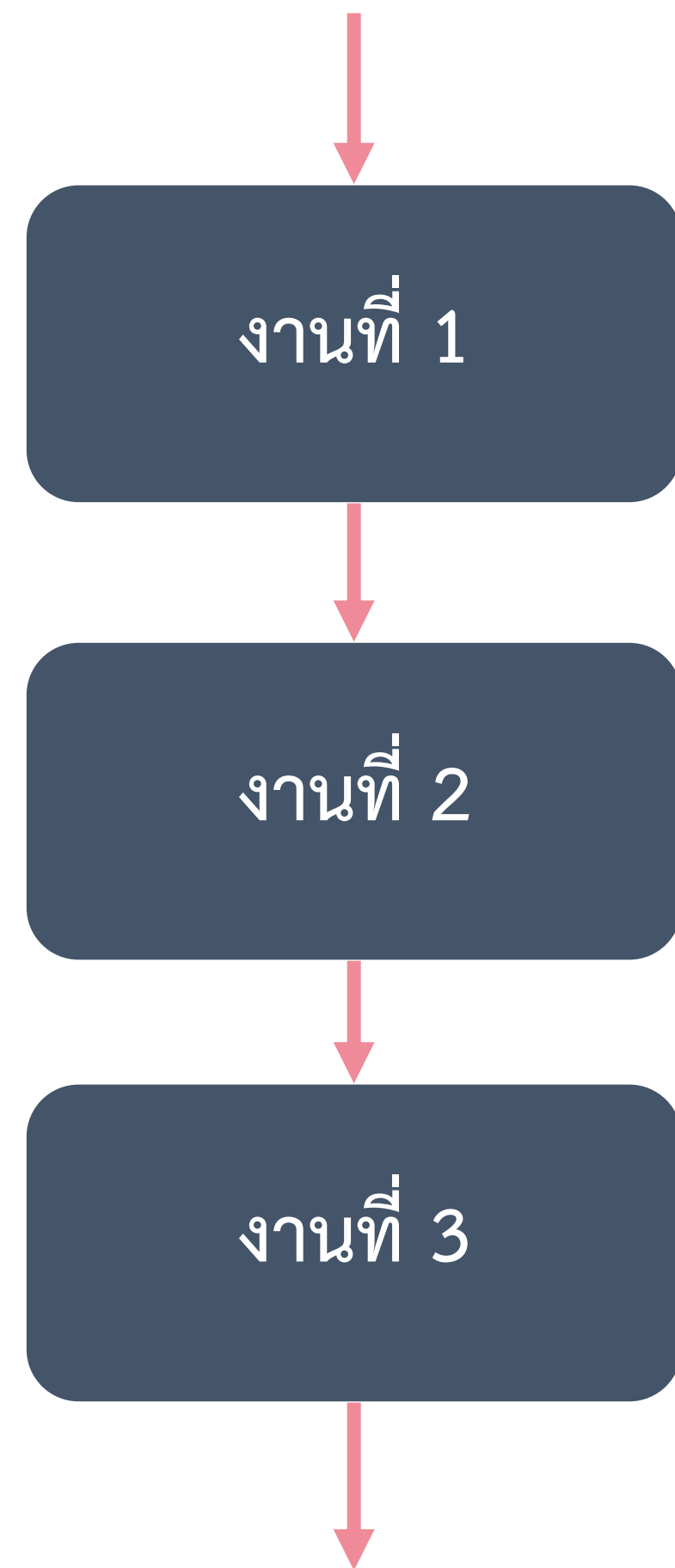


การทำงานแบบมีเงื่อนไข  
(Selection)



การทำงานแบบทำซ้ำ  
(Iteration)

# การทำงานแบบเรียงลำดับ (Sequence)



input  
จงเขียนโปรแกรมรับค่าปี พ.ศ. จากผู้ใช้งาน จากนั้นโปรแกรมจะแปลง  
How to output  
จากปี พ.ศ. ไปเป็นปี ค.ศ. และแสดงผลทางจอภาพ

รับค่าปีพ.ศ.

Scanner

แปลงพ.ศ. → ค.ศ.

แสดง ค.ศ.

System.out.println

# การทำงานแบบเรียงลำดับ (Sequence)



จงเขียนโปรแกรมรับค่าปี พ.ศ. จากผู้ใช้งาน จากนั้นโปรแกรมจะแปลง  
จากปี พ.ศ. ไปเป็นปี ค.ศ. และแสดงผลทางจอภาพ

# การทำงานแบบเรียงลำดับ (Sequence)

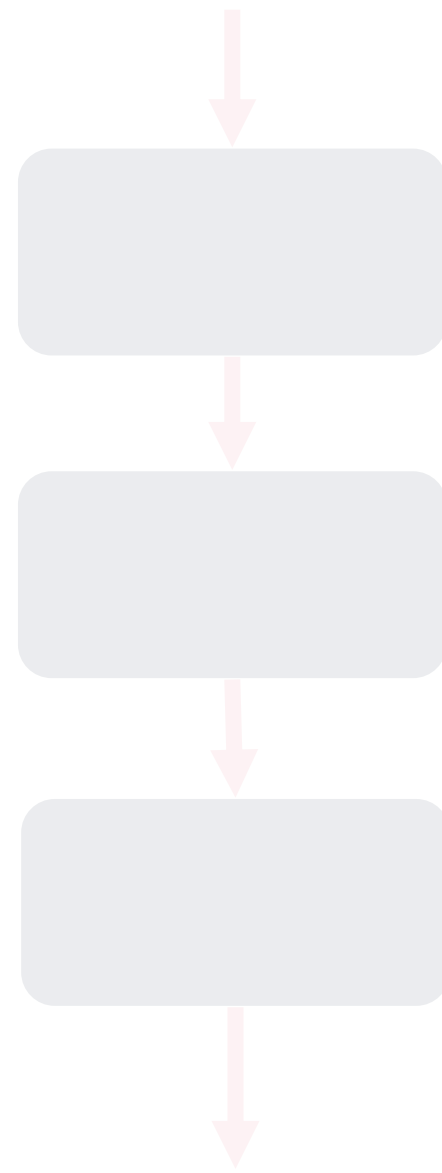


```
import java.util.*;
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int year1 = sc.nextInt();

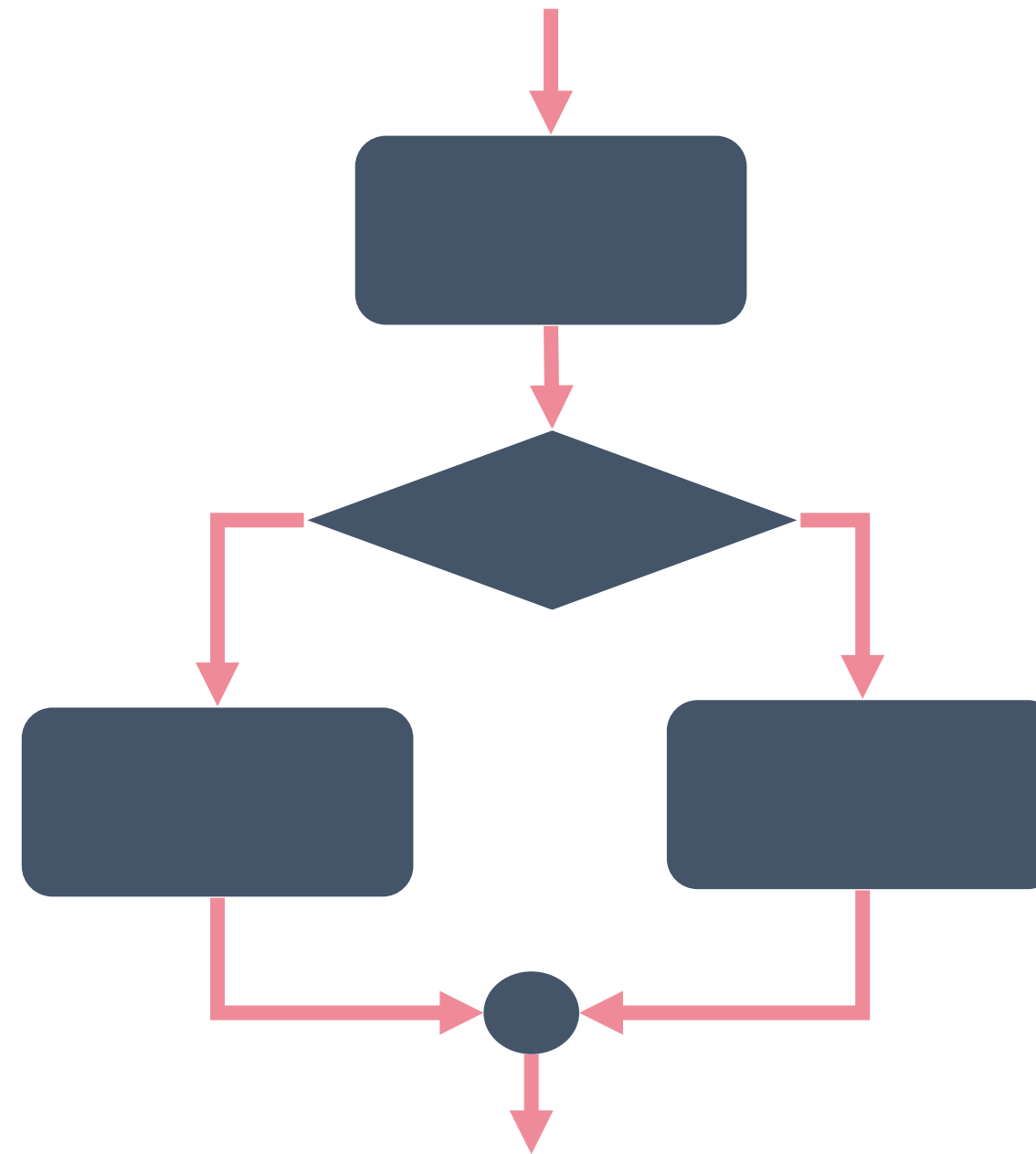
        int year2 = year1 - 543;

        System.out.println(year2);
    }
}
```

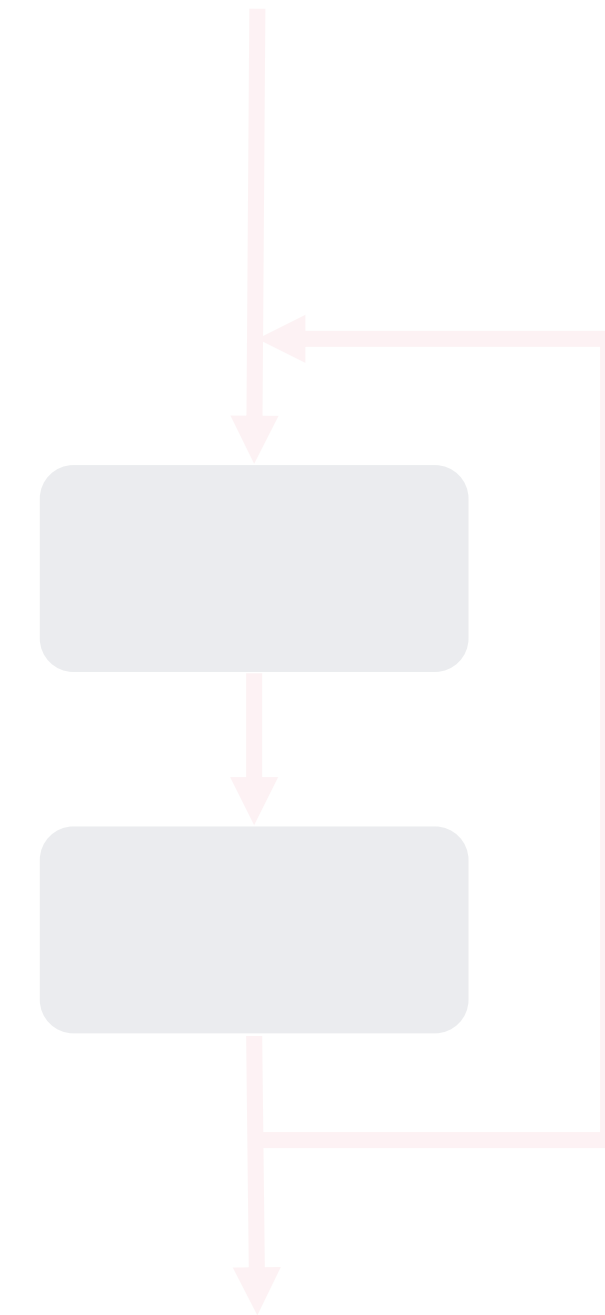
# โครงสร้างการเขียนโปรแกรม



การทำงานแบบเรียงลำดับ  
(Sequence)

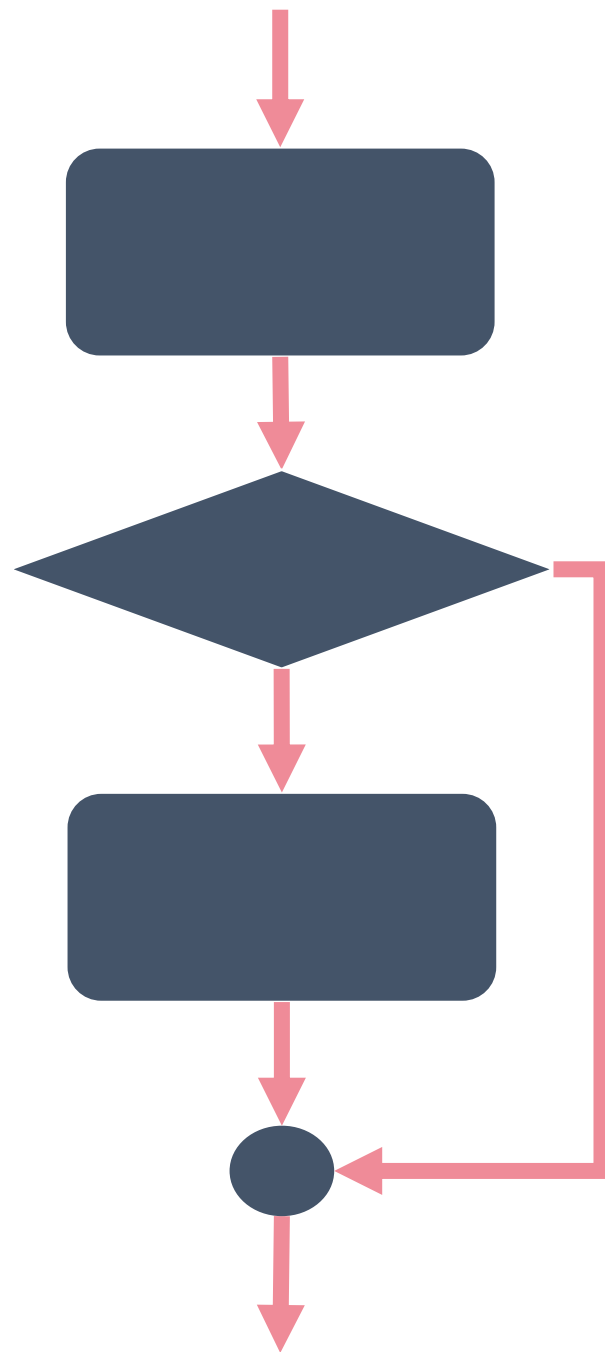


การทำงานแบบมีเงื่อนไข  
(Selection)

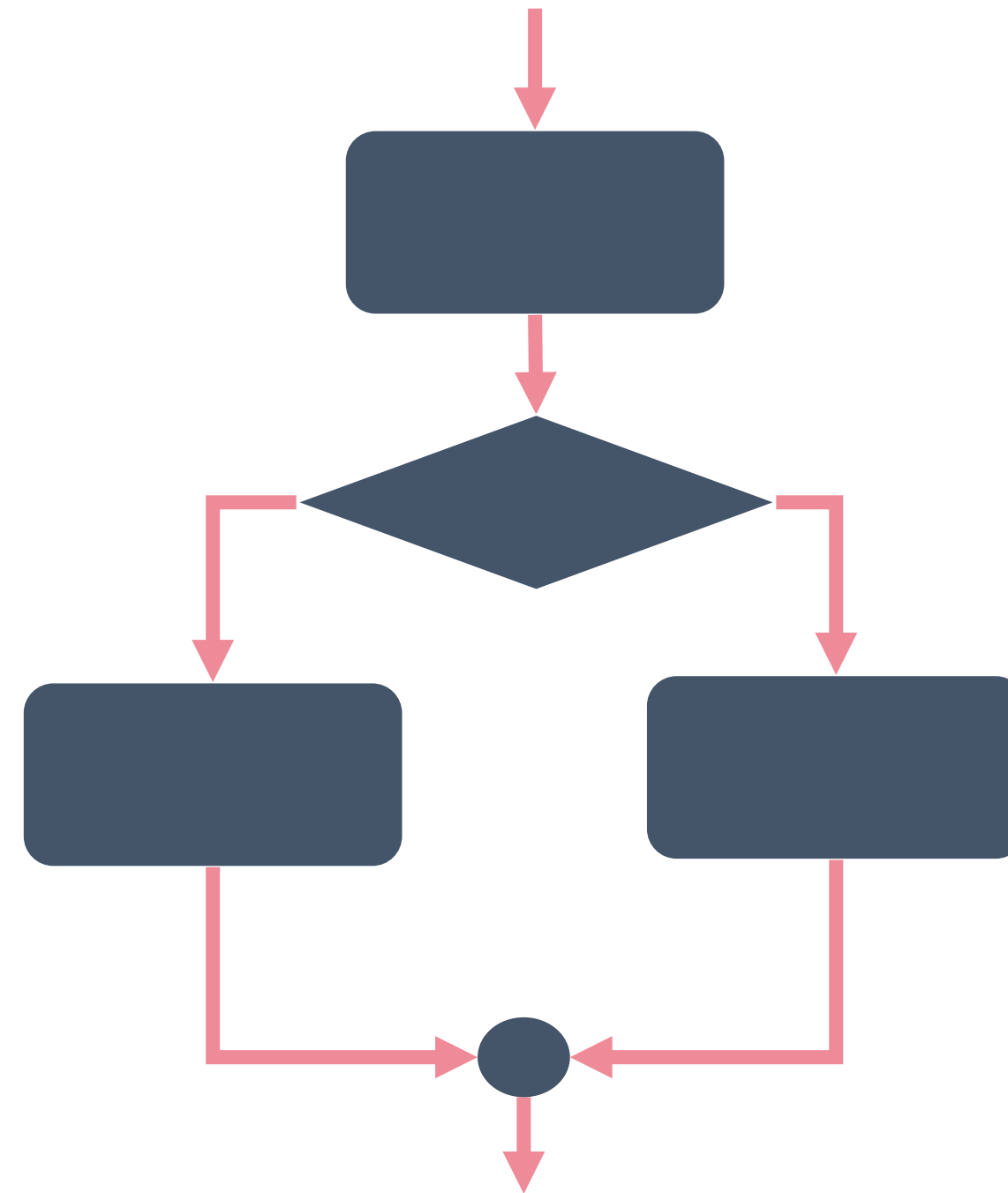


การทำงานแบบทำซ้ำ  
(Iteration)

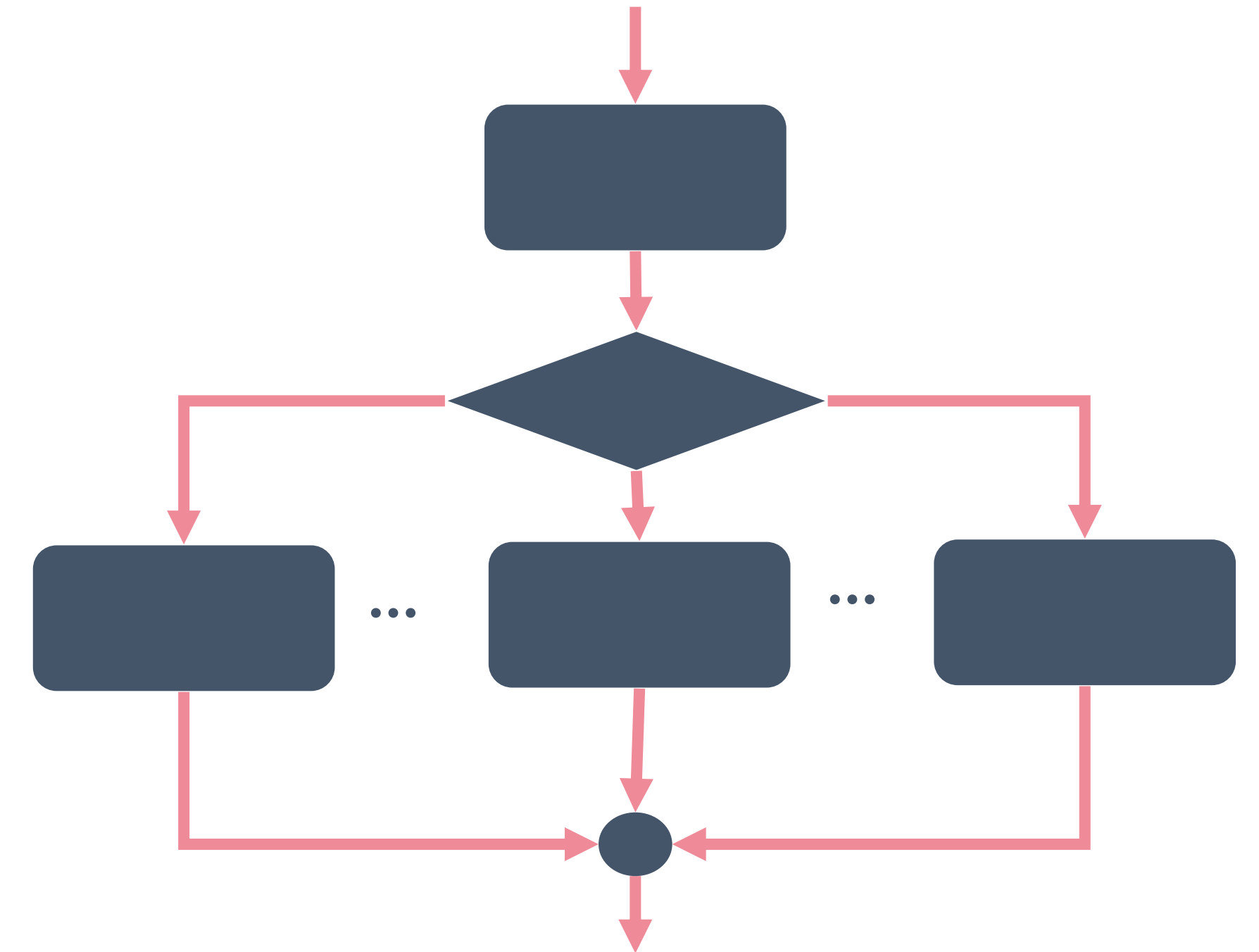
# การทำงานแบบมีเงื่อนไข (Selection)



(Single Selection)



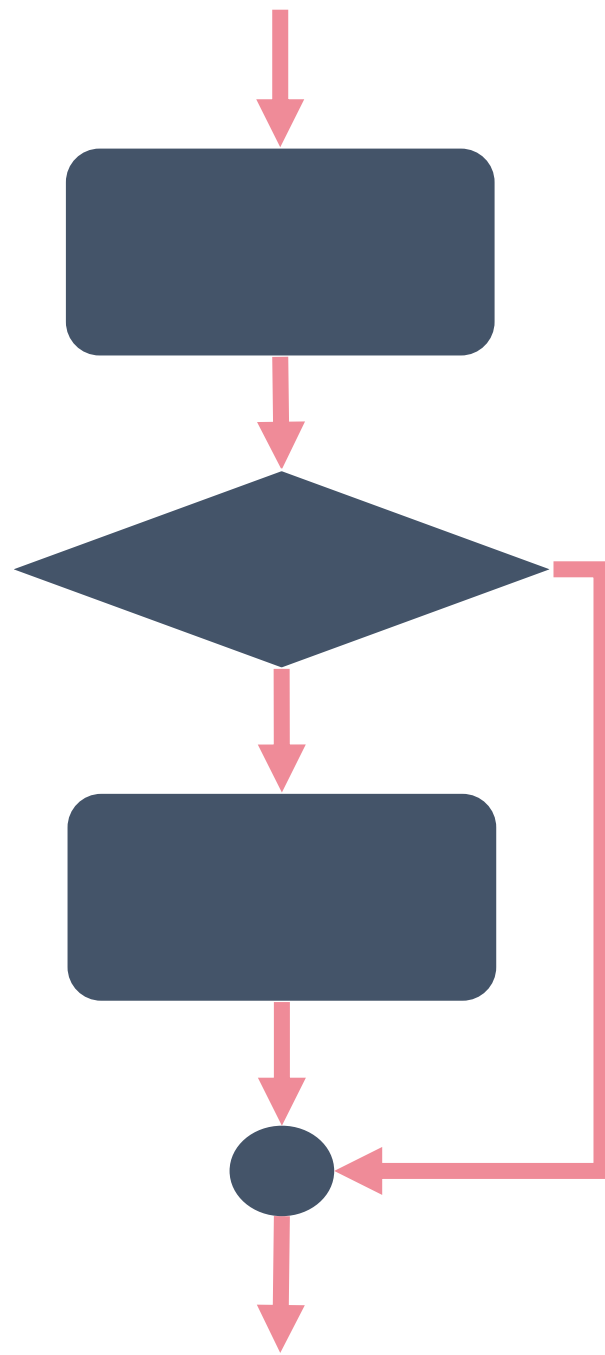
(Double Selection)



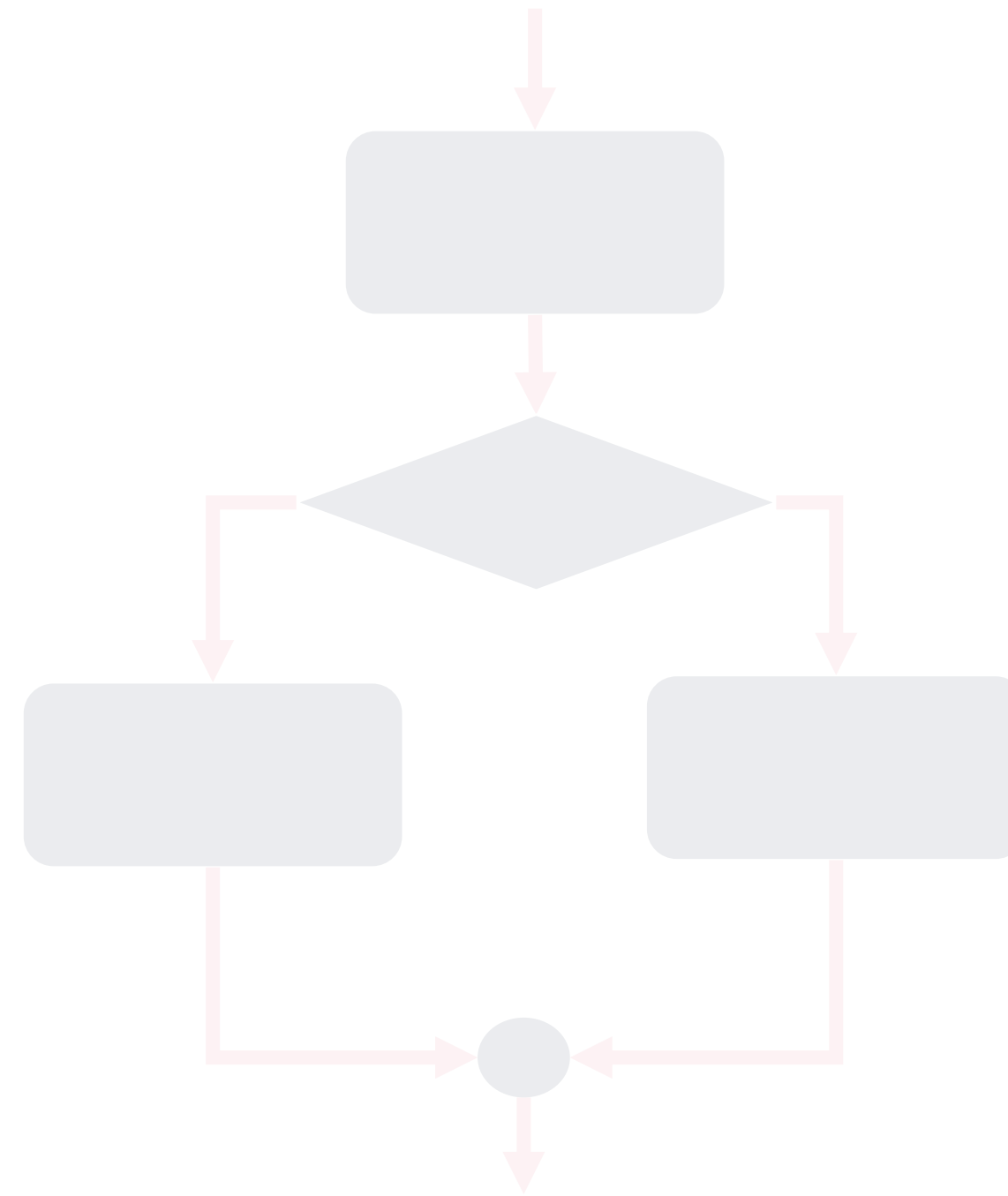
(Multiple Selection)



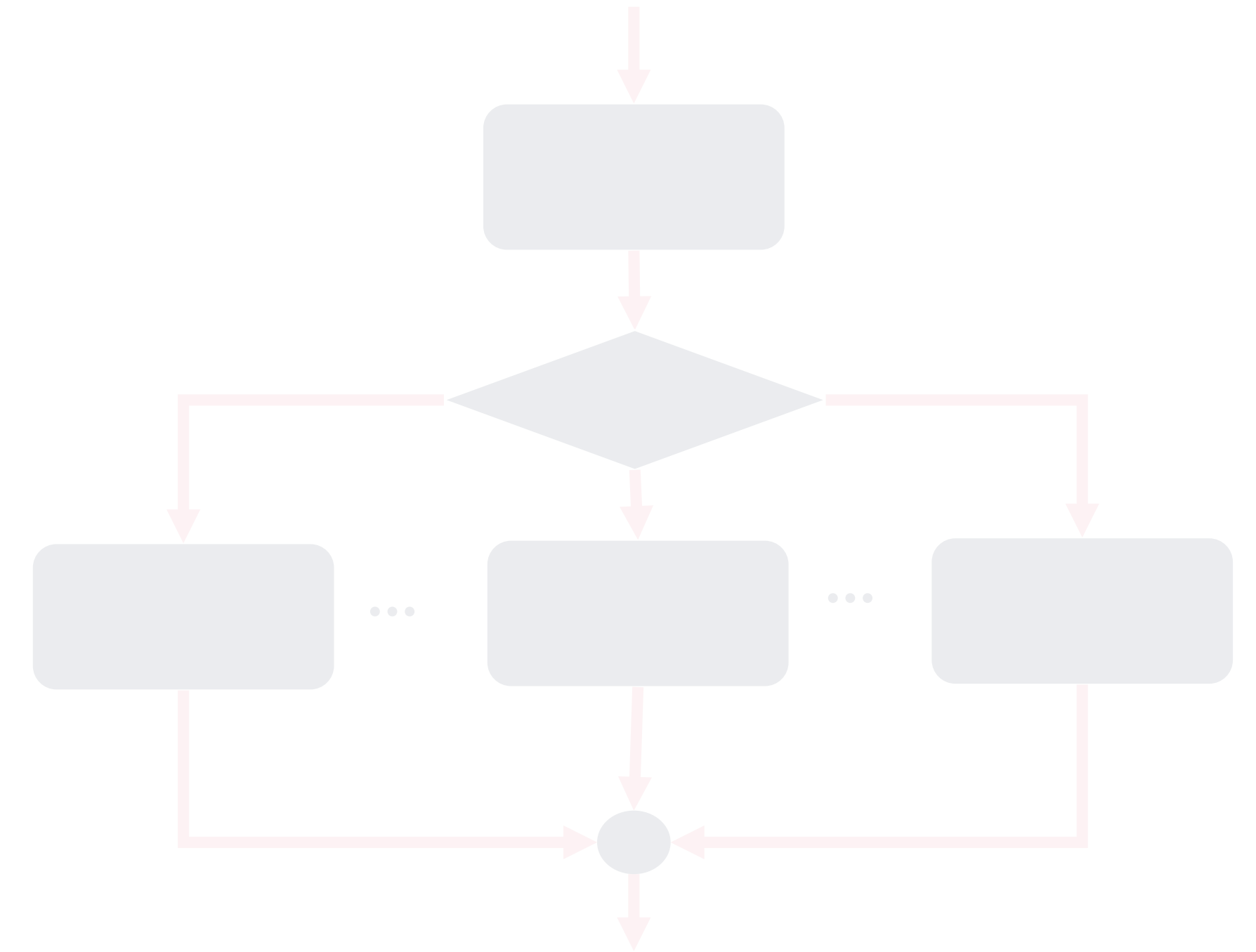
# การทำงานแบบมีเงื่อนไข (Selection)



(Single Selection)



(Double Selection)



(Multiple Selection)

# การทำงานแบบมีเงื่อนไข (Selection)

output

ให้นักศึกษาเขียนโปรแกรมเพื่อคำนวณราคาสินค้าให้กับลูกค้า โดยที่ลูกค้าที่มีรหัสสมาชิกเป็น 105

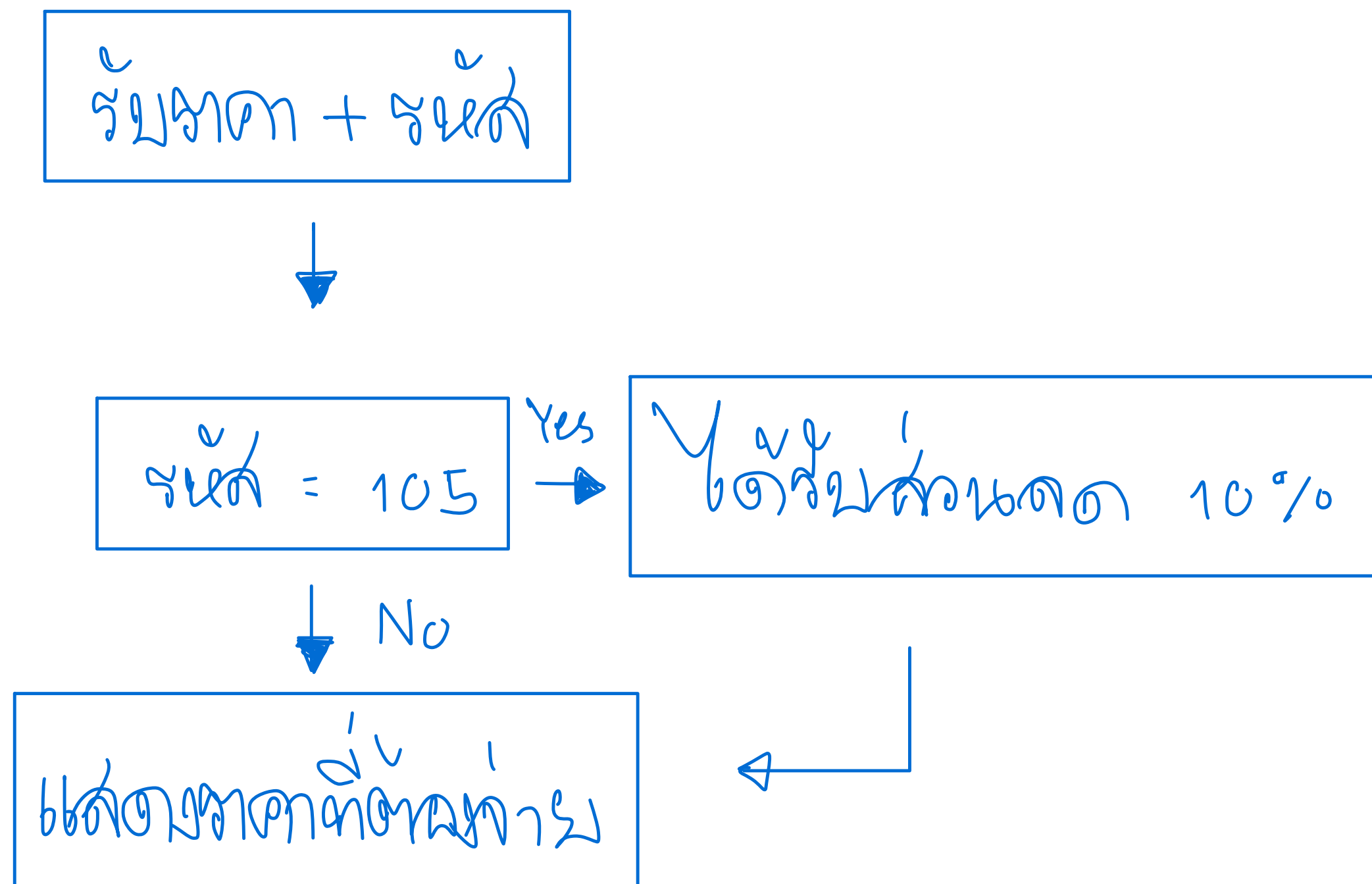
option

จะได้รับส่วนลดราคาสินค้าลง 10 % แต่ถ้าลูกค้าคนอื่น ๆ จะคิดราคาเท่าเดิม โดยจะรับค่ารหัส

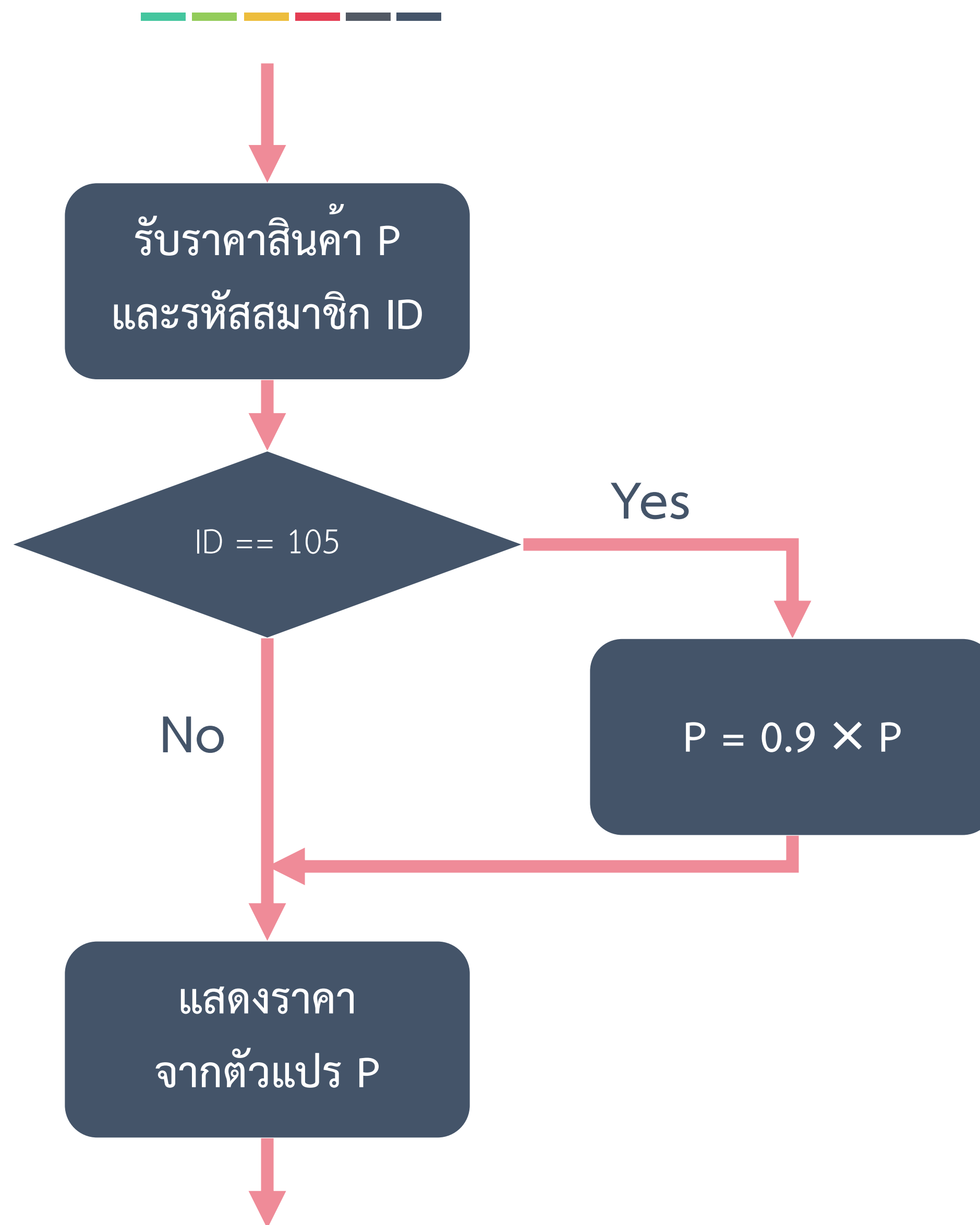
input

input

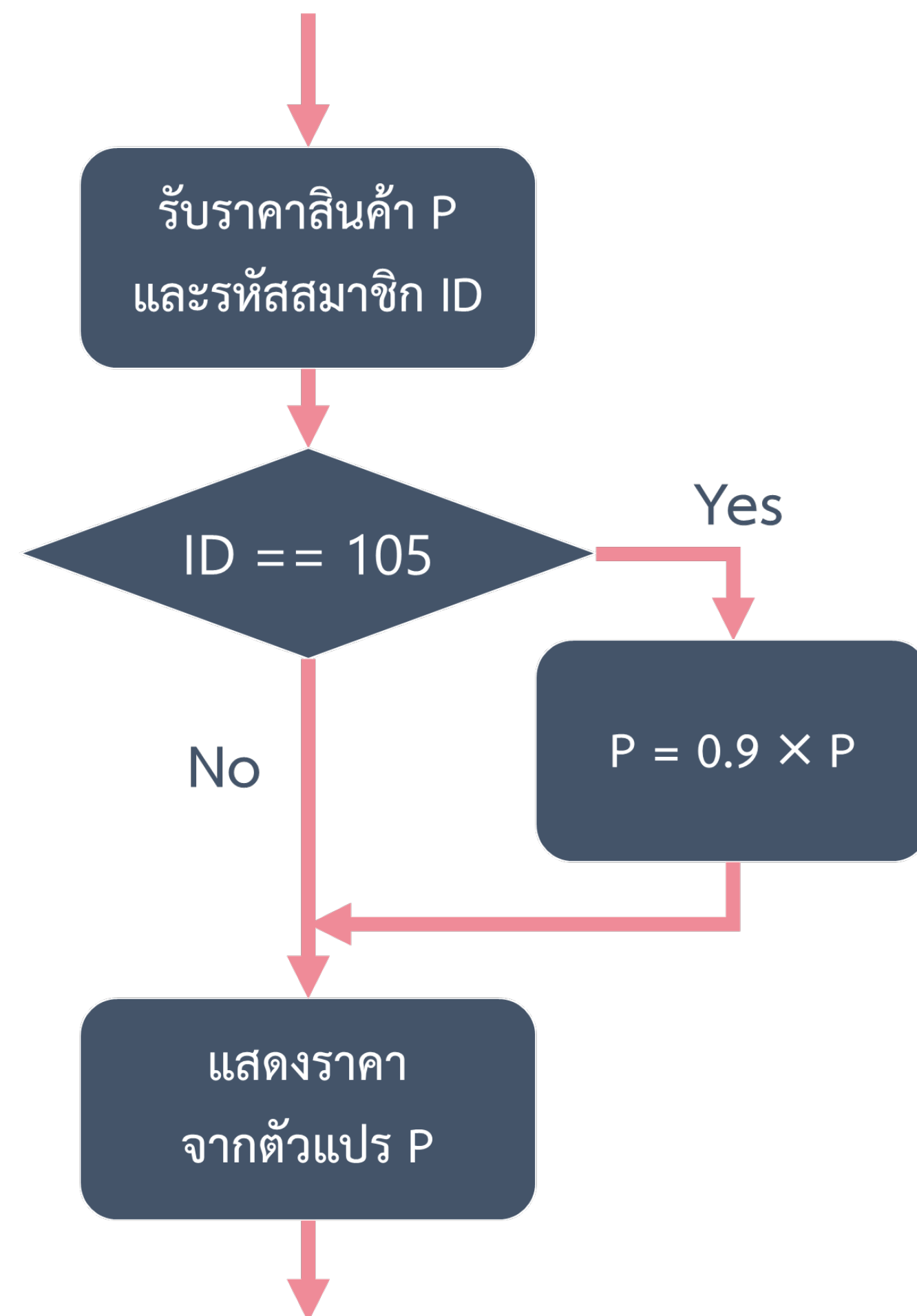
ลูกค้าและราคาสินค้าผ่านทางคีย์บอร์ด



# การทำงานแบบมีเงื่อนไข (Selection)



# การทำงานแบบมีเงื่อนไข (Selection)



รูปแบบ

```
if (เงื่อนไข)
{
    ประโยคที่จะทำงานเมื่อเงื่อนไขเป็นจริง;
}
```

ตัวอย่าง

```
if (ID == 105) {
    price = 0.9 * price;
}
System.out.println(price);
```

# ตัวอย่างโปรแกรมคำนวณส่วนลด



ให้นักศึกษาเขียนโปรแกรมเพื่อคำนวณราคาสินค้า  
ให้กับลูกค้าที่มีรหัสสมาชิกเป็น 105 ซึ่งจะลดราคา  
สินค้าลงจากเดิม 10 % แต่ถ้าลูกค้าคนอื่น ๆ จะคิด  
ราคาเท่าเดิม โดยจะรับค่ารหัสลูกค้าและราคาสินค้า  
ผ่านทางคีย์บอร์ด

```
import java.util.*;
public class MyShop01 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int ID = sc.nextInt();
        int price = sc.nextInt();

        if (ID == 105) {
            price = 0.9 * price;
        }
        System.out.println(price);
    }
}
```

# เงื่อนไข คืออะไร

`if (` **เงื่อนไข** `)`



**Boolean**

- [1] ค่าคงที่
- [2] ตัวแปร (Variable)
- [3] ตัวดำเนินการ (Operation)
- [4] เมธอด (Method)



# การทำงานแบบมีเงื่อนไข (Selection)

ค่าคงที่

**if** ( เงื่อนไข )



**Boolean**

[1] ค่าคงที่

[2] ตัวแปร (Variable)

[3] ตัวดำเนินการ (Operation)

[4] เมธอด (Method)

```
if ( true ) {  
    // command  
}  
  
// ===== Break Line =====  
  
if ( false ) {  
    // command  
}
```

# การทำงานแบบมีเงื่อนไข (Selection)

ตัวแปร (Variable)

**if** ( เงื่อนไข )



**Boolean**

[1] ค่าคงที่

[2] ตัวแปร (Variable)

[3] ตัวดำเนินการ (Operation)

[4] เมธอด (Method)

```
boolean isXXX = true;
if ( isXXX ) {
    // command
}

// ===== Break Line =====

boolean isPositive = (A > 0);
if ( isPositive ) {
    // command
}
```



# การทำงานแบบมีเงื่อนไข (Selection)



```
public class Main {  
    public static void main(String[] args) {  
        int correctedPassword = 12345;  
        int givenPassword = 12345;  
        boolean isMatch;  
  
        isMatch = (correctedPassword == givenPassword);  
  
        // The "if (isMatch)" flag not that useful.  
        // It is more appropriate than "if (isMatch == true)".  
  
        if (isMatch) {  
            System.out.println("Match");  
        }  
    }  
}
```

# การทำงานแบบมีเงื่อนไข (Selection)

## ตัวดำเนินการ (Operation)

**if** ( เงื่อนไข )



**Boolean**

- [1] ค่าคงที่
- [2] ตัวแปร (Variable)
- [3] ตัวดำเนินการ (Operation)
- [4] เมธอด (Method)

```
int age = 10;  
if ( age > 5 ) {  
    // command  
}  
  
// ===== Break Line =====  
  
int salary = 20000;  
if ( (salary > 0) && (salary <= 30000) ) {  
    // command  
}
```

# ตัวดำเนินการในภาษาจาวา

- เปรียบเทียบ (Comparison Operator)

< > <= >= == !=

- ตรรกศาสตร์ (Logical Operator)

&& || & | !

# ตัวดำเนินการเชิงเปรียบเทียบ

เครื่องหมาย	ความหมาย	ตัวอย่าง	ผลลัพธ์
<	น้อยกว่า	$3 < 4$	true
<=	น้อยกว่าหรือเท่ากับ	$3 \leq 4$	true
>	มากกว่า	$3 > 4$	false
>=	มากกว่าหรือเท่ากับ	$3 \geq 4$	false
==	เท่ากับ	$3 == 4$	false
!=	ไม่เท่ากับ	$3 != 4$	true

# ตัวอย่าง

```
public class Main {  
    public static void main(String[] args) {  
        int x = 5;  
        int y = 4;  
  
        boolean b1;  
        b1 = (x!=y);  
  
        System.out.println("x not equal to y is "+b1);  
        System.out.println("y less than 0 is "+(y<0));  
  
    }  
}
```

# ตัวดำเนินการทางตรรกศาสตร์



เครื่องหมาย

ความหมาย

!

กลับค่าทางตรรกะ

&& หรือ &

AND ค่าทางตรรกะ

|| หรือ |

OR ค่าทางตรรกะ

^

Exclusive-OR ค่าทางตรรกะ

อนุญาตให้ทำ  
 short circuit



ไม่อนุญาตให้ทำ short circuit

# ตัวดำเนินการทางตรรกศาสตร์

input 1

input 2

1 and 2

1 or 2

1 xor 2

not 1

P	Q	P && Q	P    Q	P ^ Q	! P
true	true	true	<div> true  short circuit </div>	false	false
true	false	false	true	true	false
false	true	<div> false  short circuit </div>	true	true	true
false	false	false	false	false	true

★ short circuit = เมื่อการรันถึงค่าที่มันสามารถหาค่าได้แล้วมันจะหยุด (ตัวหลังจะไม่ถูก execute)

# ตัวอย่างโปรแกรม

```
public class Ex02 {  
    public static void main(String args[]) {  
        int numA = 10;  
        boolean numB = ((10numA < 11) | (11++numA >= 11)); true or true  
        System.out.println(numA); 11  
        System.out.println(numB); true  
    }  
}
```

11

true



# ตัวอย่างโปรแกรม

```
public class Ex03 {  
    public static void main(String args[]) {  
        int numA = 10;      10      short circuit  
        boolean numB = ((numA < 11) || (++numA >= 11));  
        System.out.println(numA); 10  
        System.out.println(numB); true  
    }  
}
```

↳ not execute

10

true

# ตัวอย่างโปรแกรม

```
public class Ex04 {  
    public static void main(String args[]) {  
        int numA = 10;  
        boolean numB = ((numA > 11) & (++numA >= 11));  
        System.out.println(numA);  
        System.out.println(numB);  
    }  
}
```

```
11  
false
```

# ตัวอย่างโปรแกรม

```
public class Ex05 {  
    public static void main(String args[]) {  
        int numA = 10;  
        boolean numB = ((numA > 11) && (++numA >= 11));  
        System.out.println(numA);  
        System.out.println(numB);  
    }  
}
```

```
10  
false
```

# ตัวอย่างการใช้ตัวดำเนินการทางตรรกศาสตร์

นิพจน์

ผลลัพธ์

`(7>6) && (2<1)`

**false**

`(7>6) || (2<1)`

**true**

`! (7>6)`

**false**

# การทำงานแบบมีเงื่อนไข (Selection)

## ตัวดำเนินการ (Operation)

**if ( เงื่อนไข )**



**Boolean**

- [1] ค่าคงที่
- [2] ตัวแปร (Variable)
- [3] ตัวดำเนินการ (Operation)
- [4] เมธอด (Method)

```
int age = 10;  
if ( age > 5 ) {  
    // command  
}  
  
// ===== Break Line =====  
  
int salary = 20000;  
if ( (salary > 0) && (salary <= 30000) ) {  
    // command  
}
```

< > <= >= == !=

# การทำงานแบบมีเงื่อนไข (Selection)

## เมธอด (Method)

**if** ( เงื่อนไข )

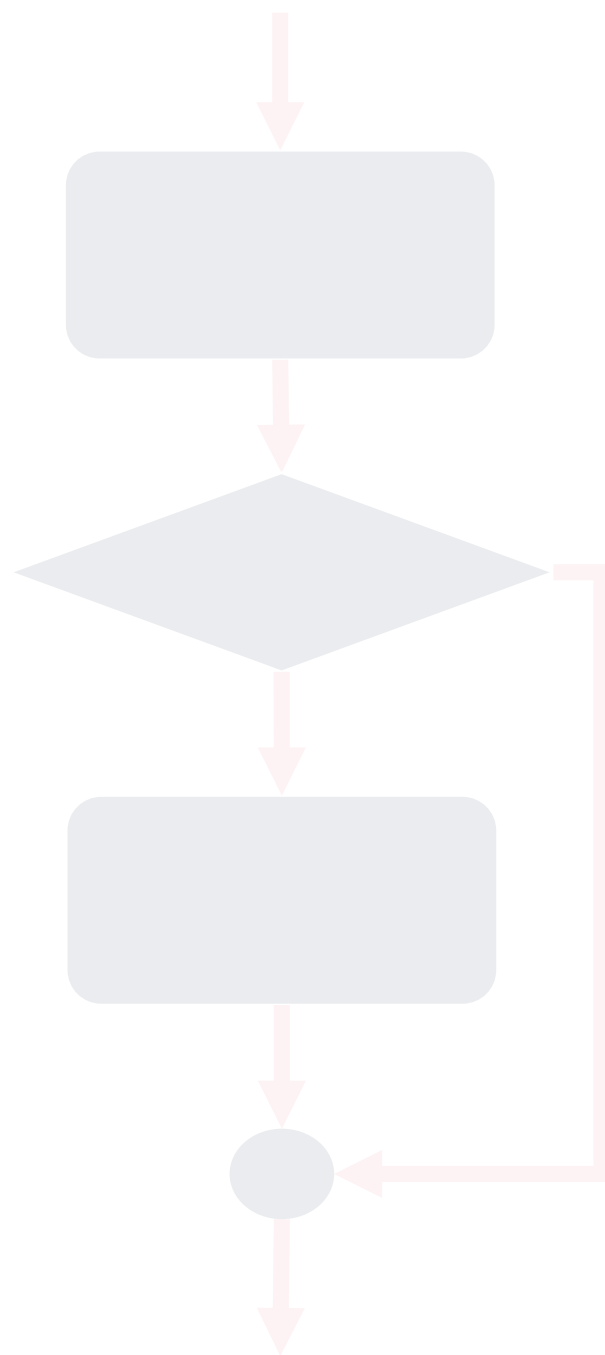


**Boolean**

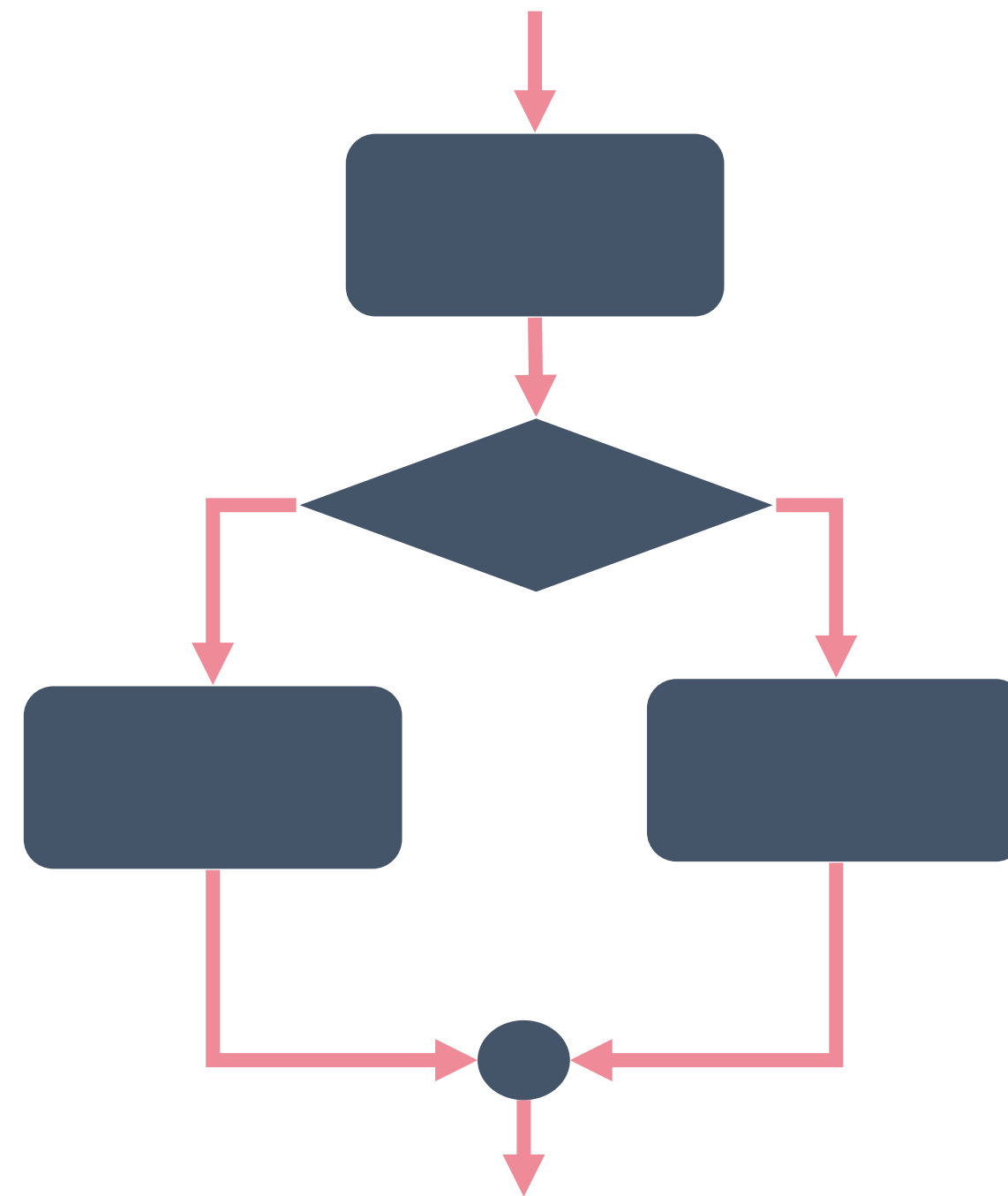
- [1] ค่าคงที่
- [2] ตัวแปร (Variable)
- [3] ตัวดำเนินการ (Operation)
- [4] เมธอด (Method)

```
public static boolean isPositive(int num) {  
    if (num > 0)  
        return true;  
    else  
        return false;  
}  
public static void main(String[] args) {  
    int num = 7;  
    if (isPositive(num)) {  
        // command  
    }  
}
```

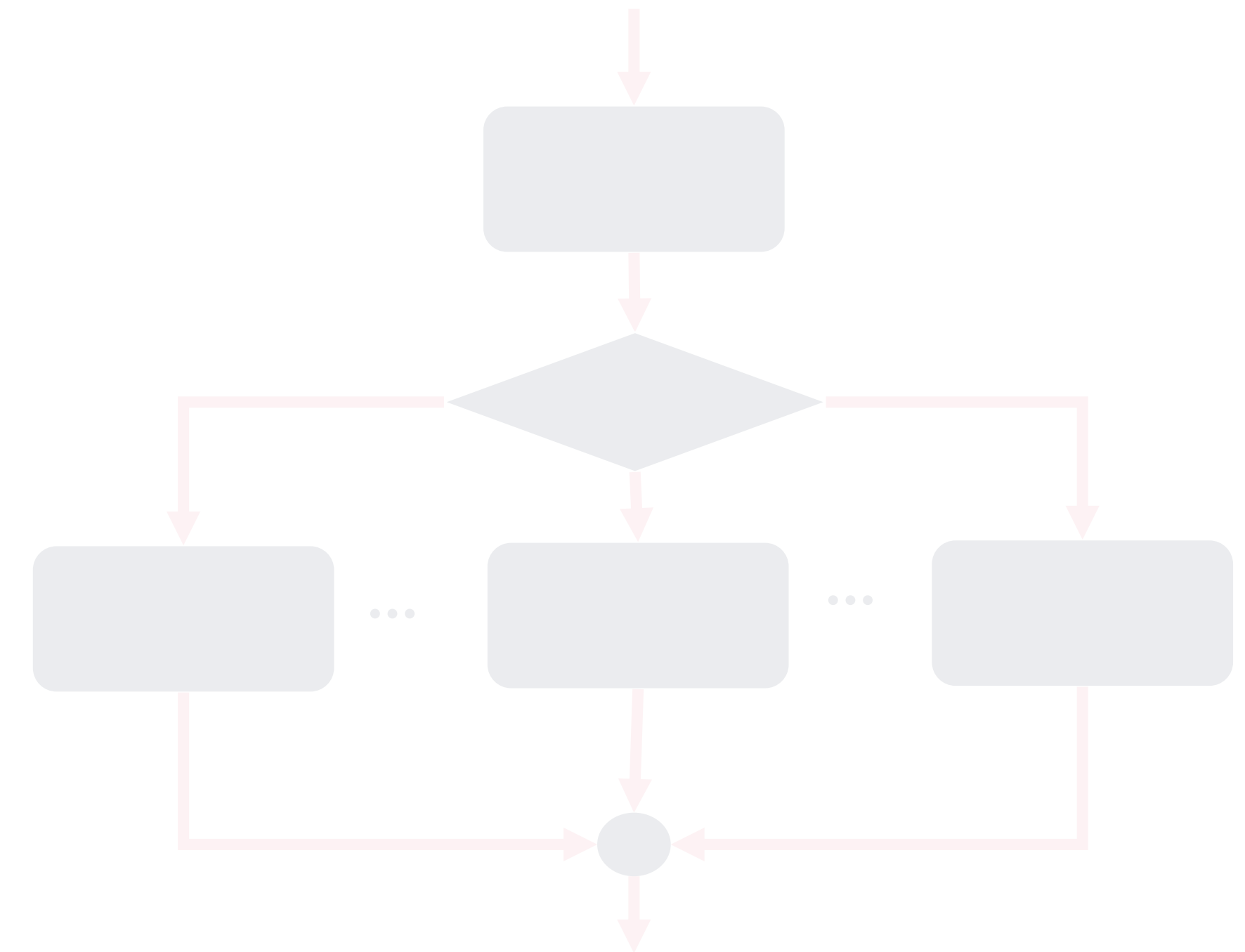
# การทำงานแบบมีเงื่อนไข (Selection)



(Single Selection)



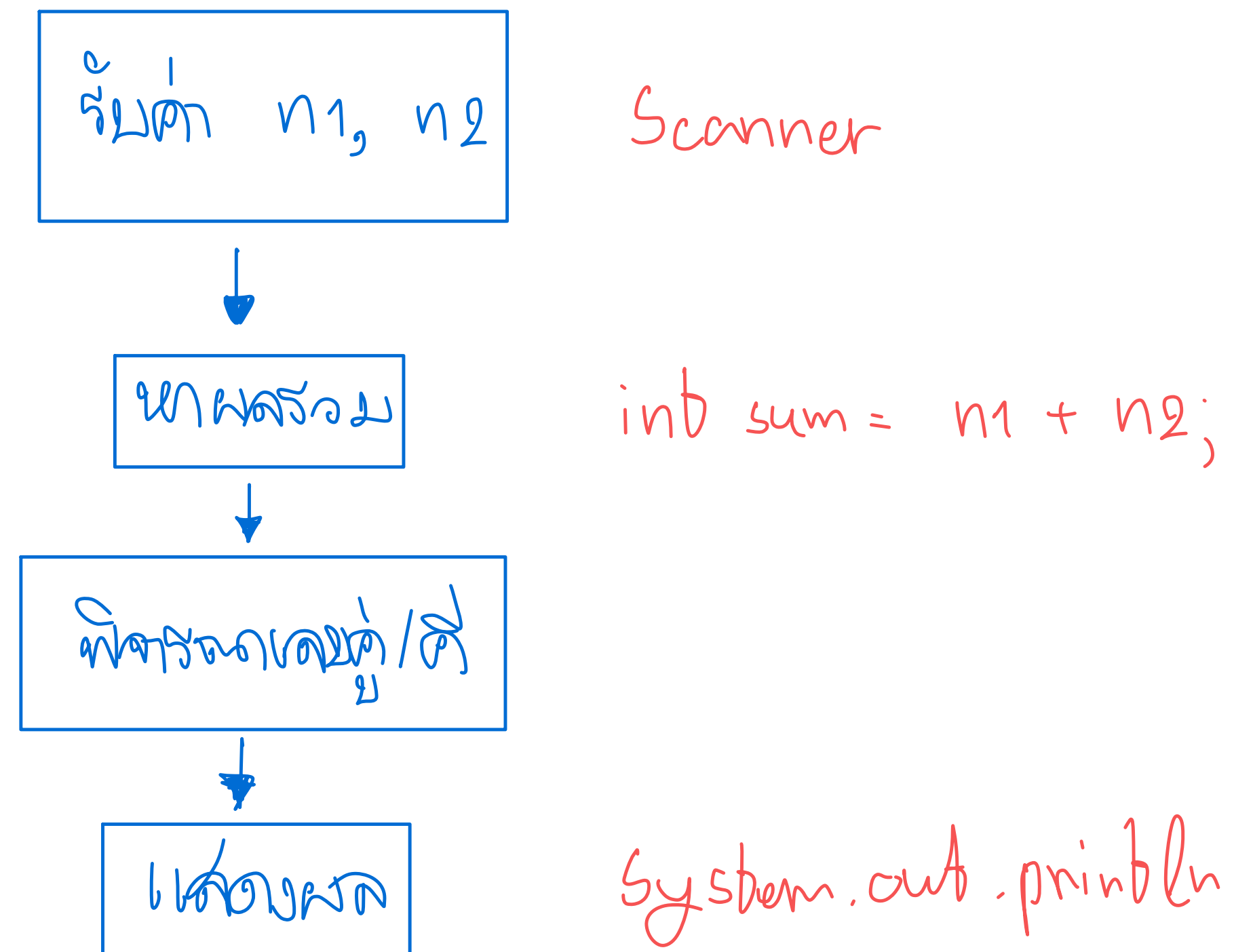
(Double Selection)



(Multiple Selection)

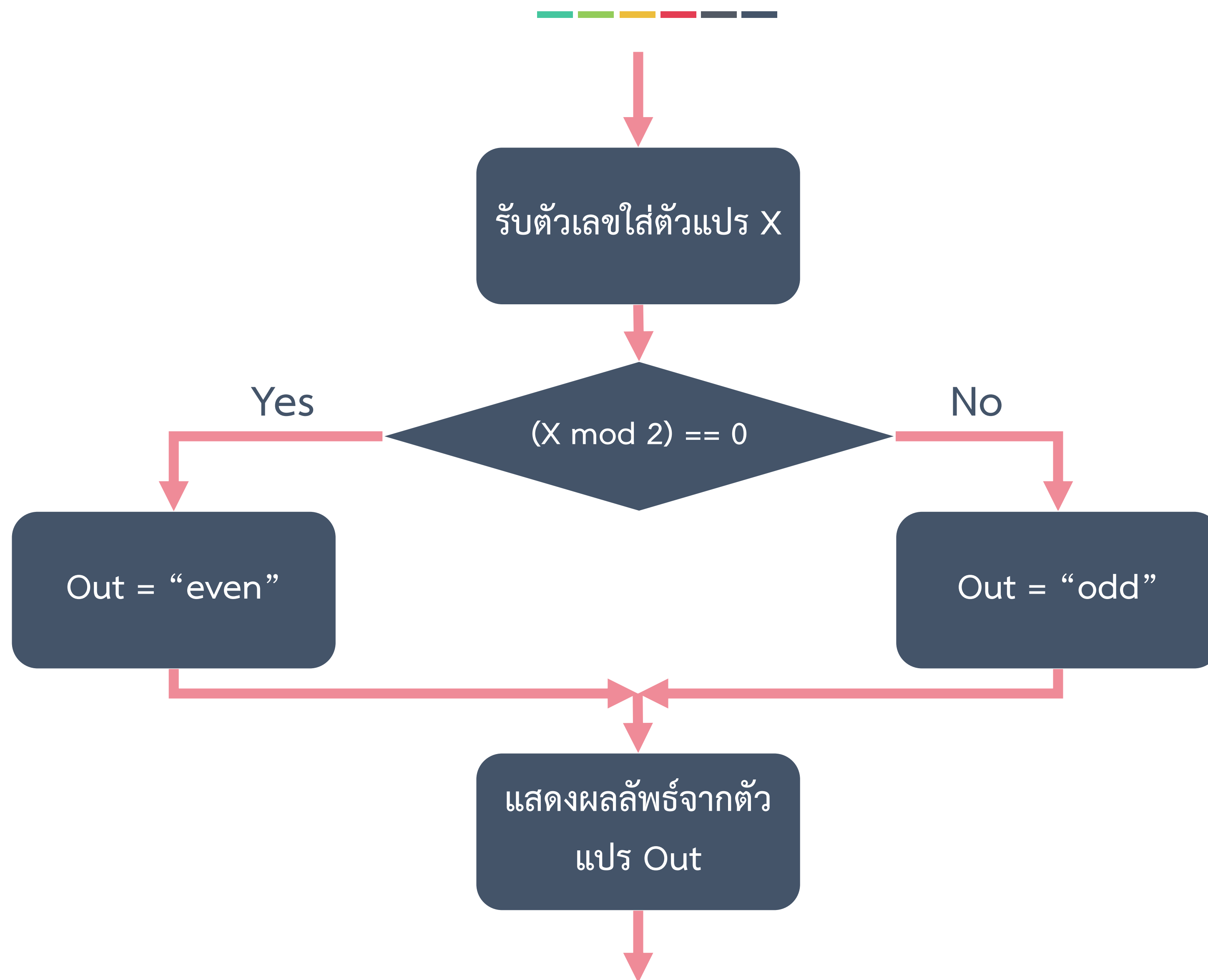
# การทำงานแบบมีเงื่อนไข (Selection)

ให้นักศึกษาเขียนโปรแกรมรับค่าผ่านคีย์บอร์ดจำนวน 2 ค่า เพื่อคำนวณหาค่าผลรวม ถ้าผลรวมมีค่าเป็นเลขคู่ ให้แสดงออกทางหน้าจอว่า “Even” แต่ถ้าผลรวมเป็นเลขคี่ ให้แสดงออกทางหน้าจอว่า “Odd”



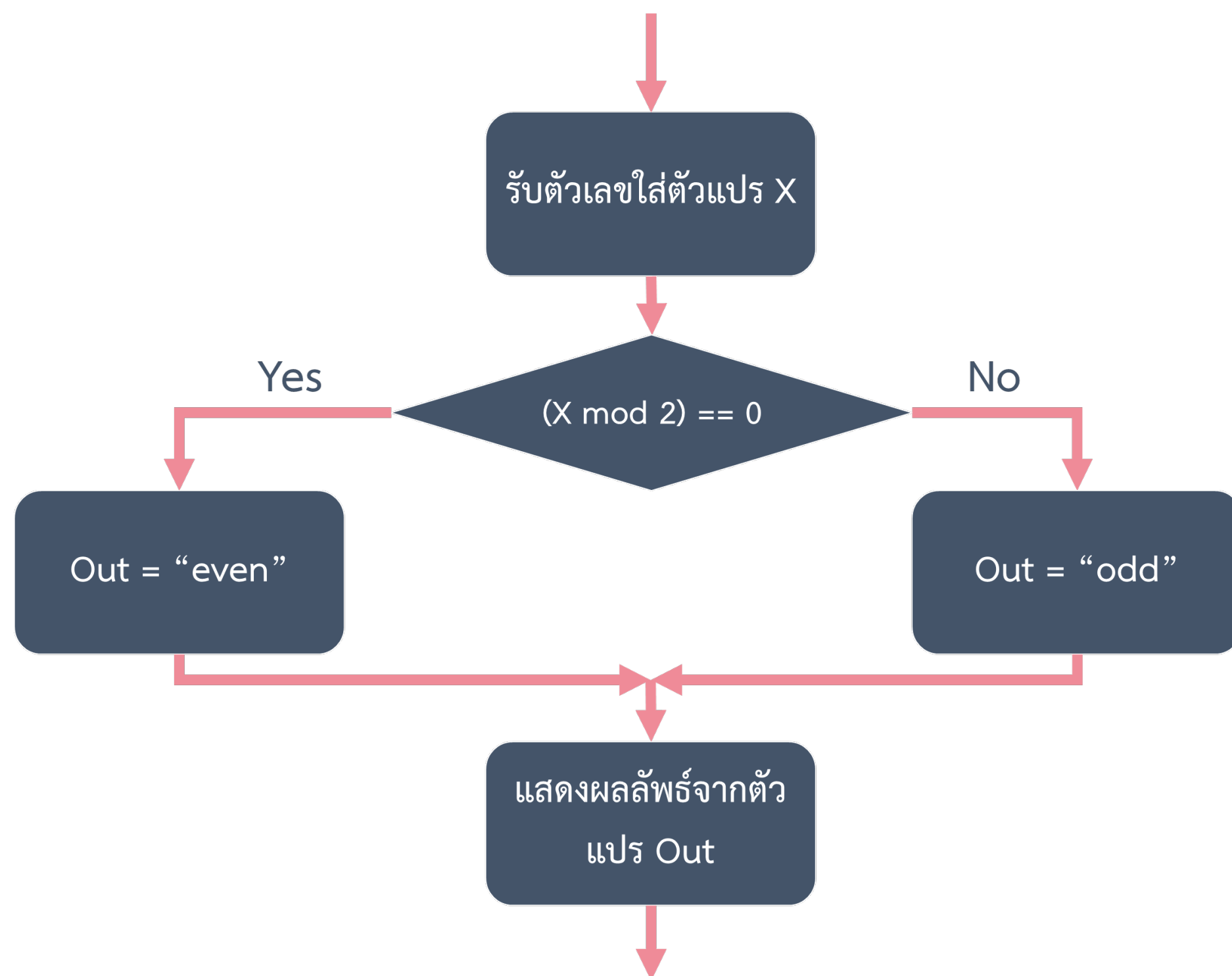


# การทำงานแบบมีเงื่อนไข (Selection)



# การทำงานแบบมีเงื่อนไข (Selection)

รูปแบบ



```

if (เงื่อนไข)
{
    ประโยคที่จะทำงานเมื่อเงื่อนไขเป็นจริง ;
}
else
{
    ประโยคที่จะทำงานเมื่อเงื่อนไขเป็นเท็จ ;
}
  
```

★ if 1 seb ข้อ 1 ความสำคัญ 1 meaning

# ตัวอย่างโปรแกรมหาเลขคู่เลขคี่

ให้นักศึกษาเขียนโปรแกรมรับค่าผ่านคีย์บอร์ดจำนวน  
2 ค่า เพื่อคำนวณหาค่าผลรวม ถ้าผลรวมมีค่าเป็น  
เลขคู่ ให้แสดงออกทางหน้าจอว่า “Even” แต่ถ้า  
ผลรวมเป็นเลขคี่ให้แสดงออกทางหน้าจอว่า “Odd”

```
import java.util.*;
public class OddEven02 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n1 = sc.nextInt();
        int n2 = sc.nextInt();
        int num = n1 + n2;
        if ( (num%2) == 0 ) {
            System.out.print("Even");
        } else {
            System.out.print("Odd");
        }
    }
}
```

# การจัดระเบียบโปรแกรม



```
public class Example01
{
public static void main( String[] args )
{
int price = 99;
int withdraw;
if( price <= 100 )
withdraw = 100;
else
withdraw = 200;
System.out.println(withdraw);
}
}
```



```
public class Example01{
    public static void main( String[] args ) {
        int price = 99;
        int withdraw;

        if( price <= 100 ){
            withdraw = 100;
        } else {
            withdraw = 200;
        }

        System.out.println(withdraw);
    }
}
```



# การเขียน Double Selection แบบย่อ



รูปแบบ

```
variable = condition ? a : b;
```

ตัวอย่าง

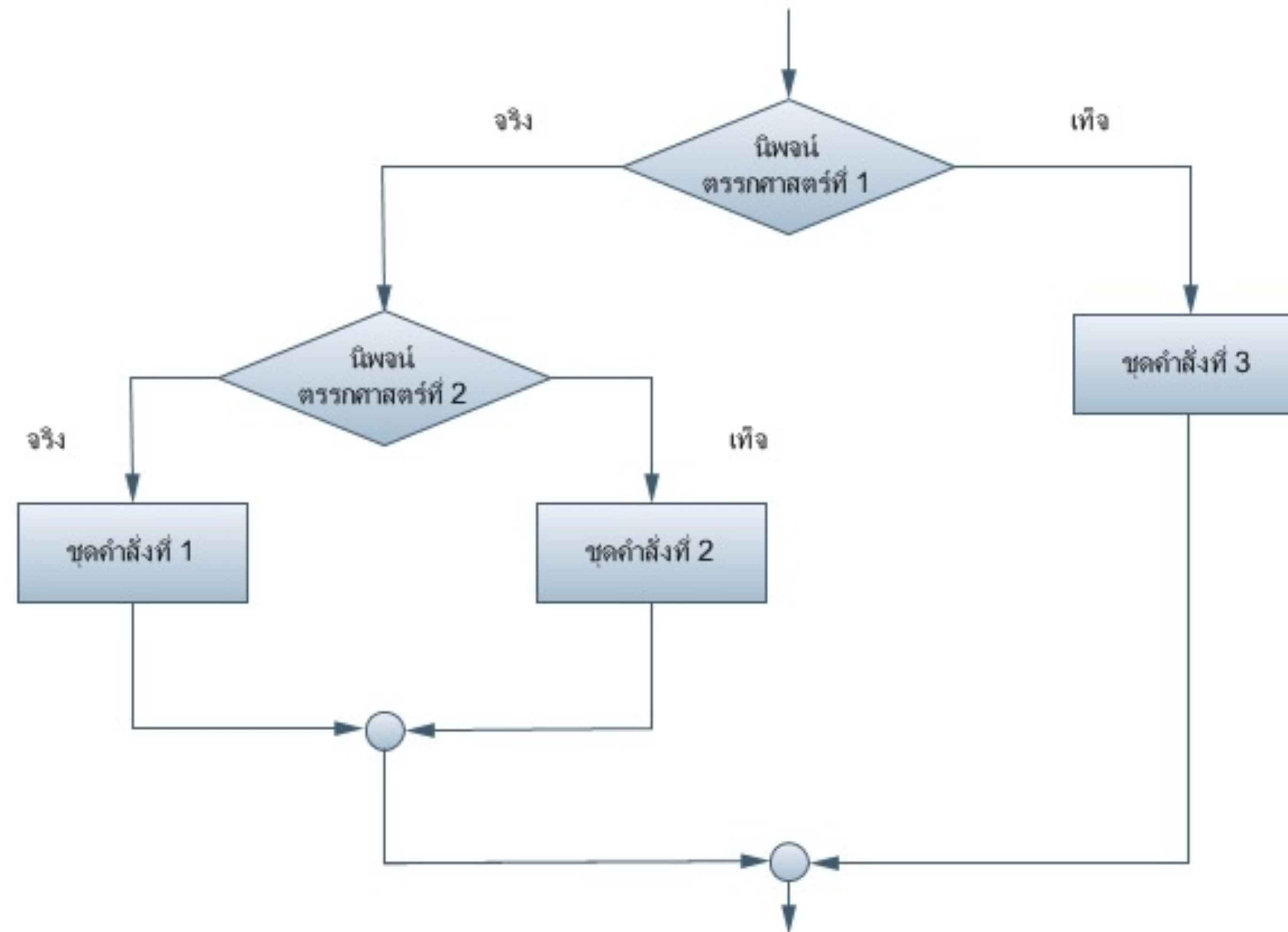
```
int price = 99;  
int withdraw = price <= 100 ? 100 : 200;
```

# รูปแบบของคำสั่ง if แบบซ้อน

เราสามารถใช้คำสั่ง `if` ที่มีหลายเงื่อนไขได้ ดังนี้

```
if (เงื่อนไข 1) {  
    if (เงื่อนไข 2) {  
        statements 1  
    } else {  
        statements 2  
    }  
} else {  
    statements 3  
}
```

# รูปแบบของคำสั่ง if แบบซ้อน



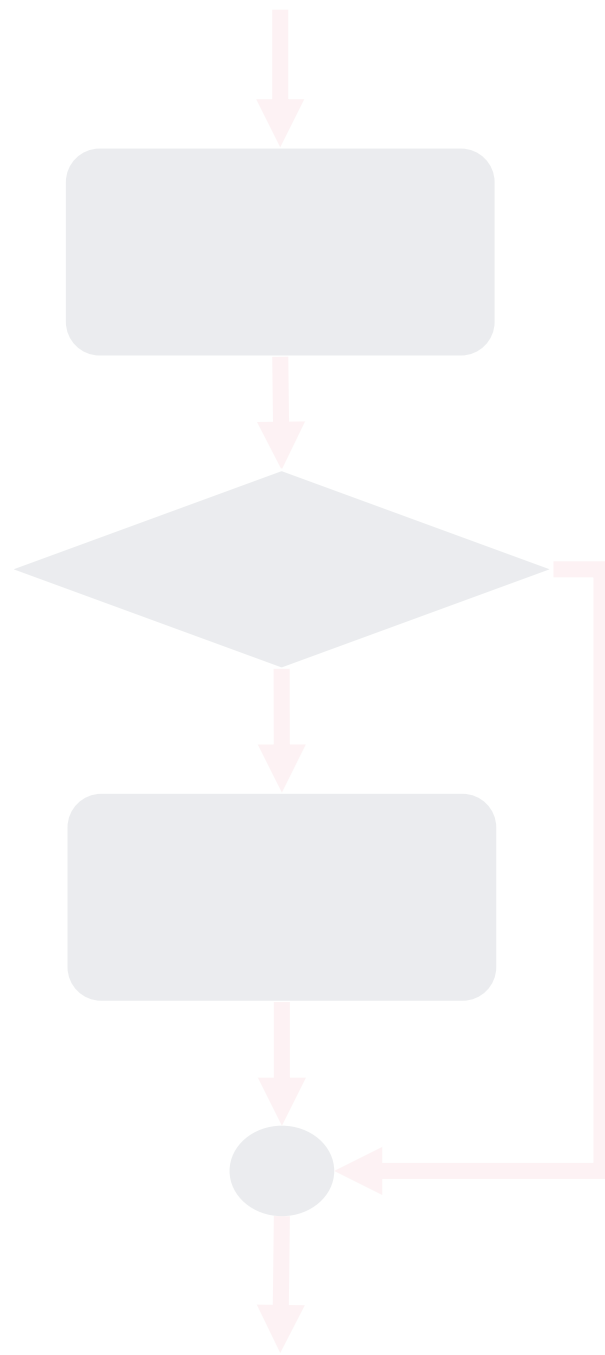
# รูปแบบของคำสั่ง if แบบซ้อน

เราสามารถใช้คำสั่ง `if` ที่มีหลายเงื่อนไขได้ ดังนี้

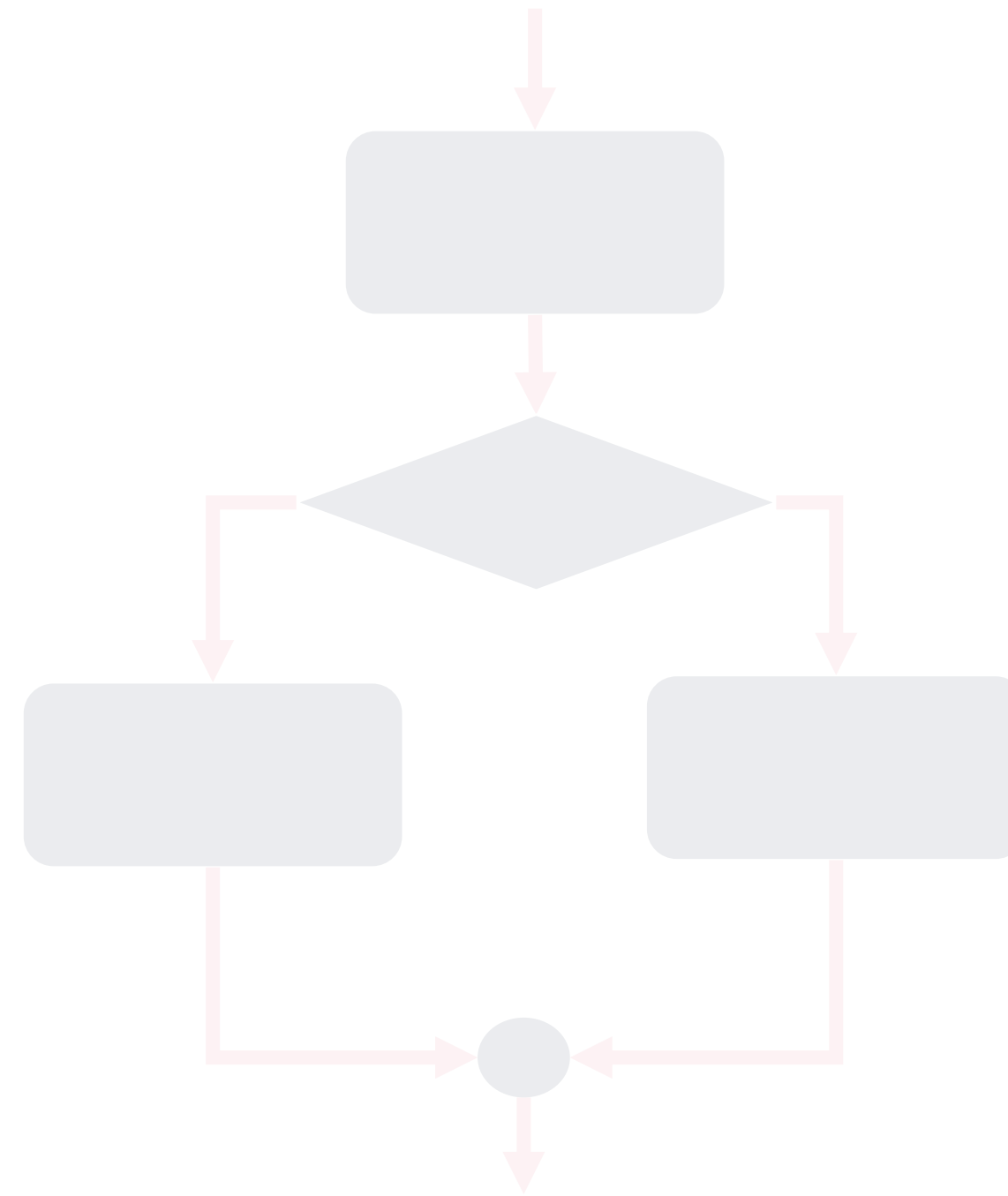
```
if (เงื่อนไข 1) {  
    statements 1  
} else {  
    if (เงื่อนไข 2) {  
        statements 2  
    } else {  
        statements 3  
    }  
}
```



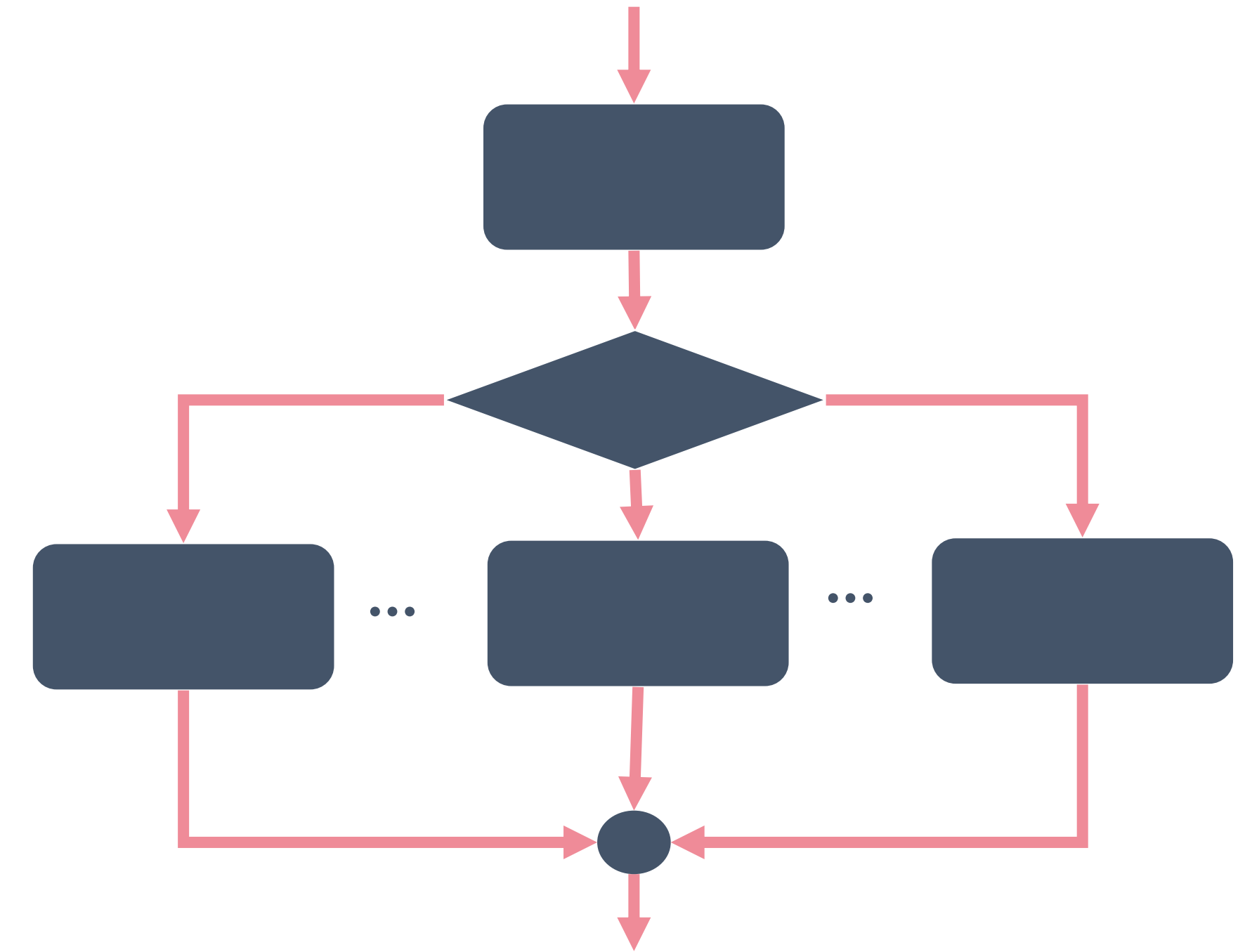
# การทำงานแบบมีเงื่อนไข (Selection)



(Single Selection)



(Double Selection)



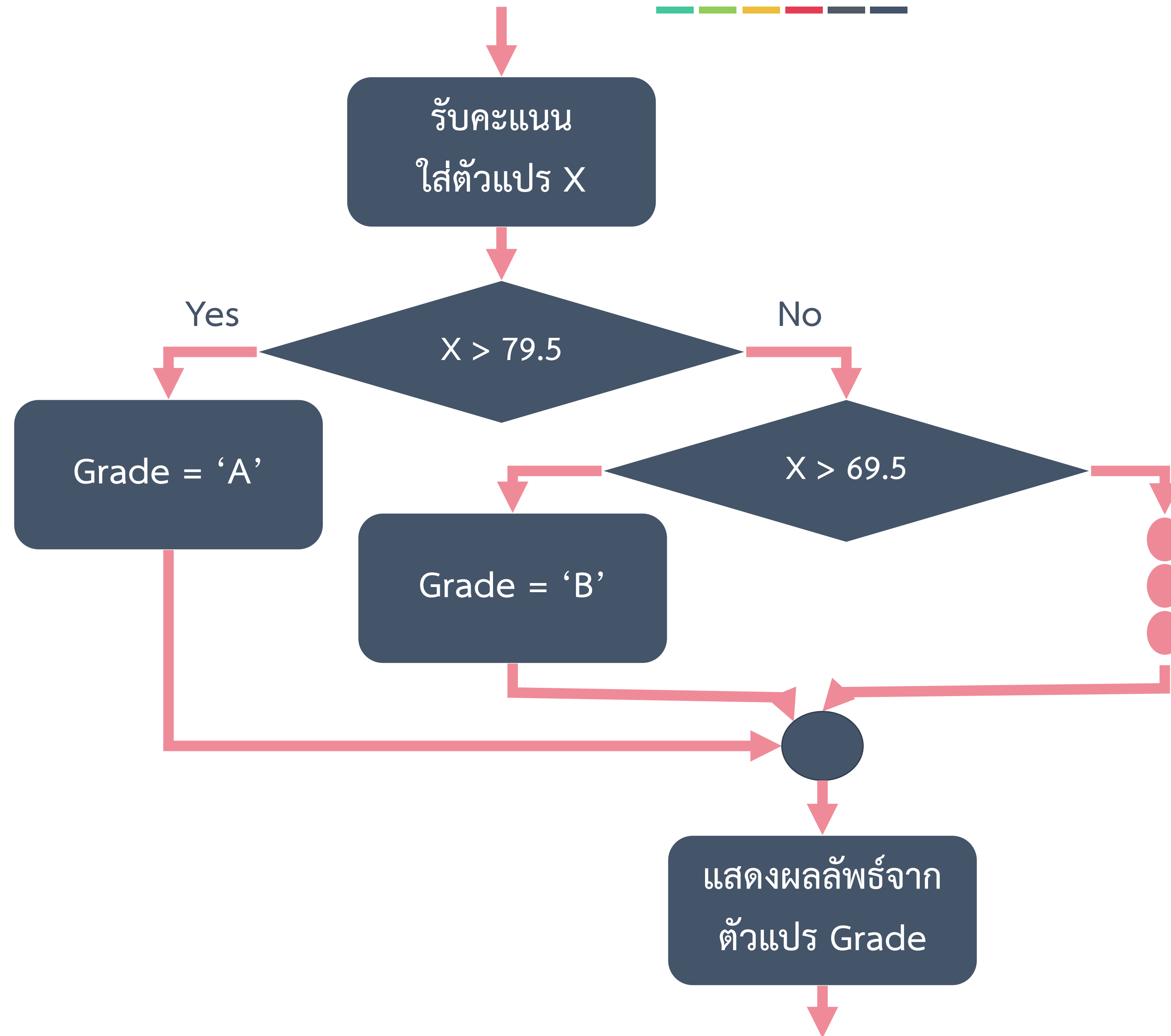
(Multiple Selection)

# การทำงานแบบมีเงื่อนไข (Selection)

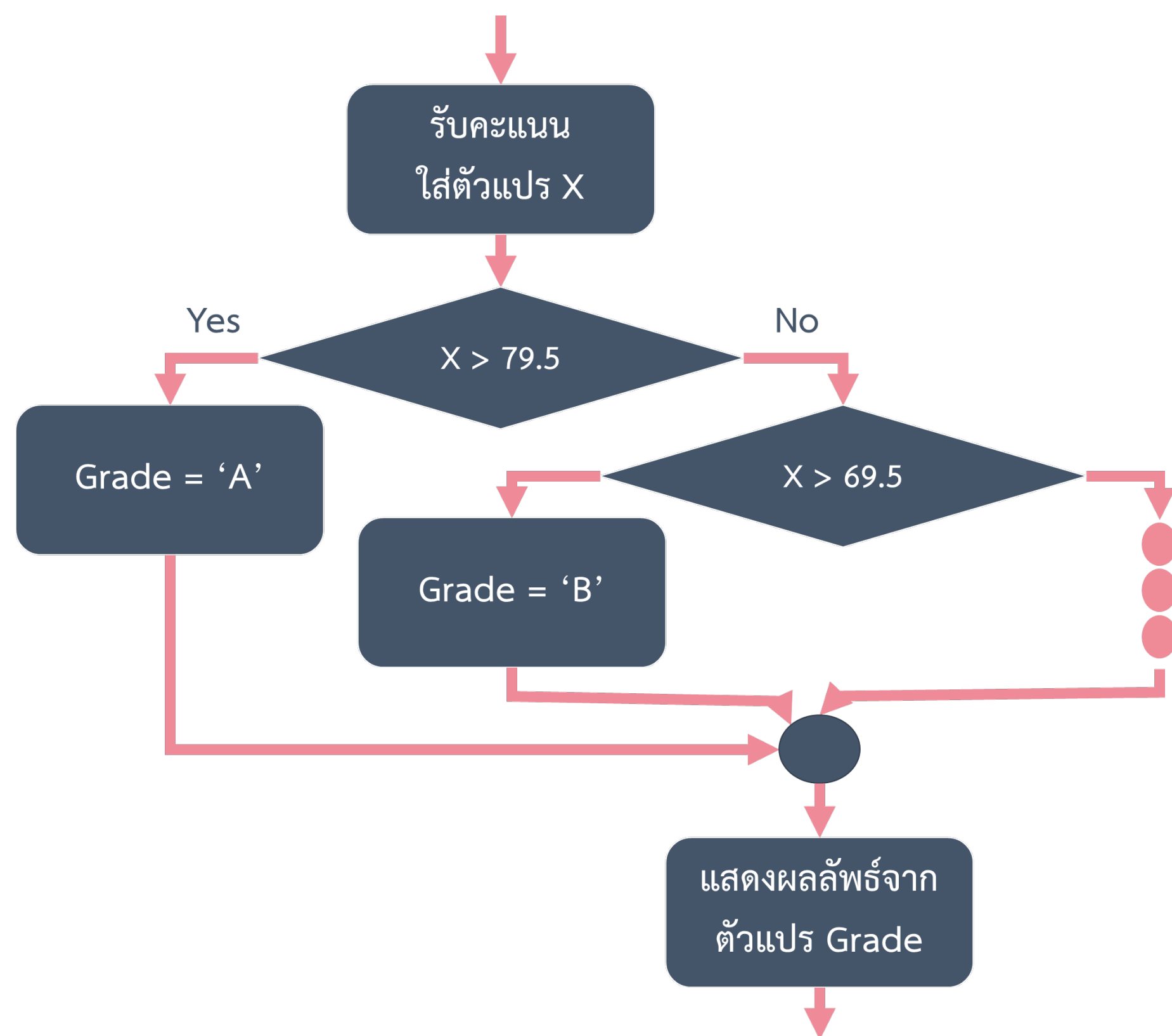
ให้นักศึกษาเขียนโปรแกรมรับคะแนนผ่านคีย์บอร์ด เพื่อคำนวณหาเกรดที่นักศึกษาจะได้ และแสดงผลออกทางจอภาพ โดยที่

- A ก็ต่อเมื่อ คะแนน  $> 79.5$
- B ก็ต่อเมื่อ คะแนน  $> 69.5$
- C ก็ต่อเมื่อ คะแนน  $> 59.5$
- D ก็ต่อเมื่อ คะแนน  $> 49.5$
- F ก็ต่อเมื่อ คะแนน  $< 49.5$

# การทำงานแบบมีเงื่อนไข (Selection)



# การทำงานแบบมีเงื่อนไข (Selection)

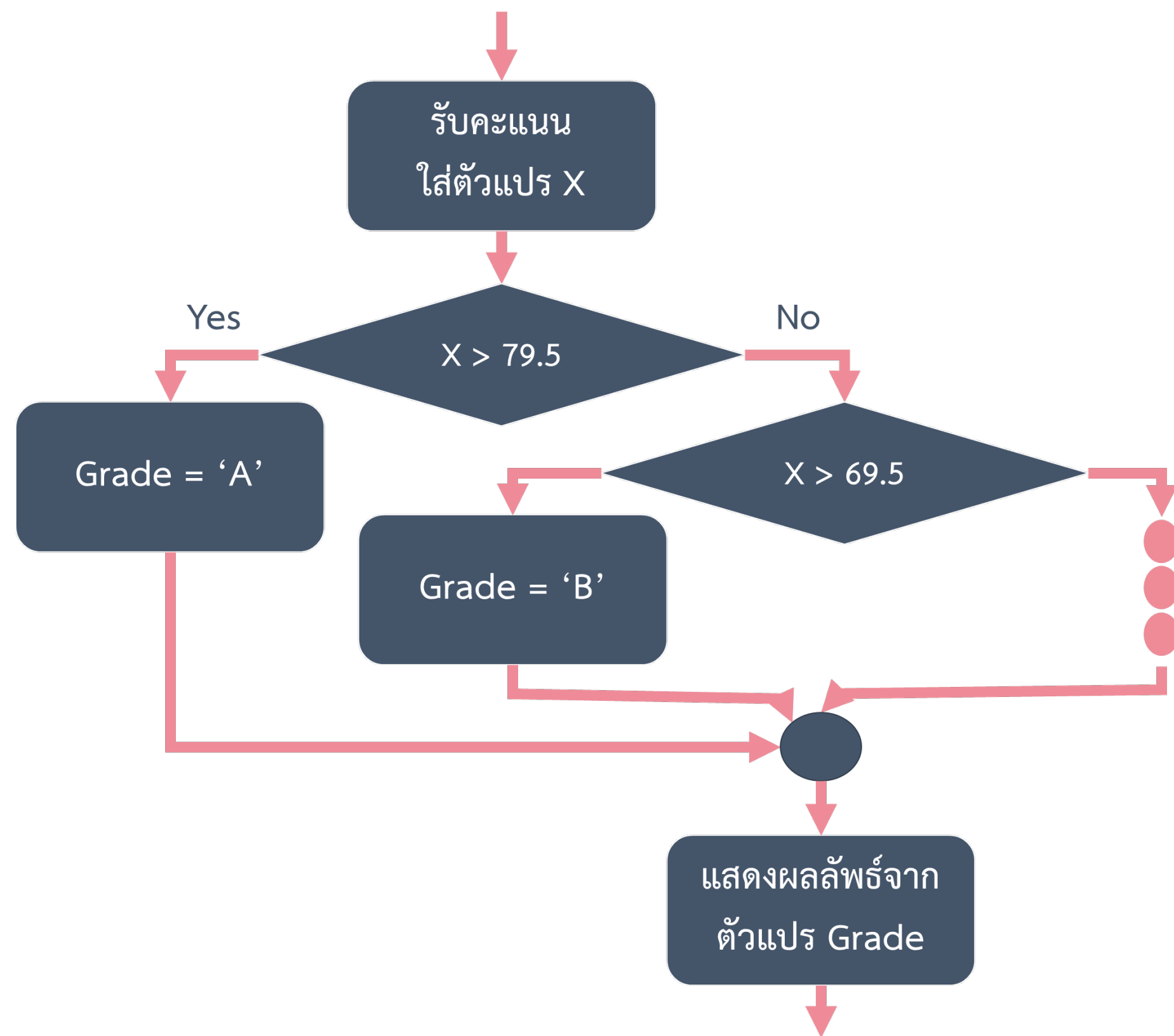


## รูปแบบ

```

if (เงื่อนไข 1) {
    คำสั่งจะทำงานเมื่อเงื่อนไข 1 เป็น จริง;
} else if (เงื่อนไข 2) {
    คำสั่งจะทำงานเมื่อเงื่อนไข 2 เป็น จริง;
} else if (เงื่อนไข 3) {
    คำสั่งจะทำงานเมื่อเงื่อนไข 3 เป็น จริง;
} else {
    คำสั่งจะทำงานก็ต่อเมื่อเงื่อนไขก่อนหน้านี้ทั้งหมดเป็น เท็จ;
}
  
```

# การทำงานแบบมีเงื่อนไข (Selection)



```

import java.util.*;
public class Example03 {
    public static void main( String[] args ) {
        Scanner sc = new Scanner(System.in);
        char grade = '\0';
        int score = sc.nextInt();
        if( score >= 79.5 ){
            grade = 'A';
        } else if( score >= 69.5 ){
            grade = 'B';
        } else if( score >= 59.5 ){
            grade = 'C';
        } else if( score >= 49.5 ){
            grade = 'D';
        } else {
            grade = 'F';
        }
        System.out.println(grade);
    }
}
  
```

# วิเคราะห์โปรแกรมต่อไปนี้

## (กรณี ใช้ if ซ้อน if ในโปรแกรม)

```
public class Main {  
    public static void main(String[] args) {  
        Scanner s = new Scanner(System.in);  
        System.out.print("กรุณาใส่คะแนน (0-100): ");  
        int score = s.nextInt();  
        char grade;  
  
        if((score <=100) && (score>=0)){  
            if(score >= 80){  
                grade = 'A';  
            }else if(score >= 70){  
                grade = 'B';  
            }else if(score >= 60){  
                grade = 'C';  
            }else if(score >= 50){  
                grade = 'D';  
            }else{  
                grade = 'F';  
            }  
            System.out.print("คุณได้เกรด "+grade +" !!!");  
        }else{  
            System.out.print("คุณใส่คะแนนไม่ถูกต้อง");  
        }  
    }  
}
```

# ความแตกต่างระหว่าง **if-else** และ **if-else if**



```
import java.util.*;
public class MyCode {
    public static void main(String[] args){
        Scanner tube = new Scanner(System.in);
        int num = tube.nextInt();
        if ( num > 0 ) {
            System.out.print("I+");
        } else {
            System.out.print("I- & 0");
        }
    }
}
```

```
import java.util.*;
public class MyCode {
    public static void main(String[] args){
        Scanner tube = new Scanner(System.in);
        int num = tube.nextInt();
        if ( num > 0 ) {
            System.out.print("I+");
        } else if ( num < 0 ) {
            System.out.print("I-");
        }
    }
}
```

# ความหลากหลายของการเขียนเงื่อนไข

1

```
if ( (num%2) == 0 ) {  
    System.out.print("Even number");  
} else {  
    System.out.print("Odd number");  
}
```

3

```
if ( (num%2) == 0 ) {  
    System.out.print("Even number");  
} else if ( (num%2) != 0 ) {  
    System.out.print("Odd number");  
}
```

2

```
if ( (num%2) == 0 ) {  
    System.out.print("Even number");  
} else if ( (num%2) == 1 ) {  
    System.out.print("Odd number");  
}
```



# การเปรียบเทียบ การเท่ากัน

int  
 char  
 boolean

} ใช้ == ใช้ได้เสมอ

double ใช้ == ใช้ไม่ได้ symbol x แต่ไม่ควรถูกใช้เพราะอาจเกิดข้อผิดพลาด

int

```

int n1 = sc.nextInt();
int n2 = sc.nextInt();
if( n1 == n2 ) {
    System.out.println("==");
} else {
    System.out.println("!=");
}
  
```

char

```

char n1 = sc.next().charAt(0);
if( n1 == 'A' ) {
    System.out.println("==");
} else {
    System.out.println("!=");
}
  
```

ใช้การเทียบผลต่างแทน

double

```

double n1 = sc.nextDouble();
double n2 = sc.nextDouble();
if( Math.abs(n1 - n2) < 0.001 ) {
    System.out.println("==");
} else {
    System.out.println("!=");
}
  
```

(ถ้าใช้ == จะเป็นการเช็ค address)

String / object

```

String n1 = sc.nextLine();
String n2 = sc.nextLine();
if( n1.equals(n2) ) { เช็ค value
    System.out.println("==");
} else {
    System.out.println("!=");
}
  
```

# วิเคราะห์โปรแกรมต่อไปนี้

(กรณี ใช้ if-else มากกว่า 1 ตำแหน่งในโปรแกรม)

```
public class Main {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        System.out.print("กรุณาใส่เลขเดือน (1-12) : ");
        int month = s.nextInt();
        System.out.print("กรุณาใส่เลขวัน (1-7) : ");
        int day = s.nextInt();

        String txt_month = "";
        String txt_day = "";

        if(month == 1)      { txt_month = "JAN" ; }
        else if(month == 2) { txt_month = "FEB" ; }
        else if(month == 3) { txt_month = "MAR" ; }
        .....
        else if(month == 12) { txt_month = "DEC" ; }

        if(day == 1)        { txt_day = "MON" ; }
        else if(day == 2)   { txt_day = "TUE" ; }
        else if(day == 3)   { txt_day = "WED" ; }
        .....
        else if(day == 7)   { txt_day = "SUN" ; }

        System.out.print( txt_month + " , " + txt_day + " !!! ");
    }
}
```

# รูปแบบของคำสั่ง switch



มีรูปแบบการใช้คำสั่ง ดังนี้

```
switch (expression) {  
    if ↔ case value-1:    statements 1  
                           break;  
    else if ↔ case value-2: statements 2  
                           break;  
                           :  
                           :  
        case value-N:    statements N  
                           break;  
    else ↔ default:      statements N+1;  
}
```

# รูปแบบของคำสั่ง switch

มีรูปแบบการใช้คำสั่ง ดังนี้

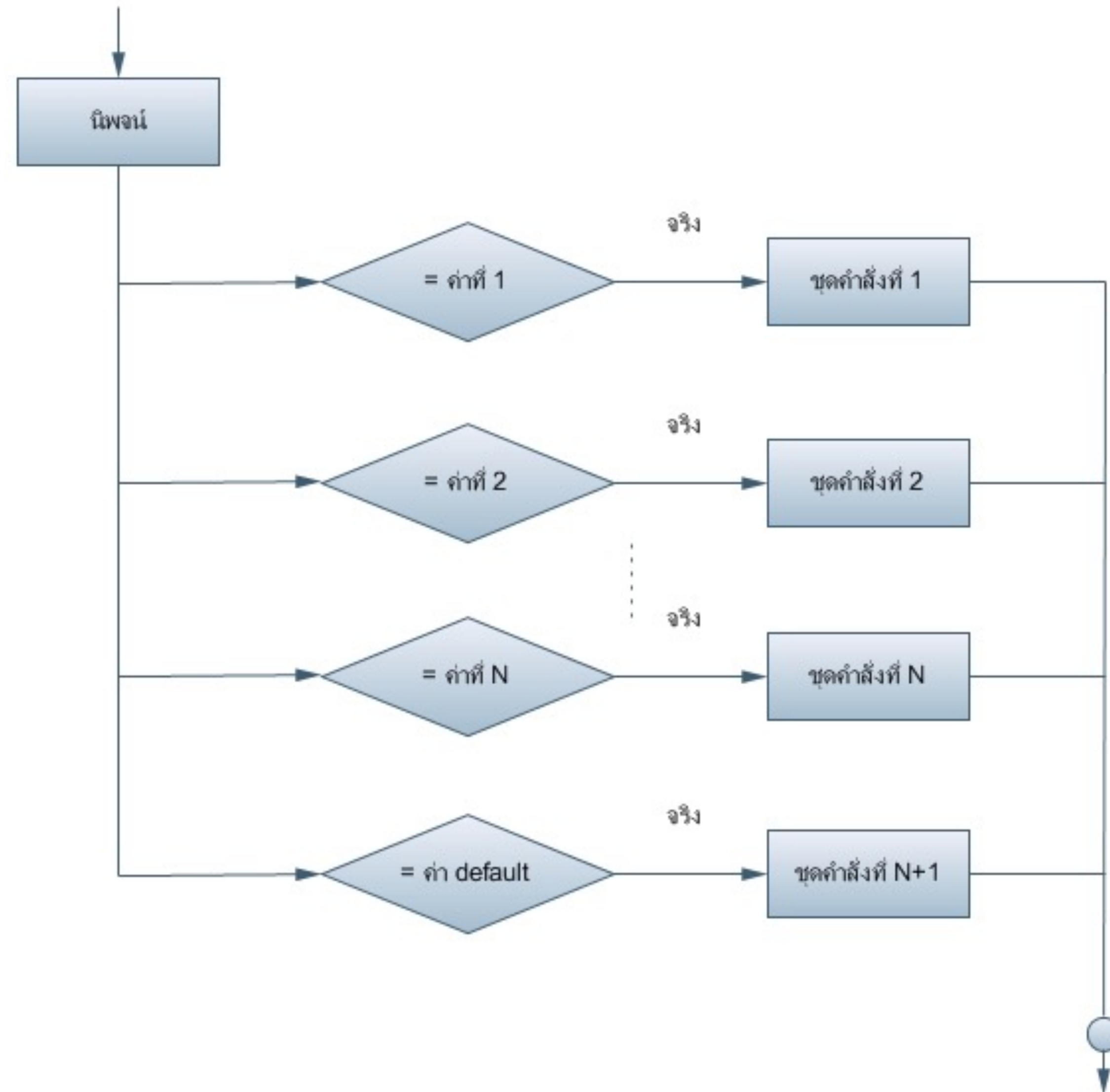
นิพจน์ต้องมีชนิดข้อมูลเป็น  
`char`, `byte`, `short`  
หรือ `int` เท่านั้น

- ชนิดข้อมูลของนิพจน์และค่าที่ 1 ถึง N ต้องเป็นชนิดเดียวกัน
- ถ้าค่าของนิพจน์ตรงกับค่าใด จะทำชุดคำสั่งของค่านั้น

- ถ้าค่าของนิพจน์ไม่ตรงกับค่าใดเลย จะทำชุดคำสั่งของ default
- default จะมีหรือไม่มีก็ได้

```
switch (expression) {  
    case value-1:      statements 1  
                        break;  
    case value-2:      statements 2  
                        break;  
                        :  
    case value-N:      statements N  
                        break;  
    default:           statements N+1;  
}
```

# รูปแบบของคำสั่ง switch



# ตัวอย่างโปรแกรมที่ใช้คำสั่ง switch

```
import java.util.Scanner;
public class Main {
    public void showDemo() {
        Scanner sc = new Scanner(System.in);
        int x = sc.nextInt();
        switch (x) {
            case 1: System.out.print("Value is one");
                     break;
            case 2: System.out.print("Value is two");
                     break;
            default: System.out.print("Other than 1 and 2");
        }
    }
}
```

# ตัวอย่างโปรแกรมที่ใช้คำสั่ง switch

```
int month = 2, 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:
        numDays = 28;
        break;
    default:
        System.out.println("Invalid month."); break;
}
System.out.println("Number of Days = " + numDays);
```



# วิเคราะห์ความแตกต่างของโปรแกรมต่อไปนี้

(กรณี ใช้ if-else if-else กับ switch-case)

```
public class Main {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        System.out.print("กรุณาใส่เลขวัน(1-7) : ");
        int day = s.nextInt();
        String txt_day = "";

        if(day == 1){
            txt_day = "จันทร์";
        }else if(day == 2){
            txt_day = "อังคาร";
        }else if(day == 3){
            txt_day = "พุธ";
        }else if(day == 4){
            txt_day = "พฤหัสบดี";
        }else if(day == 5){
            txt_day = "ศุกร์";
        }else if(day == 6){
            txt_day = "เสาร์";
        }else if(day == 7){
            txt_day = "อาทิตย์";
        }

        System.out.print("สวัสดีวัน"+txt_day + "\nขอให้มีความสุข ");
    }
}
```

```
public class Main {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        System.out.print("กรุณาใส่เลขวัน(1-7) : ");
        int day = s.nextInt();
        String txt_day = "";

        switch(day){
            case 1:
                txt_day = "จันทร์"; break;
            case 2:
                txt_day = "อังคาร"; break;
            case 3:
                txt_day = "พุธ"; break;
            case 4:
                txt_day = "พฤหัสบดี"; break;
            case 5:
                txt_day = "ศุกร์"; break;
            case 6:
                txt_day = "เสาร์"; break;
            case 7:
                txt_day = "อาทิตย์"; break;
        }

        System.out.print("สวัสดีวัน"+txt_day + " \nขอให้มีความสุข ");
    }
}
```