

複習算術運算式、條件運算式

臺北市立大學 [資訊科學系\(含碩士班\)](#)

賴阿福

搶答Q1: debug

```
int x=9, y=8, z;
```

```
if (y=x+1) z=3;
```

```
if (x-1=y+8) z=4;
```

搶答Q1: debug (answer)

```
int x=9, y=8, z;  
if (y==x+1) z=3;  
if (x-1==y+8) z=4;
```

```
int x=9, y=8, z;  
if (y==x+1)  
    z=3;  
if (x-1==y+8)  
    z=4;
```

```
int x=9, y=8, z;  
if (y==x+1)  
    { z=3; }  
if (x-1==y+8)  
    { z=4; }
```

條件Q2:邏輯運算式 (立即搶答)

`if (nn>=100 && nn<=999)`

- 符合條件數值範圍?

`If (nn>=100 || nn<=999)`

-
- 符合條件數值範圍?

條件Q3:邏輯運算式 (立即搶答)

`If (nn<100 || nn>999)`

- 符合條件數值範圍?

`If (nn<100 && nn>999)`

- 符合條件數值範圍?

`If (!(nn>=100 && nn<=999))`

- 符合條件數值範圍?

條件Q2/3:邏輯運算式 (answer)

if ($nn \geq 100 \ \&\& \ nn \leq 999$)

- $\&\&$: and (及) 交集
- 符合條件數值範圍? $100 \leq nn \leq 999$

If ($nn \geq 100 \ || \ nn \leq 999$)

- $||$: or (或) 聯集
- 符合條件數值範圍? unlimited

If ($nn < 100 \ || \ nn > 999$)

- 符合條件數值範圍? 100~999之外

If ($nn < 100 \ \&\& \ nn > 999$)

- 符合條件數值範圍? 無此

if ($!(nn \geq 100 \ \&\& \ nn \leq 999)$)

- $!$: not (反)
- 符合條件數值範圍? Same as ($nn < 100 \ || \ nn > 999$)



搶答Q4

```
int x , y , z = 5;
```

```
x = y = z + 5;
```

```
z = x + y + z;
```

```
System.out.println("z = " + z);
```

搶答Q5

```
int x1 = 9, y1 = 8, z1 = 7;  
x1 = x1 % z1;  
if (x1 < 5)  
    y1 = y1 + 1;  
else  
    y1 = x1 + 1;  
System.out.println("y1 = " + y1);
```

搶答Q6

```
x1 = 9; y1=8; z1=7;  
x1 = x1 % z1;  
if (!(x1<=5))  
    y1 = y1 + 1;  
else  
    y1 = x1 + 1;  
System.out.println("y1 = " + y1);
```

搶答Q7

```
int w=7, p = 4;
```

```
x = 3;
```

```
p = w / x * 2;
```

```
if (p<18 && p>8)
```

```
    x = (x*x)%2;
```

```
else
```

```
    x = x % 3;
```

```
System.out.println("x = " + x);
```

搶答Q8

```
w = 7; p = 4; x = 3;
```

```
p = w / x * 2;
```

```
if (p < 18 || p > 8)
```

```
    x = (x * x) % 2;
```

```
else
```

```
    x = x % 3;
```

```
System.out.println("x = " + x);
```

搶答Q9

```
w = 8; p = 6; x = 4;  
x++; //x = x + 1;  
if( w+1 == p+3) x=x+1;  
System.out.println("x = " + x);
```

搶答Q10

```
w = 8; p = 4; x = 2;
```

```
x--; //x = x-1;
```

```
if (w+1!=p+3)
```

```
    x = x--;
```

```
else
```

```
    x = x++;
```

```
System.out.println("x = " + x);
```

Q4~Q10 Answer

- Please download [test_if.java](#), compile & run
- Then try to revise and test

```
z = 25  
y1 = 9  
y1 = 3  
y1 = 8  
x = 0  
x = 1  
x = 6  
x = 1
```

Assignment statement

<code>x=x+1;</code>	<code>x=x-1;</code>
<code>x++;</code>	<code>x--;</code>
<code>++x;</code>	<code>--x;</code>
<code>x+=1;</code>	<code>x-=1;</code>

再談分支 (selection, branch) III

程式如何轉彎?

臺北市立大學 [資訊科學系\(含碩士班\)](#)

賴阿福

Review BMI診斷:分成三層次(簡化)

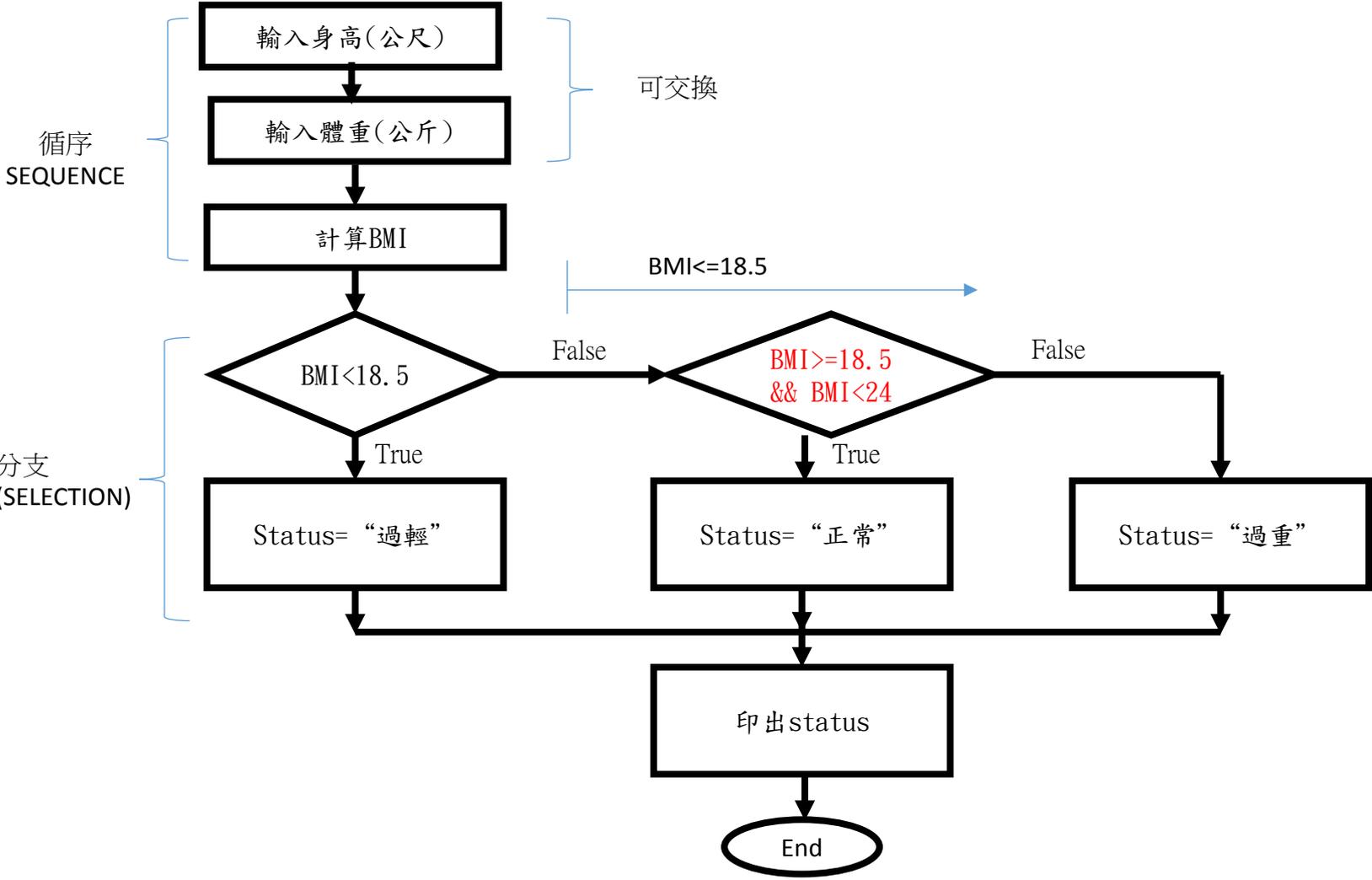
```
import java.util.Scanner;
public class BMI_2 {
public static void main(String[] args) {
    Scanner input = new Scanner(System.in);
    System.out.print("輸入身高:");
    double height = input.nextDouble();
    System.out.print("輸入體重:");
    double weight = input.nextDouble();
    double bmi = Math.round((weight/ (height*height) )* 100) / 100.0;
    String status;
```

```
    if (bmi < 18.5) {
        status = "體重過輕Underweight";
    }
    else if (bmi < 24) { //(bmi>=18.5 && bmi < 24)
        status = "正常Normal";
    }
    else
        status = "過重Overweight";
```

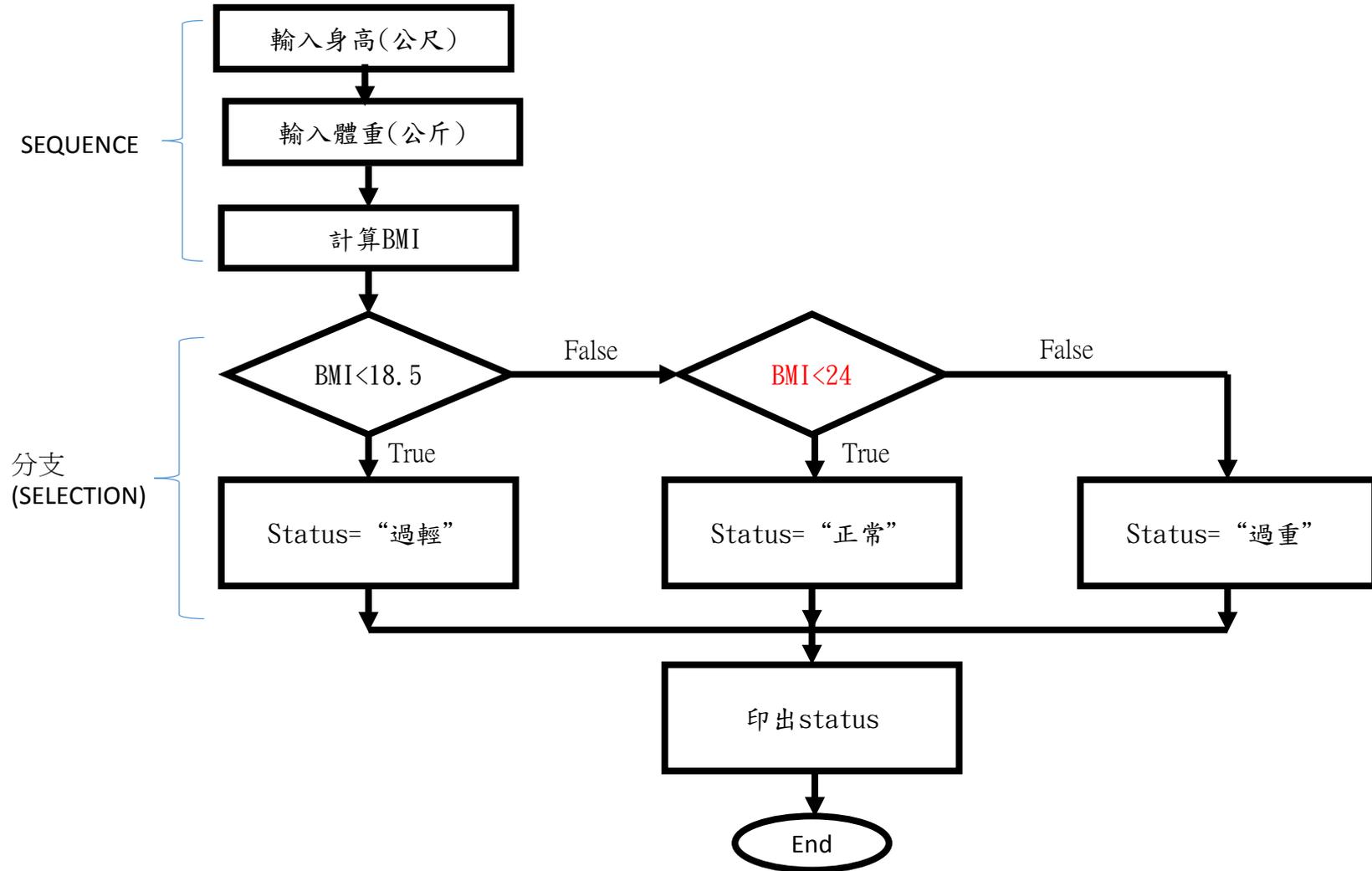
```
    System.out.println("BMI : "+bmi+" , 狀態: "+status);
} //main
} //class
```

成人肥胖定義	身體質量
體重過輕	BMI<18.5
健康體位	18.5<=BMI<24
體位異常	過重: 24<=BMI<27 輕度肥胖: 27 <= BMI < 30 中度肥胖: 30 <= BMI < 35 重度肥胖: BMI >= 35

多分支流程圖



多分支流程圖



BMI診斷

- 完整BMI診斷需要分支?
- 流程圖如何畫?

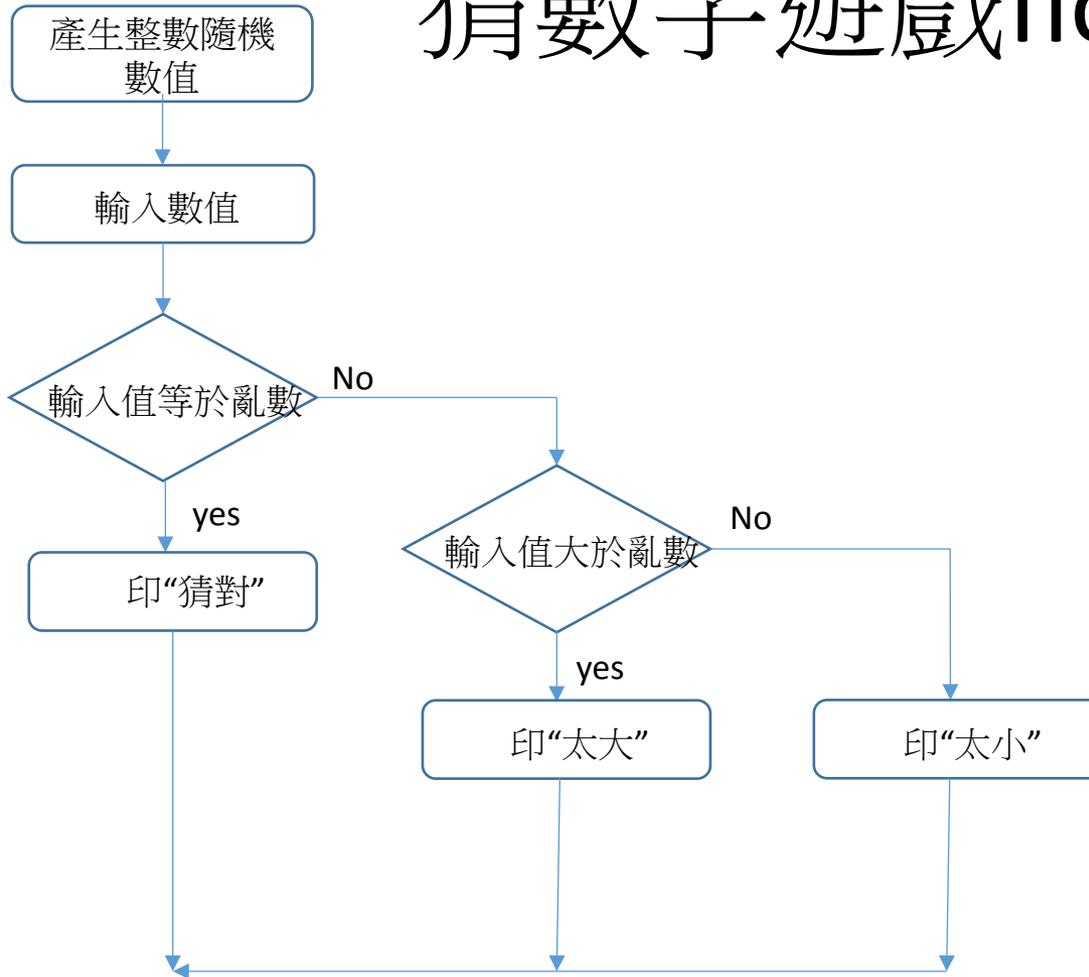
成人肥胖定義	身體質量
體重過輕	BMI<18.5
健康體位	18.5<=BMI<24
體位異常	過重：24<=BMI<27 輕度肥胖：27 <= BMI < 30 中度肥胖：30 <= BMI < 35 重度肥胖：BMI >= 35

猜數字遊戲

解題方法:

- 運用亂數函數(亂數類別)產生整數隨機數值
- 猜數字:讓使用者輸入數值
- 判斷答對與否，且輸出回饋信息

猜數字遊戲flow chart



猜數字遊戲(1):運用亂數設計猜數字遊戲程式

```
import java.security.SecureRandom;
import java.util.Scanner;

public class GuessN_0 {
    public static void main(String[] args) {

        SecureRandom sr = new SecureRandom();
        Scanner input = new Scanner(System.in);

        int a = 0;
        int b = 0;
        a = sr.nextInt(100)+1;

        System.out.print("猜數字(1~100) : ");
        b = input.nextInt();
        if (a == b) {
            System.out.println("恭喜猜對!");
        }
        else if (a > b)
            System.out.println("猜的太小囉\n");
        else
            System.out.println("猜的太大囉\n");

        System.out.println("亂數為"+a);
    } //main
} //class
```

多重分支
Multi-way if

產生1~100亂數

- 建立亂數類別SecureRandom之物件，再以此類別nextInt函數(方法)產生1~100隨機整數
 - Secure:安全，Random:亂數、隨機

```
SecureRandom sr = new SecureRandom();  
a = sr.nextInt(100)+1;
```
- new SecureRandom()以建立sr物件，再用.nextInt(100)方法/method產生0~99隨機整數
 - sr.nextInt(100) +1
- import java.security.SecureRandom;
- 如何產生10~90隨機整數?

猜數字遊戲(2):運用亂數設計猜數字遊戲程式

```
import java.security.SecureRandom;
import java.util.Scanner;

public class GuessN_1 {
    public static void main(String[] args) {

        SecureRandom sr = new SecureRandom();
        Scanner input = new Scanner(System.in);

        int a = 0;
        int b = 0;
        a = sr.nextInt(100)+1;

        System.out.print("猜數字(1~100) : ");
        b = input.nextInt();
        if (a == b) {
            System.out.println("恭喜猜對!");
        }
        else { //nested if
            if (a > b)
                System.out.println("猜的太小囉\n");
            else
                System.out.println("猜的太大囉\n");
        }
        System.out.println("亂數為"+a);
    } //main
} //class
```

巢狀分支
Nested if

猜數字遊戲(3):運用亂數設計猜數字遊戲程式

```
import java.security.SecureRandom;
import java.util.Scanner;

public class GuessN_2 {
    public static void main(String[] args) {

        SecureRandom sr = new SecureRandom();
        Scanner input = new Scanner(System.in);

        int a = 0;
        int b = 0;
        a = sr.nextInt(100)+1;

        System.out.print("猜數字(1~100) : ");
        b = input.nextInt();
        if (a!= b) { //nested if
            if (a > b)
                System.out.println("猜的太小囉\n");
            else
                System.out.println("猜的太大囉\n");
            System.out.println("亂數為"+a);
        } //if
    } else
        System.out.println("恭喜猜對!");
    } //main
} //class
```

巢狀分支
Nested if

不同分支，相同結果

- 多重Multi-way if

```
if (a == b) {  
    System.out.println("恭喜猜對!");  
}  
else if (a > b)  
    System.out.println("猜的太小囉\n");  
else  
    System.out.println("猜的太大囉\n");
```

三個分支

- 巢狀分支Nested if

```
if (a != b) { //nested if  
    if (a > b)  
        System.out.println("猜的太小囉\n");  
    else  
        System.out.println("猜的太大囉\n");  
    System.out.println("亂數為"+a);  
} //if  
else  
    System.out.println("恭喜猜對!");
```

```
if (a == b)  
    System.out.println("恭喜猜對!");  
else { //nested if  
    if (a > b)  
        System.out.println("猜的太小囉\n");  
    else  
        System.out.println("猜的太大囉\n");  
    System.out.println("亂數為"+a);  
} //else
```

巢狀分支Nested if

第一層:二個分支

```
if (a != b) { //nested if
    if (a > b)
        System.out.println("猜的太小囉\n");
    else
        System.out.println("猜的太大囉\n");
    System.out.println("亂數為"+a);
} //if
else
    System.out.println("恭喜猜對!");
```

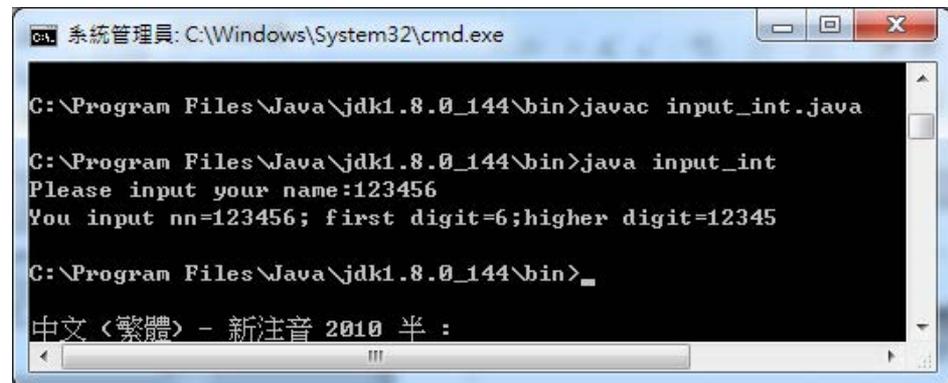
第二層:二個分支

```
if (a == b)
    System.out.println("恭喜猜對!");
else { //nested if
    if (a > b)
        System.out.println("猜的太小囉\n");
    else
        System.out.println("猜的太大囉\n");
    System.out.println("亂數為"+a);
} //else
```

猜數字遊戲之反思

- 目標: 讓初學者熟悉巢狀分支Nested if、多重分支Multi-way if
 - 邏輯變化
 - 分支結構
- 缺點: 無法讓使用者依據太大、太小的回饋訊息再次輸入數字，應重複直至使用者答對或不想玩。
 - **Loop** (迴圈) can do it

```
System.out.print("Please input a 3-digit decimal:");
Scanner ipt = new Scanner(System.in);
int nn = ipt.nextInt();
int n1=nn%10;
int n2=nn/10;
System.out.print("You input nn="+nn+"; first
digit="+n1+";");
System.out.println("higher digit="+n2);
```

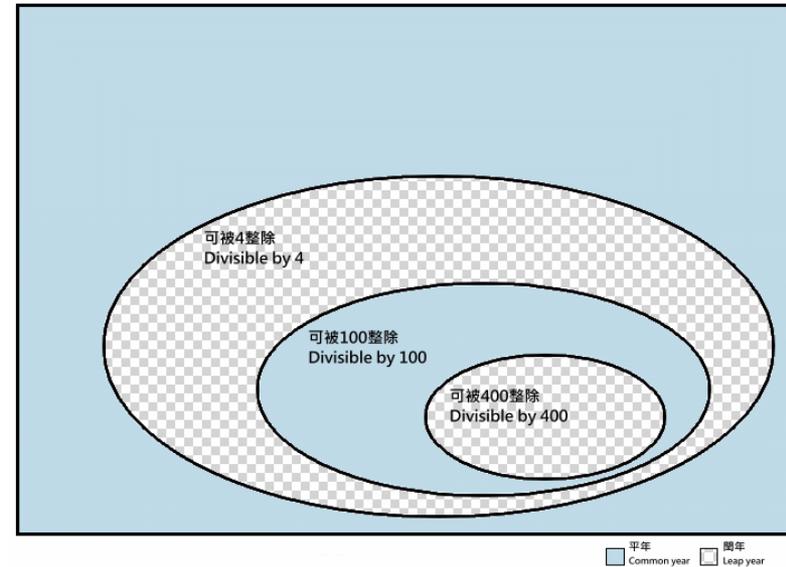


```
系統管理員: C:\Windows\System32\cmd.exe
C:\Program Files\Java\jdk1.8.0_144\bin>javac input_int.java
C:\Program Files\Java\jdk1.8.0_144\bin>java input_int
Please input your name:123456
You input nn=123456; first digit=6;higher digit=12345
C:\Program Files\Java\jdk1.8.0_144\bin>
中文 (繁體) - 新注音 2010 半 :
```

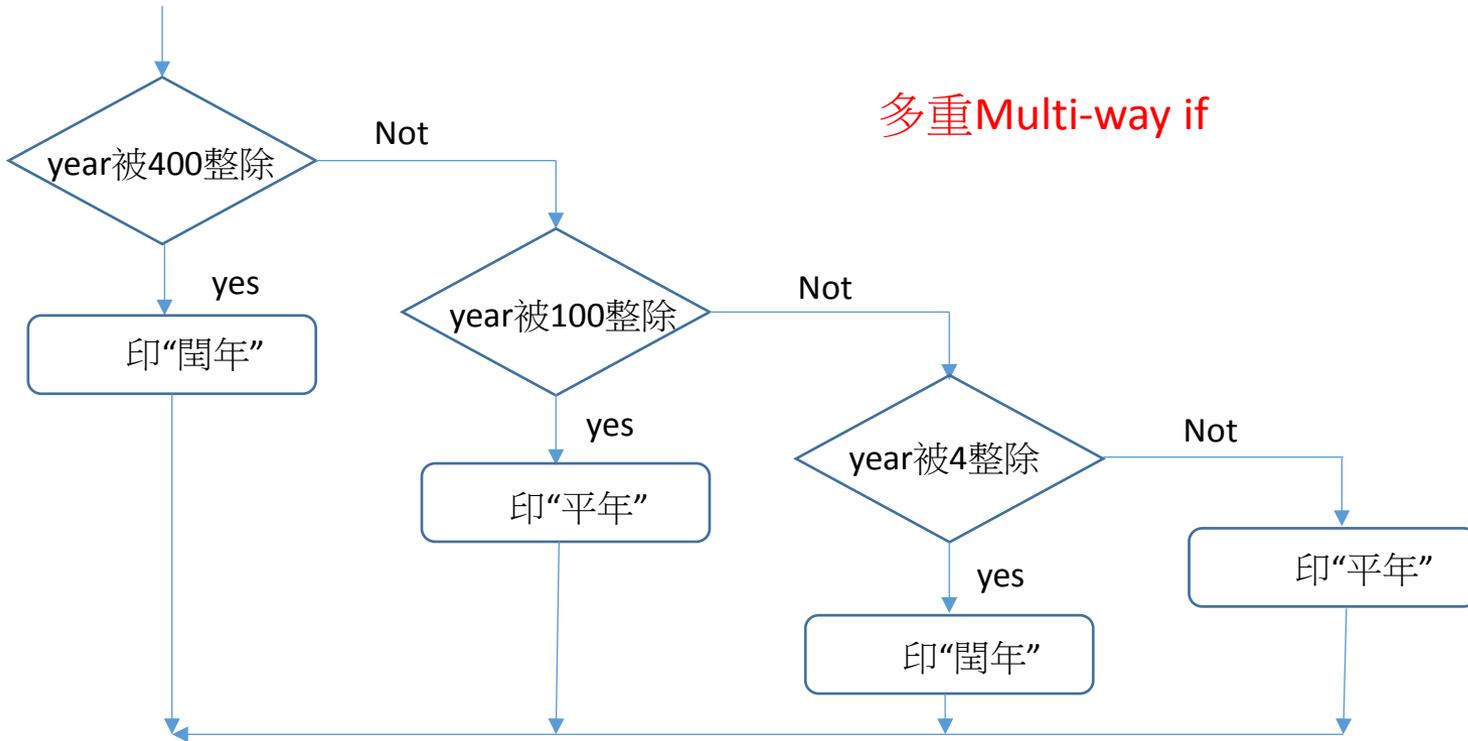
判斷閏年(leap year)、平年(common year)

閏年、平年

- 閏年:閏年是比普通年分多出一段時間的年分，目的是為了彌補人為規定的紀年與地球公轉產生的差異
- 格里高利曆(Calendarium Gregorianum)閏年規則如下：
 - 4的倍數是可能的。
 - 100的倍數是不可能的。
 - 400的倍數是可能的。
- 每逢閏年，2月分有29日，平年的2月分為28日
- 公元前之閏年出現在1, 5, 9, 13, ... BC，須將年份值減1再以「除以4」計算。（因為沒有公元0年這一年，所以公元前1, 2, 3, 4, ... 年應該是公元0, -1, -2, -3, ... 年，而公元前1, 5, 9, 13, ... 年為公元0, -4, -8, -12, ... 年，為4的倍數）
- <https://zh.wikipedia.org/wiki/%E9%97%B0%E5%B9%B4>



判斷閏年(leap year)、平年(common year)流程圖 條件式調整順序，是否正確？



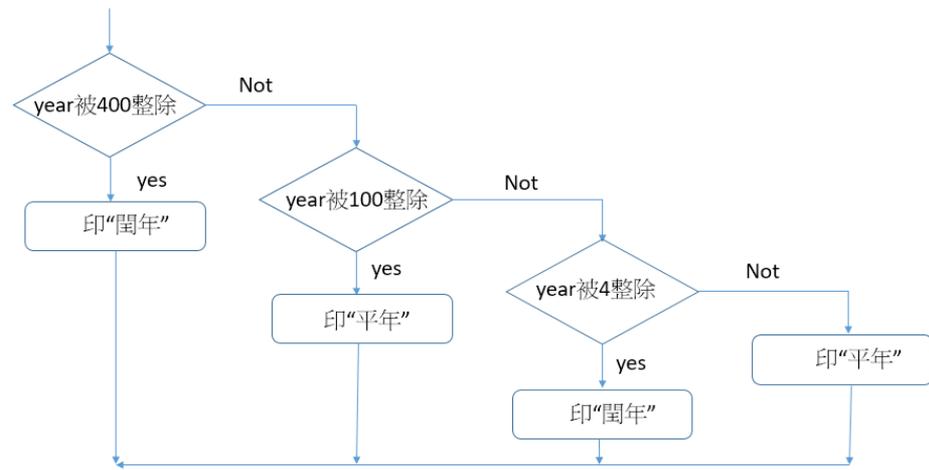
```

import java.util.Scanner;
//閏年:閏年是比普通年分多出一段時間的年分
public class leap_year_0a {

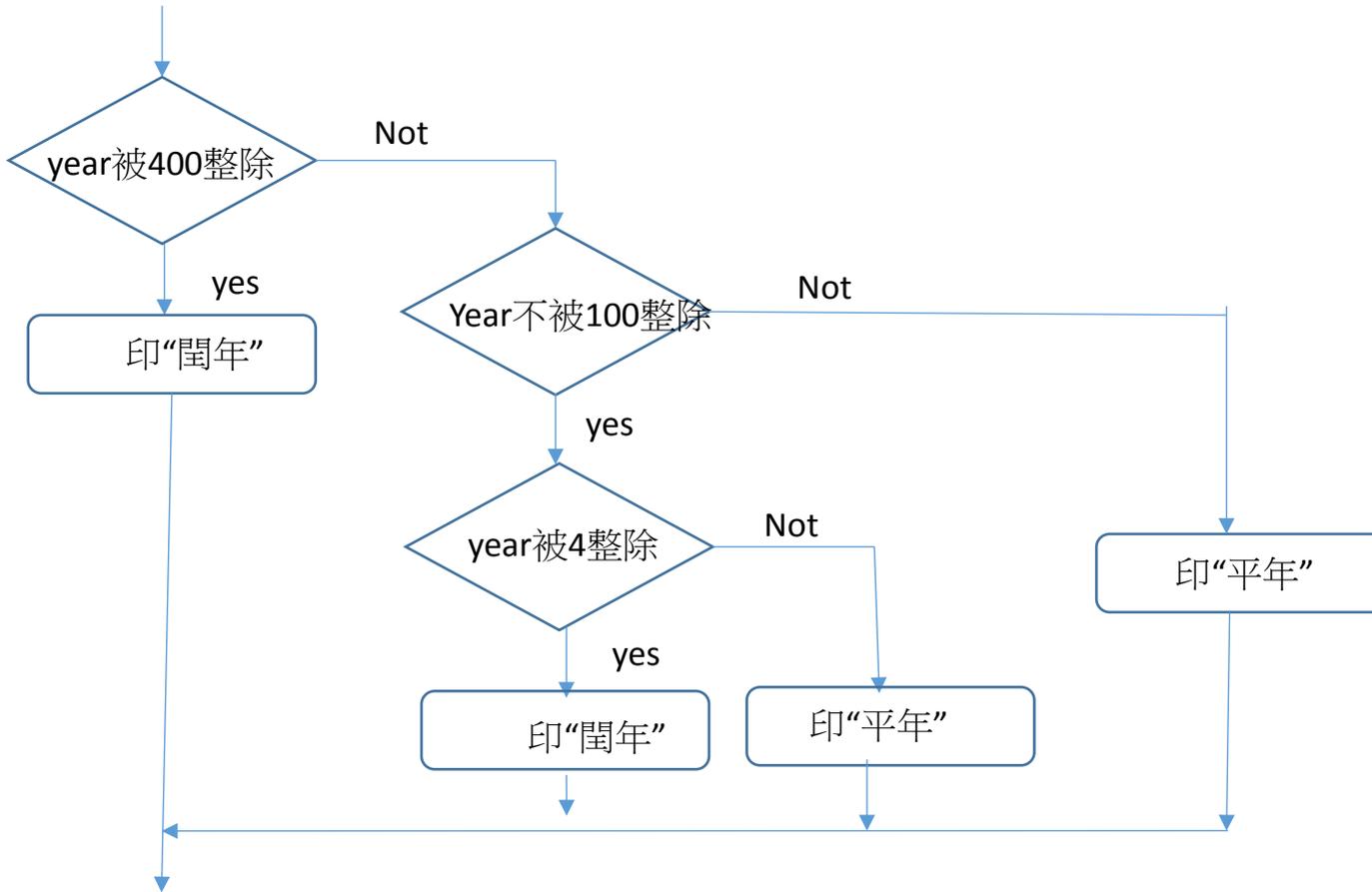
public static void main(String[] args) {
    String status;
    System.out.println("=====歡迎=====");
    Scanner input = new Scanner(System.in);
    int year=2000;
    System.out.print("輸入公元年：");
    year = input.nextInt();

    if (year%400==0)
        status = "閏年(leap year)!";
    else if (year%100==0)
        status = "平年(common year)!";
    else if (year%4==0)
        status = "閏年(leap year)!";
    else
        status = "平年(common year)!";
    if (year>=0)
        System.out.println("你輸入：公元"+year+"年，是 "+status);
    System.out.println("=====bye=====");
    }//main
} //class

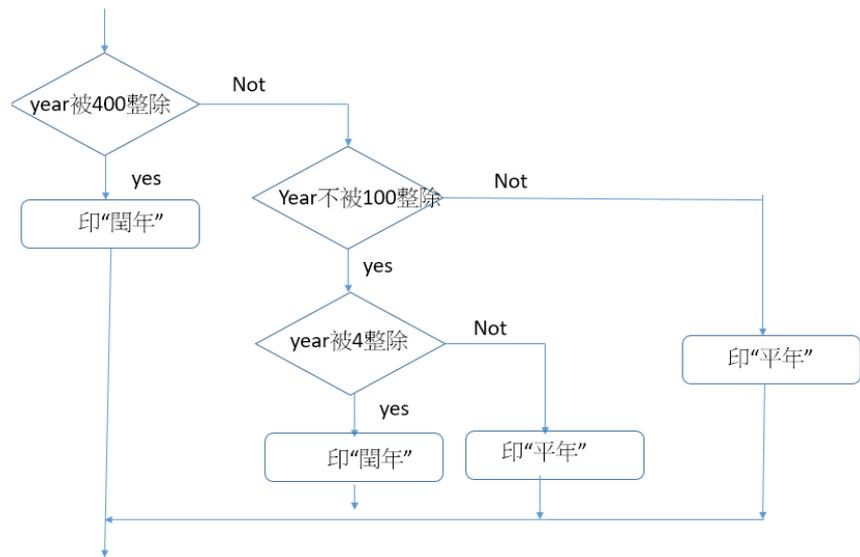
```

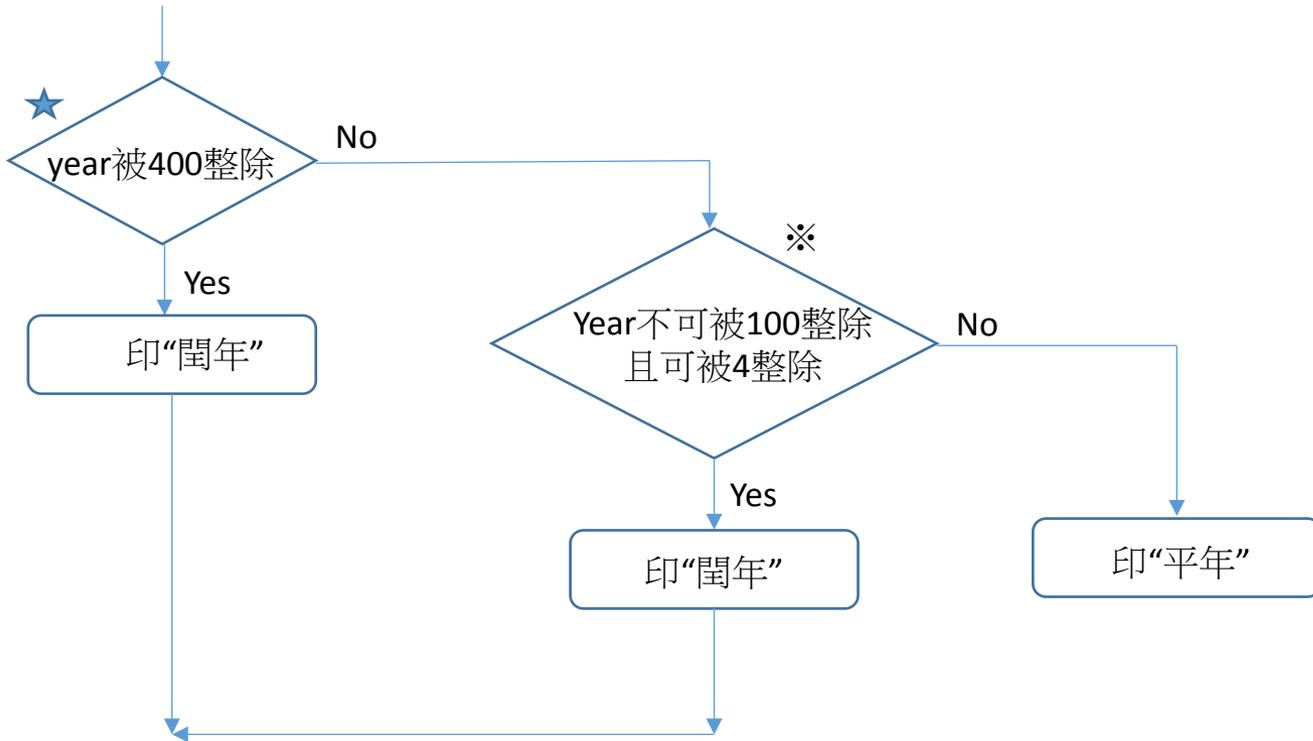


巢狀分支Nested if

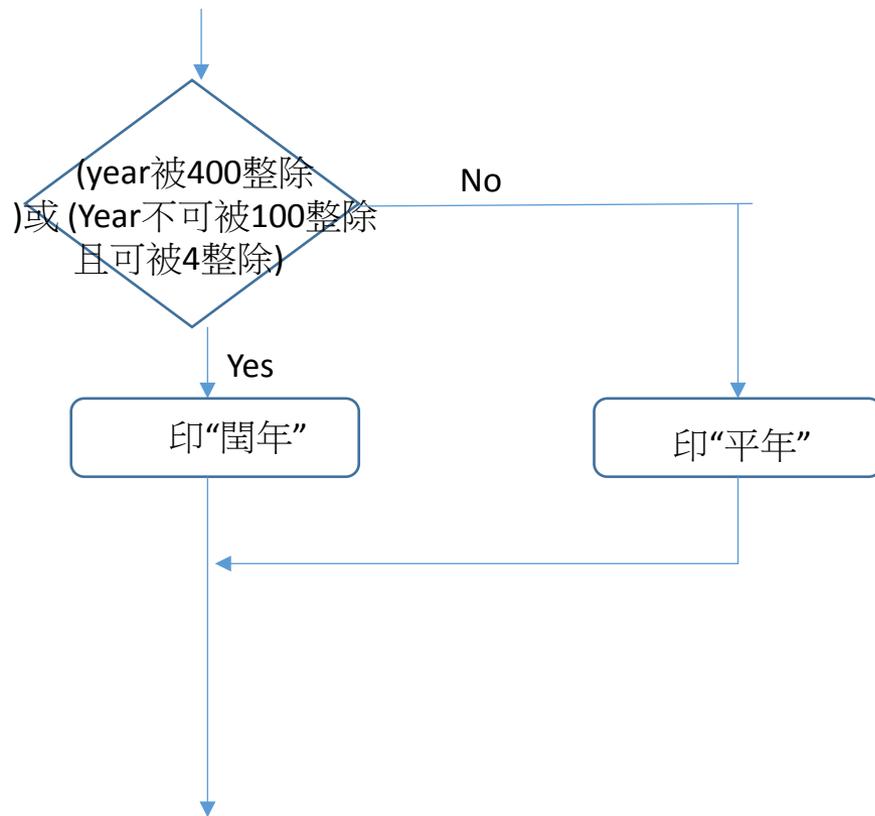


```
Scanner input = new Scanner(System.in);
int year=2000;
System.out.print("輸入公元年：");
year = input.nextInt();
if (year%400==0)
    status = "閏年(leap year)!";
else if (year%100!=0)
{
    if (year%4==0)
        status = "閏年(leap year)!";
    else //can be omitted
        status = "平年(common year)!"; //can be omitted
}
else
    status = "平年(common year)!";
if (year>=0)
    System.out.println("你輸入：公元"+year+"年，是 "+status);
else
    System.out.println("你輸入錯誤!");
```





二分支都可得閏年:或(or)



閏年規則轉成條件

- 閏年規則如下：
 - 是4的倍數
 - 不是100的倍數
 - 是400的倍數

```
if ((year%4==0 && year%100!=0) || year%400==0) {  
    status = "閏年(leap year)!";  
}  
else  
    status = "平年(common year)!";
```

```
System.out.print("輸入公元年(-1:結束)：");
year = input.nextInt();

if ((year%4==0 && year%100!=0)|| year%400==0) {
    status = "閏年(leap year)!";
}
else
    status = "平年(common year)!";
if (year>=0)
    System.out.println("你輸入：公元"+year+"年，是 "+status);
```

```
import java.util.Scanner;
//閏年:閏年是比普通年分多出一段時間的年分
public class leap_year_1 {

public static void main(String[] args) {
    String status;
    System.out.println("=====歡迎=====");
    Scanner input = new Scanner(System.in);
    int year=2000;
    while (year>=0) {
        System.out.print("輸入公元年(-1:結束):");
        year = input.nextInt();

        if ((year%4==0 && year%100!=0)|| year%400==0) {
            status = "閏年(leap year)!";
        }
        else
            status = "平年(common year)!";
        if (year>=0)
            System.out.println("你輸入:公元"+year+"年,是 "+status);
    }//while
    System.out.println("=====bye=====");
} //main
} //class
```

迴圈(loop)概念：
讓程式繞圈圈

迴圈(loop)

應用:

- 讓猜數字重複執行
 - 猜到對為止
 - 繼續猜
 - 但何時結束? 結束條件
- 重複計算BMI
- 重複判斷leap year(閏年)

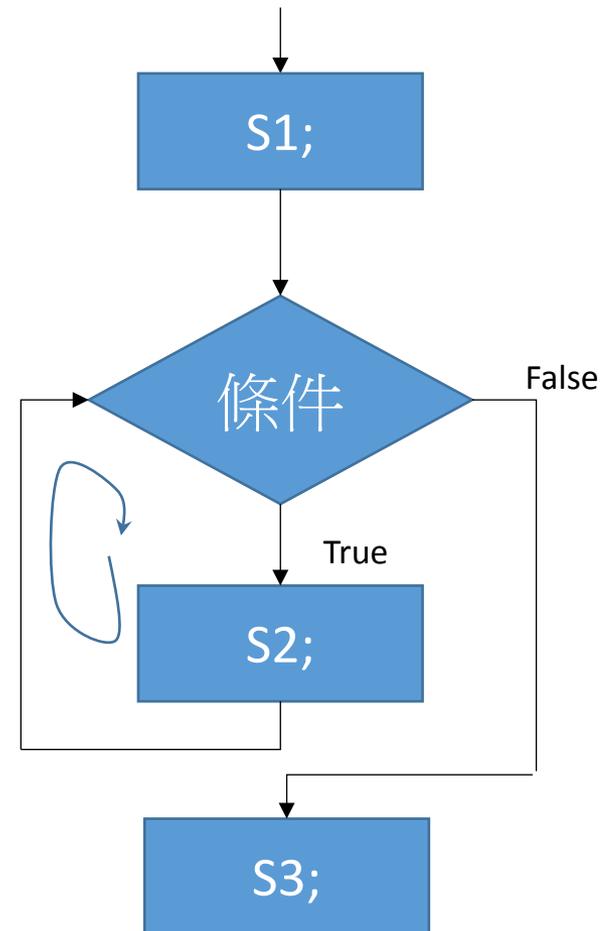
迴圈(loop)概念

- 重複不斷執行，直到條件不符合為止。

常見型態：for、while、do...while

- 一開始條件須符合，才能進入迴圈內部執行
- 無窮迴圈：條件永遠符合
- 有些環境必須是無窮迴圈：
O.S(作業系統)
* CPU特性：重複執行(耐性)

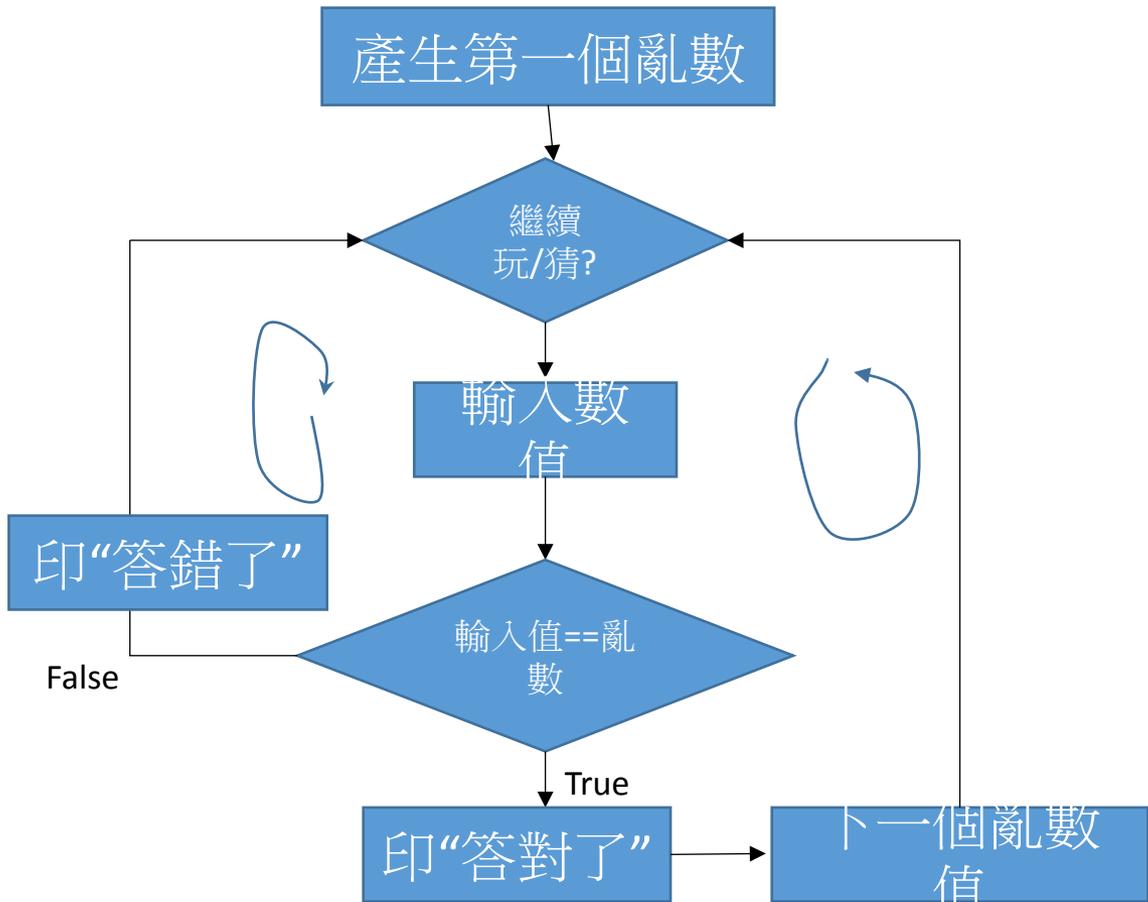
```
S1;  
while (條件) {  
    S2;  
}  
S3;
```



猜數字遊戲

- 讓猜數字重複執行
- 猜到對為止
- 繼續猜
- 但何時結束? 結束條件

建立亂數類別SecureRandom之物件
以此類別nextInt函數(方法)產生1~100隨機整數
猜數字:讓使用者輸入數值
判斷答對與否,且輸出回饋信息



```

C:\Windows\System32\cmd.exe - ja...
亂數為8
E:\java-2017\9-22\code>javac GuessN_3.java
E:\java-2017\9-22\code>java GuessN_3
猜數字(1~100) : 33
猜的太大囉
猜數字(1~100) : 22
猜的太大囉
猜數字(1~100) : 12
猜的太大囉
猜數字(1~100) : 10
猜的太大囉
猜數字(1~100) : 8
猜的太大囉
猜數字(1~100) : 5
猜的太小囉
猜數字(1~100) : 7
恭喜猜對!
猜數字(1~100) :
  
```

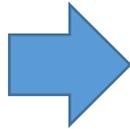
讓猜數字重複執行

```
public class GuessN_2 {
    public static void main(String[] args) {

        SecureRandom sr = new SecureRandom();
        Scanner input = new Scanner(System.in);

        int a = 0;
        int b = 0;
        a = sr.nextInt(100)+1;

        System.out.print("猜數字(1~100) : ");
        b = input.nextInt();
        if (a!= b) { //nested if
            if (a > b)
                System.out.println("猜的太小囉\n");
            else
                System.out.println("猜的太大囉\n");
            System.out.println("亂數為"+a);
        } //if
        else
            System.out.println("恭喜猜對!");
    } //main
} //class
```



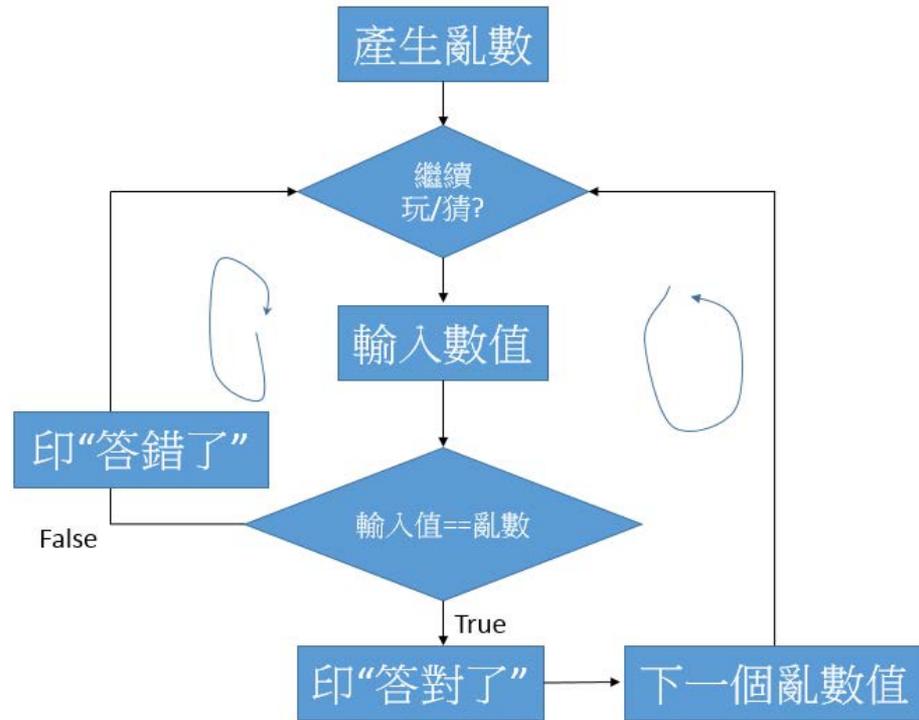
```
public class GuessN_3 {
    public static void main(String[] args) {

        SecureRandom sr = new SecureRandom();
        Scanner input = new Scanner(System.in);

        int a = 0;
        int b = 100;
        a = sr.nextInt(100)+1;
        while (b>0) {
            System.out.print("猜數字(1~100) : ");
            b = input.nextInt();
            if (a!= b) { //nested if
                if (a > b)
                    System.out.println("猜的太小囉");
                else
                    System.out.println("猜的太大囉");
                //System.out.println("亂數為"+a);
            } //if
            else {
                System.out.println("恭喜猜對!\n");
                a = sr.nextInt(100)+1;
            }
        } //while
    } //main
} //class
```

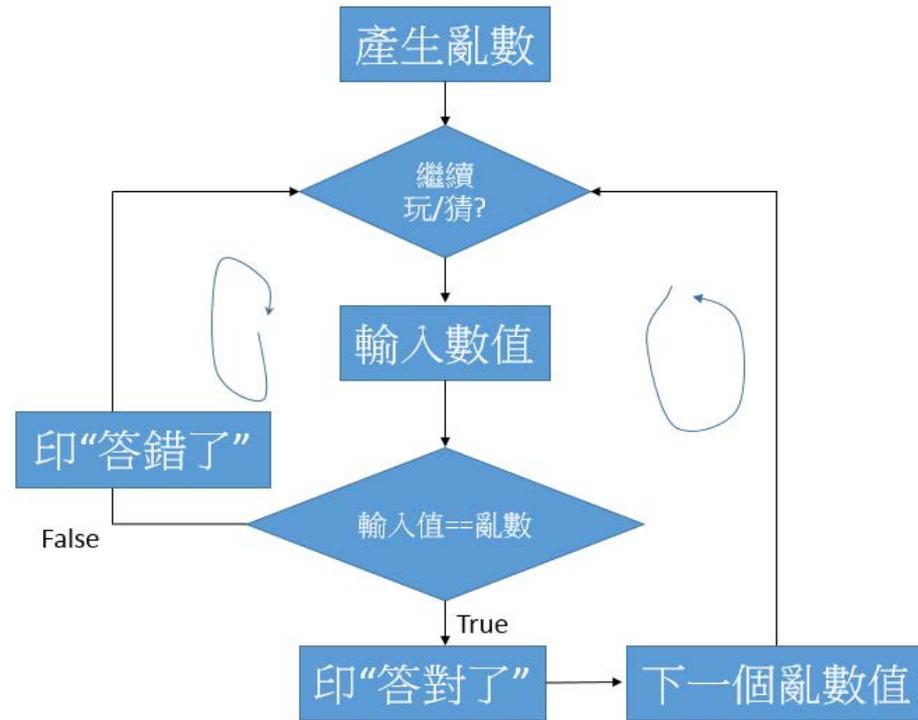
程式與流程圖對應

```
public class GuessN_3 {  
    public static void main(String[] args) {  
  
        SecureRandom sr = new SecureRandom();  
        Scanner input = new Scanner(System.in);  
  
        int a = 0;  
        int b = 100;  
        a = sr.nextInt(100)+1;  
        while (b>0) {  
            System.out.print("猜數字(1~100) : ");  
            b = input.nextInt();  
            if (a!= b) { //nested if  
                if (a > b)  
                    System.out.println("猜的太小囉");  
                else  
                    System.out.println("猜的太大囉");  
                //System.out.println("亂數為"+a);  
            } //if  
        } else {  
            System.out.println("恭喜猜對!\n");  
            a = sr.nextInt(100)+1;  
        }  
    } //while  
} //main  
} //class
```



迴圈會執行? Why?

```
import java.security.SecureRandom;
import java.util.Scanner;
public class GuessN_3 {
    public static void main(String[] args) {
        SecureRandom sr = new SecureRandom();
        Scanner input = new Scanner(System.in);
        int a = 0;
        int b = 0;
        a = sr.nextInt(100)+1;
        while (b>0) {
            System.out.print("猜數字(1~100) : ");
            b = input.nextInt();
            if (a!= b) { //nested if
                if (a > b)
                    System.out.println("猜的太小囉");
                else
                    System.out.println("猜的太大囉");
            } //if
        } else {
            System.out.println("恭喜猜對!\n");
            a = sr.nextInt(100)+1;
        } //else
    } //while
} //main
} //class
```



習題

- 習題1:輸入矩形長、寬(整數)資料，求面積及邊長、判斷為正方形或長方形，但須能重複執行，直到長或寬輸入為 ≤ 0
 - 正方形或長方形條件?
 - 如何解決? 請規畫其過程 (解法)且畫流程圖
- 習題2:輸入民國幾年，判斷該年是閏年或平年，須能重複執行，直到輸入年份為 ≤ -100