

請大家準時上課

實作輔導 2

- 日期: 3/24 (星期六) 09:10~16:00
- 地點: 臺北市立大學 臺北市中正區愛國西路一號 (中正紀念堂站7號出口)
- 公誠樓三樓 G316 電腦教室 ([資訊科學系](#))
- 可自行攜帶筆電
- 目標: 協助習題、安裝java 環境、path設定
- 參加者: 請email 給laiahfur@gmail.com 或直接到輔導地點
- 下次預定: 4/14 (星期六) 09:10~16:00

搶答

1. 宣告 W 為整數變數請寫此敘述
(指令)? 宣告目的為何?

搶答

2. 將W整數變數內容放入80；

搶答

3. 將W整數變數原本內容再增加15？

搶答

4. 將w的整數變數內容放入(set)z整數變數？

搶答

5. 將w及z整數變數內容相乘後，放入y整數變數內容？

搶答

6. 將W整數變數內容清除為0？

搶答

7. 下列哪些變數名稱是不合法的？(複選)

(1) $x-y$

(2) $x+y$

(3) 3pig

(4) $x4B$

(5) $x*y$

搶答

8.那些變數的名稱不合法invalid?

(1) a_2

(2) 123pp

(3) p+q

(4) f%3

搶答

9. $x+3=z*k$;

上述Bug在何處？如何修改才會正確？

解答

1. 宣告w為整數變數，請寫此敘述(指令)? 宣告目的為何?

- Ans: `int w;` 宣告目的在於請求系統依資料型態配給一塊記憶體

2. 將w整數變數內容放入80;

- Ans: `w=80;`

3. 將w整數變數原本內容再增加15?

- Ans: `w=w+15;`

4. 將w的整數變數內容放入(set)z整數變數? (z已宣告)

- Ans: `z=w;`

5. 將w及z整數變數內容相乘後，放入y整數變數內容? (y已宣告)

- Ans: `y=w*z;`

6. 將w整數變數內容清除為0?

- Ans: `w=0;`

解答

7. 下列哪些變數名稱是不合法的？(複選)

(1) x-y

(2) x+y

(3) 3pig

(4) x4B

(5) x*y

- Ans: 1, 2, 3, 5

8. 那些變數的名稱不合法(invalid)?

(1) a_2

(2) 123pp

(3) p+q

(4) f%3

- Ans: 2, 3, 4

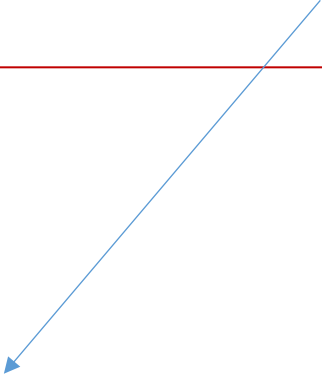
9. $x+3=z*k$;

上述Bug在何處？如何修改才會正確？

- Ans: $x_3=z*k$; 或 $x3=z*k$; 或 $x=z*k$;

搶答：變數內容追蹤，印出結果？

```
int X=10, Y=4, Z=15;  
X=X+1;  
X=X+Y;  
Z=(Y*Y)%(X-1);  
System.out.println("X="+X+", Z="+Z);
```



思考三分鐘後，請在ADOBE connect 右下角聊天區作答！

分支 (selection , branch) 程式會轉彎

臺北市立大學 [資訊科學系\(含碩士班\)](#)

賴阿福

變數(variable)概念複習

- 何謂變數?執行時，變數存在何處?
- 變數有何用途?
- 變數有哪些型態(data type)?
- 為何需有不同型態?
- 變數如何宣告?為何要宣告?
- 變數內容如何給初值?
- 變數內容如何存、取:存入(store, set)、取出(get, retrieve)

變數概念複習

- 何謂變數?變數存在何處?
 - 變數代表程式要使用一塊RAM，存在RAM
- 變數有何用途?
 - 紀錄、存放變動資料
- 變數有哪些型態?
 - 整數、浮點數、字串、boolean、char
- 為何需有不同型態?
 - 不同型態有不同用途、值域
- 變數如何宣告?
 - `int p,q,r;` →
- 變數內容如何給初值?
 - `int p=9,q;`
- 變數內容如何存、取:存入(透過=)、取出
 - 存入 `p=100;`
 - 取出 `System.out.print("p="+p);`

```
int p;  
int q;  
int r;
```

比較 (連連看)

- Math

- $x=x+1$
- $y=x+1$
- $x-1=y+8$

方程式

指定運算

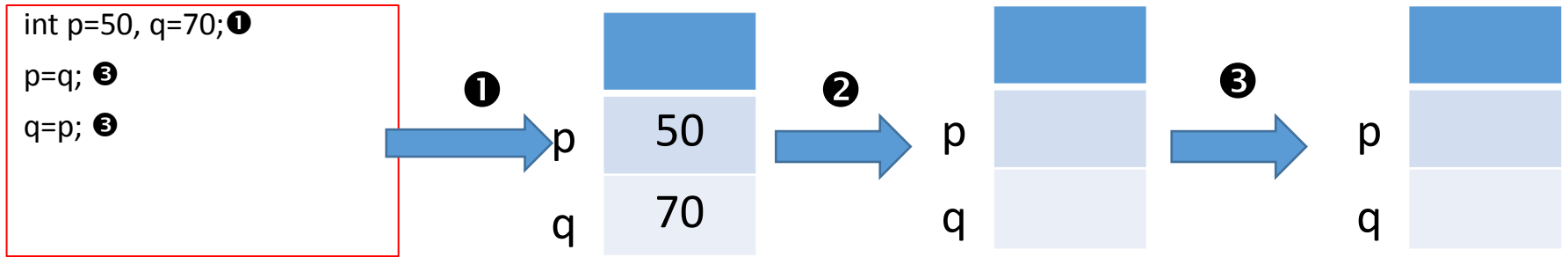
條件式(等於)

- Java

- $x=x+1;$
- $x-1=y+8$
- $\text{If } (y==x+1)$
- $\text{If } (x-1==y+8)$

不存在(bug)

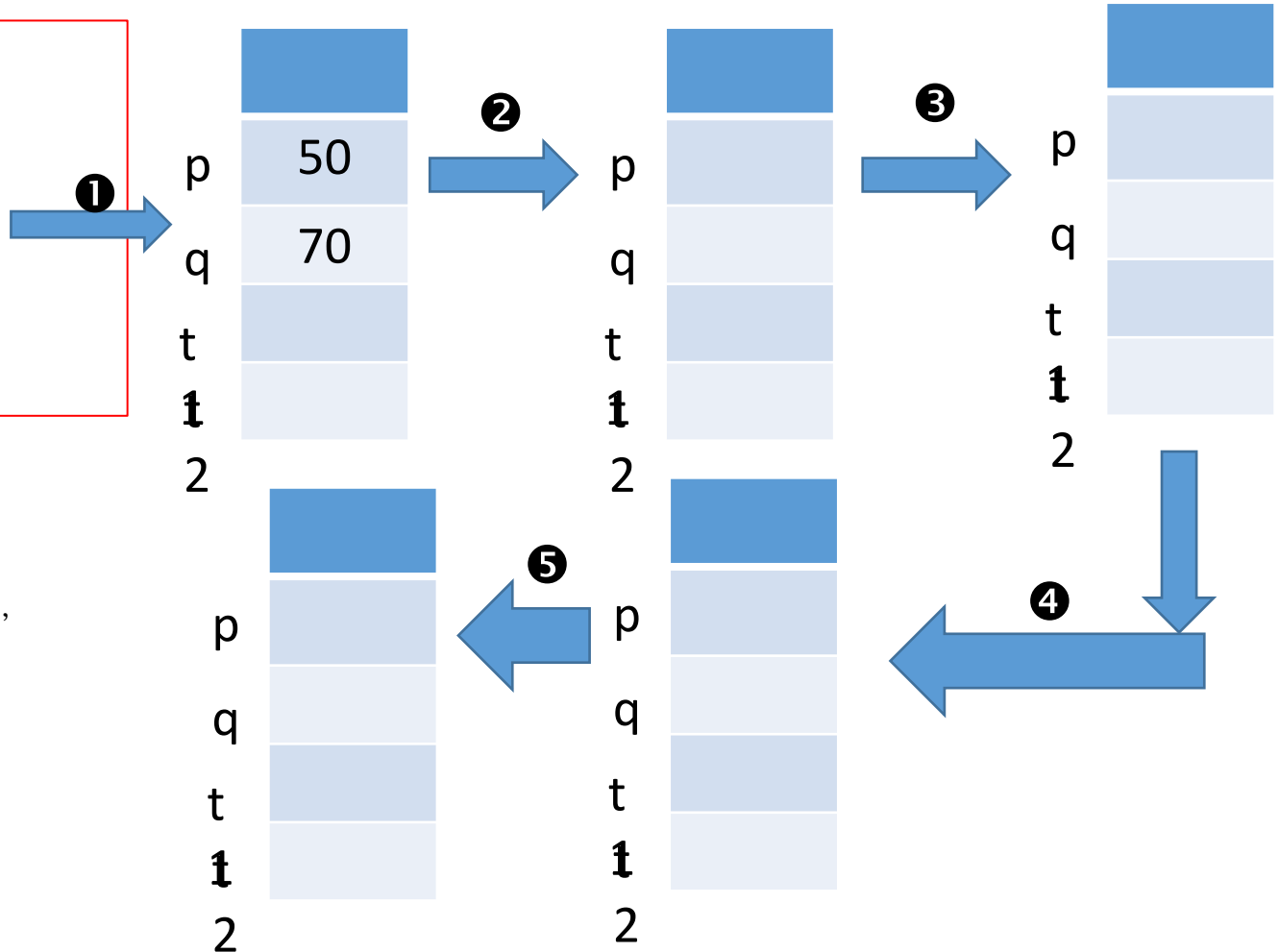
變數內容互換 (swap, exchange)



思考三分鐘!將每敘述造成改變，填入RAM
CORRECT? INCORRECT?

變數內容互換1 (swap, exchange)

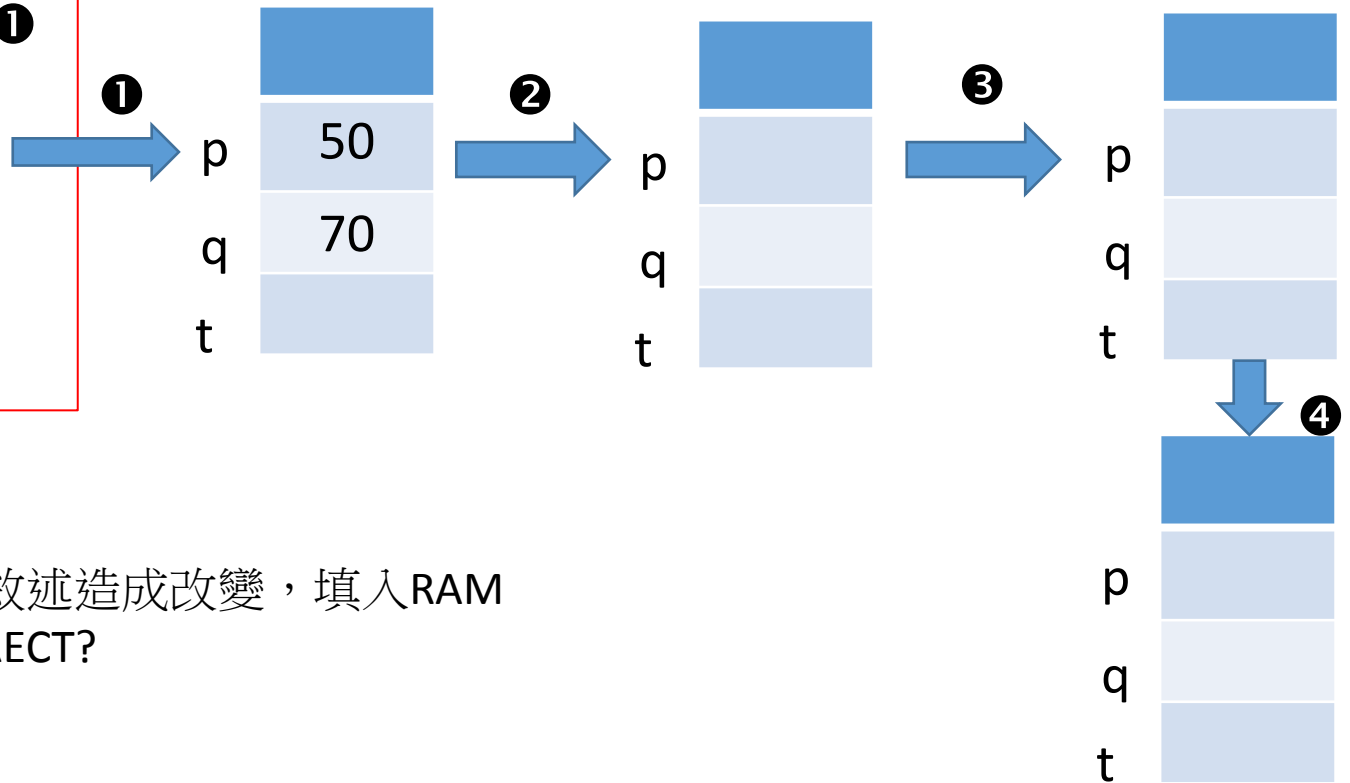
```
int p=50, q=70, t1,t2; ❶  
t1=p; ❷  
t2=q; ❸  
p=t2; ❹  
q=t1; ❺
```



思考三分鐘!將每敘述造成改變，
填入RAM
CORRECT? INCORRECT?

變數內容互換2 (swap, exchange)

```
int p=50, q=70, t; ❶  
t=p; ❷  
p=q; ❸  
q=t; ❹
```



思考三分鐘!將每敘述造成改變，填入RAM
CORRECT? INCORRECT?

少，是王道！ The less, the better!

- 變數用得越少越好？
- 敘述(code)寫得越少越好，Why?
 - 資源: CPU time, RAM
 - For Portable devices

從循序(sequence)結構到
分支/選擇

(Branch/selection)結構:

程式會轉彎

再談分支/選擇

條件式

判斷成績(SC)及格與否:

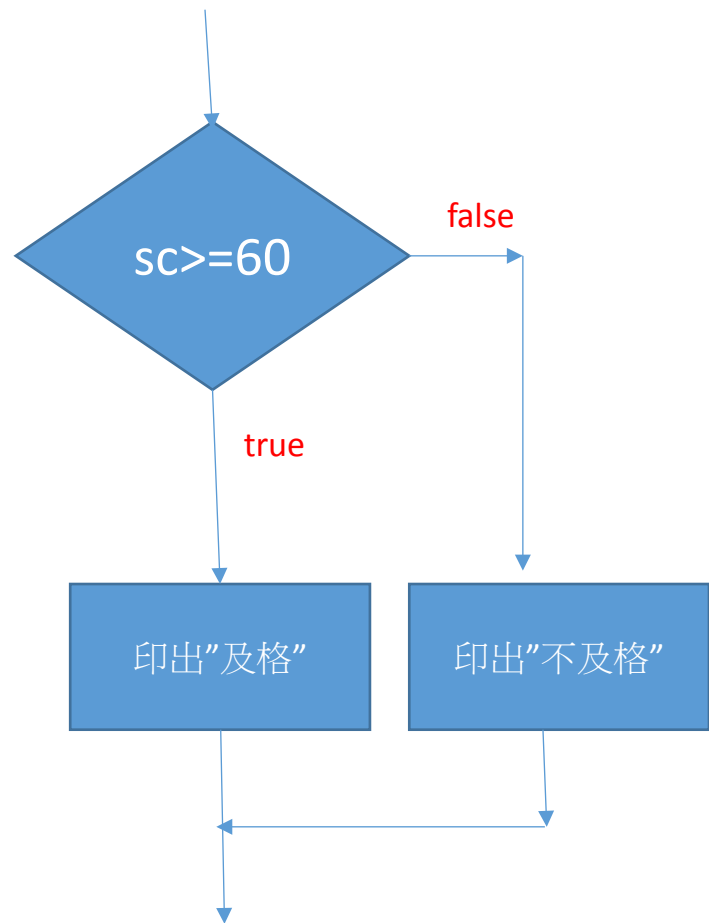
- 條件為真(true)，印出“及格”
- 否則，印出“不及格”
- $sc \geq 60$ 條件式是屬於關係運算式

```
if (sc >= 60)
```

```
    印出“及格”
```

```
else
```

```
    印出“不及格”
```

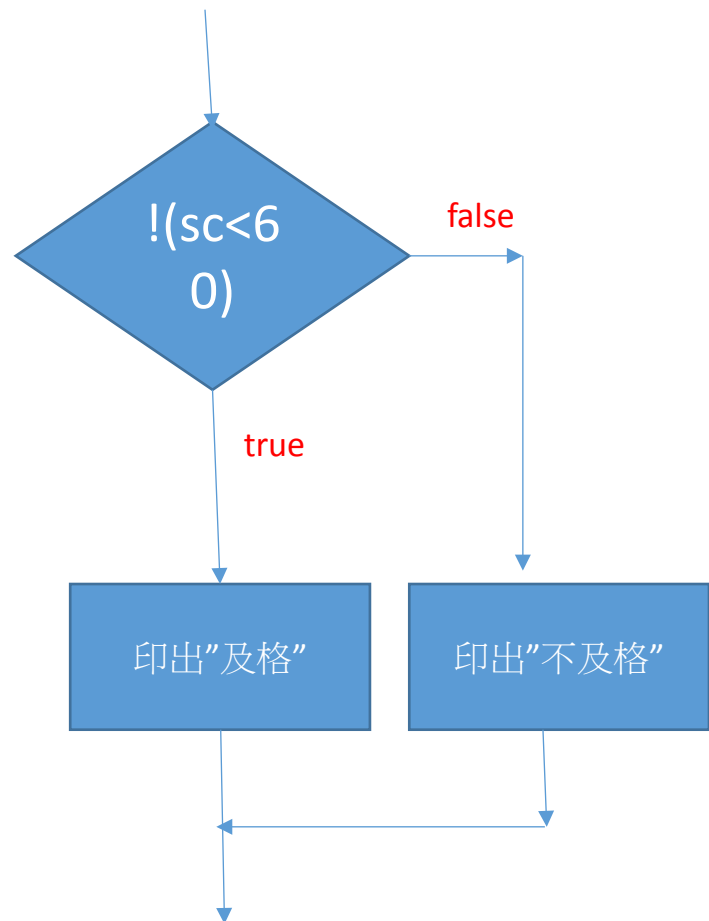


條件式

- 條件為真(true)，印出“及格”
- 否則，印出“不及格”

- !(sc<60) 條件式
 - (sc<60)是關係運算式
 - !是反向邏輯(not)

```
if !(sc<60)
    印出“及格”
else
    印出“不及格”
```



條件式

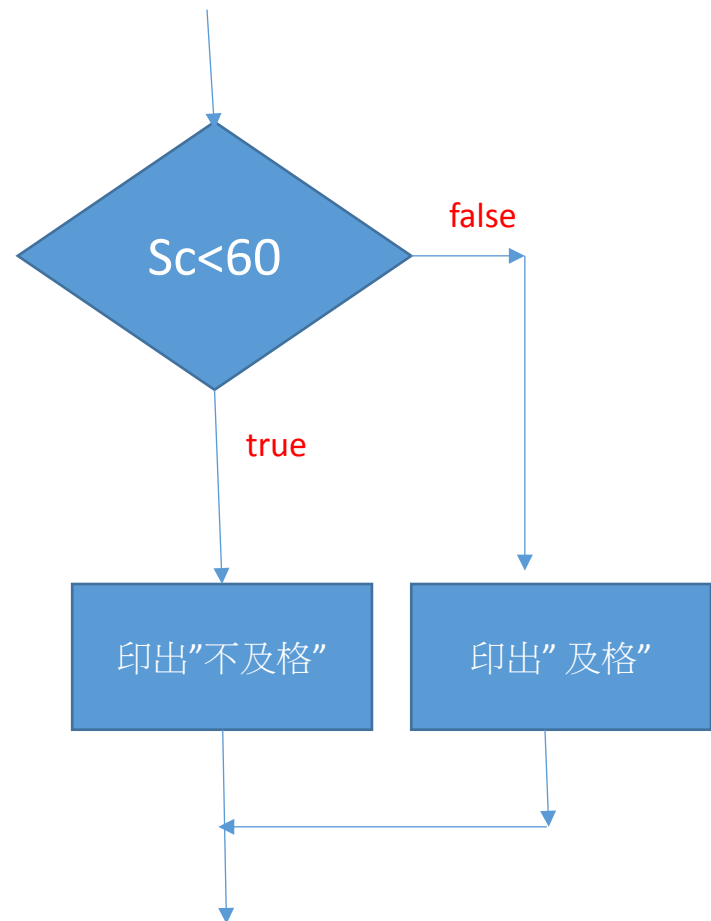
- 條件為真(true)，印出“不及格”
- 否則，印出“及格”
- $sc < 60$ 條件式是屬於關係運算式

if ($sc < 60$)

 印出“不及格”

else

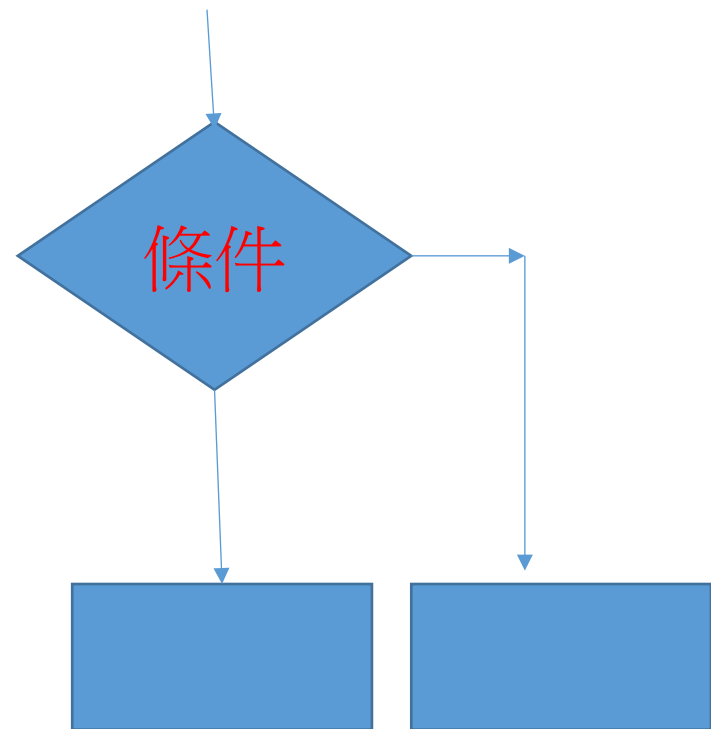
 印出“及格”



分支靠“條件”

- 條件

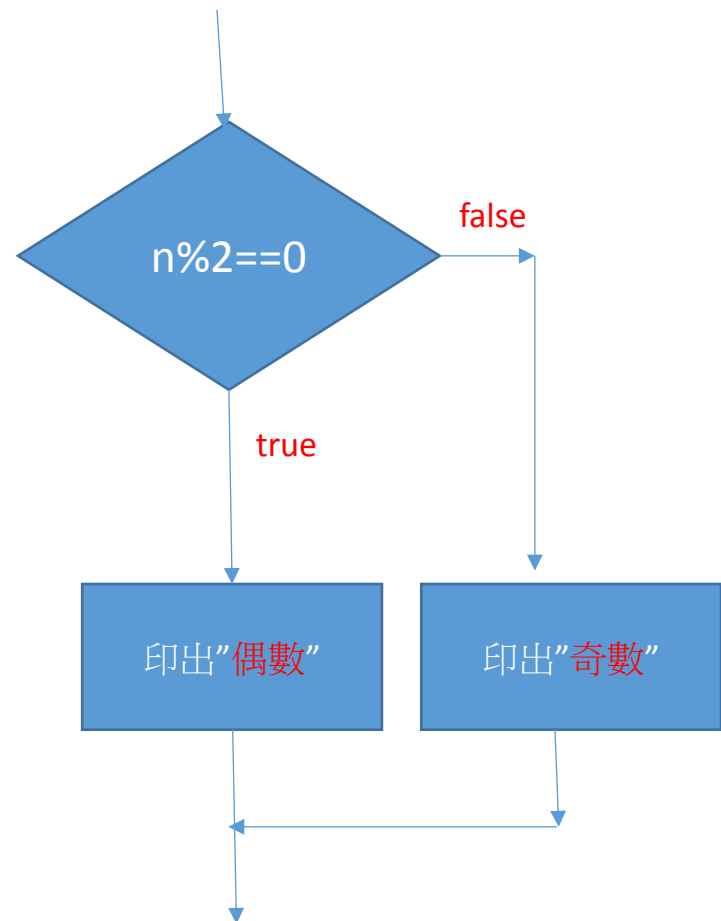
- 關係運算式
 - == 等於
 - != 不等於
 - >、<、<=、>=
- 邏輯運算式
 - && :and且
 - || :or或
 - !:not反/非
- 結果:
 - True:真
 - False:偽



判斷偶數、奇數?

- 如果條件為真(true)，印出“偶數”
- 否則，印出“奇數”
- $n\%2==0$ 條件式是屬於關係運算式

```
if (n%2==0)  
    印出“偶數”  
else  
    印出“奇數”
```



判斷偶數、奇數?

```
import java.util.Scanner;
public class evennumber {
public static void main(String[] args) {
    Scanner input = new Scanner(System.in);

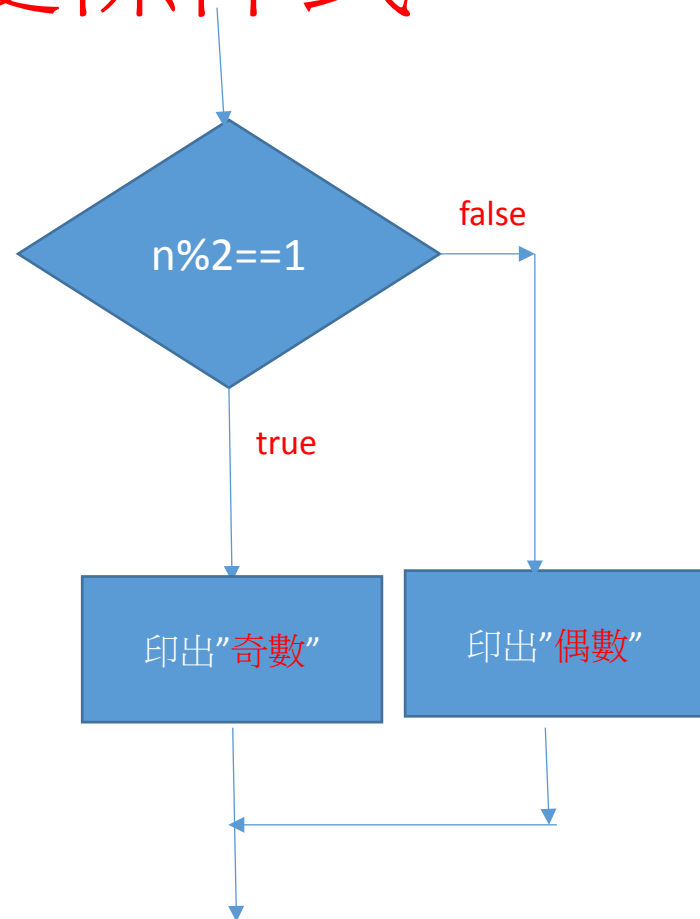
    System.out.print("輸入數值：");
    int n = input.nextInt();
    if (n%2==0)
        System.out.println(n+"是偶數!");
    else
        System.out.println(n+"是奇數!");
    }
}
```

判斷偶數、奇數? 改變條件式

如果條件為真(true)，印出“奇數”
否則，印出“偶數”

- $n\%2==1$ 條件式是屬於關係運算式

```
if (n%2==1)
    印出“奇數”
else
    印出“偶數”
```



邏輯運算式

A	B	A and B A && B	A or B B	Not A !A
F	F	F	F	T
F	T	F	T	
T	F	F	T	F
T	T	T	T	

條件:邏輯運算式 (立即搶答)

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

- `&&` : and (及)
- 符合條件數值範圍?

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

- `||` : or (或)
- 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?

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

- `!` : not (反)
- 符合條件數值範圍?

條件:邏輯運算式 (立即搶答)

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

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

if (nn>=100 || nn<=999)

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

if (nn<100 || nn>999)

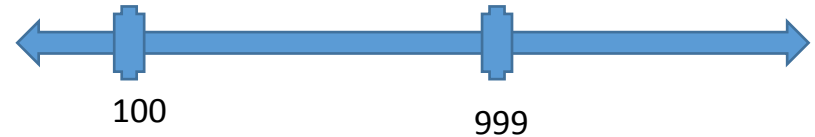
- 符合條件數值範圍? 100~999之外

if (nn<100 && nn>999)

- 符合條件數值範圍? 無此

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

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



不只算BMI，再探討條件式判斷體
位BMI

判斷體位是否標準?

選擇/分支
(selection)

```
if (BMI >= 18.5 && BMI < 24)
```

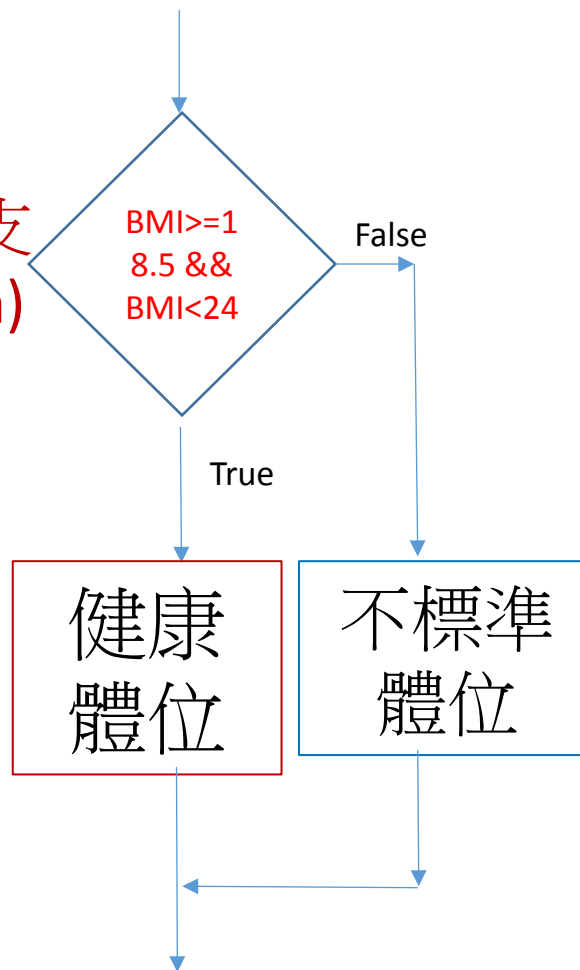
```
    System.out.print("健康體位 Normal\n\n");
```

```
else
```

```
    System.out.print("不標準體位\n\n");
```

```
//source file: BMI-2a
```

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



BMI診斷分成二層次

```
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 && bmi < 24)

        status = "正常Normal";

    else

        status = "不標準體位";

    System.out.println("BMI : "+bmi+" , 狀態: "+status);

} //main

} //class
```

將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

比較

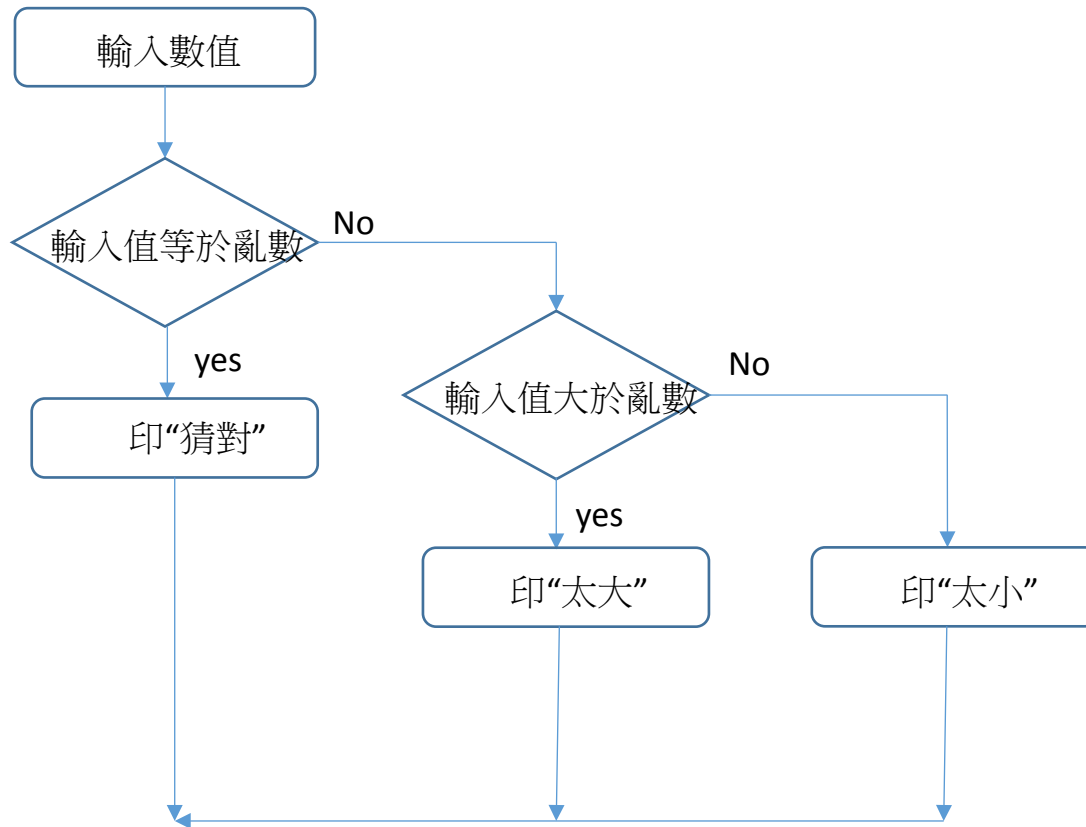
```
if (bmi < 18.5)
    status = "體重過輕Underweight";
else if (bmi >= 18.5 && bmi < 24)
    status = "正常Normal
(18.5 <= BMI < 24)";
else
    status = "過重Overweight";
```

```
if (bmi < 18.5)
    status = "體重過輕Underweight";
else if (bmi < 24)
    status = "正常Normal
(18.5 <= BMI < 24)";
else
    status = "過重Overweight";
```

猜數字遊戲

- 運用亂數產生整數隨機數值
- 猜數字:讓使用者輸入數值
- 判斷答對與否

Nested if



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

亂數

- 運用亂數類別SecureRandom，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

- 多重分支Multi-way if、巢狀分支Nested 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");
```

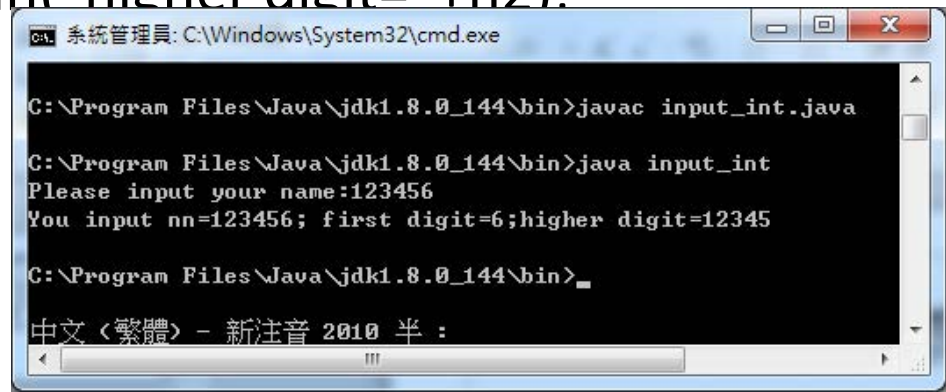
```
if (a == b)
    System.out.println("恭喜猜對!");
else { //nested if
    if (a > b)
        System.out.println("猜的太小囉\n");
    else
        System.out.println("猜的太大囉\n");
    System.out.println("亂數為"+a);
} //else
```

```
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!");
```

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
- 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，應重複直至使用者不想玩。
 - **Loop** 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);
```



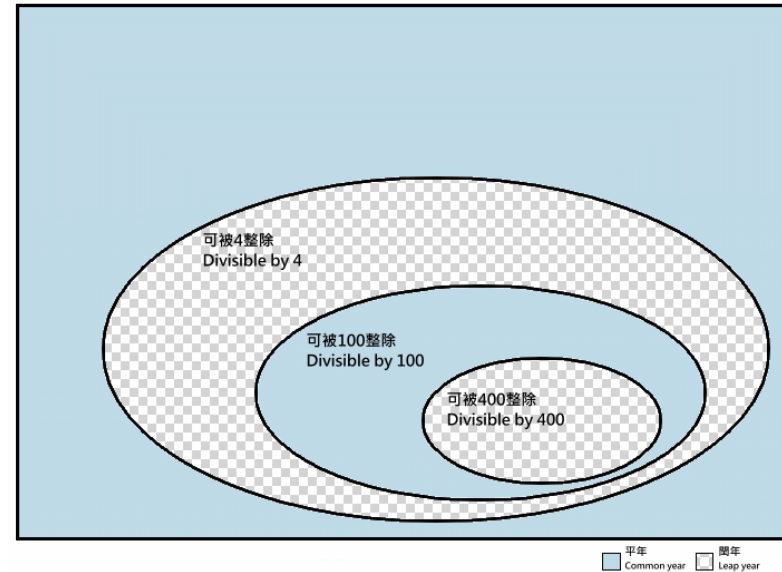
```
C:\Program Files\Java\jdk1.8.0_144\bin>javac input_int.java  
C:\Program Files\Java\jdk1.8.0_144\bin>java input_int  
Please input your name:123456  
You input nn=123456; first digit=6;higher digit=12345  
C:\Program Files\Java\jdk1.8.0_144\bin>_
```

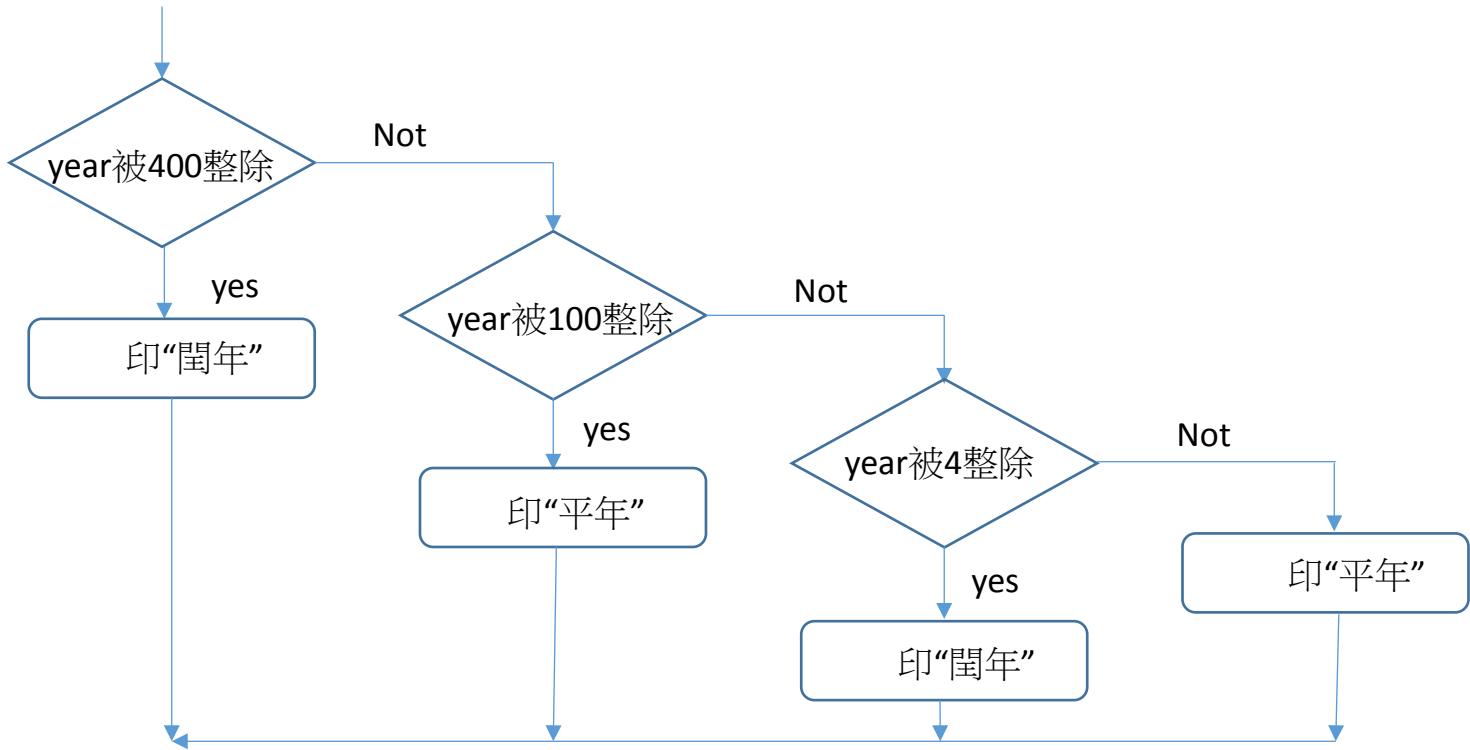
中文 (繁體) - 新注音 2010 半 :

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>





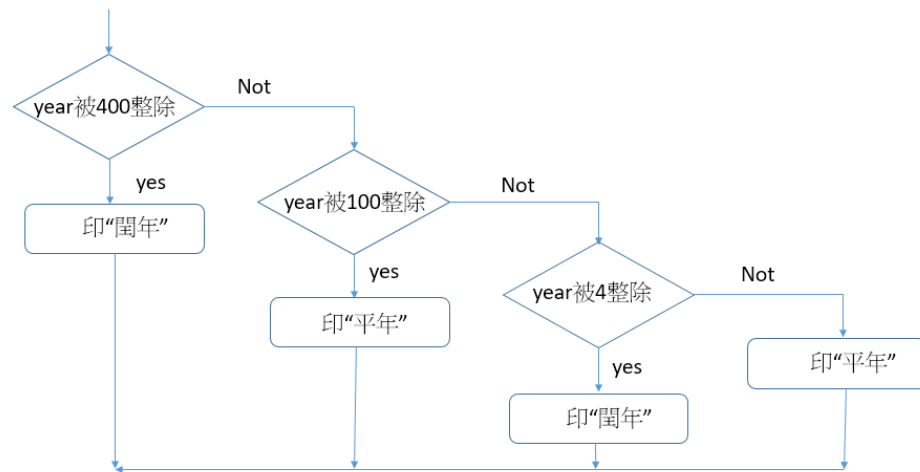
```

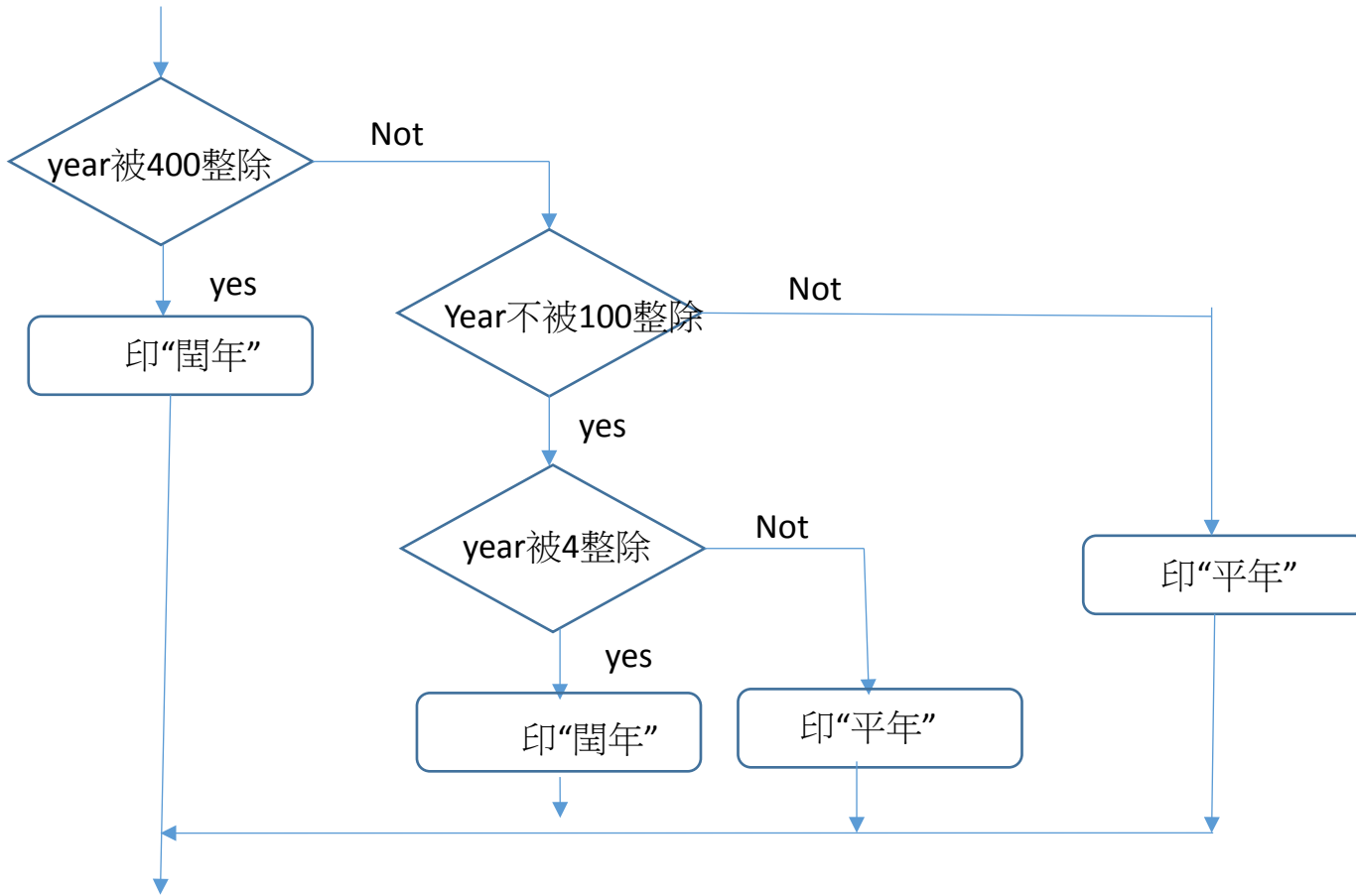
import java.util.Scanner;
//閏年:閏年是比普通年分多出一段時間的年分
public class leap_year_0a {

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

```

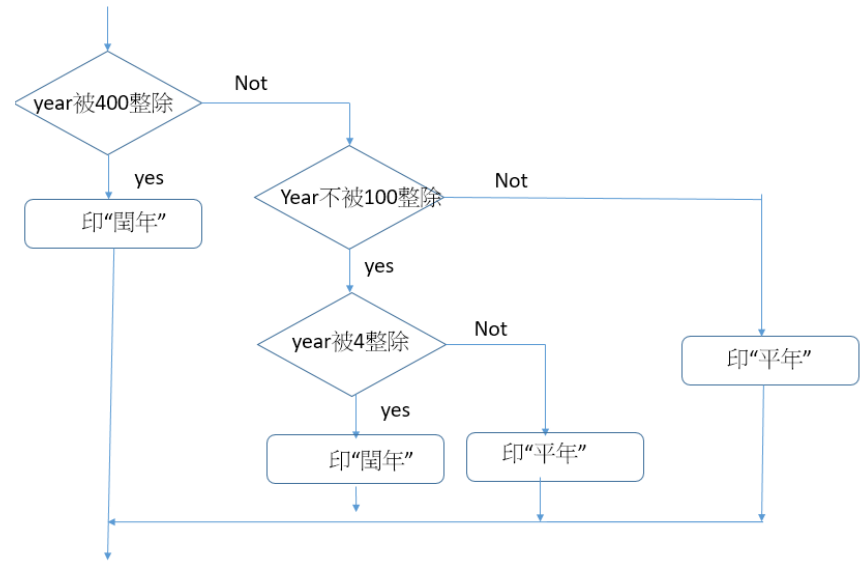


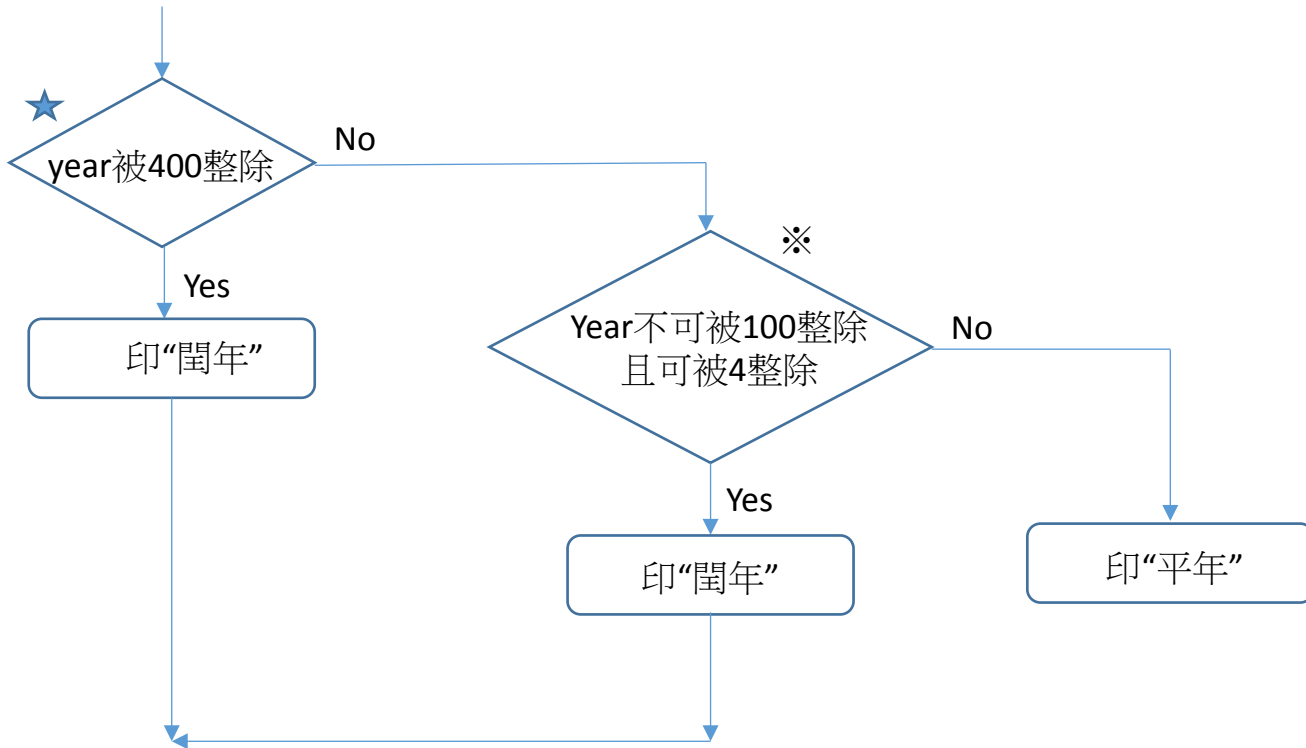


```

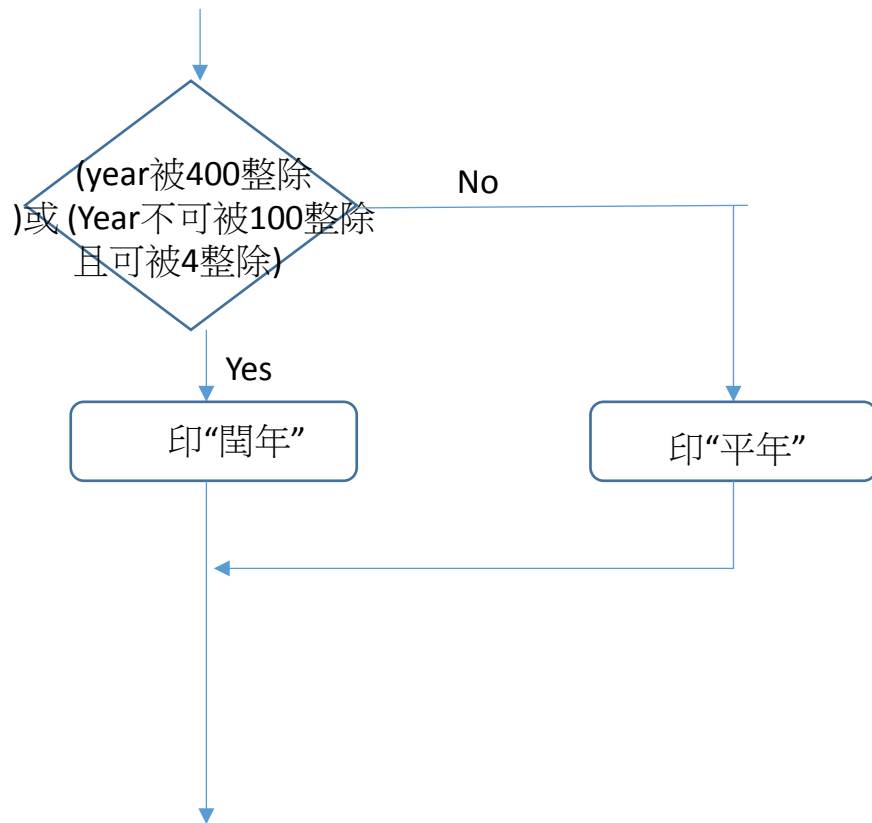
Scanner input = new Scanner(System.in);
int year=2000;
System.out.print("輸入公元年：");
year = input.nextInt();
if (year%400==0)
    status = "閏年(leap year)!";
else if (year%100!=0)
{
    if (year%4==0)
        status = "閏年(leap year)!";
    else //can be omitted
        status = "平年(common year)!"; //can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

習作:輸入收縮壓及舒張壓判斷血壓類型(依血壓定義)

說明: 簡化題目(輸入收縮壓，判斷血壓類型)

輸入收縮壓(mmHg): 145

輸入舒張壓(mmHg): 95

診斷結果:第一期高血壓

輸入收縮壓(mmHg): 161

輸入舒張壓(mmHg): 95

診斷結果:第二期高血壓

分類	收縮壓 (mmHg)		舒張壓(mmHg)
正常	<120	和	<80
高血壓前期	120-139	或	80-89
第一期高血壓	140-159	或	90-99
第二期高血壓	160-179	或	100-109
第三期高血壓	>=180	或	>=110