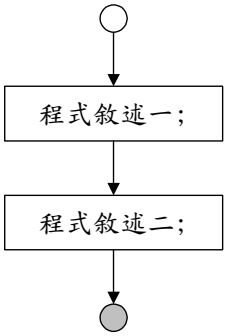
第四章流程控制

資訊科技系
林偉川

4-1-1 流程

- 流程就是程式執行的方向
- 循序執行



- 可以利用**選擇敘述**或**迴圈敘述**改變流程方向

2

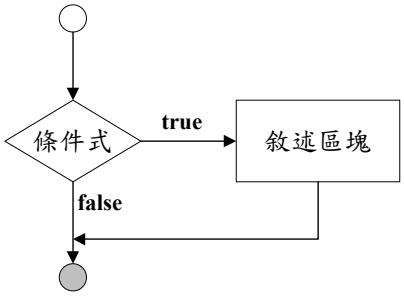
4-1-2 流程控制敘述

- 五種控制敘述：
  - **if/else**
  - **switch**
  - **for**
  - **while**
  - **do/while**

4-1-2 流程控制敘述

- 選擇敘述

```
if(條件式){
        //程式敘述
}
```

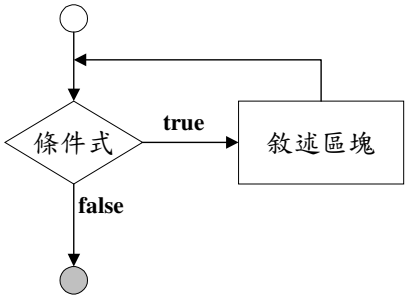


- **條件式**為一運算式，而其結果必須為布林值
- {} 包起來的程式敘述為敘述區塊

4-1-2 流程控制敘述

- 迴圈敘述

```
while(條件式) {
        //程式敘述
}
```



5

4-1-3 程式參數

- 形式參數

```
public static void main(String [] args)
public static void main(String [] s)
```

形式參數

- 參數值

```
C:\java2tb\ch04>java EX4_1 3  45 abc
```

- 將字串轉換成整數

```
int j = Integer.parseInt("6");
```

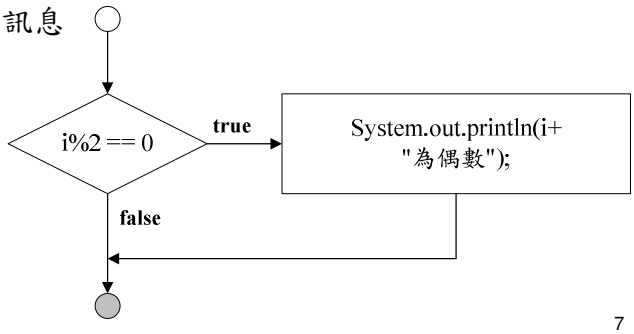
6

4-2-1 if 選擇敘述

```
if(條件式){
        //程式敘述
}
```

- 若條件式為true，執行敘述區塊。
- 若條件式為false，則跳過敘述區塊。

i若為偶數則輸出訊息



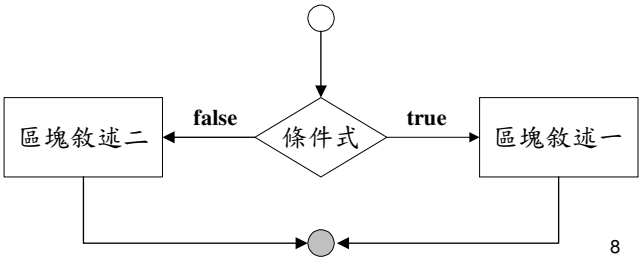
7

4-2-2 if/else敘述基本用法

```
if(條件式){
        //區塊敘述一
}
else{
        //區塊敘述二
}
```

- 若條件式為true，執行區塊敘述一。
- 若條件式為false，執行區塊敘述二。

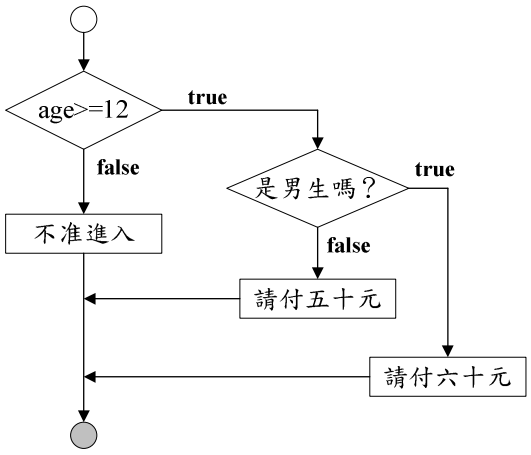
if/else 的流程圖



8

4-2-3 if/else的巢狀結構

- 若年齡夠大，再依照性別決定票價。



9

4-2 if/else選擇敘述

4-2-3 if/else的巢狀結構

- 兩層巢狀 if/else 的可能情況

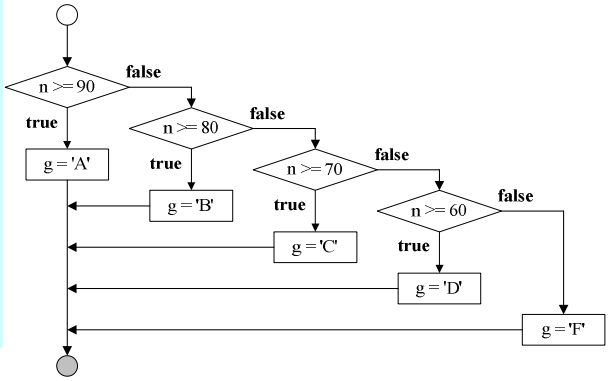
```
if(...){
    ...
    if(...){
        ...
    } else {
        ...
    }
} else {
    ...
}
```

```
if(...){
    ...
} else {
    ...
    if(...){
        ...
    } else {
        ...
    }
}
```

10

4-2-4 if/else 多層巢狀結構

```
if(n>=90){
   g='A';
}else if(n>=80){
   g='B';
}else if(n>=70){
   g='C';
}else if(n>=60){
   g='D';
} else   g='F';
```



11

4-2-5 尋找伴侶的else

- else 配對時，應由前面的先配對
- else 先和最靠近自己的 if 配對
- 若最靠近的 if 已經配對了，則找次靠近者

```
if ( i >0 )
    if ( j >0 )
        if ( k >0 )
            a = 100;
        else
            a = 200;
    else
        a = 300;
else
    a = 400;
```

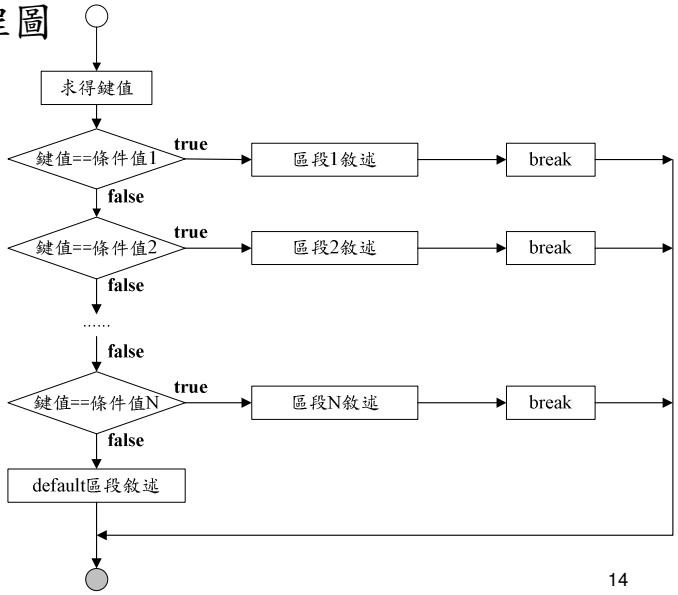
12

switch敘述語法

```
switch (鍵值){
        case 條件值1:
                // 區段1敘述
                break;
        case 條件值2:
                // 區段2敘述
                break;
        ……
        case 條件值N:
                // 區段N敘述
                break;
        default:
                // default區段敘述
}
```

13

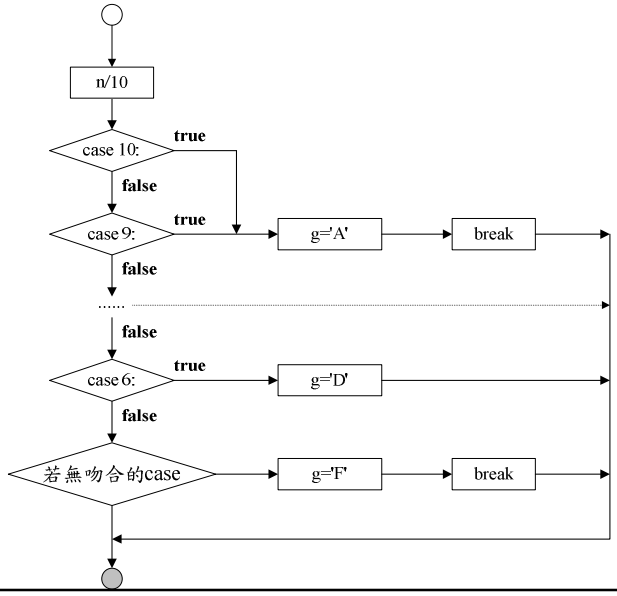
switch的流程圖



14

4-3 switch選擇敘述

- 沒有break時，繼續往下執行，並忽略標籤



15

4-3 switch選擇敘述

- 使用switch敘述時應注意的事項

  - 鍵值必須為可自動轉換成int的型別。
  - case 和 default 視為標籤，其順序沒有限制。
  - case 後面只能接**常數**或是**常數的運算式**，不能包含變數。
  - 鍵值會和所有 case 標籤一一比對。
  - 當鍵值和所有 case 標籤比對過，而且沒有符合的 case 標籤時，default 標籤以下的敘述會被執行。

16

4-4-1 for的語法與流程

- for迴圈語法

```
for (起始式; 條件式; 步進式) {
        //區塊內敘述
}
```

起始式－進入迴圈時，一開始執行的程式運算式，只執行一次。通常起始式，多為設定控制迴圈執行變數的起始值。

條件式－判斷是否重複執行迴圈的依據。如果條件式的結果為true，則繼續執行迴圈；如果為false，則跳離迴圈。

步進式－每經過一次迴圈，就會執行一次的運算式。通常步進式，多為設定控制迴圈執行變數的遞增或遞減運算式。

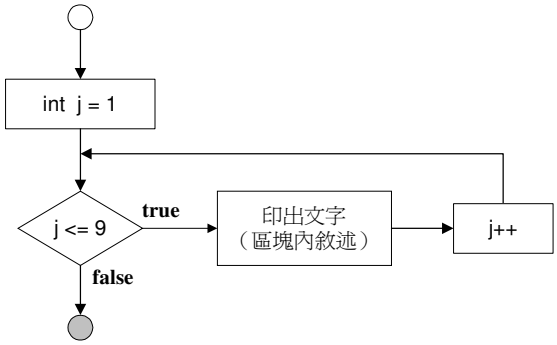
17

4-4 for迴圈

4-4-1 for的語法與流程

- 例如：

```
for (int j=1; j<=9; j++){
        System.out.print(j*j + "\t");
}
```



18

4-4-1 for的語法與流程

使用for迴圈應注意的事項

- **起始式**、**條件式**及**步進式**，可以視情況省略。
- 若**起始式**或**步進式**，有不只一個敘述時，以逗號區隔運算式。
- 括弧內用以區隔**起始式**、**條件式**及**步進式**的兩個分號是不能省略。

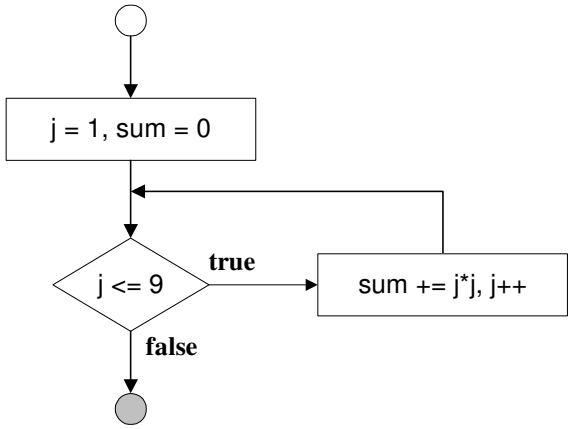
19

4-4 for迴圈

4-4-1 for的語法與流程

for迴圈的變化型

```
for (j=1, sum=0; j<=9; sum+=j*j, j++) ;
```



20

4-4-2 for巢狀結構

```
for (int i=1; i<=5; i+=2) {
  for (int j=2; j<=6; j+=2) {
    System.out.print("(" + i + "," + j + ")\t");
  }
  System.out.print("\n");
}
```

執行方向→				
外層for第1次	內層for第1次 (1,2)	內層for第2次 (1,4)	內層for第3次 (1,6)	換行
外層for第2次	內層for第4次 (3,2)	內層for第5次 (3,4)	內層for第6次 (3,6)	換行
外層for第3次	內層for第7次 (5,2)	內層for第8次 (5,4)	內層for第9次 (5,6)	換行

4-4 for迴圈

4-4-2 for巢狀結構

- 九九乘法表

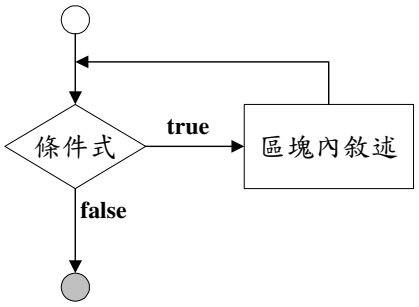
```
2X2=4   3X2=6   4X2=8   5X2=10  6X2=12  7X2=14  8X2=16  9X2=18
2X3=6   3X3=9   4X3=12  5X3=15  6X3=18  7X3=21  8X3=24  9X3=27
2X4=8   3X4=12  4X4=16  5X4=20  6X4=24  7X4=28  8X4=32  9X4=36
2X5=10  3X5=15  4X5=20  5X5=25  6X5=30  7X5=35  8X5=40  9X5=45
2X6=12  3X6=18  4X6=24  5X6=30  6X6=36  7X6=42  8X6=48  9X6=54
2X7=14  3X7=21  4X7=28  5X7=35  6X7=42  7X7=49  8X7=56  9X7=63
2X8=16  3X8=24  4X8=32  5X8=40  6X8=48  7X8=56  8X8=64  9X8=72
2X9=18  3X9=27  4X9=36  5X9=45  6X9=54  7X9=63  8X9=72  9X9=81
```

4-5 while迴圈

- 標上起始式和步進式的 **while** 迴圈

```
//起始式
while(條件式) {
        // 區塊內敘述
        // 步進式
}
```

- while的流程圖



4-5 while迴圈

- 無窮迴圈

```
i=1;
while (i<=100) { sum+=i;}
```

```
for (;;) {
        // 區塊內敘述
}
```

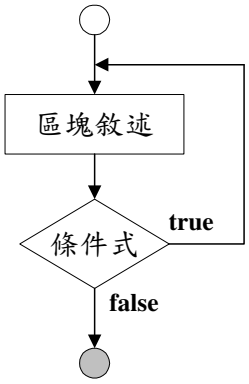
```
while (true) {
        // 區塊內敘述
}
```

- 無窮迴圈經常是在「不知道要跑幾次迴圈」時使用，常搭配跳離迴圈的關鍵字break來使用。

4-6 do/while迴圈

- do/while的語法：

```
Do {
        // 區塊內敘述
}
while (條件式);
```



- do/while的條件式擺在結構的最後面，也就表示先執行區塊敘述一次，再做條件式判斷。

25

4-7 break與continue

4-7-1 break

- break的作用是跳離敘述區塊

```
i=1;
while(true) {
  if(i==7)
    break;
  System.out.print(i+"\t");
  i++;
}
```

i 為 7 時跳出 while 迴圈

26

4-7-2 continue

- continue的功能是跳到迴圈的起始處

```
for(int j=0; j<10; j++)
{
        if(j==4)
                continue;    // 跳到迴圈的起始處
        System.out.print(j + "\t");
}
```

j 為 4 時跳到迴圈的起始處

4-7-3 迴圈標籤

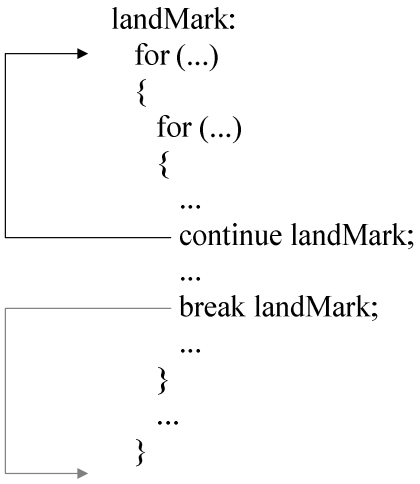
- 迴圈標籤由識別字和冒號組成

```
landMark:
```

- 使用迴圈標籤的方式
  - 方式一：標籤設定在迴圈敘述的前面。
  - 方式二：迴圈敘述為標籤區塊的第一個敘述。

4-7-3 迴圈標籤

- 標籤後緊接著迴圈敘述

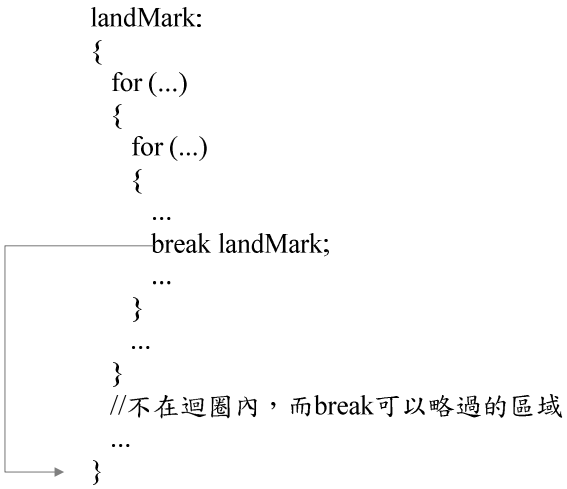


29

4-7 break與continue

4-7-3 迴圈標籤

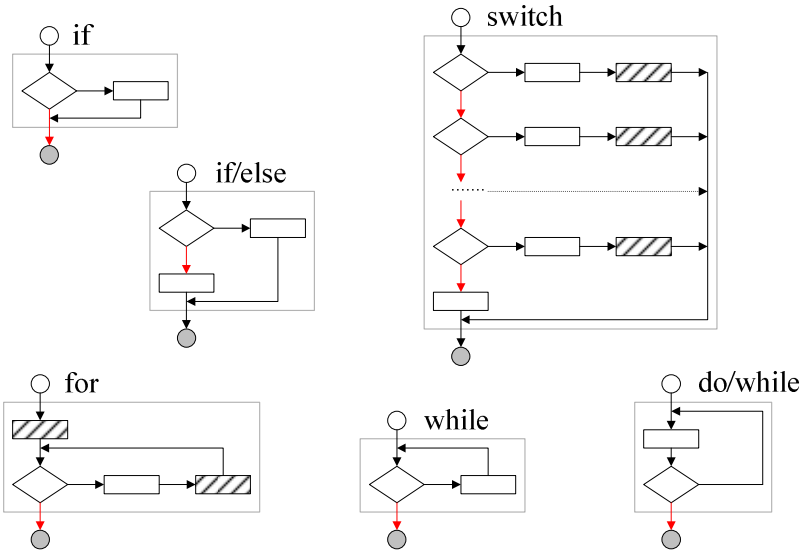
- 標籤區塊內的第一個敘述為迴圈敘述



30

4-8 流程敘述組合

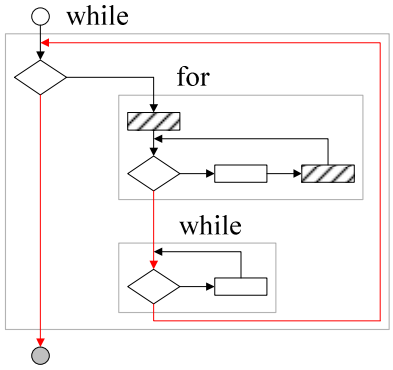
- 各種流程控制敘述簡圖



31

4-8 流程敘述組合

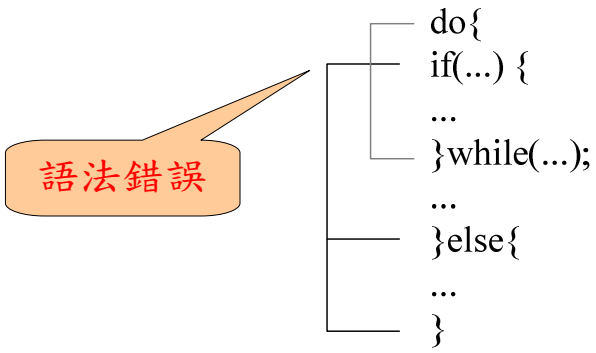
- 利用**依序**和**巢狀**，可以撰寫出結構化程式




```
while(...)
{
    for(...)
    {
        ...
    }
    while(...)
    {
        ...
    }
}
```

32

- 兩個單元的範圍線不能重疊



33