Java

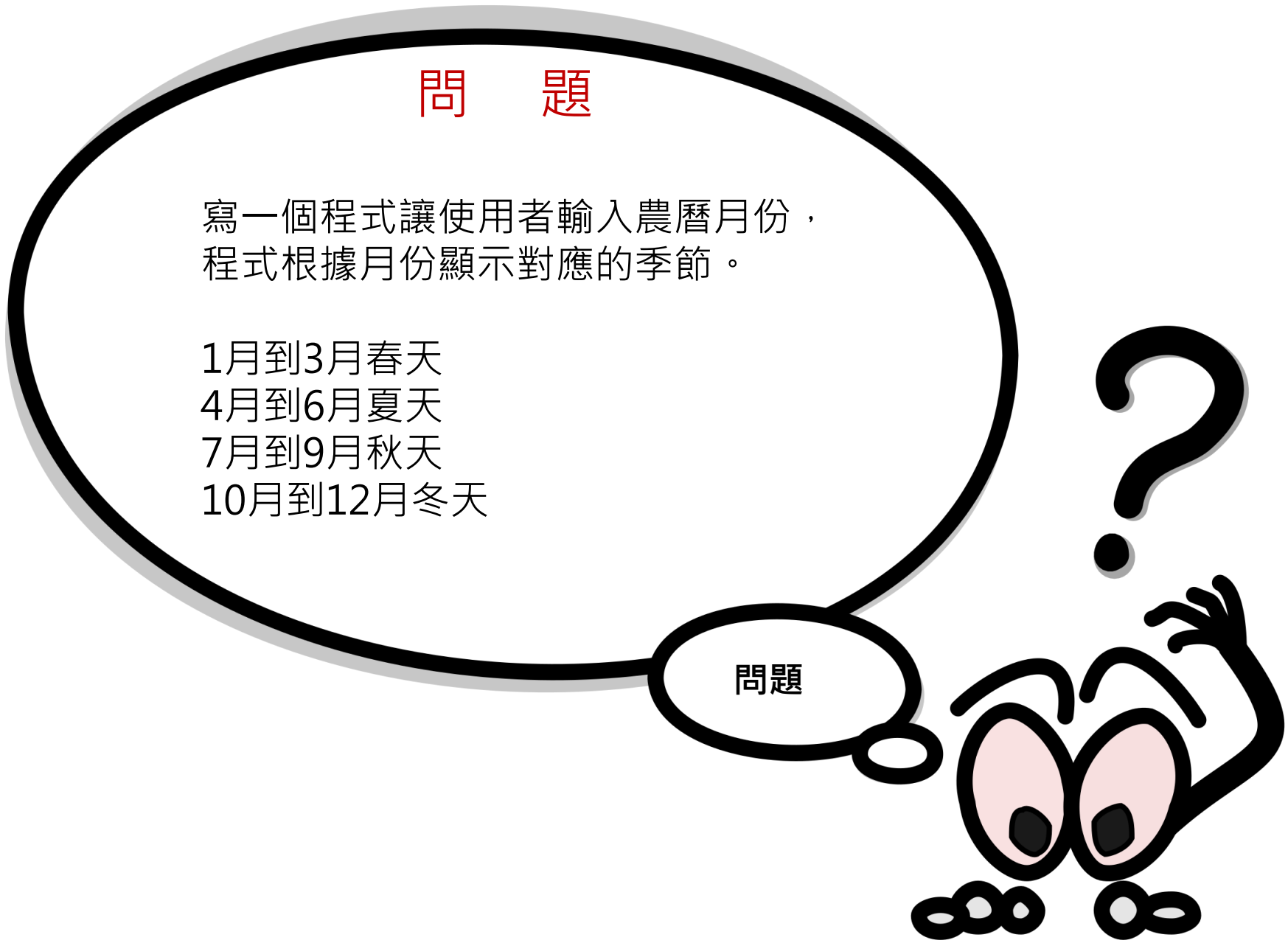
多重選擇 switch


還能怎麼選?

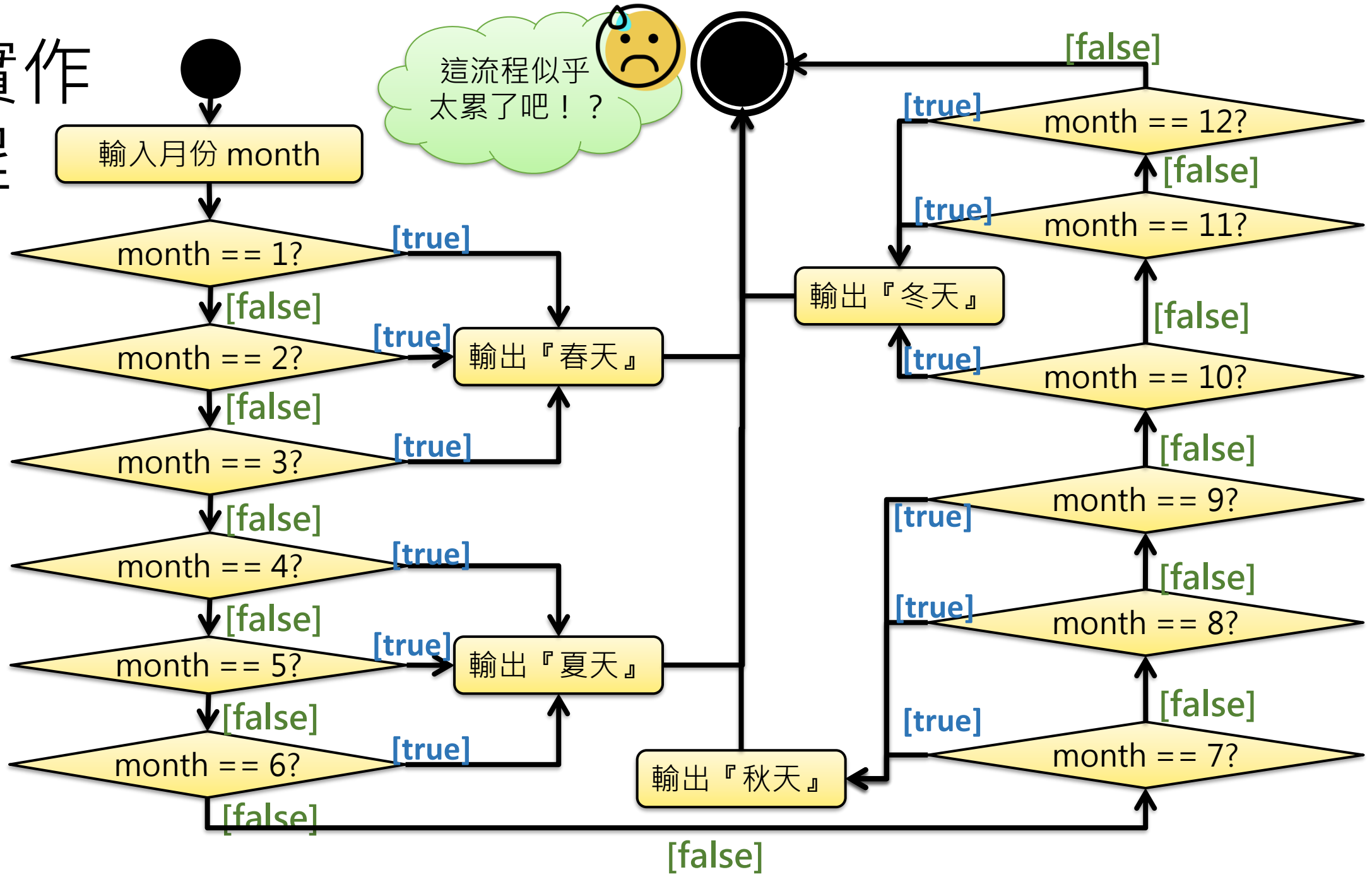
問　題

寫一個程式讓使用者輸入農曆月份，
程式根據月份顯示對應的季節。

1月到3月春天
4月到6月夏天
7月到9月秋天
10月到12月冬天

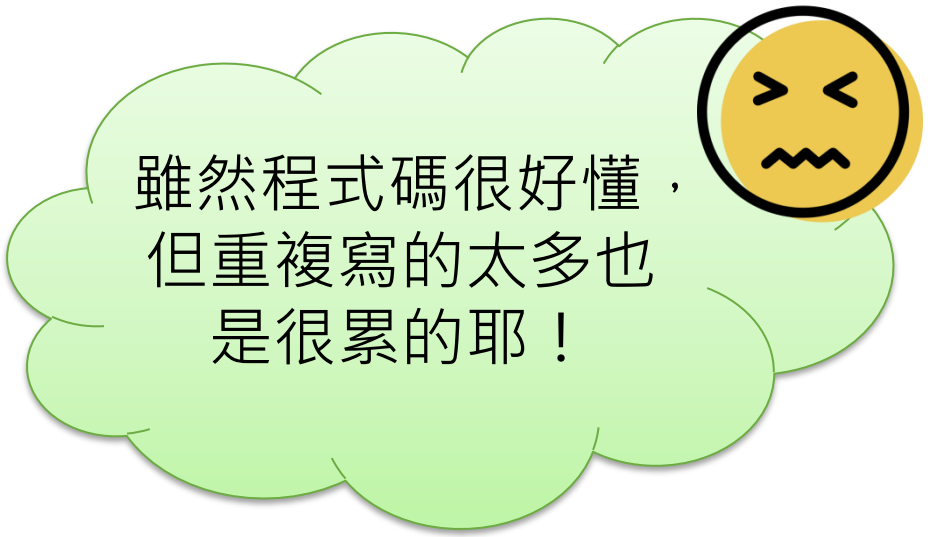


用 if 實作的流程


這流程似乎太累了吧！？
輸入月份 month
month == 1?
[true]
[false]
month == 2?
[true]
輸出『春天』
[false]
month == 3?
[true]
[false]
month == 4?
[true]
[false]
month == 5?
[true]
輸出『夏天』
[false]
month == 6?
[true]
[false]
[false]
輸出『秋天』
[true]
month == 7?
[false]
[true]
month == 8?
[false]
[true]
month == 9?
[false]
[true]
month == 10?
[false]
[true]
輸出『冬天』
[true]
month == 11?
[false]
[true]
month == 12?
[false]

用 if 實作

使用虛擬碼表達看看




```
如果                月份 == 1
    輸出『春天』
第2種如果           月份 == 2
    輸出『春天』
第3種如果           月份 == 3
    輸出『春天』
第4種如果           月份 == 4
    輸出『夏天』
第5種如果           月份 == 5
    輸出『夏天』
第6種如果           月份 == 6
    輸出『夏天』
第7種如果           月份 == 7
    輸出『秋天』
第8種如果           月份 == 8
    輸出『秋天』
第9種如果           月份 == 9
    輸出『秋天』
第10種如果          月份 == 10
    輸出『冬天』
第11種如果          月份 == 11
    輸出『冬天』
第12種如果          月份 == 12
    輸出『冬天』
```

另一種框架：switch

```
switch (變數名稱或運算式) {
        case 數字(或字元)1:
            → 符合數字(或字元)1時執行的敘述式
            break;
        case 數字(或字元)2 :
            → 符合數字(或字元)2時執行的敘述式
            break;

        default:
            → 上面各個 case 都不符合時執行的敘述式

    }
```

switch 框架

1 switch 根據數字值或字元值進行判斷

```
switch ( 變數名稱或運算式 ) {
    case 數字(或字元)1:
        → 符合數字(或字元)1時執行的敘述式
        break;
    case 數字(或字元)2 :
        → 符合數字(或字元)2時執行的敘述式
        break;

    default:
        → 上面各個 case 都不符合時執行的敘述式
}
```

2 case 列出可以接受常數值(整數或字元)

3 case 值符合時，進會進入此區塊執行，會一直執行直到碰到break ，因此若 case 條件1符合但未在 case 條件1區塊放入 break，就會繼續執行 case 2 區塊。

4 default 處理未被 case 列出的情況

switch 多重選擇

- **switch** 會依照變數或運算式的**值**與 **case 條件值**一一作比較，直到找到符合條件值時，就會執行該區塊的程式碼，如果條件值都不成立，就執行**default**區塊的程式碼。
- 當多條件且條件為常數時，建議使用 switch 多條件判斷控制流程。

```
switch (變數名稱或運算式) {
  case 數字(或字元)1:
     → 符合數字(或字元)1時執行的敘述式
     break;
  case 數字(或字元)2 :
     → 符合數字(或字元)2時執行的敘述式
     break;
  default:
     → 上面各個case都不符合時執行的敘述式
     }
```

switch 運算式型態

if 判斷式是根據運算結果的 **boolean** 值判斷，因此只要判斷結果是得到 true 或 false 就可以作為 if 的條件判斷式。

```
if (條件判斷式) {
    → 判斷符合時要執行的程式碼 ←
}
```

switch 的判斷式是根據**整數或字元**值作為判斷，只要判斷的運算式可以得到整數或字元結果，就可以作為 switch 的判斷運算式。

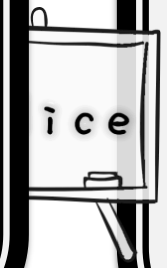
可接受的資料型態：short、byte、int 、char

```
switch (變數名稱或運算式) {
        case 數字(或字元)1:
            → 符合數字(或字元)1時執行的敘述式
            break;
        case 數字(或字元)2 :
            → 符合數字(或字元)2時執行的敘述式
        }
```

switch 運算式數量

一個判斷式在 if 框架裡只能有兩種結果區塊(if 區塊或 else 區塊)，若使用 else if，要得到 N 個結果區塊也需要使用 N-1 個判斷式。

```
if (判斷式1) {
    符合條件式1時執行的敘述式
} else if (判斷式2) {
    符合條件式2時執行的敘述式
} else if (判斷式3) {
    符合條件式3時執行的敘述式
} else {
    都不符合時執行的敘述式
}
```



一個 switch 只會有**一個**運算式，搭配 **case** 將運算式結果可能涵蓋的常數值一一列出。

```
switch (變數名稱或運算式) {
  case 數字(或字元)1:
    → 符合數字(或字元)1時執行的敘述式
    break;
  case 數字(或字元)2 :
    → 符合數字(或字元)2時執行的敘述式
    break;
  default:
    → 上面各個case都不符合時執行的敘述式
}
```

用 switch 答題 (1)

這個程式碼感覺也挺辛苦的耶！

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

      month = sc.nextInt();
      switch (month) {
         case 1:
            System.out.println("春天");
            break;
         case 2:
            System.out.println("春天");
            break;
         case 3:
            System.out.println("春天");
            break;
         case 4:
            System.out.println("夏天");
            break;
         case 5:
            System.out.println("夏天");
            break;
         case 6:
            System.out.println("夏天");
            break;
         case 7:
            System.out.println("秋天");
            break;
         case 8:
            System.out.println("秋天");
            break;
         case 9:
            System.out.println("秋天");
            break;
         case 10:
            System.out.println("冬天");
            break;
         case 11:
            System.out.println("冬天");
            break;
         case 12:
            System.out.println("冬天");
            break;
         default:
            System.out.println("沒有這個月份喔!");
      }
   }
}
```

用 switch 答題 (2)

- 前面介紹 break 時有提過進入某 case 區塊後會一直執行值到碰到 break，若符合的 case 區塊沒有 break，即使下個 case 區塊不符合，仍會繼續執行下個 case 區塊內的程式碼。

- 若某幾個 case 都會執行相同的程式碼，就可以將這幾個 case 當作一個群組放在一起，只有在這個群組最後一個 case 加上 break。

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

      month = sc.nextInt();
      switch (month) {
         case 1:
            System.out.println("春天");
            break;
         case 2:
            System.out.println("春天");
            break;
         case 3:
            System.out.println("春天");
            break;
         case 4:
            System.out.println("夏天");
            break;
         ……
      }
   }
}
```

用 switch 答題 (3)

```
switch (month) {
  case 1:
    System.out.println("春天");
    break;
  case 2:
     System.out.println("春天");
     break;
  case 3:
     System.out.println("春天");
      break;
}
```

1、2、3月都是屬於春天，因此可以將1、2、3月當作一個群組，這群組的前兩個 case 區塊內都不放入程式碼與 break，只在這群組的最後一個 case 放入程式碼與 break。

```
switch (month) {
  case 1:
  case 2:
  case 3:
     System.out.println("春天");
      break;
}
```

用 switch 答題(4)

- 依照 case 群組的概念，將春夏秋冬各自所屬月份群組後，就可得到簡化後的 switch 判斷了！

```
import java.util.Scanner;
class web{
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      int month;
      month = sc.nextInt();
      switch (month) {
         case 1:
         case 2:
         case 3:
            System.out.println("春天");
            break;
         case 4:
         case 5:
         case 6:
            System.out.println("夏天");
            break;
         case 7:
         case 8:
         case 9:
            System.out.println("秋天");
            break;
         case 10:
         case 11:
         case 12:
            System.out.println("冬天");
            break;
         default:
            System.out.println("沒有這個月份喔!");
      }
   }
}
```

群組1

群組2

群組3

群組4

JAVA

延伸的概念

概念1：if 與 switch

	if	switch
必搭配的關鍵字		case
可搭配的關鍵字	else if else	default break
運算式結果型態	boolean	short、byte、int 、char
運算式數量	不限	1個
適用情境	•各種比較都可以 •也可以搭配 &&、||、!，可以處理範圍問題(例如: n>90 & n<80)	•可以條理列出的具體數值 •數值量是有限的（當數值太多時，最降低程式的可讀性）
優點	可使用的情境較廣	具體數值以及數值量不多的情境使用 switch 會有較佳的執行效率
缺點	判斷式沒設計好時容易降低程式的可讀性	可使用的情境有限

概念2：case 寫法

數字比較

是數字，所以要不用加單引號。

```
int month = 6;
switch (month) {
   case 1:
      System.out.println("春天");
      break;
   case  6:
       System.out.println("夏天");
       break;
}
```

字元比較

是**字元**，所以要**加單引號**。

```
char alph = 'b';
switch (alph) {
   case 'a':
      System.out.println("apple");
      break;
   case  'b':
       System.out.println("banana");
       break;
}
```

與變數值定義時一樣， case 後面接的值會依照資料型態有不同寫法，如果是字元就需要在前後加上單引號，數字就不需要。

概念3：switch 不接受 long 資料型態

數字型態： short、byte、int

```
int month = 6;
switch (month) {
   case  1:
      System.out.println("春天");
      break;
   case  6:
       System.out.println("夏天");
       break;
}
```

數字型態：long

```
long month = 6;
switch (month) {
  case 1:
    System.out.println("春天");
    break;
  case 6:
    System.out.println("夏天");
    break;
}
```

switch 是使用 int 數值進行判斷的，short 與 byte 的數值範圍都有在 int 涵蓋的數值範圍內，所以可以使用 switch 可以接受 short 與 byte 進行判斷，但 long 涵蓋的數值範圍比 int 多，若使用 long 進行 switch 判斷，可能會有超出 int 涵蓋範圍的數值無法處理的情況，所以 switch 不接受 long 資料型態。