

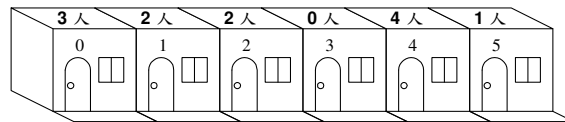
第五章陣列

資訊科技系
林偉川

5-1 何謂陣列

① 陣列- 「多個擁有相同名稱且同型別的變數集合」

② 生活上的例子：住宿情形



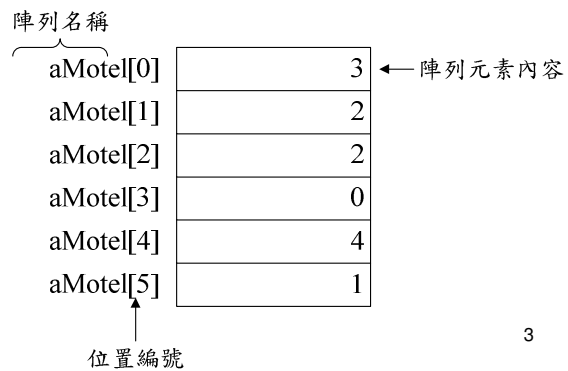
5-1 何謂陣列

◎ 生活上的例子：汽車旅館

0號房間有3個房客
1號房間有2個房客
2號房間有2個房客
3號房間有0個房客
4號房間有4個房客
5號房間有1個房客

■ 擁有六個int型別的陣列

■ `int aMotel[]=new int[6];`



5-1 何謂陣列

◎ 陣列元素

- 陣列元素以陣列名稱，後接一對中括弧“[]”，中括弧內放入位置編號。
- 位置編號一般也稱為索引，索引一律由0開始，接著是1、2、3...至元素個數減一。
- 陣列的索引必須是大於或等於0的整數，或結果為整數的運算式。

5-2-1 陣列的宣告與配置

①陣列的宣告方式：//28,31

```
資料型別 陣列名稱[]; 資料型別 []陣列名稱;資料型別[] 陣列名稱;
```

②同時宣告兩個int型別

```
int[] arr1, arr2;  int arr1[],arr2[];  int []arr1, []arr2;
```

③配置陣列：

```
陣列名稱 = new 資料型別[元素個數];
```

④宣告並配置：

```
資料型別 陣列名稱[] = new 資料型別[元素個數];
```

5

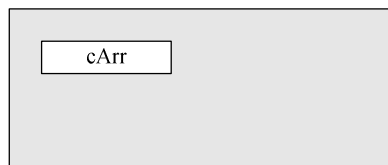
5-2-1 陣列的宣告與配置

①陣列變數其實是一種參照變數(reference variable)

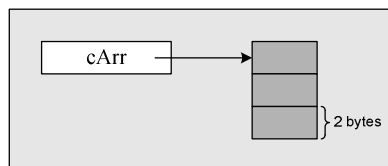
②陣列名稱是指向陣列實體的一個參照變數//18

③JVM中陣列的宣告與配置

```
char cArr[];
```



```
cArr = new char[3];
```



6

5-2-2 使用陣列元素

Ⓒ 使用陣列元素的語法：

陣列名稱[索引]

Ⓒ 使用陣列的敘述：

```
int iArr[] = new int[3]; //宣告並配置陣列
iArr[1] = 123;           //設定第2個元素值為123
iArr[2] = 456;           //設定第3個元素值為456
```

//將第2和第3個元素值相加後設定給第1個元素

```
iArr[0] = iArr[1] + iArr[2];
```

Ⓒ 取得陣列的長度

陣列名稱.length

7

5-2-2 使用陣列元素

Ⓒ 元素預設值//2

- 陣列在配置後，元素的內容就會被填入預設值。
- 陣列的基底型態為整數者預設值為0。
- 浮點數則為0.0f或0.0。
- boolean型別為false。
- 物件為null。

8

5-2-3 以大括號配置陣列並設值

Ⓒ 宣告並設定陣列內容

```
資料型別 陣列名稱[] = new 資料型別[] {元素值1, 元素值2, ..., 元素值N};
```

```
資料型別 陣列名稱[] = {元素值1, 元素值2, ..., 元素值N};
```

Ⓒ 例子：

```
int myArray[] = new int[] {10, 20, 30, 40, 50, 60}; //6
```

```
int myArray[] = {10, 20, 30, 40, 50, 60}; //6
```

Ⓒ 宣告並設定初值時，勿在中括號內填入元素的個數

```
int myArray[3] = {1, 2, 3}; // 錯誤
```

Ⓒ 已宣告的陣列不可以使用大括號配置陣列及設值

```
int myArray[];
myArray = {1, 2, 3}; //錯誤
```

9

5-2-4 索引勿超出範圍

Ⓒ 索引超出範圍時，編譯的過程不會有錯誤訊息。

Ⓒ 執行時會發生例外。

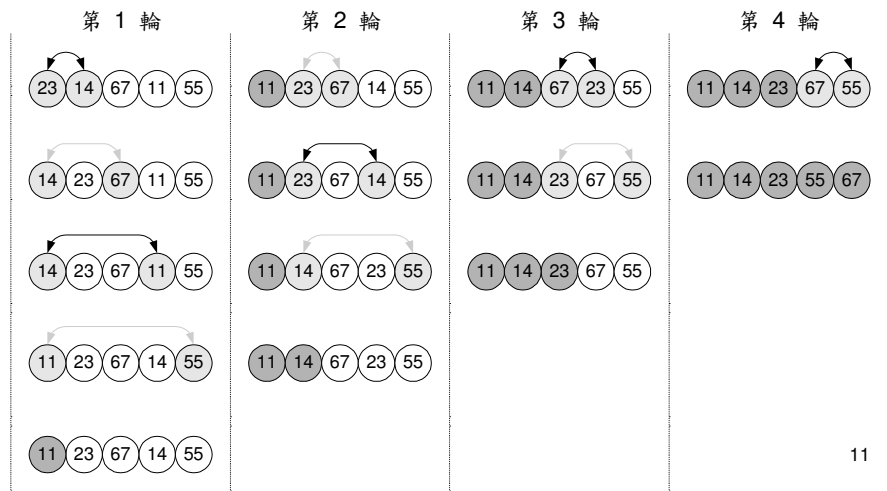
Ⓒ 使用if過濾：

```
if(ch<0 || ch>=xmen.length){//98,105,125
    System.out.println("超出範圍了，請重來!");
    System.exit(0);
}
```

10

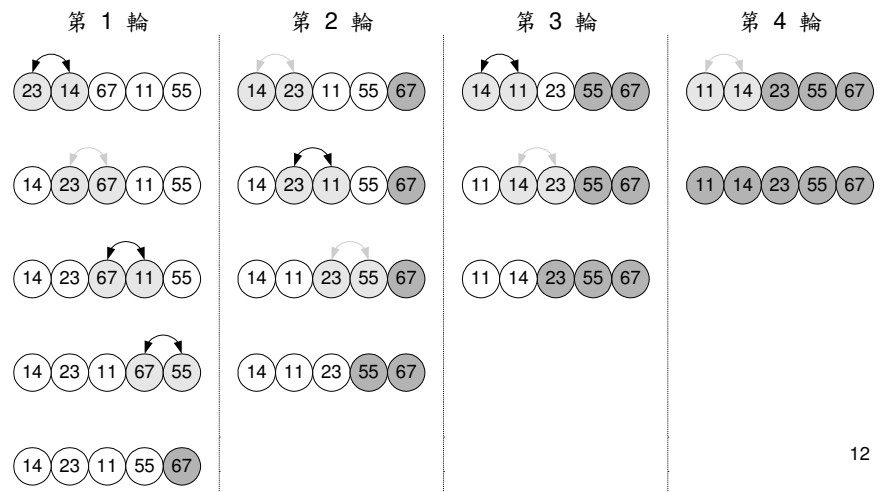
5-3-1 排序

◎ 選擇排序法



5-3-1 排序

◎ 氣泡排序法



5-3-2 搜尋

◎ 循序搜尋

- 從頭到尾一一做比對

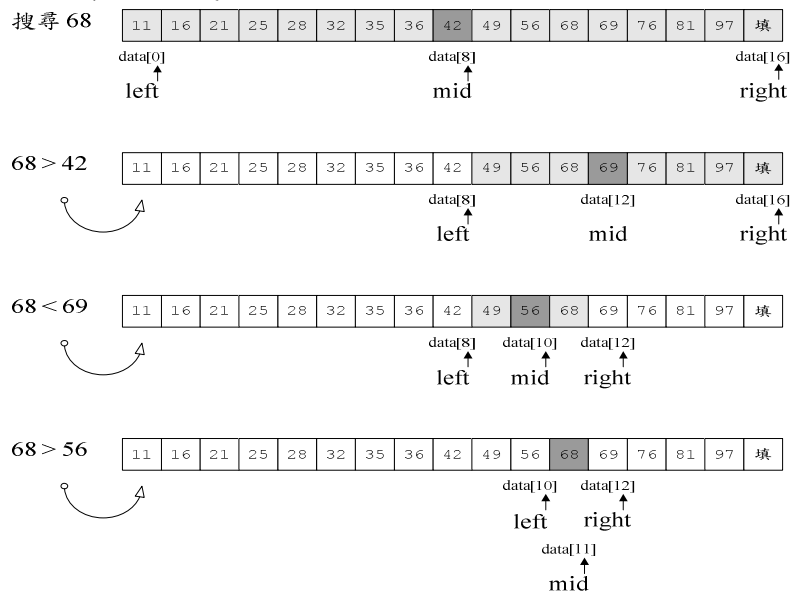
◎ 二元搜尋

- 資料必須先經過排序
- 二分法
- 猜數字遊戲，數字範圍為1~99

| | |
|--------|---------|
| 乙：「50」 | 甲：「大一點」 |
| 乙：「75」 | 甲：「小一點」 |
| 乙：「62」 | 甲：「大一點」 |
| 乙：「68」 | 甲：「小一點」 |
| 乙：「65」 | 甲：「答對了」 |

13

◎ 二元搜尋的方式



14

5-4 陣列的「=」運算

基本型別使用指定運算子時，會將變數的值存入另一個變數的記憶位址。

```
int a = 5, b;  
b = a;           //將a設定給b，a為5，b為5  
b = 6;          //將6設定給b，a為5，b為6
```

指定運算子使用在陣列或物件時，與「以值指定」使用相同的機制，不過其值不是一般的基本型別常數，而是參照(Reference)

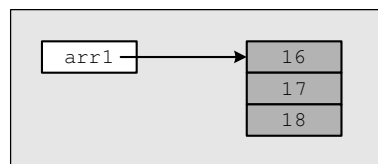
```
int[] arr1 = {16, 17, 18}; //宣告配置arr1並設值 SCJP 18  
int[] arr2;                //宣告arr2陣列  
arr2 = arr1;               //arr1參照指定給arr2
```

15

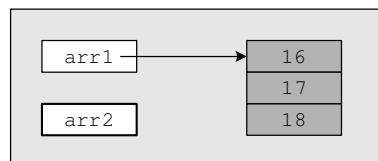
5-4 陣列的「=」運算

兩個陣列名稱指向相同的陣列實體

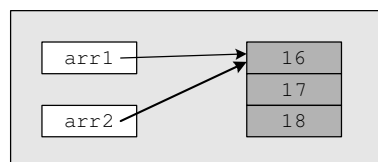
```
int[] arr1 = {16, 17, 18};
```



```
int[] arr2;
```



```
arr2 = arr1;
```



16

5-4 陣列的「=」運算

- ④ 兩個陣列的**基底型別**不同時，**參照不可相互指定**

```
int[] arr1 = {16, 17, 18};  
long[] arr3 = arr1;           //錯誤
```

17

5-5 陣列複製

- ④ 使用System.arraycopy方法

```
System.arraycopy(來源陣列, 來源起始索引,  
                目的陣列, 目的起始索引, 陣列長度);
```

- ④ **沒有配置空間的陣列**，**只能指定參考**，**不能**用System.arraycopy複製陣列。

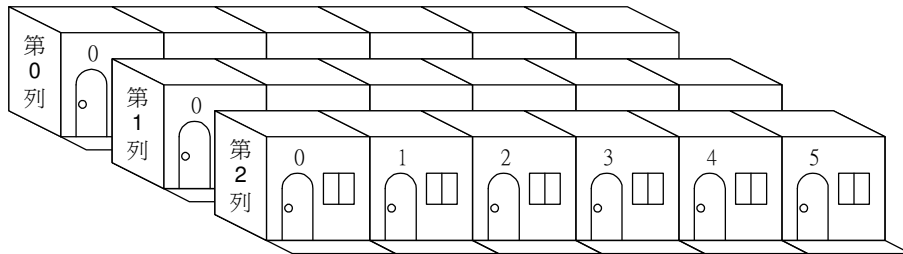
- ④ 使用System.arraycopy的例子

```
int[] a1 = {10, 20, 30, 40, 50};  
int[] a2 = new int[a1.length];  
System.arraycopy(a1, 0, a2, 0, a1.length);
```

18

5-6 二維陣列

◎ 二維陣列可以想成有好幾列的房間



◎ `int a[][]=new int[3][6];`

◎ `int a[][]={{0,0,0,0,0,0},{0,0,0,0,0,0},{0,0,0,0,0,0}};`

19

5-6 二維陣列

5-6-1 二維陣列的宣告與配置

◎ 宣告二維陣列的語法//51,62,85,131

資料型別 陣列名稱[][];

資料型別[]陣列名稱[];

資料型別[][]陣列名稱;

◎ 配置陣列：

陣列名稱 = new 資料型別[列數][行數];

◎ 宣告並配置的例子：

```
int[][] arr = new int[3][5]; int[] arr[] = new int[3][5]; int arr[][] = new int[3][5];  
int[] [] arr = new int[3][5]; int [][] arr = new int[3][5]; int [] arr[ ] = new int[3][5];
```

20

5-6-2 使用元素

@宣告並配置：

```
int[][] arr = new int[3][4];
```

@元素的表示法：

| | 行0 | 行1 | 行2 | 行3 |
|----|---------|---------|---------|---------|
| 列0 | a[0][0] | a[0][1] | a[0][2] | a[0][3] |
| 列1 | a[1][0] | a[1][1] | a[1][2] | a[1][3] |
| 列2 | a[2][0] | a[2][1] | a[2][2] | a[2][3] |

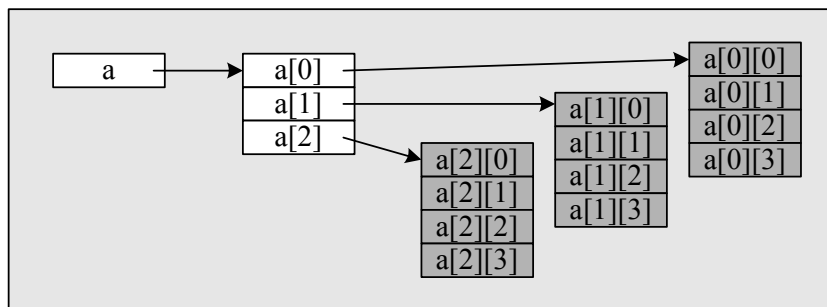
21

5-6-3 JVM 中的二維陣列

@宣告並配置：

```
int[][] arr = new int[3][4];
```

@JVM 中的二維陣列 a

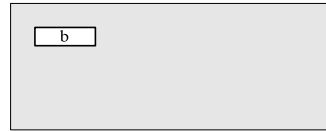


5-6 二維陣列

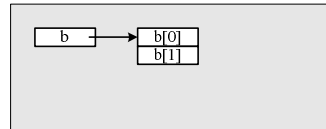
5-6-3 JVM 中的二維陣列

④ 建立陣列實體，
再指定給參照變數

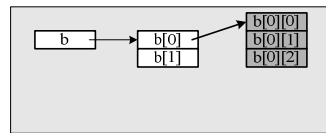
```
int b[][];
```



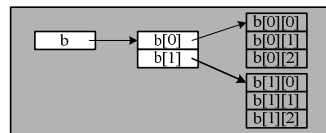
```
b = new int[2][];
```



```
b[0] = new int[3];
```



```
b[1] = new int[3];
```



5-6 二維陣列

5-6-4 使用大括號建立二維陣列

④ 非矩形二維陣列

```
int[][] a = {{1, 3, 5}, {2, 4}, {9, 8, 7, 6}};
```

| | 行0 | 行1 | 行2 | 行3 |
|----|----|----|----|----|
| 列0 | 1 | 3 | 5 | |
| 列1 | 2 | 4 | | |
| 列2 | 9 | 8 | 7 | 6 |

5-6-4 使用大括號建立二維陣列

ⓐ 若某列只有一個元素時，同樣必須以大括號包起來

```
int[][] a = {{1, 3, 5}, {2, 4}, {9}};
```

→

```
int a[][]=new [3][];
```

```
int a[0]=new int[3]; int a[1]=new int[2]; int a[2]=new int[1];
```

```
int[][] a = {{1, 3, 5}, {2, 4}, 9};
```

//錯誤

25

5-6-4 使用大括號建立二維陣列

ⓐ 範例5_13

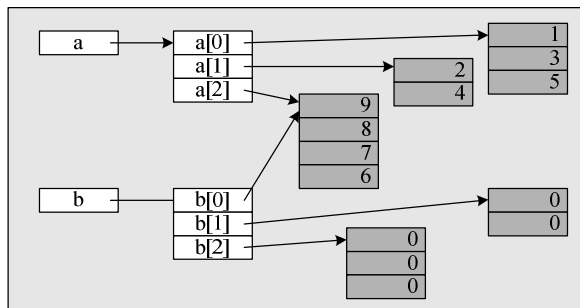
```
int[][] a = {{1, 3, 5}, {2, 4}, {9, 8, 7, 6}};
```

```
int b[][] = new int[3][];
```

```
b[0] = a[2]; //{9, 8, 7, 6}
```

```
b[1] = new int[2];
```

```
b[2] = new int[3];
```



26

5-7 多維陣列

◎N維陣列的宣告

```
資料型別 陣列名稱[][]...[];
```

```
資料型別[][]...[]陣列名稱;
```

◎配置陣列：

```
陣列名稱 = new 資料型別[個數1][個數2]...[個數N];
```

◎使用元素

```
陣列名稱[索引1][索引2]...[索引N]
```

◎建立三維陣列的例子：

```
int[][][]ar = new int[3][4][5];  
int[]b[][] = {{{9}}};
```

27

5-8 for/in迴圈

◎for/in迴圈的語法(J2SE1.5)

```
for(迭代變數宣告:陣列或物件){  
    // 區塊敘述  
}
```

◎for/in迴圈用在一維陣列

```
int[] a = {1, 3, 5, 7, 9};  
for(int i: a) System.out.println(i);
```

◎for/in迴圈的迭代變數並不是陣列的索引值，而是元素值。for/in的內部運作是將元素值指定給迭代變數，**迭代變數本身並不表示任何元素。**

```
int[][] a = {{1, 3}, {5, 7, 9}}; // 二維陣列  
for(int[] i: a) {  
    for (int j:i) System.out.print(j);  
    System.out.println(""); }  
}
```

28

ⓐ 使用for/in迴圈時所應注意的事項：

- 迭代變數宣告，不可以省略；迭代變數也不可以和既有的變數同名。
- 迭代變數宣告，只包含型別和變數名稱，可以使用final宣告。
- for/in迴圈內，修改迭代變數的值，並不會影響陣列的元素值。
- 迭代變數宣告時，型別必須和陣列元素相符合。
- for/in迴圈可以使用在陣列，或者實作java.lang.Iterable介面的物件上。