

程式設計 (**Programming**)

真理大學 資訊工程系 吳汶涓老師

CH10

結構、Unions、位元
處理以及列舉型別



本章綱要

10-1 簡介

10-2 結構定義

10-3 結構的初始值設定

10-4 存取結構的成員

10-5 使用結構與函式

10-6 Typedef

10-7 範例：高效率洗牌和發牌模擬器

10-8 Unions

10-9 位元運算子

~~10-10 位元欄位~~

10-11 列舉型別常數

10.6 typedef

■ typedef

- 為之前定義的資料型別建立別名
- 用typedef建立較短的类型名稱
- 範例：

```
struct card{  
    char *face;  
    char *suit;  
};  
typedef struct card CC;  
CC deck[52];
```

※注意: 為了強調，**typedef** 的第一個字母通常大寫

10.7 範例：高效率的洗牌和發牌

■ 程式

- 建立一個card 結構型態的陣列
- 將撲克牌放到一副牌中
- 洗牌
- 發牌

Four of Clubs	Three of Hearts
Three of Diamonds	Three of Spades
Four of Diamonds	Ace of Diamonds
Nine of Hearts	Ten of Clubs
Three of Clubs	Four of Hearts
Eight of Clubs	Nine of Diamonds
Deuce of Clubs	Queen of Clubs
Seven of Clubs	Jack of Spades
Ace of Clubs	Five of Diamonds
Ace of Spades	Five of Clubs
Seven of Diamonds	Six of Spades
Eight of Spades	Queen of Hearts
Five of Spades	Deuce of Diamonds
Queen of Spades	Six of Hearts
Queen of Diamonds	Seven of Hearts
Jack of Diamonds	Nine of Spades
Eight of Hearts	Five of Hearts
King of Spades	Six of Clubs
Eight of Diamonds	Ten of Spades
Ace of Hearts	King of Hearts
Four of Spades	Jack of Hearts
Deuce of Hearts	Jack of Clubs
Deuce of Spades	Ten of Diamonds
Seven of Spades	Nine of Clubs
King of Clubs	Six of Diamonds
Ten of Hearts	King of Diamonds

```

1  /* Fig. 10.3: fig10_03.c */
3  #include <stdio.h>
4  #include <stdlib.h>
5  #include <time.h>
8  struct card {
9      const char *face;
10     const char *suit;
11 };
13 typedef struct card Card;
14
15
16 void fillDeck( Card * const wDeck, const char * wFace[], const char * wSuit[] );
17 void shuffle( Card * const wDeck );
18 void deal( const Card * const wDeck );
19
20
21 int main( void )
22 {
23     Card deck[ 52 ];
24     const char *face[] = { "Ace", "Deuce", "Three", "Four", "Five",
25                             "Six", "Seven", "Eight", "Nine", "Ten", "Jack", "Queen", "King" };
26     const char *suit[] = { "Hearts", "Diamonds", "Clubs", "Spades" };
27     srand( time( NULL ) );
28     fillDeck( deck, face, suit );
29     shuffle( deck );
30     deal( deck );
31     return 0;
32 }

```

每一張 card 都有花色和點數

現在Card是 struct card 的別名了

```

44 void fillDeck( Card * const wDeck, const char * wFace[],
45     const char * wSuit[] )
46 {
47     int i;
48
49     for ( i = 0; i <= 51; i++ ) {
50         wDeck[ i ].face = wFace[ i % 13 ];
51         wDeck[ i ].suit = wSuit[ i / 13 ];
52     }
53 }
54 }

```

常數指標，指向可改變的
Cards陣列

給每張**Card**一個花色和一個點數，填滿整副牌

```

58 void shuffle( Card * const wDeck )
59 {
60     int i;
61     int j;
62     Card temp;
63
64     for ( i = 0; i <= 51; i++ ) {
65         j = rand() % 52;
66         temp = wDeck[ i ];
67         wDeck[ i ] = wDeck[ j ];
68         wDeck[ j ] = temp;
69     }
70 }
71 }

```

每張牌都隨機地跟另一張牌交換，以這種方式將牌洗好

```
75 void deal( const Card * const wDeck )
76 {
77     int i;
80     for ( i = 0; i <= 51; i++ ) {
81         printf( "%5s of %-8s%c", wDeck[ i ].face, wDeck[ i ].suit,
82             ( i + 1 ) % 2 ? '\t' : '\n' );
83     }
85 }
```



常見的程式設計錯誤 10.7

參考到結構陣列中的個別結構時，忘記加上陣列的下標，會造成語法錯誤。

練習

- 撰寫程式試將五位學生之基本資料紀錄於巢狀結構中，並使用函式進行學生ID的搜尋，最後將找到的一筆資料列印出。學生資料有ID, Name, Birth day, height, weight等。

A122334002, David, 88/05/23, 173, 68

M112785994, John, 89/07/22, 178, 75

L222333885, Sue, 88/09/18, 162, 45

R217882495, Sally, 88/03/11, 166, 88

A185224569, Kevin, 89/12/01, 183, 92

練習

- 撰寫程式試使用結構與函式將上例五位學生之分數依平均 **bubble** 排序，並顯示排序前與排序後的結果
(學生資料請參考Ch10-p1投影片第12頁)



Before sorting:				
Name	Math	English	Program	Average
David	74	80	66	73.33
John	72	90	77	79.67
Sue	77	65	60	67.33
Sally	65	58	74	65.67
Kevin	81	79	68	76.00
After sorting:				
Name	Math	English	Program	Average
Sally	65	58	74	65.67
Sue	77	65	60	67.33
David	74	80	66	73.33
Kevin	81	79	68	76.00
John	72	90	77	79.67
請按任意鍵繼續 . . .				

10.8 Unions

■ 範例: (與結構相似)

```
union Number{  
    int x;  
    float y;  
};  
union Number value;
```

- union的成員會共用相同空間
- 以節省儲存空間
- 它會限制成員無法同時存在於記憶體中
- 兩個union不能用==或!=來比較
- 使用. 存取union成員，使用-> 存取指標union成員

```
1  /* Fig. 10.5: fig10_05.c */
```

```
3  #include <stdio.h>
```

```
6  union number {
```

```
7      int x;
```

```
8      double y;
```

```
9  };
```

Union 定義

Union 定義必須以分號作為結尾

```
11 int main( void )
```

```
12 {
```

```
13     union number value;
```

```
15     value.x = 100;
```

請注意y沒有值

```
16     printf( "%s\n%s\n%s\n %d\n\n%s\n %f\n\n\n",
```

```
17         "Put a value in the integer member",
```

```
18         "and print both members.",
```

```
19         "int:", value.x,
```

```
20         "double:", value.y );
```

```
22     value.y = 100.0;
```

```
23     printf( "%s\n%s\n%s\n %d\n\n%s\n %f\n",
```

```
24         "Put a value in the floating member",
```

```
25         "and print both members.",
```

```
26         "int:", value.x,
```

```
27         "double:", value.y );
```

```
29     return 0;
```

```
31 }
```

設值給 y，就會移除 x 的值

