

6 TIPI DI DATI DEFINITI DALL'UTENTE

6.1 Le strutture

Le strutture vengono impiegate per riunire insieme informazioni correlate. Esse sono formate da una serie di variabili a cui si fa riferimento con un unico nome.

La dichiarazione della struttura costituisce il modello per creare le istanze della struttura. Le variabili che compongono tale modello sono chiamate *membri* o *campi*.

Nell'esempio seguente viene dichiarata una struttura, tramite la parola chiave `struct`, che presenta, come campi, informazioni tipiche di una anagrafica:

```
struct info {  
    char Nome[35];  
    char Indirizzo[50];  
    char Citta[30];  
    unsigned long int CAP;  
    char Nazione[20];  
};
```

La dichiarazione deve terminare con un punto e virgola in quanto essa è una istruzione.

Si osservi che la scrittura precedente definisce semplicemente il modello della struttura dati, ma non dichiara alcuna variabile che adopera tale struttura. È come se fosse stato definito un nuovo tipo di dato, ovvero, uno *specificatore di tipo*. Per generare una variabile conforme al tipo `info` occorre scrivere:

```
struct info Nominativo;
```

Questa riga dichiara una variabile di tipo `info` chiamata `Nominativo`.

È possibile incorporare la dichiarazione di una o più variabili conformi ad una struttura nella dichiarazione stessa della struttura:

```
struct info {  
    char Nome[35];  
    char Indirizzo[50];  
    char Citta[30];  
    unsigned long int CAP;  
    char Nazione[20];  
} Nominativo1, Nominativo2, Nominativo3;
```

La scrittura dichiara un tipo di struttura, chiamato `info`, e le variabili `Nominativo1`, `Nominativo2` e `Nominativo3` che appartengono a tale tipo.

6.1.1 Accesso ai membri delle strutture

Per accedere ai singoli membri di una struttura si utilizza il nome della variabile definita per il tipo di struttura, seguito da un punto e dal nome del membro. La forma generale per l'accesso al membro di una struttura è quindi la seguente:

variabile_struttura.nome_membro

Ad esempio, il seguente frammento di codice assegna il valore 70100 al campo CAP della variabile **Nominativo** dichiarata precedentemente:

```
Nominativo.CAP=70100;
```

6.1.2 Assegnamenti ad una struttura

Le informazioni contenute in una struttura possono essere assegnate ad un'altra struttura dello stesso tipo utilizzando un'unica istruzione di assegnamento. Ciò significa che non è necessario assegnare separatamente i valori di ogni singolo membro.

Ad esempio, considerate le variabili **Nominativo1** e **Nominativo2** dichiarate precedentemente, il seguente codice ricopia le informazioni di **Nominativo2** in **Nominativo1**:

```
Nominativo1 = Nominativo2;
```

ovvero, equivale a scrivere:

```
strcpy(Nominativo1.Nome, Nominativo2.Nome);  
strcpy(Nominativo1.Indirizzo, Nominativo2.Indirizzo);  
strcpy(Nominativo1.Citta, Nominativo2.Citta);  
Nominativo1.CAP = Nominativo2.CAP  
strcpy(Nominativo1.Nazione, Nominativo2.Nazione);
```

6.2 Gli array di strutture

Le strutture vengono generalmente impiegate negli array. Per dichiarare un array di strutture, occorre, in primo luogo, definire il tipo della struttura e poi dichiarare una variabile come un array di tale tipo. La scrittura seguente dichiara un array di 50 elementi di strutture di tipo **info**:

```
struct info Nominativi[50];
```

Per accedere ad un determinato elemento dell'array basta indicizzare la variabile **Nominativi**. Nel seguente esempio viene stampato il codice CAP dell'elemento 5 dell'array **Nominativi**:

```
printf("%d", Nominativi[4].CAP);
```

6.3 Passaggio di strutture a funzioni

6.3.1 Passaggio dei membri di una struttura ad una funzione

Quando si passa ad una funzione un membro di una struttura, la funzione riceve solo il valore del membro. In altri termini, viene passata solo la variabile membro e non tutto il gruppo di variabili che costituiscono la struttura.

Vengono indicati di seguito alcuni passaggi di membri di una struttura alla funzione **Stampa**:

```
Stampa(Nominativo.Nome); /* passa il valore della stringa Nome */  
Stampa(Nominativo.Citta); /* passa il valore della stringa Citta */  
Stampa(Nominativo.CAP); /* passa il valore dell' intero CAP */
```

6.3.2 Passaggio di una intera struttura ad una funzione

Quando una struttura viene utilizzata come argomento di una funzione, essa viene passata interamente alla funzione utilizzando il metodo della chiamata per valore.

Nel seguente esempio viene, dapprima, dichiarata nella `main` la variabile `stcData` di struttura `pippo`; successivamente i membri di tale struttura vengono inizializzati e stampati mediante la chiamata alla funzione `Stampa`. Si osservi che il passaggio della struttura alla funzione avviene per valore, il che significa che i membri di `stcData` vengono semplicemente ricopiati nei rispettivi membri di `stcPrint`, ma le due strutture restano indipendenti.

```
#include <stdio.h>

struct pippo {
    int Codice;
    char Descrizione;
}

void Stampa(struct pippo stcPrint);

int main() {
    struct pippo stcData;

    stcData.Codice=3;
    stcData.Descrizione='B';

    Stampa(stcData);

    return 0;
}

void Stampa(struct pippo stcPrint) {
    printf("%d",stcPrint.Codice);
    printf("%c",stcPrint.Descrizione);
}
```

Come si può notare, se si dichiarano parametri formali di una funzione che corrispondono a strutture, occorre rendere globale la dichiarazione del tipo di struttura. In tale maniera la tipologia della struttura risulta visibile a tutte le funzioni e quindi utilizzabile in ogni parte del programma. Se la struttura fosse stata dichiarata all'interno di `main` non sarebbe stata utilizzabile da `Stampa`.