

Fondamenti di Programmazione A

Appunti per le lezioni – Gianfranco Rossi

Tipi strutturati - tabelle

La struttura dati *struct*, combinata con *array*, può essere usata per realizzare la struttura dati (astratta) tabella.

Tabella $m \times n$: disposizione rettangolare di m righe ed n colonne, dove ciascuna colonna contiene elementi (tutti) di tipo t_j ed e' individuata da un nome A_j (*attributo*), con $1 \leq j \leq n$, mentre ciascuna riga è individuata da un intero i (*indice*), con $0 \leq i < m$.

Graficamente

T	A_1	A_2	...	A_n
0			...	
1			...	
$m-1$...	

Ciascuna riga è una n -upla (o *record*), A_1, \dots, A_n sono gli *attributi* della n -upla e t_1, \dots, t_n i *domini* di ciascun attributo.

Più formalmente, una tabella T può essere definita come un sottoinsieme del prodotto cartesiano dei domini t_1, \dots, t_n e cioè

$$T \subseteq t_1 \times t_2 \times \dots \times t_n = \{ \langle a_1, a_2, \dots, a_n \rangle : a_1 \in t_1, \dots, a_n \in t_n \}$$

(ovvero una *relazione* n -aria su t_1, \dots, t_n). n.b. La cardinalità dell'insieme T corrisponde al numero di righe nella tabella.

Esempi

Tabella *esami*

<i>esami</i>	<i>materia</i>	<i>voto</i>	<i>lode</i>	<i>data_es</i>
0				
1				

49				

dove: l'attributo *materia* è di tipo *stringa*, *voto* di tipo numero *intero*, *lode* di tipo *booleano*, e *data_es* di tipo *data*.

Tabella *amici*

<i>amici</i>	<i>nome</i>	<i>cognome</i>	<i>indirizzo</i>	<i>tel</i>	<i>data_nascita</i>
0					
1					
99					

dove: gli attributi *nome*, *cognome* e *indirizzo* sono di tipo *stringa*, *tel* di tipo numero *intero* e *data_nascita* di tipo *data*.

La struttura dati astratta tabella $m \times n$ può essere realizzata in C++ tramite un *array* di *struct* (struttura dati concreta) nel modo seguente:

```

struct n_upla
{t1 A1;
  t2 A2;
  ...
  tn An;
};
n_upla T[m];

```

Esempio – tabella esami

```

struct dati_esame
{char materia[100];
  int voto;
  bool lode;
  data data_esame;
};
dati_esame esami[50];

```

in cui si assume che: la lunghezza massima del titolo dell'esame sia *100* caratteri, il numero massimo di esami (= righe della tabella) sia *50*, e *data* sia il solito tipo definito da utente che realizza la data del giorno (tripla $\langle g,m,a \rangle$).

Rappresentazione grafica come *array*

	0	1	2	...	49
materia					
voto					
lode					
data_es					

Esempio (programma completo C++ che realizza e gestisce una tabella)

Leggere una sequenza di dati relativi agli esami sostenuti da uno studente:

- titolo esame
- voto (in trentesimi, tra 18 e 30)
- eventuale lode
- data esame

fino a leggere (un esame con titolo) "stop": Memorizzare i dati letti in un'opportuna tabella `esami`, stampare la tabella letta, e quindi effettuare uno o più elaborazioni sulla tabella (ad es., ricerca dell'esame con il voto massimo) e stampare i relativi risultati.

```
#include <iostream>
#include <cstring>

using namespace std;

const int dim_tabella = 50; // capacita' max della tabella esami
const int dim_titolo = 100; // capacita' max del nome dell'esame
struct data {
    int giorno;
    int mese;
    int anno;
};
struct dati_esame {
    char materia[dim_titolo];
    int voto;
    bool lode;
    data data_esame;
};

int main()
{
    dati_esame esami[dim_tabella]; // tabella esami
    int i=0; // i: indice array esami;
    do {
        char titolo[dim_titolo];
        cout << "\nDai nome dell'esame (stop per smettere): ";
        cin.getline(titolo,dim_titolo);
        if (strcmp(titolo,"stop")==0) break;
        else strcpy(esami[i].materia,titolo);
```

```

int voto;
cout << "Dai voto in trentesimi: ";
do cin >> voto; while (voto < 0 || voto > 30);
esami[i].voto = voto;

char lode;
if (esami[i].voto == 30) {
    cout << "Lode? (s/n) ";
    do cin >> lode; while (lode != 'n' && lode != 's');
    if (lode == 'n') esami[i].lode = false;
    else esami[i].lode = true;
}

data data_esame;
cout << "Dai data esame (g m a): ";
cin >> data_esame.giorno
    >> data_esame.mese
    >> data_esame.anno;
// aggiungere controlli su validita' data !!
esami[i].data_esame = data_esame;
i++;
cin.ignore(256, '\n'); // elimina "a capo" successivo
} // ultimo intero immesso
while(i < dim_tabella);

int n=i;
// stampa della tabella dei voti
for (int i=0; i<n; i++) {
    cout << "\nMateria: " << esami[i].materia << endl;
    cout << "Voto: " << esami[i].voto;
    if (esami[i].voto == 30 && esami[i].lode) cout << " con lode";
    cout << endl;
    cout << "Data: " << esami[i].data_esame.giorno << '.'
        << esami[i].data_esame.mese << '.'
        << esami[i].data_esame.anno << endl;
}
cout << endl;

// calcolo del massimo (senza tener conto delle lodi)
int max=0;
for (int i=0; i<n; i++)
    if (esami[i].voto > max) max = esami[i].voto;

cout << "Il voto massimo e' " << max << endl << endl;

system("pause");
return 0;
}

```