8 - Tipi definiti dall'utente Enumerativi e Array

Tipi Ordinali

I tipi ordinali sono una categoria di tipi di dati in cui i valori sono ordinati e finiti. Questo significa che possiamo parlare di un “primo” valore, di un valore “successivo” e di un valore “precedente”.
Permettono operazioni come:

- Confronto
- Assegnamento
- Funzioni predecessore-successore
  - pred(x)
  - succ(y)

Gli esempi più comuni di tipi ordinali sono gli interi, i caratteri e i valori booleani.
Questi tipi sono predefiniti in molti linguaggi di programmazione, ma è possibile definirne di nuovi, come i tipi enumerativi e i sottocampi

Tipi enumerati

I tipi enumerativi sono definiti elencando esplicitamente tutte le possibili costanti del nuovo tipo
La specifica del dominio avviene per elencazione del nuovo tipo T
Esempio
Tipo T=($v1, v2, \ldots, v_n$)

v_n rappresentano i valori che una variabile dichiarata di quel tipo può assumere, questi insieme di valori sono definiti e ordinati;
Per ogni tipo T devono valere delle regole:

- I valori sono distinti,ogni valore è unico
- L'ordine dipende dall'elencazione
- Gli unici valori del tipo sono quelli elencati
  Generalemente questi valori occupano 1 byte nella memoria

Tipi Sottocampo (subrange)

Un tipo sottocampo è un intervallo di valori di un tipo ordinale esistente, cioè quando un dato può assumere un intervallo di valori che è un sottoinsieme dei valori descritti da un certo tipo ordinale esistente il suo tipo può essere definito come un subrange di quel tipo ospite, i suoi valori si definiscono specificando il più piccolo e il più grande dello stesso tipo

Esempi

✓Enumerativo
- tipo giorno : (lun, mar, mer, gio, ven, sab, dom)
  - x: giorno (dichiarazione della variabile con identificatore x e di tipo giorno)
  - x <- gio (assegnazione del valore gio alla variabile x)
  - se x < sab allora …
  - pred(gio) = mer
  - succ(gio) = ven

✓Subrange
- **tipo** feriale : [lun … sab] (tipo ospite è giorno)
  - y: feriale y=mar corretto y=dom non è corretto
- **tipo** paga : [1000 … 5000] (tipo ospite è integer)
  - z: paga z=2000 corretto z=6000 non è corretto

N.B. Non è permesso definire un subrange del tipo reale poichè esso non è un tipo ordinale

Dati Strutturati

I dati strutturati sono aggregati di dati elementari che vengono trattati come un'unica unità, indicati collettivamente da un unico nome. Si presuppone che tra essi esista una struttura, legata:

- All'organizzazione
- Al tipo di valori che compongono l'insieme
- Alle operazioni per estrarre dati dall'insieme
  È fondamentale il modo in cui i dati componenti vengono individuati, alcuni esempi includono tabelle, matrici, schede, etc.

Struttura di dati

Una struttura di dati è un insieme di dati correlati che possono non essere tutti dello stesso tipo, i dati sono legati ad un organizzazione ed è fondamentale il modo in cui i dati componenti vengono individuati.
La struttura di dati è costituita anche da un insieme di componenti su cui sono definiti un insieme di operatori:

- Selezionare un componente
- Cancellare un componente
- Aggiungere una componente

Variabili strutturate

I dati strutturati e le strutture di dati vengono memorizzati in variabili strutturate, una variabile strutturata possiede più di una componente, la dichiarazione prevede l'indicazione di nome e tipo della struttura, insieme al tipo delle sue componenti, alcune strutture sono già previste

dai linguaggi di programmazione poiché molto comuni, altri permettono di crearle da zero per tipi di dati personalizzati o strutture dati complesse

La **variabile strutturata** è quindi un agglomerato (significativo) in cui sono riuniti dati da elaborare con un particolare tipo, caratterizzato più dall'organizzazione imposta agli elementi componenti che dal tipo degli elementi stessi.
Prevede:

- Un modo sistematico per organizzare i dati
- Un insieme di operatori per manipolare gli elementi della struttura e/o aggregare elementi per costruirne altre

Strutture dati - Linearità, Accesso e Dimensione

Le strutture dati possono essere

- Lineari (array),può essere vista come una sequenza di dati in cui è possibile individuare le N componenti fino ad esaurire la lunghezza della sequenza.
- Non lineari (matrici)

Si possono avere poi modi differenti per poter accedere alle diverse componenti:

- Accesso diretto, possibilità di accedere ad ogni componente della struttura senza ciclare interamente ma prelevando l'identificatore di ogni componente
- Accesso sequenziale, iterazione attraverso il ciclo per visualizzare e/o esaminare ogni elemento della struttura
- Accesso diretto, che avviene solo ad alcuni componenti nelle strutture di dati nidificate o con dati complessi, dove possiamo avere accesso alle strutture o al dato complesso ma non direttamente ai dati presenti al loro interno

Le dimensioni delle strutture poi possono essere:

- Fissa, il numero delle componenti non può variare
- Variabile, il numero delle componenti può variare

Array/Vettore

Una struttura vera e propria è il vettore o array, un array permette di allocare un insieme di elementi dello stesso tipo in zone contigue della memoria.
Il vettore si caratterizza con una tabella monodimensionale con:

- Una struttura lineare
- Dimensione fissa
- Accesso diretto mediante un indice

Le operazioni consentite sono:

- Selezione (o lettura): reperire valore di un elemento
- Sostituzione (o scrittura): sostituire il valore di un elemento con un nuovo valore
  I componenti dell'array sono esplicitamente denotati tramite un selettore detto indice, non si è legati ad uno specifico indice

L'array è definito da:

- Tipo degli elementi
- Numero degli indici
- Tipo degli indici
  Ogni cella dell'array si comporta come una variabile tradizionale

Ogni elemento dell'array è individuato dal nome della variabile seguito dal numero della cella detto indice, la sua notazione è $NomeVettore[i]$.
Gli indici di un array vanno da 0 a N-1, poiché il primo elemento è identificato dalla cella 0 e così via.
L'indice è fondamentale anche per l'accesso ad una singola cella dell'array, che avviene tramite la specifica della posizione del componente, nel caso dell'access diretto vi è la stessa definizione precedente, nell'accesso ai valori singoli si usa una funzione:

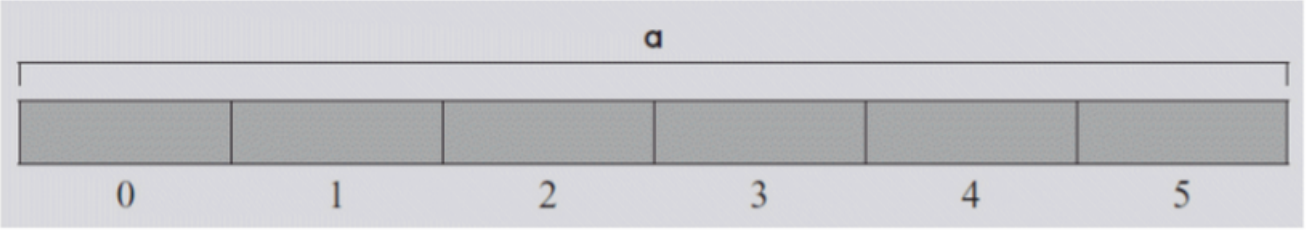
Data nota la posizione della prima componente I_0, la j-esima componente ha posizione $I_j = I_0 + J + S$, dove S è l'occupazione di memoria del tipo base

Un elemento di un array può a sua volta essere un tipo di dato strutturato, la dimensione di un array dipende dal numero di elementi, che deve essere dichiarata e non modificabile successivamente

Il tipo di dati contenuti di un array viene detto **tipo di array**, la sua dichiarazione avviene in questo modo:

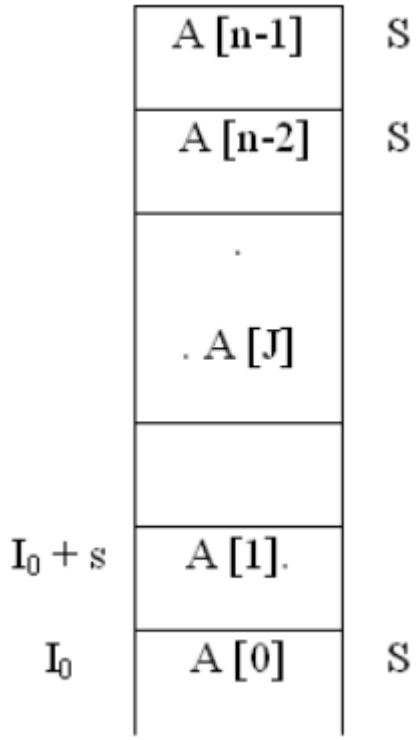
```
int a[6]
```

Si dichiara in primis il dato dell'array, ovvero il tipo che avranno i componenti al suo interno, successivamente indichiamo il nome della variabile e tra le parentesi quadre la lunghezza della struttura, il quale valore deve essere intero positivo



Dimostriamo su questa base la formula precedente dell'accesso ai singoli valori:
partiamo dall'indirizzo base I_0 e allochiamo il primo indirizzo in memoria in modo tale che i prossimi avvengano in modo contiguo.
L'indirizzo `a[0]` sarà I_0
L'indirizzo `a[1]` sarà I_0 + S

quindi l'indirizzo a[j] sarà $I_0 + J * S$



S occupa 1 componente
L'occupazione di memoria sarà quindi data da $n \cdot s$, dove n sono le celle di memoria.

Relativamente agli array

Dipende da un linguaggio di programmazione:

- La modalità di dichiarazione
- La modalità di definizione di variabilità dell'indice
- La disponibilità di operatori globali (agiscono su tutte le componenti della struttura e le componenti sono sempre elaborabili singolarmente tramite gli operatori del proprio tipo)
- Controllo di dimensionalità