

# 10 - Record

In molte situazioni, il tipo array non è adeguato per modellare le entità da trattare per colpa del vincolo della omogeneità (tutti i dati nell'array devono essere del tipo dichiarato).

I linguaggi di programmazione offrono un costruttore di tipo per aggregare attributi eterogenei, il tipo scheda (o struct), questo tipo permette, combinato con i puntatori, di essere utilizzato per la creazione di liste, pile, code e alberi ed è usato spesso per definire elementi da memorizzare nei file.

## Uso di schede

La scheda è una sorta di contenitore di campi di tipo disomogeneo che raggruppa dati più semplici (ne rende più ordinata la gestione evitando confusioni, l'elemento di una scheda a sua volta può essere di un tipo strutturato).

I campi della scheda non sono visibili direttamente, il loro nome deve essere preceduto da quello della scheda a cui appartengono, i singoli campi non sono identificatori (ma sono identificatori locali all'interno di una variabile di tipo strutturato, da usarsi come suffissi).

Le operazioni consentite in una struttura sono la selezione o lettura (reperimento del valore di un elemento) e sostituzione o scrittura (sostituzione del valore di un certo elemento con un nuovo valore).

## Rappresentazione della scheda

Le schede hanno le loro componenti allocate in posizioni di memoria contigue e nell'ordine in cui sono specificate nella dichiarazione, occupando una memoria complessiva pari alla somma di occupazione di ciascun campo (dati noti in fase di compilazione).

La posizione di ciascun campo all'interno del record è calcolata a partire dalla posizione iniziale del record e sommando le dimensioni dei campi precedenti.

## Record

Un record raggruppa le principali caratteristiche di un'entità in una **n-upla** (insieme ordinato di elementi), ha una struttura non lineare e una dimensione fissa.

Ogni record ha un numero definito di componenti, stabilito in fase di definizione. di tipi potenzialmente diversi, ciascuna esplicitamente denotata ed indirizzata tramite un selettore (campo), paragonabile ad una variabile ordinaria.

I campi sono identificati tramite un nome unico all'interno del record (selettore).

Ogni campo ha una descrizione composta da:

- Il tipo.
- I limiti di valore che può assumere.
- L'identificatore (nome del campo).

# Array vs Record

## Array

Gli **array** sono collezioni omogenee di dati. Questo significa che tutti gli elementi in un array devono essere dello stesso tipo (ad esempio, un array di numeri interi o di stringhe).

- **Caratteristiche principali:**

1. **Dimensione fissa:** Una volta dichiarata la lunghezza dell'array, questa non può essere modificata.
2. **Accesso diretto tramite indice:** Ogni elemento è accessibile tramite un indice numerico (es. `array[0]` , `array[1]` ).
3. **Sequenzialità:** Gli elementi sono memorizzati in memoria in posizioni contigue, con un ordine ben definito.

Gli array sono semplici da gestire e ideali per lavorare con dati dello stesso tipo e consentono operazioni veloci di accesso e modifica grazie agli indici.

## Record

I **record**, invece, sono progettati per raggruppare dati di tipi diversi in un'unica struttura. Ogni campo del record rappresenta un attributo specifico dell'entità.

- **Caratteristiche principali:**

1. **Tipi eterogenei:** I campi di un record possono avere tipi diversi, ad esempio un campo stringa per il nome, un campo intero per l'età, ecc.
2. **Accesso diretto tramite identificatore:** I campi sono accessibili tramite il nome del campo (es. `record.Nome` ).
3. **Più generale:** I record possono rappresentare entità complesse in modo chiaro e organizzato.

I record consentono di rappresentare oggetti complessi con più attributi e ogni campo ha un nome che migliora la leggibilità del codice e riduce gli errori, ma non possono essere facilmente manipolati come collezioni omogenee.

## Array e Record

Un array di record è una sequenza di record simili fra loro, ottenendo una struttura ordinata per gestire entità complesse, consentendo di gestire una collezione di entità complesse con attributi strutturati, mantenendo una struttura ordinata.

È possibile accedere direttamente sia al singolo record tramite indice, sia a un campo specifico all'interno di un record ed è facile aggiungere nuovi record senza modificare la struttura esistente (purché la dimensione sia sufficiente).