

7 - Dati

Tipo di dato

Il dato è un valore che la variabile può assumere, il tipo di dato è un modello matematico che indica i valori e le operazioni annesse.

L'attributo di una variabile individua:

- L'insieme dei valori che può assumere
 - L'insieme di operazioni effettuabili su di esso
 - Il modo in cui questi valori e operazioni sono memorizzati e rappresentati in memoria
- Un programma lavora su due tipi di dati:

- **Costanti**: se il loro valore non cambia mai nel tempo, il programma ci accede unicamente in lettura
- **Variabili**: se il loro valore cambia nel tempo, il programma può accedervi in lettura e scrittura

Da un punto tecnico tutto quello che deve essere elaborato dal microprocessore di un computer deve risiedere nella memoria RAM, bisogna sapere quanti bit la compongono e quale è la sua organizzazione all'interno

Dichiarazione di tipo

Le istruzioni operative di un programma operano su rappresentazioni delle entità di un programma, quindi occorrono delle istruzioni dichiarative che consentono di definire come rappresentare tali entità in variabili mediante una caratterizzazione che accade tramite **un nome e un tipo**

Dichiarando le variabili diamo definizione del **dominio** del programma, comprendiamo il funzionamento dell'algoritmo e verifichiamo la correttezza del programma.

Il tipo di dato definisce quanto spazio di memoria sarà necessario per la variabile utilizzata e che valori sono permessi (come il null), nei linguaggi moderni è permesso creare variabili con un insieme di tipo comune e nuovi tipo di dati.

Definizione di tipo

Ci sono istruzioni e costrutti linguistici per informare l'esecutore su il dominio di una variabile, l'insieme delle operazioni effettuabili su di essa e il modo a cui ci si può riferire ad essa.

A volte si ricorre all'indicazione esplicita dei valori permessi per ogni variabile (le costanti di tipo)

Tipi Primitivi

I moderni linguaggi di programmazione mettono a disposizione un insieme di tipi di uso più comune, chiamate predefinite, standard o primitivi

I tipi più comuni di variabili sono:

- **Interi**
- **Reali** (o float), sono discretizzati poiché è impossibile rappresentarli, non bisogna superare X bit
- **Logici**, che possono essere di tipo vero o falso (o comunemente di valore 1 per il vero o 0 per il falso). Nel C non esistono come tipo, per questo si usa un valore intero combinandoli con gli operatori (NOT, AND, OR in questo esatto ordine per priorità decrescente nel confronto)
- **Caratteri**
Esistono due tipi di valori rappresentabili:
 - Tipo numerico
 - Tipo alfanumerico

Espressioni

Un'espressione è una combinazione valida di operatori, variabili, costanti e funzioni la cui valutazione produce un valore oppure non termina, in questo caso si dice espressione indefinita.

Le espressioni, insieme ai comandi e alle dichiarazioni, costituiscono uno dei componenti base di ogni linguaggio di programmazione.

Si basa fortemente sull'algebra, con le regole usuali:

- Regola della parentesi: valutare le espressioni separatamente tra parentesi
- Regola della priorità dell'operatore: gli operatori binari hanno priorità maggiore a quelli aritmetici (% vs +)
- Regola dell'associatività: all'interno di un'espressione se la priorità è la stessa il calcolo inizia dall'operatore che sta più a sinistra e prosegue a destra

Tipizzazione del linguaggio

I linguaggi di programmazione sono detti a **tipizzazione forte** quando

- Dispongono di un determinato tipo di dati primitivi
- Consentono di definire nuovi tipi primitivi
- Consentono di costruire tipi strutturati offrendo diversi metodi di strutturazione (array, record, insieme di sequenza) e un certo numero di operatori

Il vantaggio di avere un linguaggio di programmazione fortemente tipizzato è quello di poter prevenire errori di tipo durante la compilazione come previene l'uso di dati in modi che non sono stati esplicitamente consentiti (come sommare delle stringhe con un operando aritmetico), riducendo così il rischio di bug difficili da individuare e migliorando la sicurezza e affidabilità del codice.

Vantaggi del concetto del tipo

Un vantaggio di concetto del tipo risiede nel:

- Sapere quanta memoria riservare alle variabili in base al loro tipo ($\text{char}=8 \text{ bit}, 2^8 = 256$)

Tipizzazione del C

In linea di massima il C persegue la tipizzazione forte, però è molto permissivo, infatti permette ad un operando che non è del tipo atteso subire una conversione automatica di tipo (cast implicito)

Per esempio `int i` e `float f` son due dati diversi, per poterli sommare il compilatore effettua la conversione di `i` in `float` e poi la somma

Tipi primitivi nel C

I tipi primitivi di C sono:

- Interi (`int` e i suoi modificatori `unsigned`, `long`, `short`)
- Reali (`float` e `double`)
- Logici (non disponibili in C)
- Caratteri alfanumerici (`char`)

Interi

La memorizzazione dei numeri interi è alla base della maggior parte dei linguaggi di programmazione. Gli interi possono essere positivi o negativi e le operazioni basilari includono somma, prodotto, differenza, quoziente e resto. Tuttavia, i valori interi non sono infiniti nei calcolatori e dipendono dall'architettura della CPU.

La capacità di rappresentare numeri interi dipende dal numero di bit utilizzati dalla CPU. Ad esempio, una CPU a 16 bit può rappresentare numeri da -32768 a 32767 (con segno) o da 0 a 65535 (senza segno).

I numeri relativi (interi con segno) richiedono la rappresentazione del segno. Questo viene fatto utilizzando un bit per il segno e i restanti bit per la grandezza del numero.

Reali

I numeri reali non possono essere rappresentati esattamente nei calcolatori e richiedono una rappresentazione approssimata. Esistono due forme principali di rappresentazione:

- **Virgola fissa**
- **Virgola mobile**

Il linguaggio C fornisce tre tipi di dati floating point:

- `float` : precisione singola
- `double` : precisione doppia

- `long double` : precisione estesa

In C, gli operatori per i dati `float`, `double` e `long double` includono `+`, `-`, `*`, `/`, operatori di uguaglianza e relazionali, e l'assegnamento. La libreria standard di C fornisce numerose funzioni matematiche per questi tipi di dati.

Logici

I valori logici rappresentano valori di verità (vero o falso), rappresentate come uniche costanti. Sono tipicamente usati nelle condizioni (selezioni e costruttori iterativi) e possono essere ottenuti come risultato di confronti o operatori logici.

In C, non esiste un tipo logico nativo. Si utilizza il tipo `int`, dove 0 rappresenta falso e ogni valore diverso da 0 rappresenta vero.

Caratteri

I caratteri sono memorizzati come numeri utilizzando codifiche standard come ASCII o Unicode. Ogni carattere ha una rappresentazione numerica univoca.

Il codice ASCII è una codifica che utilizza 1 byte per rappresentare i caratteri, permettendo di rappresentare fino a 256 valori. La forma ristretta su 7 bit riduce l'insieme a 128 valori.

Alcuni caratteri non sono stampabili, come l'invio o lo spazio. Inoltre, i caratteri delle cifre sono diversi dalle cifre stesse, ad esempio, `'7'` e `7` hanno rappresentazioni interne diverse.