

## Tipi di informazione

- Tipi *semplici* (atomici)

l'informazione **non** può essere ulteriormente decomposta

Es:

numero intero

carattere alfabetico

- Tipi *strutturati*

l'informazione può essere decomposta in tipi più semplici

Es:

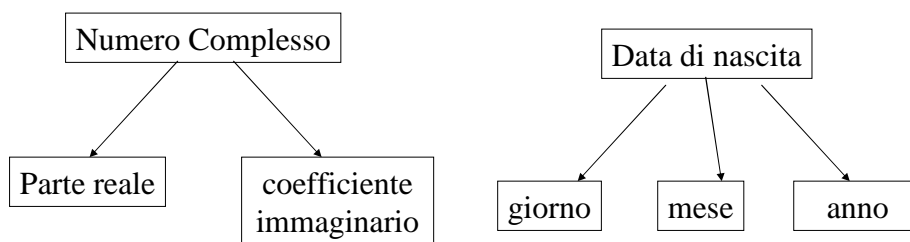
data (composta da: giorno, mese, anno)

numero complesso (parte reale, coefficiente immaginario)

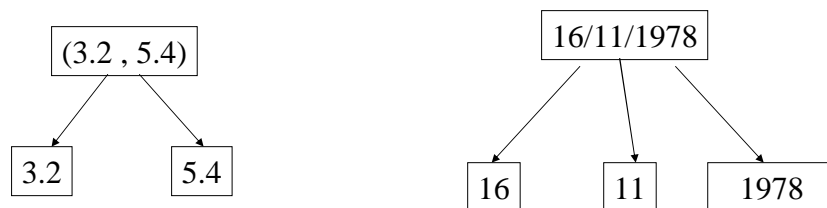
generalità anagrafiche (cognome, nome, data nascita, indirizzo)

## Tipi strutturati

**Struttura:** indica i tipi componenti e le relazioni tra essi



**Esemplare :** indica un valore dell'informazione strutturata



## Tipi semplici

- Un tipo è definito tramite
  - enumerazione
  - proprietà caratteristiche
  - operazioni

i vari linguaggi di programmazione hanno tipi predefiniti (primitivi)

## Il tipo Intero

tipo *intero* = {  $i$  |  $i$  è un numero intero,  $m < i < M$  }

**NON** coincide con l'insieme dei numeri interi:  $m \neq -\infty$ ,  $M \neq +\infty$

$m$  ed  $M$  variano col linguaggio di programmazione (e con l'architettura dell'elaboratore)

Esempi di  $m$  e  $M$ :

$m = -2^{15} = -32768$ ,  $M = (+2^{15} - 1) = +32767$   
tipo intero *short*

$m = (-2^{31}) = -2147483648$ ,  $M = (+2^{31} - 1) = +2147483647$   
tipo intero *long*

## Il tipo Intero

### **Operazioni:**

+ somma                      – sottrazione

\* moltiplicazione   / divisione

% modulo

Sono operazioni interne al tipo cioè il risultato assume ancora un valore di tipo intero

L'operatore % denota il resto della divisione intera.

Esempio:

supposto che:  $n1 = 12$    e    $n2 = 10$

$n1 \% n2 = 2$

La divisione intera ha come risultato un intero:  $n1 / n2 = 1$

## Il tipo Intero

Per le operazioni aritmetiche le precedenze sono quelle definite in matematica:

le operazioni \*, /, e % hanno precedenza sulle operazioni + e –

L'uso delle parentesi può cambiare l'ordine di valutazione

## Il tipo Intero

### *Attenzione:*

proprietà valide per i numeri interi non lo sono (sempre) per i tipi interi

Esempio:

l'uguaglianza tra le due espressioni

$$A + (B - C) \quad \text{e} \quad (A + B) - C$$

è vera per i numeri interi, ma *non* è (sempre) vera per i tipi interi

se il tipo intero è definito come [-10, 10] e si ha:

$$A=8, B=7, C=6$$

in questo caso la prima espressione vale 9,

ma la seconda è *indefinita*

... *OVERFLOW* ...

## Il tipo Intero

Il tipo intero è un tipo ordinato,

tra i suoi elementi esiste una relazione d'ordine (dato un suo elemento è possibile stabilirne il predecessore ed il successore)

Su un tipo ordinato sono definiti gli operatori di relazione:

= uguale	<> diverso
< minore	> maggiore
<= minore uguale	>= maggiore uguale

Le operazioni di confronto tra elementi del tipo intero, ottenute usando gli operatori di relazioni, restituiscono un risultato esterno al tipo

## Il tipo Reale

tipo *reale* =

{un sottoinsieme finito dell'intervallo di numeri reali  $[m, M]$ , tale che detto  $rs$  un qualsiasi numero reale con  $m \leq rs \leq M$ , esiste un valore  $r$  del tipo che lo approssima con precisione assegnata  
 $Abs[(rs-r)/r] < eps$ }

.... Ogni valore del tipo reale rappresenta tutti i componenti di un intervallo di numeri reali ...

- è un tipo primitivo
- è un tipo ordinato
- $m, M, eps$  variano col linguaggio e col processore  
cardinalità del tipo e precisione sono fortemente variabili  
(... *numero di bit e rappresentazione interna* ...)

## Il tipo Reale

in quasi tutti i linguaggi vengono messi a disposizione due tipi reali:

tipo reale semplice (float in C, C++)

tipo reale doppia precisione (double in C, C++)

*cardinalità e precisione del secondo sono maggiori di quelle del primo*

le operazioni base del tipo reale sono gli operatori aritmetici

interne al tipo: +, -, \*, /

*... naturalmente definite a meno di overflow ...*

operazioni di confronto, esterne al tipo

## Il tipo Carattere

- Il tipo carattere contiene l'insieme dei simboli con cui un elaboratore comunica verso il mondo esterno attraverso i dispositivi di input/output
- E' l'insieme dei caratteri riproducibili su unità di ingresso ed uscita (ad esempio la tastiera)
- prevale come tipo primitivo dei linguaggi l'insieme dei caratteri occidentali
- è possibile considerarlo come l'unione di quattro "sotto-tipi":

*Lettere Minuscole, Lettere Maiuscole, Cifre Decimali,  
Caratteri di Interpunzione e Speciali*

## Il tipo Carattere

Ordinamento:

- concettualmente, è un insieme parzialmente ordinato (non ha senso definire un ordine tra cifre e lettere), ma per motivi pratici si è imposto un ordine

$a < b < c < \dots < z$

$A < B < C < \dots < Z$

$0 < 1 < 2 < \dots < 9$

- non esiste ordinamento sugli altri caratteri

! " £ \$ % & / ( ) = ? ^ \* ° § ç ; : \_ , . - > < @ #

- stabilito un ordinamento totale sulla base codice ASCII

### Il tipo enumerativo

- Il tipo è definito enumerando gli elementi che lo compongono ... più corretto dire che si tratta di un costruttore di tipi di utente ...
- in particolare si possono definire tipi di utente semplici ed ordinali
- l'utente definisce il *NOME* del tipo che intende introdurre ed "enumera" in ordine crescente i valori (*costanti*) del tipo
- è possibile applicare gli operatori relazionali a variabili e costanti di un tipo enumerativo (operazioni esterne al tipo)
- Es:  
colore\_semaforo {ROSSO, GIALLO, VERDE};  
corso\_univ {FIS, INF, MAT, ECO};  
seme\_carta {coppe, denari, bastoni, spade};  
settimana {lunedì, martedì, mercoledì, giovedì, venerdì, sabato, domenica}

### Il tipo Logico (Boolean)

L'insieme del tipo booleano è costituito dai due valori di verità, *true* (vero) e *false* (falso):

Tipo booleano = (false, true)

E' un tipo ordinato: false < true

Operazioni = {OR, AND, NOT), operazioni interne al tipo

**OR:** disgiunzione logica o somma logica

risultato *vero* se *almeno un* operando è *vero*

**AND:** congiunzione o prodotto logico

risultato *vero* se *tutti* gli operandi sono pari a *vero*

**NOT:** negazione logica (operazione unaria)

risultato *vero* se l'operando è *falso* (e viceversa)

## Tipi ordinati

- Sono ordinati: interi, reali, caratteri
- Su un tipo ordinato sono definiti gli operatori di relazione
  - = uguale
  - $\neq$  diverso
  - < minore
  - $\leq$  minore o uguale
  - > maggiore
  - $\geq$  maggiore o uguale
- Restituiscono **0** quando il risultato è falso,
- **1** quando il risultato è vero