

# Operazioni

venerdì 29 ottobre 2021

01:23

Quale assegnamenti sono corretti?

- `Int x = 5.0` (errore)
- `Double pi = 0.314e1` -> Corretto
- `Public static final PI_GRECO = 3.14;`  
`PI_GRECO = 3.1415`
- `Public static final piGreco = 3.14159` -> Corretto

Indicare quali cast NON danno errore in compilazione

- `Int ova = 6;`  
`Long torte = uova;` -> Giusto
- `Short gambe = 4;`  
`Float dita = 10;`  
`Gambe = dita;`
- `Short gambe = 4;`  
`Float dita = 10.0;` -> Sbagliato, non si può fare questo assegnamento  
`Gambe = (short) dita;`
- `Char c = '5'`  
`Int x = (int) c;` -> Giusto

Qual è il valore della seguente espressione?

$9 * (3 / 9) - 3 * (2/6) = 2$  (no, è uguale 0 per qualche motivo lol)

Qual è il valore di

$(3*4)/4+(2*(4*5/2))-4(5/2) = 15$

Trovate le corrispondenze tra i diversi operatori booleani:

And -> `&&`

Or -> `||`

Not -> `!`

Quali sono i 6 operatori relazioni che possono essere usati nelle espressioni condizionali?

- `!=`
- `<`
- `>=`
- `==`
- `&&` -> Questo è un operatore logico, non relazionale
- `<=`
- `>`
- `!` -> Questo è un operatore logico, non relazionale

Stabilire il valore di verità della seguente espressione booleana:

`!((2+7) == (11-2)) && 4 < 6`

Qual è il valore della seguente espressione:

$5 + 200 * (4 / 8) - 7 * (-3+1) = 19$  (wtf)

Affinché l'istruzione

`x = ((y*z) < z) && x ;`

sia valida, di che tipo devono essere dichiarate le variabili x, y e z?

- X boolean, x e z double
- X boolean, y z int

Dato il seguente frammento di codice:

```
int i=1, j=2;
```

```
double u=2, v=2;
```

```
boolean b = true;
```

Quale dei seguenti assegnamenti è valido?

- `B=v`
- `J = 3.1 + i`
- `V = j/i*u` -> Vero
- `I = u+j*v`

(La divisione `j/i` avviene come divisione intera (senza resto) e la

moltiplicazione di un intero per un double promuove di tipo l'intero a

double e restituisce un double.

Dato che il tipo della variabile v è double,

l'assegnamento può essere

effettuato.

Infatti le promozioni di tipo (ovvero i passaggi da un tipo a un tipo

più "capiente") sono ammesse automaticamente,

mentre il passaggio a un

tipo meno "capiente" richiede un cast esplicito.)