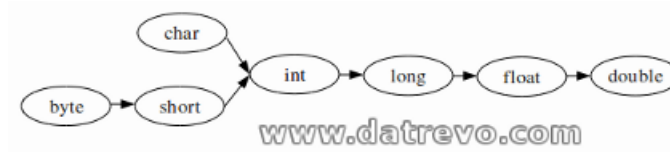


Le **operazioni di cast** sono fondamentali in ogni linguaggio di programmazione, così come nei linguaggi più diffusi, anche in Java esistono le conversioni di tipo che vengono fatte automaticamente ed altre che richiedono un cast esplicito. Con il termine **cast** si intende l'operazione di passaggio di una variabile da un tipo di dato ad un altro. Su questo argomento Java non differisce molto dal C, vediamo lo schema di conversione dei casting che possono avvenire in maniera esplicita:



Se c'è un percorso (basta seguire il verso delle frecce) nel [grafo](#) viene fatto un casting implicito, ovvero è possibile evitare di scrivere il cast, lo inserisce il compilatore automaticamente. Si può notare che il tipo **boolean** non fa parte del grafo, ciò significa che il tipo boolean non può essere mai convertito. Per i **litterali interi**, ovvero i numeri veri e propri, che rientrano nel range di valore è possibile evitare di scrivere un casting esplicito.

Esempi:

```
byte a = 5; // 5 rientra nel range di tipo byte, è quindi possibile evitare di scrivere il casting esplicito.
```

```
int b = a; // Esiste un percorso nel grafo da a a b quindi non c'è bisogno di un casting esplicito
```

```
int c = 5;
```

```
long d = c; // Esiste un percorso nel grafo da c a d quindi non c'è bisogno di un casting esplicito
```

```
long d=10; int c=d; // Errore, d non è un letterale
```

```
float c = 1.45; // Errore, sono per i litterali interi
```

Se non esiste un percorso nel grafo proposto, abbiamo bisogno di fare un casting esplicito (con probabile perdita di dati).

Esempi:

```
int a = 5;
```

```
byte b = a; // Errore, non esiste un percorso nel grafo
```

```
byte c = (byte) a; // Ok, non c'è neanche perdita di dati
```

```
byte d = (byte) 1000 // Ok, ma con perdita di dati (1000 non rientra nel range del tipo byte)
```