

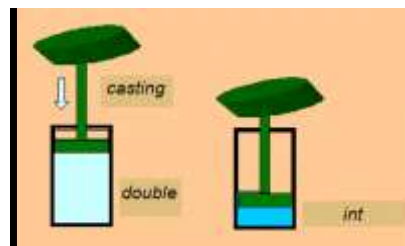
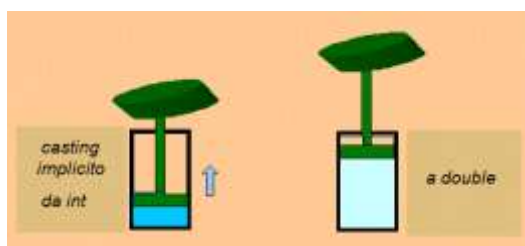
## Conversioni di tipo

La *conversione di tipo* è l'operazione con cui si converte una variabile da un tipo di dato ad un altro: questo passaggio è effettuato per avvantaggiarsi di alcune caratteristiche delle gerarchie dei tipi.

Per esempio, i valori di un intervallo limitato (come quello degli **interi**) possono essere immagazzinati in piccole quantità di memoria, per poi essere convertiti in un formato diverso che potenzialmente permette nuove operazioni, come la divisione con varie cifre decimali di precisione.



Esistono due tipi di *conversione di tipo*: **implicita** ed **esplicita**. Il termine – a volte – utilizzato per riferirsi alla conversione implicita è **coercion** (*coercizione* o *casting implicito* realizzato automaticamente dal compilatore), mentre il metodo più comune di conversione esplicita è chiamato **casting**<sup>1</sup>.



La **conversione esplicita** può essere eseguita anche con delle funzioni di conversione definite separatamente.



Tipico è il caso in cui il **tipo d'arrivo** è più “piccolo” di quello di **partenza**

Ad esempio si disporrà di *metodi* per convertire una variabile da tipo *stringa* a tipo numerico



<sup>1</sup> Vedremo in seguito che, per brevità, si usa il termine di *casting*, invece dei più appropriati *upcasting* e *downcasting*, anche quando si forza la conversione tra *oggetti*.

## Conversioni di tipo in JAVA

### • Tipi di dati :

- Otto tipi di dati PRIMITIVI: **interi** (byte, short, int, long)  
**reali** (float, double)  
**char** cioè caratteri Unicode 16 bit  
**boolean** cioè variabili con valore logico *true* o *false*
- Tipo Classe (esempio di dichiarazione della variabile cognome di tipo Classe [String](#) che si inizializza con Biasotti: `String cognome = "Biasotti";` )
- Tipo Interfaccia (specifiche astratte di comportamenti che una classe può implementare)
- Tipo [array](#) (serie di variabili componenti dello stesso tipo, oggetto della classe Object) esempio di dichiarazione e creazione di un oggetto array che contiene elementi di tipo String: `String nomi[] = new String [10];` oppure `String[] nomi = new String [10];`

### Conversioni di tipo

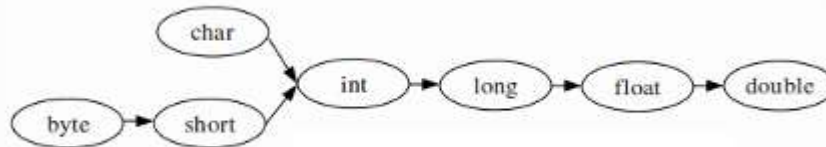
- Tra tipi primitivi:



se il tipo d'arrivo è più "grande" di quello di partenza avviene **casting implicito**: il compilatore lo inserisce automaticamente ad esempio



seguendo le frecce del *grafo* sotto riportato:



se il tipo d'arrivo è più "piccolo" di quello di partenza si converte un tipo primitivo *forzandolo* mediante un'operazione di **casting esplicito** con sintassi:

*(<nome\_tipo>) operando\_da\_convertire*

con perdita di informazione (non è possibile convertire da booleano ad altro tipo primitivo).

- Per convertire una variabile da tipo *stringa*<sup>2</sup> a tipo numerico si usano i metodi **parseInt()** della classe **Integer**, **parseFloat()** della classe **Float** o **parseDouble()** della classe **Double**.
- Per convertire una variabile da tipo numerico a tipo *stringa* si usa il metodo **valueOf()** della classe **String** oppure i metodi **toString** della classe *wrapper* opportuna (**Integer** per convertire da intero a tipo stringa etc...)

Per [approfondire](#)

<sup>2</sup> Errori in esecuzione se la variabile stringa non è solo numerica (es: "20") ma composta anche da caratteri alfabetici (es: "20anni")

```

/**
 * Conversioni.java application
 * tutte le classi wrapper (Integer, Double ...) e String
 * sono definite nel package java.lang (caricato di default)
 * @author 3INF
 */
public class Conversioni {
/**
 * metodo intToString
 * converte in stringa usando più modalità e stampa a video
 * @param i – numero intero
 */
public static void intToString(int i){
    System.out.println(i);
    System.out.println("Conversione implicita: " + i); // conversione implicita
                                                    // con accodamento

    String s = Integer.toString(i);
    System.out.println("Converto con Integer.toString: " + s);
    String s1 = String.valueOf(i);
    System.out.println("Converto con String.valueOf: " + s1);
}

/**
 * metodo doubleToString
 * converte in stringa usando più modalità e stampa a video
 * @param d – numero reale a doppia precisione
 */
public static void doubleToString(double d){
    System.out.println(d);
    System.out.println("Conversione implicita: " + d); // conversione implicita
                                                    // con accodamento

    String s = Double.toString(d);
    System.out.println("Converto con Double.toString: " + s);
    String s1 = String.valueOf(d);
    System.out.println("Converto con String.valueOf: " + s1);
}

/**
 * flusso principale
 */
public static void main(String[] args) {
    int intero = 10;
    intToString(intero);
    System.out.println("");
    double reale = 20.4;
    doubleToString(reale);
    int small = (int) reale; // operazione di casting esplicito solo tra tipi numerici
    System.out.println("\ncasting da reale a intero: " + small);
}
}

```

