

Può un metodo di Java modificare un proprio parametro ?

Eugenio G. Omodeo



UNIVERSITÀ
DEGLI STUDI DI TRIESTE

Trieste, 12/10/2015

ESERCIZIO:

Scrivere un metodo che ricevendo tramite il suo parametro un numero intero, raddoppi tale numero *nel* parametro.

ESERCIZIO:

Scrivere un metodo che ricevendo tramite il suo parametro un numero intero, raddoppi tale numero *nel* parametro.

Tener conto della preposizione “nel”

```
public static void raddoppia( int val ){  
  
    val = 2 * val;  
  
    return; // funziona?  
}
```

```
class Radd0 {  
  
    public static void main( String[] inutile ){  
  
        int wal = 2;  
  
        raddoppia( wal );  
  
        System.out.println( wal );  
    }  
}
```

eccetera

Perché stampa 2 ?

Forse a causa del nome del diverso nome della variabile ?

```
public static int raddoppia( int val ){  
    return 2 * val;  
}
```

```
class Radd1 {  
  
    public static void main( String[] inutile ){  
  
        int wal = 2;  
  
        wal = raddoppia( wal ); // rispetta le specifiche?  
  
        System.out.println( wal );  
    }  
}
```

eccetera

Ora stampa 4 !

Forse c'è una difformità: viene raddoppiato il val. *del* parametro

ALLORA PROVIAMO CON UN *array*... -I

```
public static void raddoppia( int[] val ){  
  
    int zal[] = { 2 * val[ 0 ] };  
  
    val = zal;  
  
    return;  
}
```

```
class Radd2 {  
  
    public static void main( String[] inutile ){  
  
        int[] wal = { 2 };  
  
        raddoppia( wal );  
  
        System.out.println( wal );  
  
        System.out.println( wal[ 0 ] );  
    }  
}
```

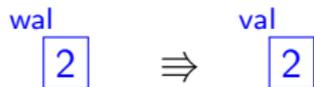
eccetera

Stampa di nuovo 2 !

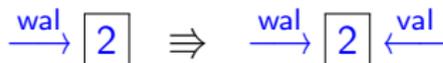
Abbiamo davvero cambiato rispetto a `Radd0` ?

CHE SIGNIFICA COPIARE ?

- 1 Come si copia un **int**:



- 2 Come si copia un **int[]**



```
public static void raddoppia( int[] val ){  
    val[ 0 ] = 2 * val[ 0 ];  
}
```

```
class Radd3 {  
  
    public static void main( String[] inutile ){  
  
        int[] wal = { 2 };  
  
        raddoppia( wal );  
  
        System.out.println( wal );  
  
        System.out.println( wal[ 0 ] );  
    }  
}
```

eccetera

Stampa davvero 4 !

Questa soluzione soddisfa i requisiti