

VALUTAZIONE A CIRCUITO-CORTO DELLE ESPRESSIONI BOOLEANE

espressioni aritmetiche

$$3 + 5 - 2 * 4 \rightsquigarrow 0$$

espressioni booleane

$$\underbrace{3 == 5} \text{ || } \underbrace{1 == 1}$$

false || true

true

~~$$\underbrace{3 == 5} \text{ \&\& } \underbrace{1 == 1}$$

false \&\& true

false~~

$$\underbrace{3 == 5} \text{ \&\& } \underbrace{1 == 1}$$

false \&\& ?

false

In realtà $1 == 1$ non lo calcola

$$\boxed{3 == 5 \text{ \&\& } 2/0 == 1}$$

false

false

Non viene valutato perché è inutile, Sappiamo già che il risultato è falso

$$\underbrace{3 == 5} \text{ || } \underbrace{2/0 == 1}$$

false

ERRORE

Questo meccanismo per calcolare l'AND e l'OR di espressioni booleane si chiama VALUTAZIONE A CIRCUITO-CORTO.

Viene usato SOLO per AND e OR.

$$\boxed{3 == 3 \text{ || } 2/0 == 1}$$

true

true

Non valute $2/0 == 1$ perché inutile

$$0 \cdot \sqrt{\ln 5 - \sqrt[3]{37}} = 0$$

UN ESSERE UMANO, USANDO LA LA NAUTA A CIRCO CIRCUITO, CONCLUDE CHE IL RISULTATO È ZERO SENZA CALCOLARE LA PARTE SOTTO RADICE.

OVERLOADING (SOVRACCARICO) DI METODI

RICORSIONE

Un algoritmo ricorsivo è un algoritmo che richiama se stesso su una versione ridotta dell'input.

DEFINIZIONE ITERATIVA

$$n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot n$$

$$n! = \begin{cases} 1 & \text{se } n=1 \\ (n-1)! \cdot n & \text{se } n>1 \end{cases}$$

$$\begin{aligned} 1! &= 1 \\ 2! &= 1 \cdot 2 = 1! \cdot 2 \\ 3! &= 1 \cdot 2 \cdot 3 \\ 4! &= 1 \cdot 2 \cdot 3 \cdot 4 \end{aligned}$$

DEFINIZIONE RICORSIVA

Conto alla rovescia (risolto in maniera ricorsiva)

Scrivere un metodo che prende in input un numero n, e stampa tutti i numeri da n fino a 0.

n=4	4	n=2	2
	3		1
	2		0
	1		
	0		

" Per fare il conto alla rovescia e partire da n, è sufficiente stampare n e poi fare un conto alla rovescia e partire da n-1"

```

7 public static void conto(int n) {
8     if (n == 0)
9         // caso base
10        System.out.println(0);
11    else {
12        // casi ricorsivo
13        System.out.println(n);
14        conto(n-1);
15    }
16 }
17
18 Run | Debug
19 public static void main(String[] args) {
20     conto(3);
21 }

```

Args: ~~MAIN~~

~~M: 3
Indirizzo RITORNO: CONTO RIGA 13~~

~~M: 2
Indirizzo RITORNO: CONTO RIGA 14~~

~~M: 1
RITORNO: CONTO RIGA 14~~

~~M: 0
Indirizzo di ritorno: CONTO RIGA 14~~

OUTPUT:

3
2
1
0

```

7 public static void conto(int n) {
8     if (n == 0)
9         // caso base
10        System.out.println(0);
11    else {
12        // casi ricorsivo
13        System.out.println(n);
14        conto(n-1);
15        System.out.println("Ciao");
16    }
17 }
18
19 Run | Debug
20 public static void main(String[] args) {
21     conto(2);
22 }

```

Args: ~~MAIN~~

~~M: 2
Ind. ritorno: RIGA 20 CONTO~~

~~M: 1
Ind. ritorno: RIGA 14 CONTO~~

~~M: 0
Ind. ritorno: RIGA 14 CONTO~~

OUTPUT:

2
1
0
Ciao
Ciao

```
10 public static int fattorialeRec(int n) {
11     if (n == 1)
12         return 1;
13     else
14         return n * fattorialeRec(n - 1);
15 }
16
17 Run | Debug
18 public static void main(String[] args) {
19     System.out.println(fattorialeRec(3));
20 }
```

OUTPUT: 6

