

Context Switch (Cambio di Contesto)

1. Cos'è un Context Switch

Un **context switch** (cambio di contesto) è il passaggio dell'esecuzione da un punto del programma a un altro, salvando lo stato corrente e caricando quello nuovo.

In termini semplici:

Quando il programma "salta" da un metodo a un altro, deve ricordare dove era e cosa stava facendo, per poterci tornare dopo.

```
public static void main(String[] args)
{ // Contesto 1
    int x = 5;
    int risultato = raddoppia(x); // CONTEXT SWITCH → verso raddoppia
    System.out.println(risultato);
}

static int raddoppia(int n)
{ // Contesto 2
    int doppio = n * 2;
    return doppio; // CONTEXT SWITCH → ritorno al main
}
...
```

Cosa Succede Durante il Context Switch

Fase 1: Prima della Chiamata (main attivo)

...

Stack:

main

x = 5	
risultato = ?	

...

Fase 2: Chiamata a raddoppia (context switch OUT dal main)

...

Salva:

- Indirizzo di ritorno (riga dopo la chiamata)
- Variabili locali del main (x, risultato)
- Puntatore allo stack frame

Prepara:

- Nuovo stack frame per raddoppia
- Copia il valore di x nel parametro n

...

Fase 3: Esecuzione di raddoppia (raddoppia attivo)

...

Stack:

raddoppia	← contesto attivo
n = 5	
doppio = 10	
main	← contesto sospeso
x = 5	
risultato = ?	

...

Fase 4: Return (context switch IN verso main)

...

Salva:

- Valore di ritorno (10)

Ripristina:

- Stack frame del main
- Indirizzo di ritorno
- Variabili locali del main

Rimuove:

- Stack frame di raddoppia (distrutto)
- ...

Fase 5: Dopo il Return (main attivo)

...

Stack:

main		
x = 5		
risultato = 10		← valore ricevuto

3. Lo Stack delle Chiamate (Call Stack)

Lo **stack** è la struttura dati che gestisce i context switch.

Caratteristiche dello Stack

LIFO (Last In, First Out):

- L'ultimo metodo chiamato è il primo a terminare
- Come una pila di piatti: aggiungi in cima, togli dalla cima

```
public static void main(String[] args) {  
    System.out.println("Inizio main");  
    metodoA();  
    System.out.println("Fine main");  
}
```

```
static void metodoA() {
```

```

    System.out.println("Inizio A");
    metodoB();
    System.out.println("Fine A");
}

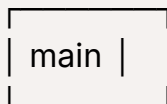
static void metodoB() {
    System.out.println("Inizio B");
    metodoC();
    System.out.println("Fine B");
}

static void metodoC() {
    System.out.println("Sono in C");
}
...

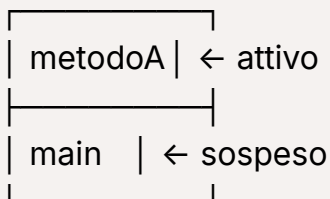
```

Evoluzione dello Stack
...

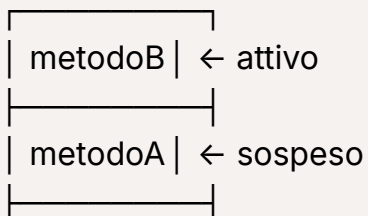
1. main inizia



2. main chiama metodoA (context switch)



3. metodoA chiama metodoB (context switch)



| main | ← sospeso

4. metodoB chiama metodoC (context switch)

| metodoC | ← attivo
| metodoB | ← sospeso
| metodoA | ← sospeso
| main | ← sospeso

5. metodoC termina (context switch)

| metodoB | ← riattivato
| metodoA | ← sospeso
| main | ← sospeso

6. metodoB termina (context switch)

| metodoA | ← riattivato
| main | ← sospeso

7. metodoA termina (context switch)

| main | ← riattivato

8. main termina

Stack vuoto

...

****Output del programma:****

...

Inizio main

Inizio A

Inizio B

Sono in C

Fine B

Fine A

Fine main