

Thread in Java

Un thread in Java ha:

- stack proprio;
- heap condiviso;
- scheduling gestito dalla JVM (prime versioni) o dal sistema operativo (consente esecuzione parallela, comportamento osservabile differente a seconda dell'SO).

I thread sono oggetti che estendono la classe astratta **Thread**, implementando il metodo **void run()**. È fornito il metodo **start()** per avviare il thread in modo concorrente (**run** lo esegue sequenzialmente).

Alternativamente, una classe che implementa l'interfaccia **Runnable** fornendo il metodo **void run()** può essere passata come parametro al costruttore **Thread**.

Stato di un thread

ready da **start** in poi, quando non è né bloccato né in esecuzione (in attesa di tempo di esecuzione dallo scheduler);

blocked **Thread.sleep(n)**, **t.join()**;

running

Esempio

```
// classe anonima
Runnable r = new Runnable() {
    public void run() { ... }
}

// lambda expression
Runnable u = () -> { ... };

Thread t = new Thread(r);
Thread s = new Thread(s);

// class ConcurrentTimer extends Thread { void run() { ... } }
Thread v = new ConcurrentTimer();

t.start();
s.start();
v.start();

try {
    Thread.sleep(1000);
} catch (InterruptedException ie) {
    ...
}

t.join();
s.join();
v.finish();
```