

# Java Threads e Blocchi/Metodi synchronized

Si crei un programma in Java che contenga due classi: MainTester e MioThread.

MainTester ha un attributo intero **pubblico** chiamato “maxSleep”, uno **privato** chiamato “sleepCounter” ed un costruttore con un parametro intero chiamato “n”. L’accesso dall’esterno a “sleepCounter” è quindi regolato da metodi implementati con gli **opportuni modificatori di accesso**. MainTester è la classe che contiene il main.

MioThread eredita da Thread e il suo costruttore prende 2 parametri: un id del thread (intero) ed un riferimento all’oggetto di classe MainTester che lo ha creato.

MainTester crea n threads di tipo MioThread. E li lancia tutti. Ciascun MioThread dorme per un numero di millisecondi “numCasuale” pari a 100+NumeroRandom(0-50). Quando si sveglia, accede a sleepCounter incrementandone il valore. Inoltre, sostituisce il valore di maxSleep se il tempo passato a dormire “numCasuale” è più grande del valore corrente di maxSleep; tale sostituzione deve avvenire in mutua esclusione definita da MioThread (non tramite un metodo synchronized di MainTester).

Il MainTester **aspetta** che tutti i thread abbiano finito per poi stampare il valore di maxSleep ed il valore del contatore di thread che si sono svegliati.

NOTE:      `Random rnd = new Random(seed);      int numCasuale = rnd.nextInt(maxValue);`