

/\*\* P184   【補足】 インナークラスでスレッドを動かす

\* ファイル名… ThreadTimer2.java

\* コンパイル… javac ThreadTimer2.java

\* 実行       … java ThreadTimer2

\*/

```
class ThreadTimer2 {
```

```
    public static void main(String[] args) {
```

```
        long startTime = 0; // 計測開始時間
```

```
        long stopTime = 0; // 計測終了時間
```

```
        // Thread クラスを拡張した内部(インナー)クラスの宣言
```

```
        class ThreadA extends Thread {
```

```
            @Override
```

```
            public void run() {
```

```
                System.out.println("(-_-)zzz…\n");
```

```
                try {
```

```
                    // 適当な時間 sleep させる
```

```
                    sleep((long)(Math.random()*9+1)*1000);
```

```
                } catch (InterruptedException e) {
```

```
                }
```

```
                System.out.println("(;ΦωΦ) ハッ!");
```

```
            } // run メソッド
```

```
        } // ThreadA(内部クラス)
```

```
        // 内部クラスのインスタンス生成
```

```
        ThreadA th = new ThreadA();
```

```
        // スレッド実行開始
```

```
        th.start();
```

```
System.out.println("計測開始");

// currentTimeMillis()で現在(開始時)の時間を取得(戻り値は long 型)
// (currentTimeMillis は 1970 年元旦 0 時から現在までをミリ秒で表す)
startTime = System.currentTimeMillis();


// run メソッド終了まで main メソッド一時停止
try {
    th.join();
} catch (InterruptedException e) {
}


// currentTimeMillis()で現在(終了時)の時間を取得
stopTime = System.currentTimeMillis();
System.out.println("計測終了");
System.out.println("sleep 時間は、" + ((stopTime - startTime) / 1000) + " 秒でした");


} // main


} // ThreadTimer2 クラス
```