

## Tutorial: Start Threading

### 1. MainThread.java

```
public class MainThread {  
  
    public static void main(String[] args) {  
        System.out.println(Thread.currentThread().getName());  
        System.out.println(Thread.currentThread().getId());  
    }  
}
```

### 2. MyThread

```
class MyThread extends Thread {  
  
    public void run() {  
        System.out.println("running...");  
    }  
  
    public static void main(String args[]) {  
MyThread t1 = new MyThread();  
        MyThread t2 = new MyThread();  
        System.out.println("Name of t1:" + t1.getName());  
        System.out.println("Name of t2:" + t2.getName());  
        System.out.println("id of t1:" + t1.getId());  
  
        t1.start();  
t2.start();  
  
        t1.setName("STIW3054");  
        System.out.println("After changing name of t1:" + t1.getName());  
    }  
}
```

### 3. MyThreadThread

```
public class MyThreadThread extends Thread {
```

```

    public static void main(String[] args) {
    new Thread(new MySleep()).start();
    new Thread(new MySleep()).start();
    }

```

```

    @Override
    public void run() {
        try {
            for (int x = 0; x < 1000; x++) {
                System.out.println(x);
                sleep(2000);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

#### 4. MySleep

```

public class MySleep extends Thread {

    public static void main(String[] args) {
        new Thread(new MySleep()).start();
    }
}

```

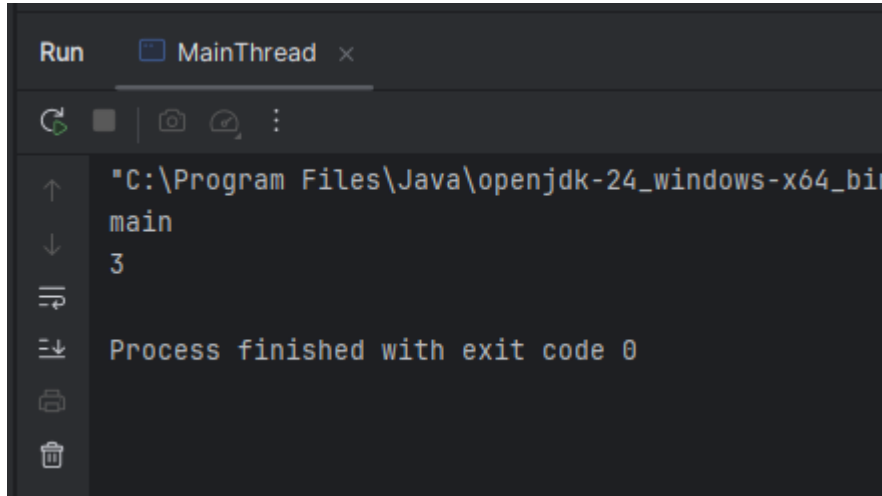
```

    @Override
    public void run() {
        try {
            for (int x = 0; x < 1000; x++) {
                System.out.println(x);
                sleep(2000);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

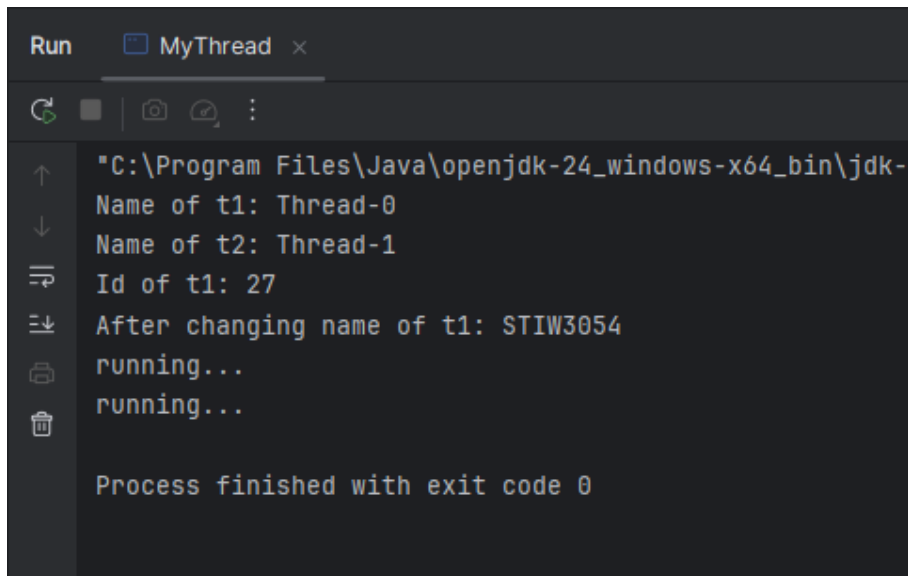
Output

MainThread.java



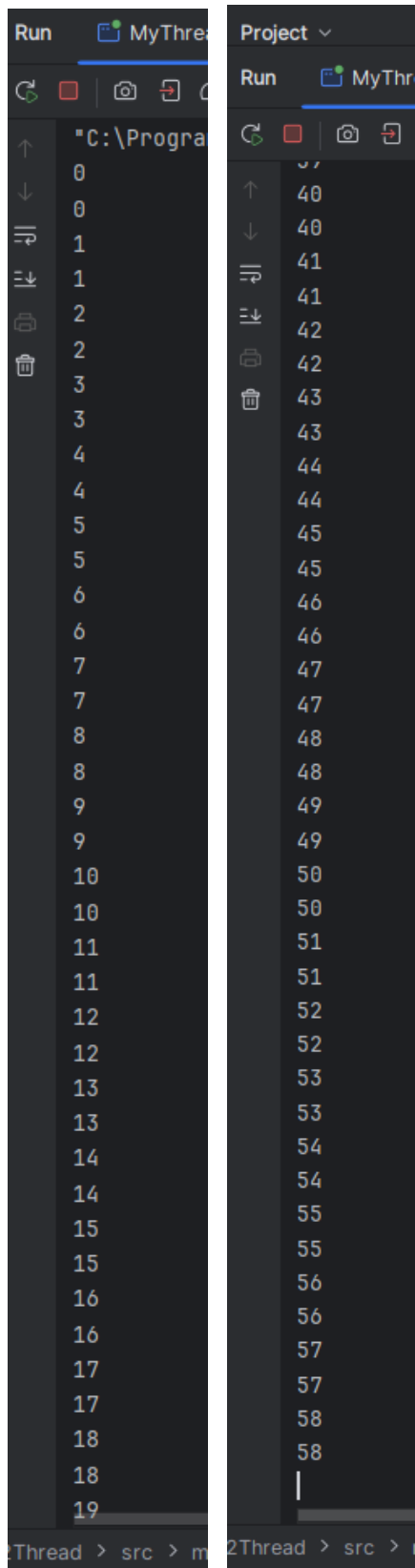
```
Run    MainThread x
C:\Program Files\Java\openjdk-24_windows-x64_bin\jdk-24\bin\java.exe
main
3
Process finished with exit code 0
```

MyThread.java



```
Run    MyThread x
C:\Program Files\Java\openjdk-24_windows-x64_bin\jdk-24\bin\java.exe
Name of t1: Thread-0
Name of t2: Thread-1
Id of t1: 27
After changing name of t1: STIW3054
running...
running...
Process finished with exit code 0
```

## MyThreadThread.java



The image displays two side-by-side screenshots of an IDE's Run console, showing the output of a multi-threaded Java program. The left screenshot shows the output of the first thread, and the right screenshot shows the output of the second thread. Both threads appear to be performing a similar operation, possibly calculating Fibonacci numbers, as the sequence of values matches the Fibonacci sequence.

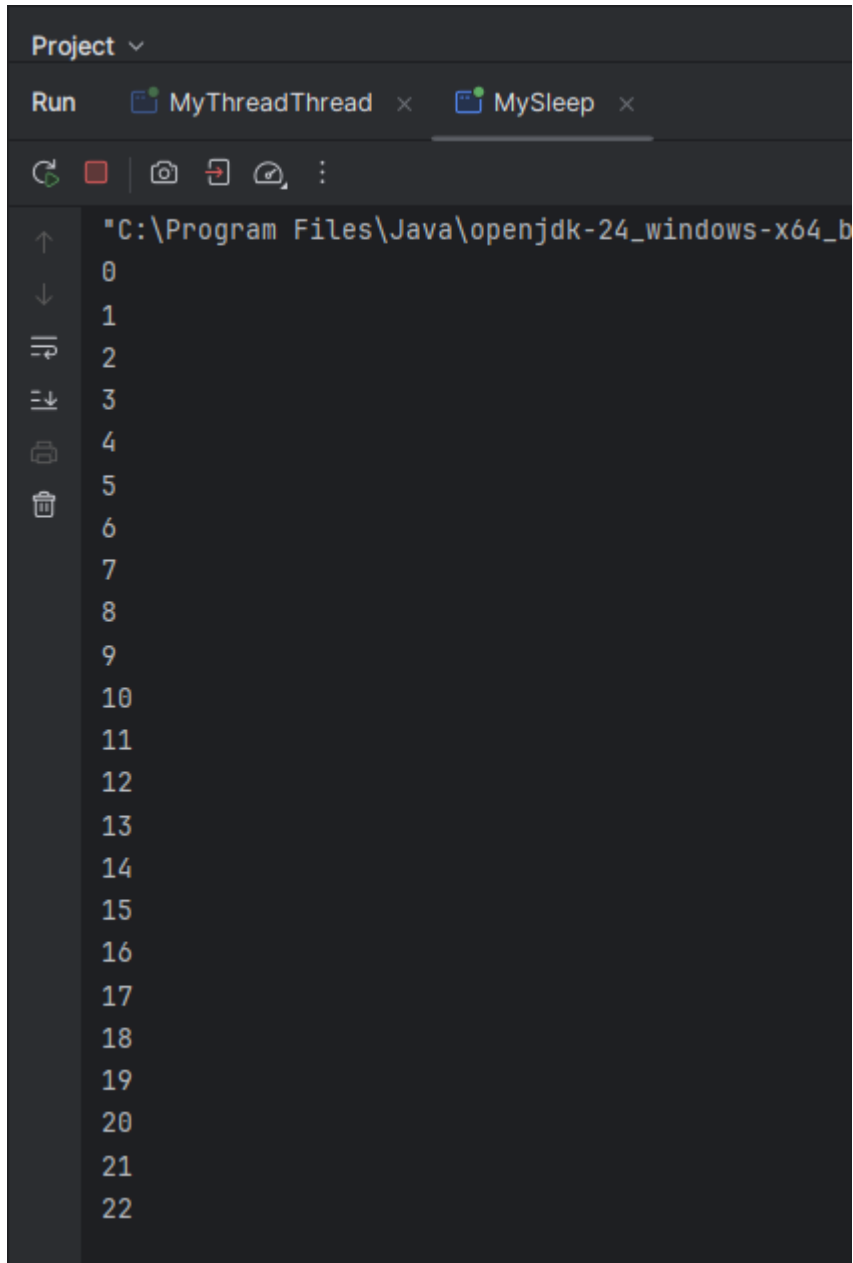
**Left Screenshot (Thread 1):**

```
0
0
1
1
2
2
3
3
4
4
5
5
6
6
7
7
8
8
9
9
10
10
11
11
12
12
13
13
14
14
15
15
16
16
17
17
18
18
19
```

**Right Screenshot (Thread 2):**

```
40
40
41
41
42
42
43
43
44
44
45
45
46
46
47
47
48
48
49
49
50
50
51
51
52
52
53
53
54
54
55
55
56
56
57
57
58
58
```

MySleep.java



```
Project ▾
Run MyThreadThread × MySleep ×
C:\Program Files\Java\openjdk-24_windows-x64_b
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
```