

com.example.project.InnerClass >

JavaApplication1 (run) × JavaApplication1 (run) #2 ×

```
run:
Hello from InnerClass
BUILD SUCCESSFUL (total time: 0 seconds)
```

<default config>

422.5/517.0MB

Services

es
ackage>
ole.project
e

<empty>

args)

Start Page × AwtExample.java × Main.java ×

Source History

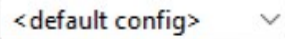
```
1 package com.example.project;
2
3 public class Main {
4
5     public static void main(String[] args) {
6         System.out.println("Hello from a package!");
7     }
8 }
```

com.example.project.Main >

Output ×

JavaApplication1 (run) × JavaApplication1 (run) #2 ×

run:
Hello from a package!
BUILD SUCCESSFUL (total time: 0 seconds)



Services

```
es
ackage>
mple.java
ss.java
ole.project
e
```

▼ <empty>

read

args)

Start Page × AwtExample.java × Main.java × NewClass.java ×

Source

History

```

1 class ThreadDemo extends Thread {
2     @Override
3     public void run() {
4         try {
5             // Moving thread to Timed Waiting state
6             Thread.sleep(150);
7         } catch (InterruptedException e) {
8         }
9         System.out.println("State after completion: " + Thread.currentThread().getState());
10    }
11
12    public static void main(String[] args) throws InterruptedException {
13        ThreadDemo t1 = new ThreadDemo();
14        System.out.println("State when created: " + t1.getState());
15        t1.start();
16        System.out.println("State when started: " + t1.getState());
17        // waiting for thread to die
18        t1.join();
19        System.out.println("State after thread ended its task: " + t1.getState());
20    }
21 }

```

ThreadDemo > run > try > catch InterruptedException e >

Output \times

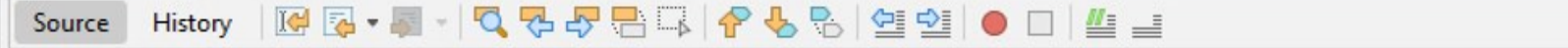
JavaApplication1 (run) × JavaApplication1 (run) #2 ×

```
run:
State when created: NEW
State when started: RUNNABLE
State after completion: RUNNABLE
State after thread ended its task: TERMINATED
BUILD SUCCESSFUL (total time: 0 seconds)
```

JavaApplication1 (run)

running_

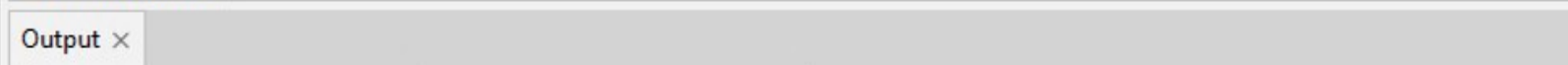
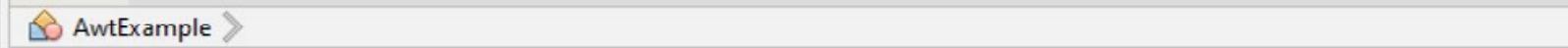
8:1



```

1  import java.awt.*;
2  import java.awt.event.*;
3
4  public class AwtExample extends Frame {
5
6      public AwtExample() {
7          // create a button
8          Button b = new Button("Click Me");
9          b.setBounds(100, 100, 80, 30); // setting button position
10
11         // add button to the frame
12         add(b);
13
14         setSize(300, 300); // frame size 300 width and 300 height
15         setLayout(null); // no layout manager
16         setVisible(true); // now frame will be visible
17
18         // close the frame when close button is clicked
19         addWindowListener(new WindowAdapter() {
20             public void windowClosing(WindowEvent e) {

```



run:

