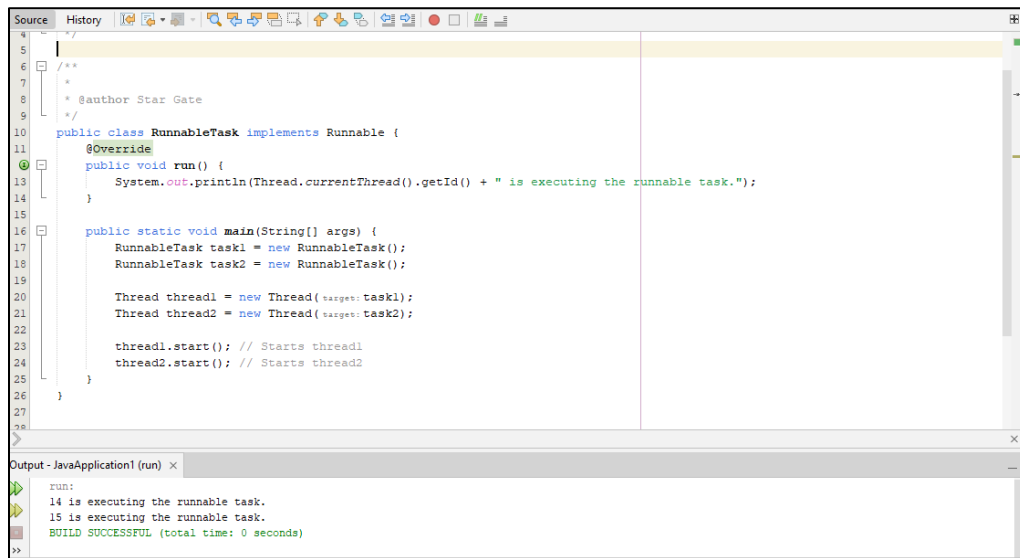


- RunnableTask



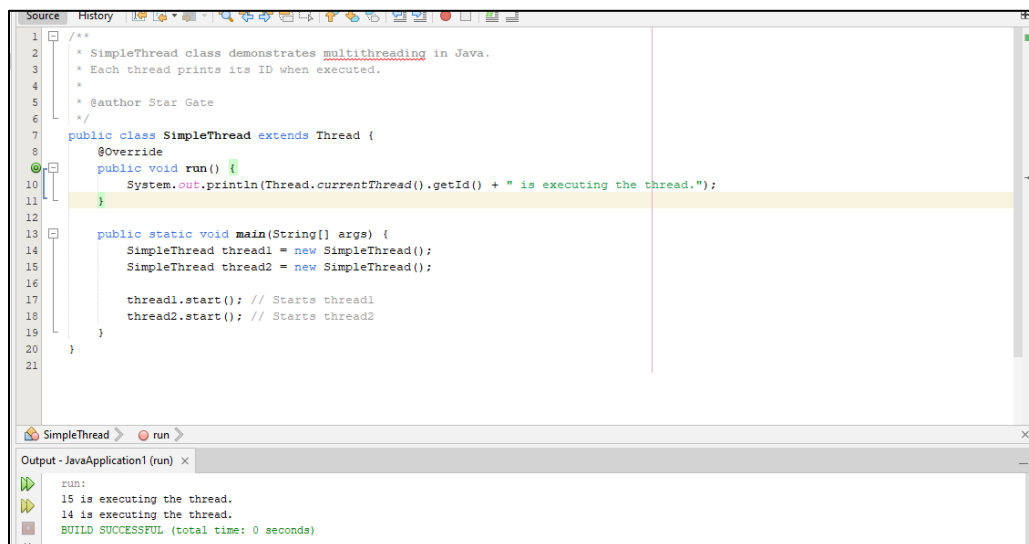
The screenshot shows an IDE with a source editor and an output console. The source editor contains the following Java code:

```
1  /**
2   *
3   * @author Star Gate
4   */
5
6  public class RunnableTask implements Runnable {
7      @Override
8      public void run() {
9          System.out.println(Thread.currentThread().getId() + " is executing the runnable task.");
10     }
11
12     public static void main(String[] args) {
13         RunnableTask task1 = new RunnableTask();
14         RunnableTask task2 = new RunnableTask();
15
16         Thread thread1 = new Thread(task1);
17         Thread thread2 = new Thread(task2);
18
19         thread1.start(); // Starts thread1
20         thread2.start(); // Starts thread2
21     }
22 }
```

The output console shows the following output:

```
run:
14 is executing the runnable task.
15 is executing the runnable task.
BUILD SUCCESSFUL (total time: 0 seconds)
```

- Simple Thread



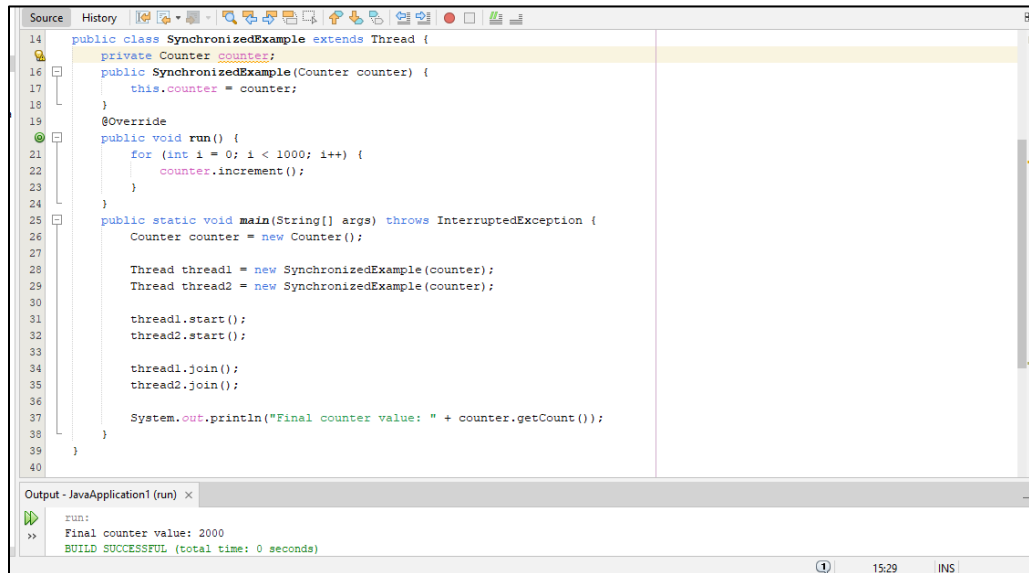
The screenshot shows an IDE with a source editor and an output console. The source editor contains the following Java code:

```
1  /**
2   * SimpleThread class demonstrates multithreading in Java.
3   * Each thread prints its ID when executed.
4   *
5   * @author Star Gate
6   */
7
8  public class SimpleThread extends Thread {
9      @Override
10     public void run() {
11         System.out.println(Thread.currentThread().getId() + " is executing the thread.");
12     }
13
14     public static void main(String[] args) {
15         SimpleThread thread1 = new SimpleThread();
16         SimpleThread thread2 = new SimpleThread();
17
18         thread1.start(); // Starts thread1
19         thread2.start(); // Starts thread2
20     }
21 }
```

The output console shows the following output:

```
run:
15 is executing the thread.
14 is executing the thread.
BUILD SUCCESSFUL (total time: 0 seconds)
```

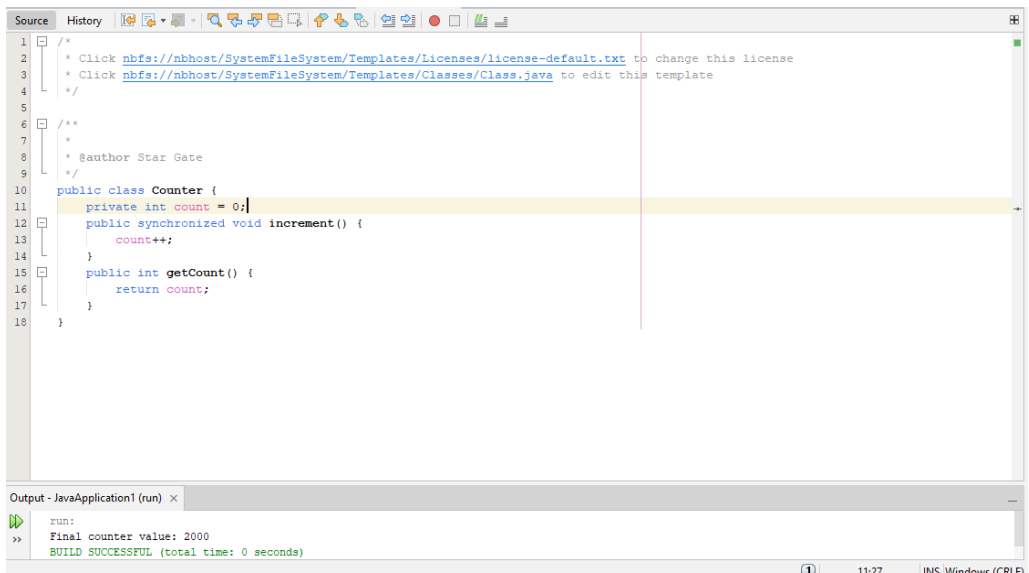
- Synchronizing Thread



```
14 public class SynchronizedExample extends Thread {
15     private Counter counter;
16     public SynchronizedExample(Counter counter) {
17         this.counter = counter;
18     }
19     @Override
20     public void run() {
21         for (int i = 0; i < 1000; i++) {
22             counter.increment();
23         }
24     }
25     public static void main(String[] args) throws InterruptedException {
26         Counter counter = new Counter();
27
28         Thread thread1 = new SynchronizedExample(counter);
29         Thread thread2 = new SynchronizedExample(counter);
30
31         thread1.start();
32         thread2.start();
33
34         thread1.join();
35         thread2.join();
36
37         System.out.println("Final counter value: " + counter.getCount());
38     }
39 }
40
```

Output - JavaApplication1 (run) ×

```
run:
>> Final counter value: 2000
BUILD SUCCESSFUL (total time: 0 seconds)
```



```
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4   */
5
6  /**
7   *
8   * @author Star Gate
9   */
10 public class Counter {
11     private int count = 0;
12     public synchronized void increment() {
13         count++;
14     }
15     public int getCount() {
16         return count;
17     }
18 }

```

Output - JavaApplication1 (run) ×

```
run:
>> Final counter value: 2000
BUILD SUCCESSFUL (total time: 0 seconds)
```

- Thread Pool Example

