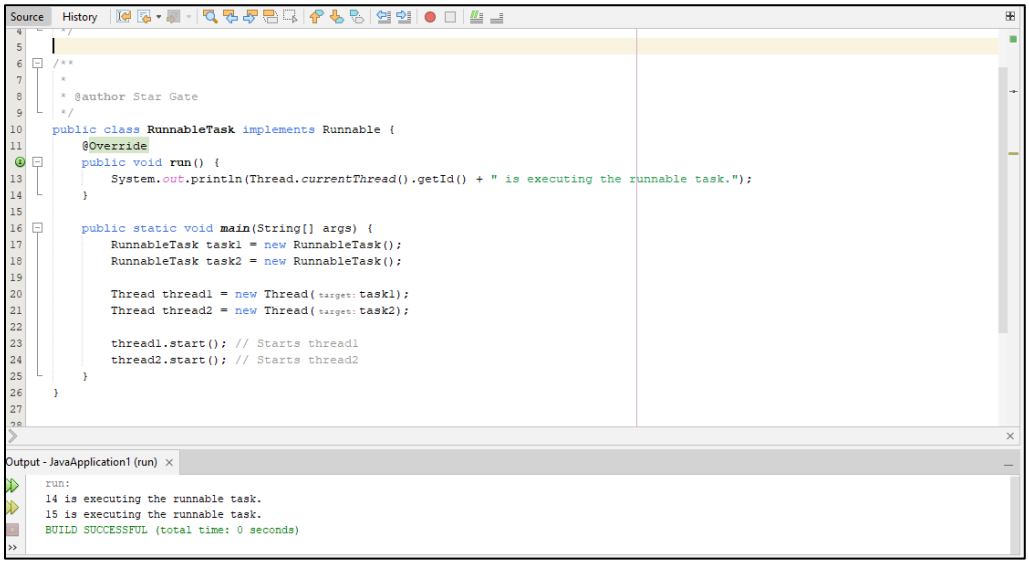
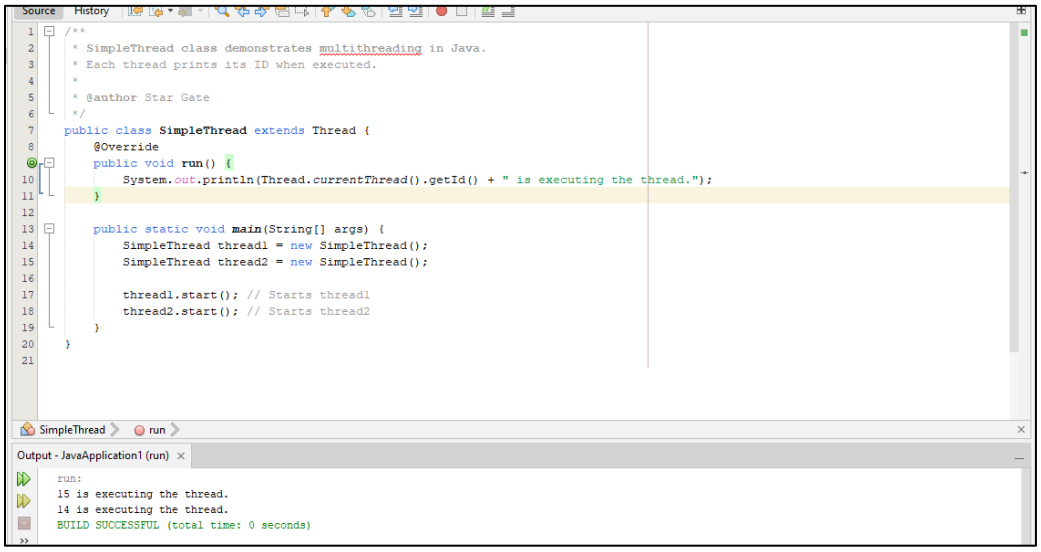
- **RunableTask**



```
/**
 *
 * @author Star Gate
 */
public class RunnableTask implements Runnable {
    @Override
    public void run() {
        System.out.println(Thread.currentThread().getId() + " is executing the runnable task.");
    }

    public static void main(String[] args) {
        RunnableTask task1 = new RunnableTask();
        RunnableTask task2 = new RunnableTask();

        Thread thread1 = new Thread( target: task1);
        Thread thread2 = new Thread( target: task2);

        thread1.start(); // Starts thread1
        thread2.start(); // Starts thread2
    }
}
```

```
Output - JavaApplication1 (run)
run:
14 is executing the runnable task.
15 is executing the runnable task.
BUILD SUCCESSFUL (total time: 0 seconds)
```

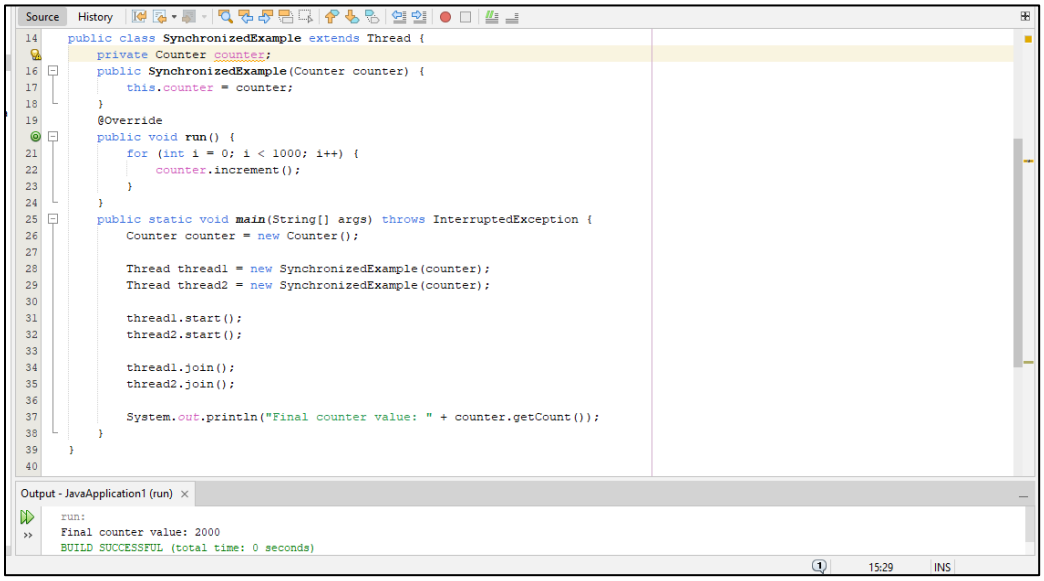
- **Simple Thread**



```
/**
 * SimpleThread class demonstrates multithreading in Java.
 * Each thread prints its ID when executed.
 *
 * @author Star Gate
 */
public class SimpleThread extends Thread {
    @Override
    public void run() {
        System.out.println(Thread.currentThread().getId() + " is executing the thread.");
    }

    public static void main(String[] args) {
        SimpleThread thread1 = new SimpleThread();
        SimpleThread thread2 = new SimpleThread();

        thread1.start(); // Starts thread1
        thread2.start(); // Starts thread2
    }
}
```

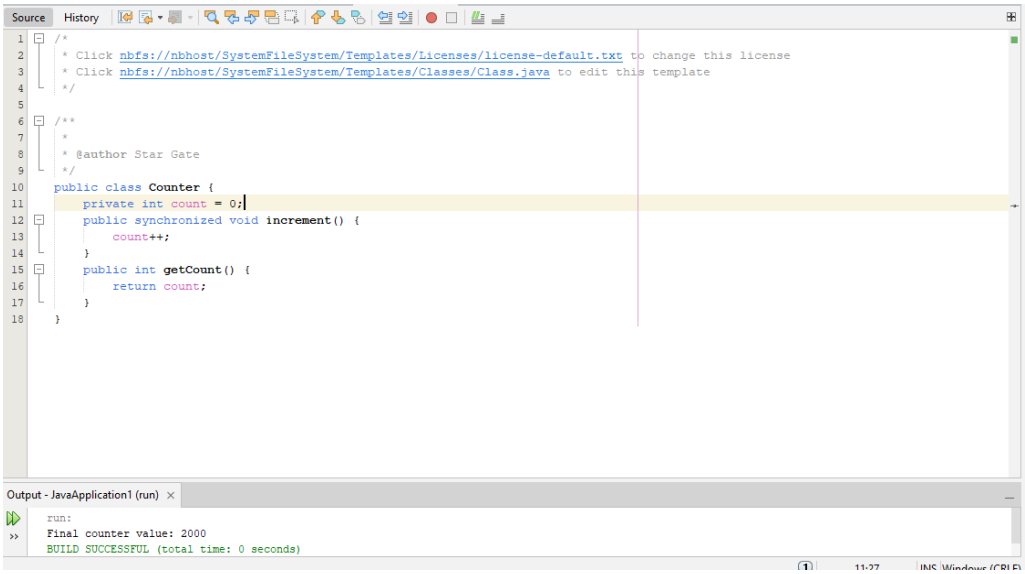
```
Output - JavaApplication1 (run)
run:
15 is executing the thread.
14 is executing the thread.
BUILD SUCCESSFUL (total time: 0 seconds)
```

- **Synchronizing Thread**



```
public class SynchronizedExample extends Thread {
    private Counter counter;
    public SynchronizedExample(Counter counter) {
        this.counter = counter;
    }
    @Override
    public void run() {
        for (int i = 0; i < 1000; i++) {
            counter.increment();
        }
    }
    public static void main(String[] args) throws InterruptedException {
        Counter counter = new Counter();

        Thread thread1 = new SynchronizedExample(counter);
        Thread thread2 = new SynchronizedExample(counter);

        thread1.start();
        thread2.start();

        thread1.join();
        thread2.join();

        System.out.println("Final counter value: " + counter.getCount());
    }
}
```

```
Output - JavaApplication1 (run)
run:
Final counter value: 2000
BUILD SUCCESSFUL (total time: 0 seconds)
```



```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */

/**
 *
 * @author Star Gate
 */
public class Counter {
    private int count = 0;
    public synchronized void increment() {
        count++;
    }
    public int getCount() {
        return count;
    }
}
```

```
Output - JavaApplication1 (run)
run:
Final counter value: 2000
BUILD SUCCESSFUL (total time: 0 seconds)
```

- **Thread Pool Example**