Demo Outputs

Callables <host> <port>

CallablesAwaitTermination <host> <port>

CallablesRNG <host> <port> <num_threads>

CallablesRNGInvokeAll <host> <port> <num_threads>

CallablesTimeout <host> <port>

Cancels <host> <port>

Runnables <host> <port> <num_threads>

RunablesRNGShutdown <host> <port> <num_threads>

- Top windows are worknode terminals; bottom windows are running the demo

Callables <host> <port>

Submits 6 SleepyHelloWorldCall callable threads to the executor service that will each return "I'm Done!" after 7 seconds.

```
MINGW32:/c/Users/Jonathan/DistributedExecutorService          —    □    ✕

$ java -jar WorkNode.jar localhost 5555 d
Debug Activated
~I have 0 Tasks~
Node ready: 2e24263f:15faedc522c:-7fff
~I have 0 Tasks~
~I have 0 Tasks~
Feeling Sleepy...
~I have 6 Tasks~
~I have 6 Tasks~
~I have 6 Tasks~
~I have 6 Tasks~
Hello World!
Feeling Sleepy...
~I have 5 Tasks~
~I have 5 Tasks~
~I have 5 Tasks~
~I have 5 Tasks~
Hello World!
Feeling Sleepy...
~I have 4 Tasks~
~I have 4 Tasks~
~I have 4 Tasks~
Hello World!
~I have 4 Tasks~
Feeling Sleepy...
~I have 3 Tasks~
~I have 3 Tasks~
~I have 3 Tasks~
Hello World!
Feeling Sleepy...
~I have 2 Tasks~
~I have 2 Tasks~
~I have 2 Tasks~
~I have 2 Tasks~
Hello World!
Feeling Sleepy...
~I have 1 Tasks~
~I have 1 Tasks~
~I have 1 Tasks~
~I have 1 Tasks~
Hello World!
~I have 0 Tasks~
~I have 0 Tasks~
~I have 0 Tasks~
~I have 0 Tasks~
~I have 0 Tasks~
~I have 0 Tasks~
~I have 0 Tasks~
~I have 0 Tasks~
```

```
MINGW32:/c/Users/Jonathan/DistributedExecutorService          —    □    ✕

Jonathan@DESKTOP-AML7NU4 ~/DistributedExecutorService (master)
$ ./rundemo.sh Callables localhost 5555
building Callables
Note: Some input files use unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
Running Callables with parameters localhost:5555...
[Ljava.lang.String;@4e50df2e
On return we got I'm Done!I'm Done!I'm Done!I'm Done!I'm Done!I'm D
one!

Done!
```
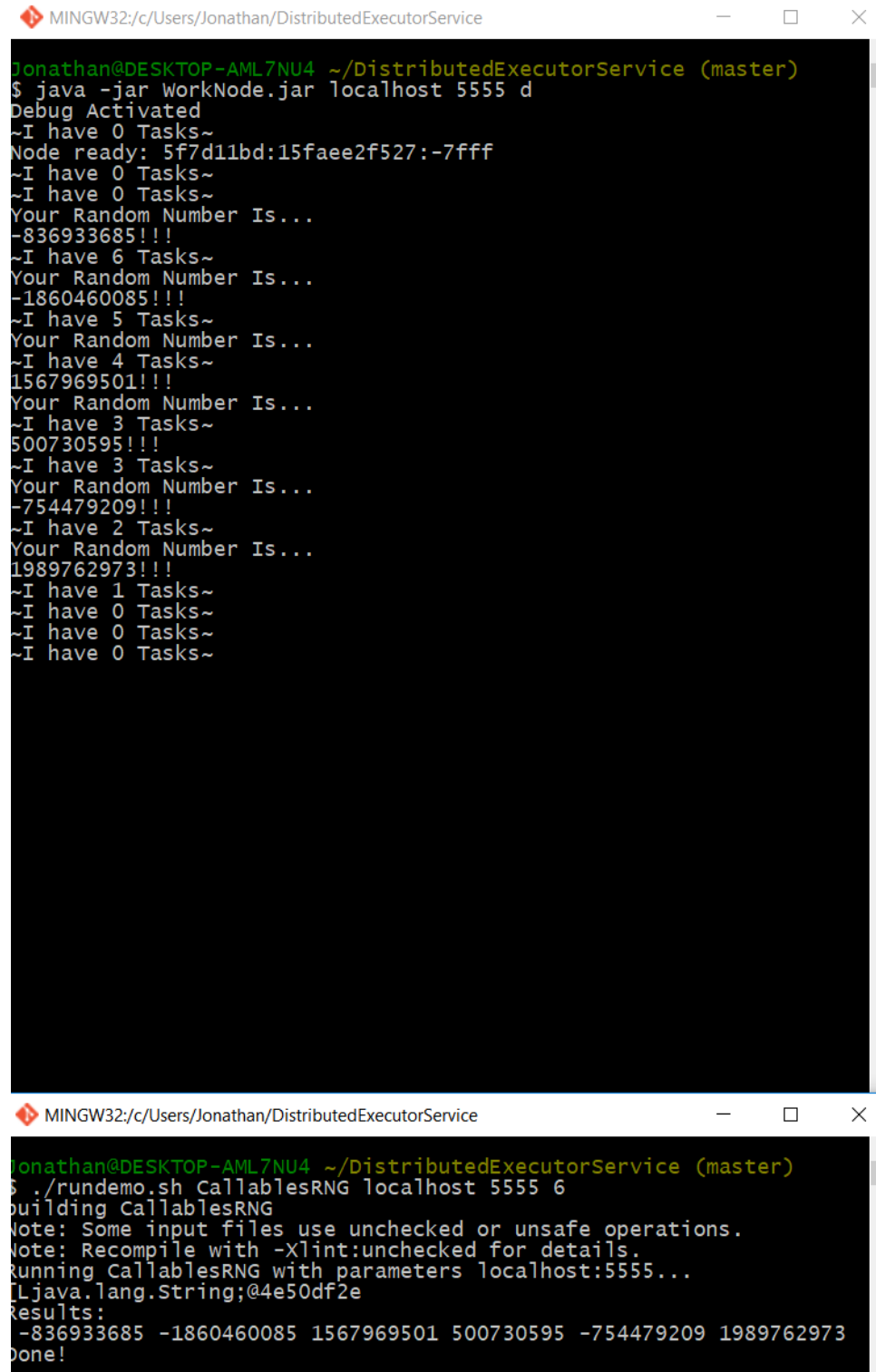
CallablesAwaitTermination <host> <port>

Submits 4 SleepyHelloWorldCall threads to the executor service, then terminates the executor service. Will await the termination of each thread from the worknodes. Will print out every 4 seconds that it is waiting.

```
MINGW32:/c/Users/Jonathan/DistributedExecutorService
$ java -jar WorkNode.jar localhost 5555 debug
Debug Activated
~I have 0 Tasks~
Node ready: -610756cb:15faeb73574:-7fff
~I have 0 Tasks~
~I have 0 Tasks~
~I have 0 Tasks~
~I have 0 Tasks~
~I have 0 Tasks~
~I have 0 Tasks~
~I have 0 Tasks~
~I have 0 Tasks~
~I have 0 Tasks~
~I have 0 Tasks~
Feeling Sleepy...
~I have 4 Tasks~
~I have 4 Tasks~
~I have 4 Tasks~
~I have 4 Tasks~
Hello World!
Feeling Sleepy...
~I have 3 Tasks~
~I have 3 Tasks~
~I have 3 Tasks~
~I have 3 Tasks~
Hello World!
Feeling Sleepy...
~I have 2 Tasks~
~I have 2 Tasks~
~I have 2 Tasks~
Hello World!
~I have 2 Tasks~
Feeling Sleepy...
~I have 1 Tasks~
~I have 1 Tasks~
~I have 1 Tasks~
Hello World!
~I have 0 Tasks~
~I have 0 Tasks~
~I have 0 Tasks~
~I have 0 Tasks~
~I have 0 Tasks~
~I have 0 Tasks~



Jonathan@DESKTOP-AML7NU4 /c/Sites
$ cd ~/DistributedExecutorService/

Jonathan@DESKTOP-AML7NU4 ~/DistributedExecutorService (master)
$ ./rundemo.sh CallablesAwaitTermination localhost 5555
building CallablesAwaitTermination
Note: Some input files use unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
Running CallablesAwaitTermination with parameters localhost:5555...
[Ljava.lang.String;@4e50df2e
Tasks Have been submitted! i'm going to shutdown and await termination for 4 seconds..
4 MORE Seconds have passed... Have we terminated? false
4 MORE Seconds have passed... Have we terminated? false
4 MORE Seconds have passed... Have we terminated? false
4 MORE Seconds have passed... Have we terminated? false
4 MORE Seconds have passed... Have we terminated? false
4 MORE Seconds have passed... Have we terminated? false
4 MORE Seconds have passed... Have we terminated? false
4 MORE Seconds have passed... Have we terminated? false
4 MORE Seconds have passed... Have we terminated? false
4 MORE Seconds have passed... Have we terminated? true

Done!

Page 1 of 1    3 words    LI^
```

CallablesRNG \<host\> \<port\> \<num_threads\>

Creates num_threads number of RandomNumber callable threads and submits them to the executor service in the order they were submitted. Calls get() on the futures in the order they were submitted, and prints the returned random numbers as they are generated.

```
MINGW32:/c/Users/Jonathan/DistributedExecutorService                    —    □    ×

Jonathan@DESKTOP-AML7NU4 ~/DistributedExecutorService (master)
$ java -jar WorkNode.jar localhost 5555 d
Debug Activated
~I have 0 Tasks~
Node ready: 5f7d11bd:15faee2f527:-7fff
~I have 0 Tasks~
~I have 0 Tasks~
Your Random Number Is...
-836933685!!!
~I have 6 Tasks~
Your Random Number Is...
-1860460085!!!
~I have 5 Tasks~
Your Random Number Is...
~I have 4 Tasks~
1567969501!!!
Your Random Number Is...
~I have 3 Tasks~
500730595!!!
~I have 3 Tasks~
Your Random Number Is...
-754479209!!!
~I have 2 Tasks~
Your Random Number Is...
1989762973!!!
~I have 1 Tasks~
~I have 0 Tasks~
~I have 0 Tasks~
~I have 0 Tasks~
```

```
MINGW32:/c/Users/Jonathan/DistributedExecutorService                    —    □    ×

Jonathan@DESKTOP-AML7NU4 ~/DistributedExecutorService (master)
$ ./rundemo.sh CallablesRNG localhost 5555 6
building CallablesRNG
Note: Some input files use unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
Running CallablesRNG with parameters localhost:5555...
[Ljava.lang.String;@4e50df2e
Results:
-836933685 -1860460085 1567969501 500730595 -754479209 1989762973
Done!
```

CallablesRNGInvokeAll <host> <port> <num_threads>

Given num_threads as the third paramter, creates that many new threads that generate and return a random number. This Demo calls invokeAll with a list of threads to execute instead of submitting them individually. Returns an array of Futures that are then printed to the terminal.

```
Jonathan@DESKTOP-AML7NU4 ~/DistributedExecutorService (master)
$ java -jar WorkNode.jar localhost 5555 debug
Debug Activated
~I have 0 Tasks~
Node ready: 7b54c51:15faec509f2:-7fff
~I have 0 Tasks~
~I have 0 Tasks~
Your Random Number Is...
~I have 5 Tasks~
-1999975412!!!
~I have 5 Tasks~
Your Random Number Is...
-974404467!!!
~I have 4 Tasks~
Your Random Number Is...
1534106585!!!
~I have 3 Tasks~
Your Random Number Is...
~I have 2 Tasks~
409160980!!!
Your Random Number Is...
~I have 1 Tasks~
931828049!!!
~I have 1 Tasks~
~I have 0 Tasks~
~I have 0 Tasks~
```

```
Jonathan@DESKTOP-AML7NU4 ~/DistributedExecutorService (master)
$ ./rundemo.sh CallablesRNGInvokeAll localhost 5555 5
building CallablesRNGInvokeAll
Note: Some input files use unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
Running CallablesRNGInvokeAll with parameters localhost:5555...
[Ljava.lang.String;@4e50df2e
Results:
-1999975412
-974404467
1534106585
409160980
931828049

Done!
```

CallablesTimeout <host> <port>

Submits 4 SleepyHelloWorldCall threads to the executor service and calls get() on the first future with a 3 second timeout. The expected output is "First task: null" followed with "Final Results: I'm Done!I'm Done!I'm Done!I'm Done!" on the next line. This is because the First task is printed after 3 seconds, and doesn't block, so it has not yet received a value from the future.

MINGW32:/c/Users/Jonathan/DistributedExecutorService

```
Jonathan@DESKTOP-AML7NU4 ~/DistributedExecutorService (master)
$ java -jar WorkNode.jar localhost 5555 debug
Debug Activated
~I have 0 Tasks~
Node ready: 45396c1b:15faeca4da6:-7fff
~I have 0 Tasks~
~I have 0 Tasks~
Feeling Sleepy...
~I have 4 Tasks~
~I have 4 Tasks~
~I have 4 Tasks~
~I have 4 Tasks~
Hello World!
Feeling Sleepy...
~I have 3 Tasks~
~I have 3 Tasks~
~I have 3 Tasks~
Hello World!
~I have 3 Tasks~
Feeling Sleepy...
~I have 2 Tasks~
~I have 2 Tasks~
~I have 2 Tasks~
Hello World!
Feeling Sleepy...
~I have 1 Tasks~
~I have 1 Tasks~
~I have 1 Tasks~
~I have 1 Tasks~
Hello World!
~I have 0 Tasks~
~I have 0 Tasks~
```

MINGW32:/c/Users/Jonathan/DistributedExecutorService

```
Jonathan@DESKTOP-AML7NU4 ~/DistributedExecutorService (master)
$ ./rundemo.sh CallablesTimeout localhost 5555
building CallablesTimeout
Note: Some input files use unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
Running CallablesTimeout with parameters localhost:5555...
[Ljava.lang.String;@4e50df2e
Submitted tasks, timeout for the first task is THREE seconds
First task: null
Final Results: I'm Done!I'm Done!I'm Done!I'm Done!

Done!
```

Cancels <host> <port>

Submits one SleepyHelloWorldCall thread and calls cancel on its future immediately. Will print "Future Cancelled" when get() is called. Instead of taking 3-4 '~I have X Tasks~' prints in the work node, it only takes up 1, because it gets cancelled immediately.

MINGW32:/c/Users/Jonathan/DistributedExecutorService

```
Jonathan@DESKTOP-AML7NU4 ~/DistributedExecutorService (master)
$ java -jar WorkNode.jar localhost 5555 debug
Debug Activated
~I have 0 Tasks~
Node ready: 2dd6f2fe:15faecd21c0:-7fff
~I have 0 Tasks~
~I have 0 Tasks~
Feeling Sleepy...
~I have 1 Tasks~
~I have 0 Tasks~
~I have 0 Tasks~
~I have 0 Tasks~
```

MINGW32:/c/Users/Jonathan/DistributedExecutorService

```
Jonathan@DESKTOP-AML7NU4 ~/DistributedExecutorService (master)
$ ./rundemo.sh Cancels localhost 5555
building Cancels
Note: Some input files use unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
Running Cancels with parameters localhost:5555...
[Ljava.lang.String;@4e50df2e
On return we got
Future Cancelled

Done!
```

Runnables <host> <port> <num_threads>

Given a third parameter num_threads, it submits that many SleepyHelloWorld Runnable threads to the executor service. Since it does not wait for futures, it can finish before the worknode is complete.

MINGW32:/c/Users/Jonathan/DistributedExecutorService

```
Jonathan@DESKTOP-AML7NU4 ~/DistributedExecutorService (master)
$ java -jar WorkNode.jar localhost 5555 debug
Debug Activated
~I have 0 Tasks~
Node ready: 62c6a760:15faecf2567:-7fff
~I have 0 Tasks~
Going to Sleep...
~I have 4 Tasks~
~I have 4 Tasks~
... Hello World!
~I have 4 Tasks~
Going to Sleep...
~I have 3 Tasks~
```

MINGW32:/c/Users/Jonathan/DistributedExecutorService

```
Jonathan@DESKTOP-AML7NU4 ~/DistributedExecutorService (master)
$ ./rundemo.sh Runnables localhost 5555 4
building Runnables
Note: Some input files use unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
Running Runnables with parameters localhost:5555...
[Ljava.lang.String;@4e50df2e
Finished!
```

RunablesRNGShutdown <host> <port> <num_threads>

Submits a given num_threads number of RandomNumberRun threads to the executor service. It waits 10 seconds before shutting down the executor service and terminating any active threads. One should observe work nodes being interrupted and having their tasks cancelled. Prints out the number of RandomNumberRun threads that the worknodes didn't get to.


```
MINGW32:/c/Users/Jonathan/DistributedExecutorService

Jonathan@DESKTOP-AML7NU4 ~/DistributedExecutorService (master)
$ java -jar WorkNode.jar localhost 5555 debug
Debug Activated
~I have 0 Tasks~
Node ready: -5341481b:15faed15dad:-7fff
~I have 0 Tasks~
Your Random Number Is...
~I have 12 Tasks~
1346815445!!!
~I have 12 Tasks~
Your Random Number Is...
-516539179!!!
~I have 11 Tasks~
Your Random Number Is...
1485041393!!!
~I have 10 Tasks~
Your Random Number Is...
~I have 9 Tasks~
1194808593!!!
Your Random Number Is...
I've been interrupted! :(
-719594136!!!
~I have 8 Tasks~
~I have 6 Tasks~
~I have 2 Tasks~
~I have 0 Tasks~
~I have 0 Tasks~
~I have 0 Tasks~
```

```
MINGW32:/c/Users/Jonathan/DistributedExecutorService

Jonathan@DESKTOP-AML7NU4 ~/DistributedExecutorService (master)
$ ./rundemo.sh RunablesRNGShutdown localhost 5555 12
building RunablesRNGShutdown
Note: Some input files use unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
Running RunablesRNGShutdown with parameters localhost:5555...
[Ljava.lang.String;@4e50df2e
SHUTTING DOWN
We missed 8 random numbers

Done!
```

Multi RunablesRNGShutdown

This is another RunablesRNG example with two work nodes and two distributed executor service (ES) programs running RunnablesRNGShutdown. The ES on the bottom-left ran about 2 seconds after the one on the bottom-right. We see the work nodes each initially get 6 tasks, they complete two of them, but then receive 6 more to have 10 tasks each. This is the second distributed ES starting up. A few seconds later however, the first ES gets an immediate shutdown and one can see the first 'I've been interrupted! ☹" where each work node had a running task that was cancelled. From here the tasks quickly dip down to 6 each because those requests were made by the second ES and are still active in the work node's queue. The second ES also does an immediate shutdown about 2 seconds after the first, hence the second interrupt message. After this each work node's task queue dips to 0. It appears the first ES was able to get 8 random numbers before shutting down, while the second was only able to get 2 random numbers.

```
MINGW32:/c/Users/Jonathan/DistributedExecutorService

Jonathan@DESKTOP-AML7NU4 ~/DistributedExecutorService (master)
$ java -jar WorkNode.jar localhost 5555 d
Debug Activated
~I have 0 Tasks~
Node ready: -415a20ca:15faed6d9db:-7fff
~I have 0 Tasks~
~I have 0 Tasks~
~I have 0 Tasks~
~I have 0 Tasks~
Your Random Number Is...
~I have 6 Tasks~
860245938!!!
~I have 6 Tasks~
Your Random Number Is...
166681538!!!
~I have 5 Tasks~
Your Random Number Is...
67124018!!!
~I have 10 Tasks~
Your Random Number Is...
~I have 9 Tasks~
1572811221!!!
Your Random Number Is...
I've been interrupted! :(
-1154147631!!!
~I have 8 Tasks~
Your Random Number Is...
~I have 6 Tasks~
-700182392!!!
~I have 6 Tasks~
Your Random Number Is...
I've been interrupted! :(
-1867330660!!!
~I have 4 Tasks~
~I have 0 Tasks~
~I have 0 Tasks~
~I have 0 Tasks~
~I have 0 Tasks~
```

```
MINGW32:/c/Users/Jonathan/DistributedExecutorService

Jonathan@DESKTOP-AML7NU4 ~/DistributedExecutorService (master)
$ java -jar WorkNode.jar localhost 5555 d
Debug Activated
~I have 0 Tasks~
Node ready: 6dd31928:15faed6d86f:-7fff
~I have 0 Tasks~
~I have 0 Tasks~
~I have 0 Tasks~
~I have 0 Tasks~
Your Random Number Is...
~I have 6 Tasks~
1267733297!!!
~I have 6 Tasks~
Your Random Number Is...
-1621012402!!!
~I have 5 Tasks~
Your Random Number Is...
927945151!!!
~I have 10 Tasks~
Your Random Number Is...
~I have 9 Tasks~
-1251366058!!!
Your Random Number Is...
I've been interrupted! :(
-1527166361!!!
~I have 8 Tasks~
Your Random Number Is...
~I have 6 Tasks~
-851376601!!!
~I have 6 Tasks~
Your Random Number Is...
I've been interrupted! :(
-1743683560!!!
~I have 5 Tasks~
~I have 1 Tasks~
~I have 0 Tasks~
~I have 0 Tasks~
~I have 0 Tasks~
```

```
MINGW32:/c/Users/Jonathan/DistributedExecutorService

Jonathan@DESKTOP-AML7NU4 ~/DistributedExecutorService (master)
$ ./rundemo.sh RunablesRNGShutdown localhost 5555 12
building RunablesRNGShutdown
Note: Some input files use unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
Running RunablesRNGShutdown with parameters localhost:5555...
[Ljava.lang.String;@4e50df2e
SHUTTING DOWN
We missed 10 random numbers

Done!
```

```
MINGW32:/c/Users/Jonathan/DistributedExecutorService

Jonathan@DESKTOP-AML7NU4 ~/DistributedExecutorService (master)
$ ./rundemo.sh RunablesRNGShutdown localhost 5555 12
building RunablesRNGShutdown
Note: Some input files use unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
Running RunablesRNGShutdown with parameters localhost:5555...
[Ljava.lang.String;@4e50df2e
SHUTTING DOWN
We missed 4 random numbers

Done!
```