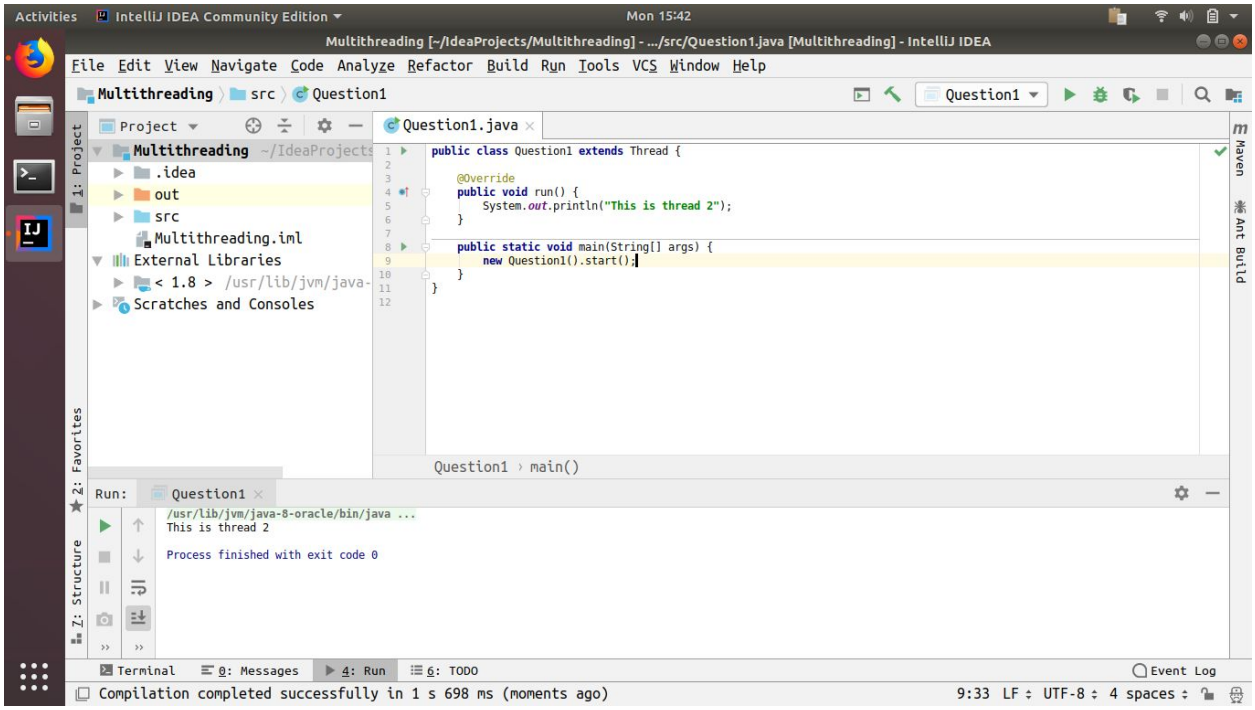
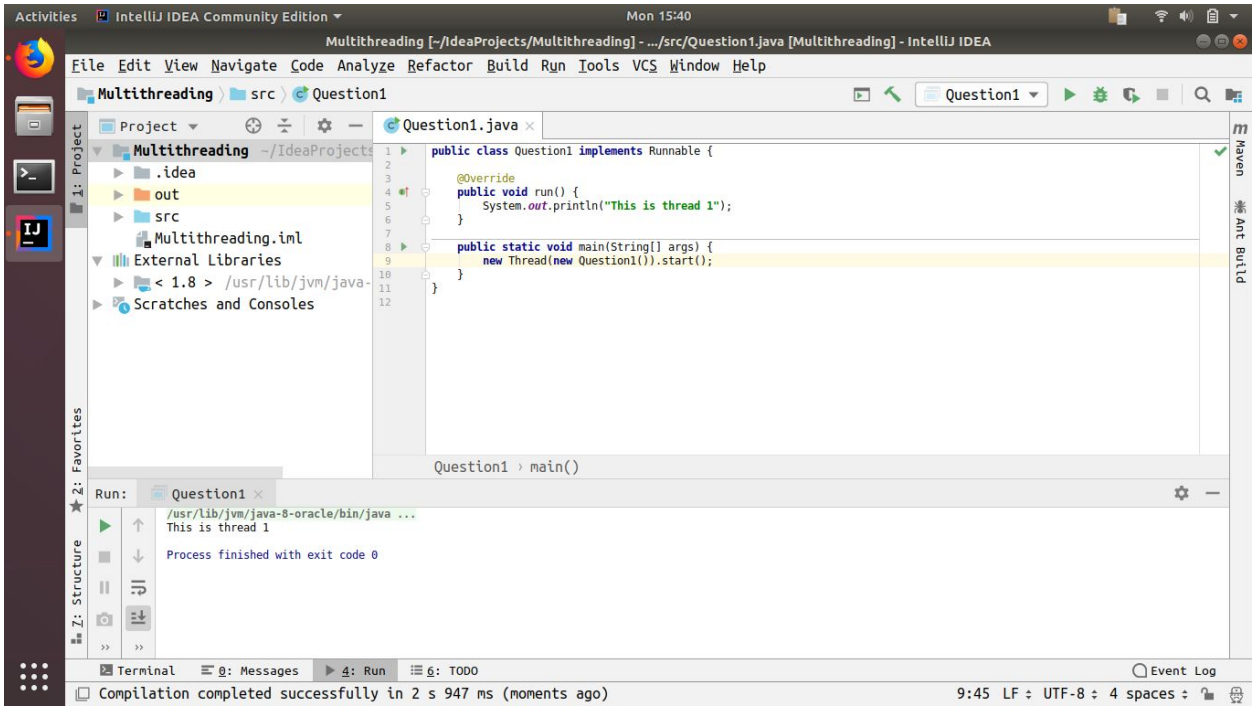
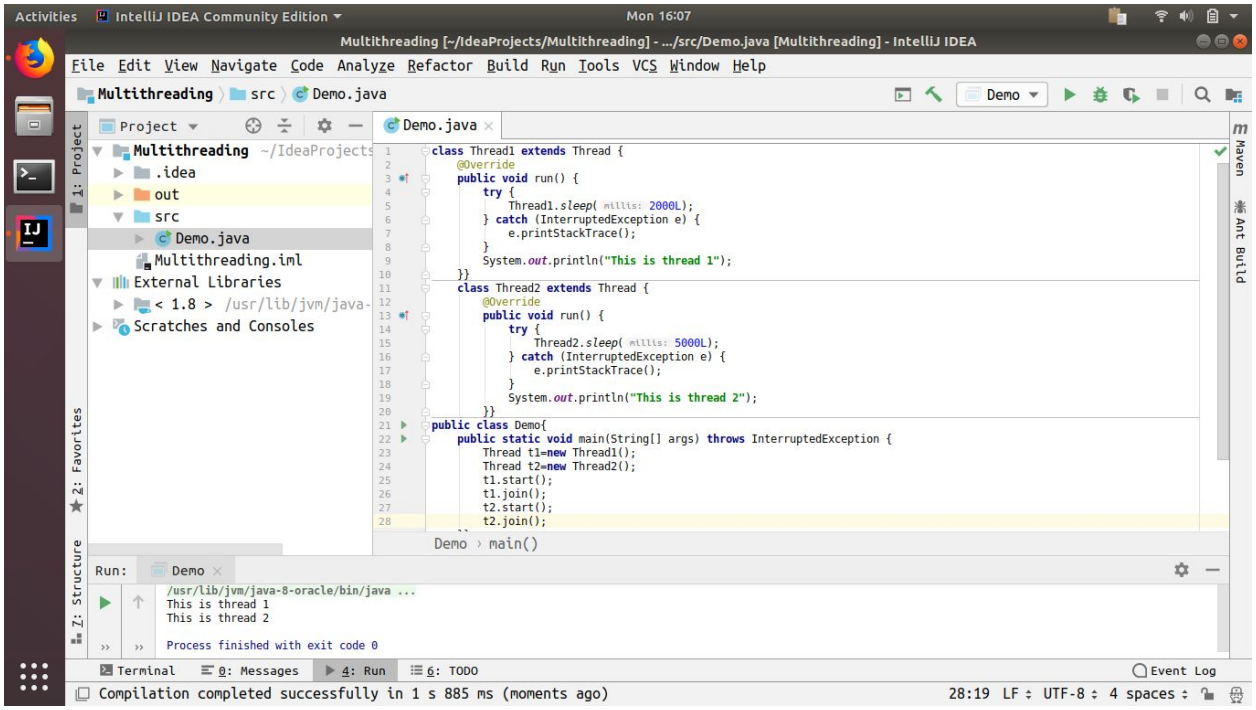
- Create and Run a Thread using Runnable Interface and Thread class.

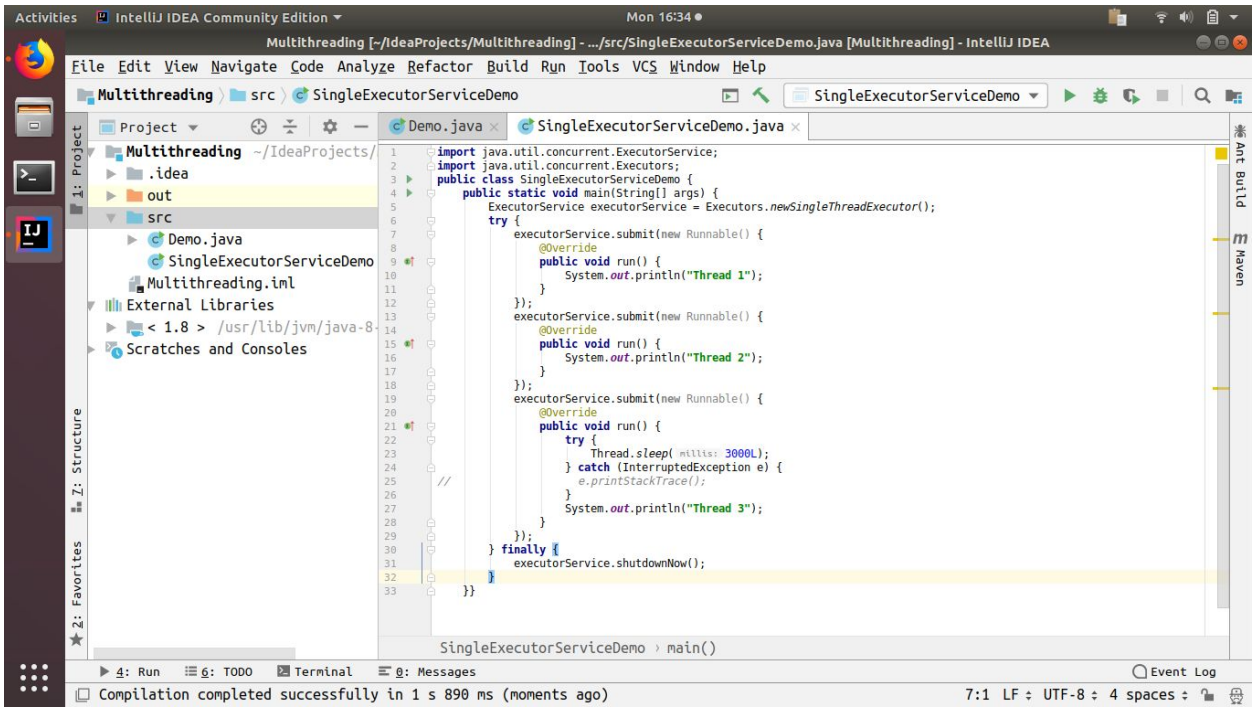


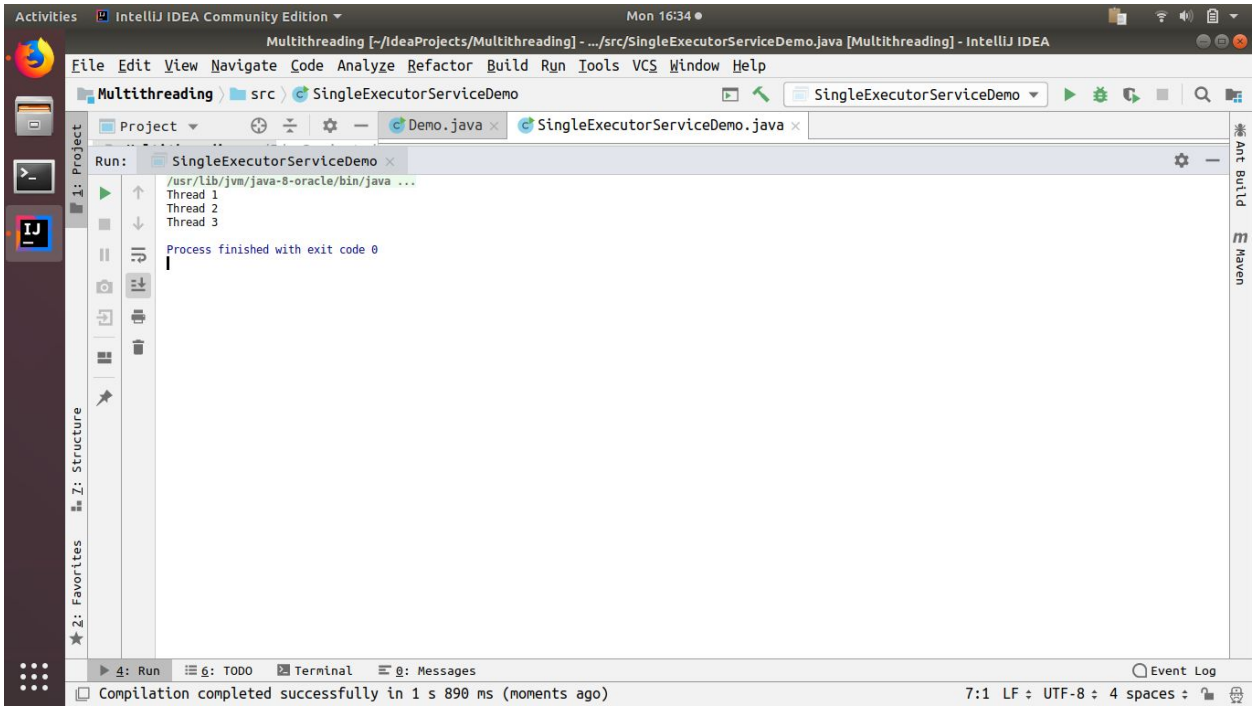


- Use sleep and join methods with thread.



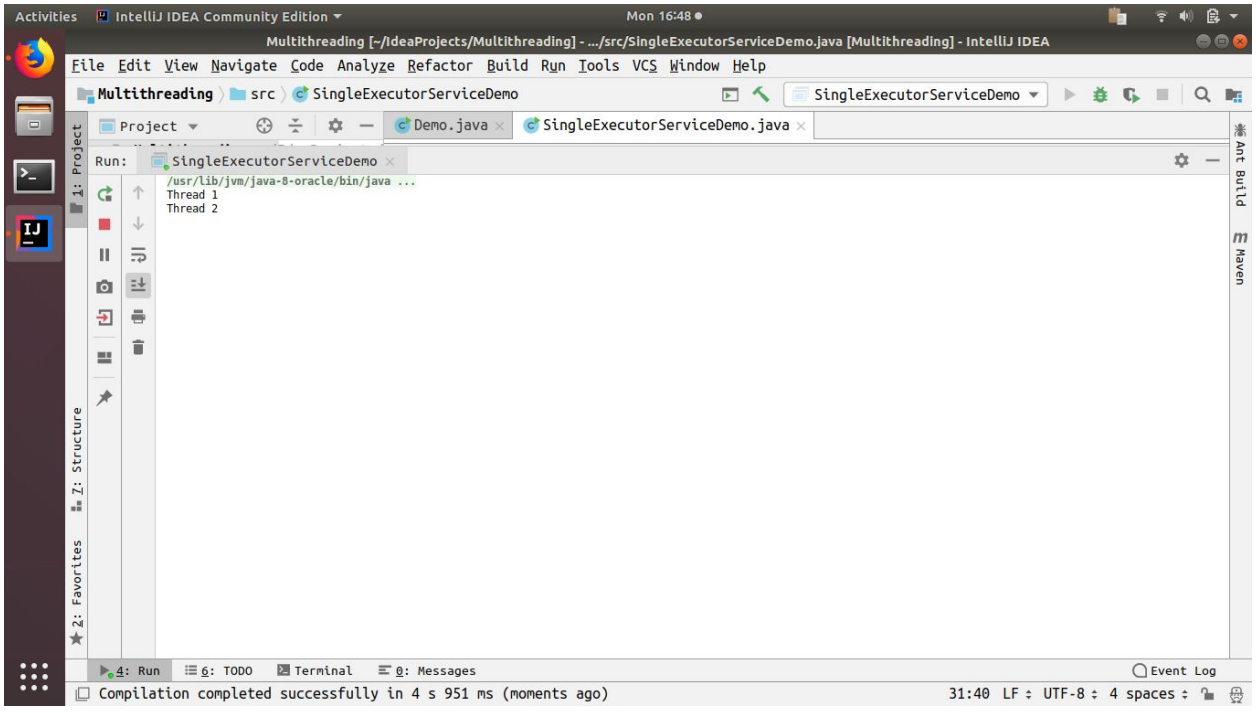
- Use a singleThreadExecutor to submit multiple threads.

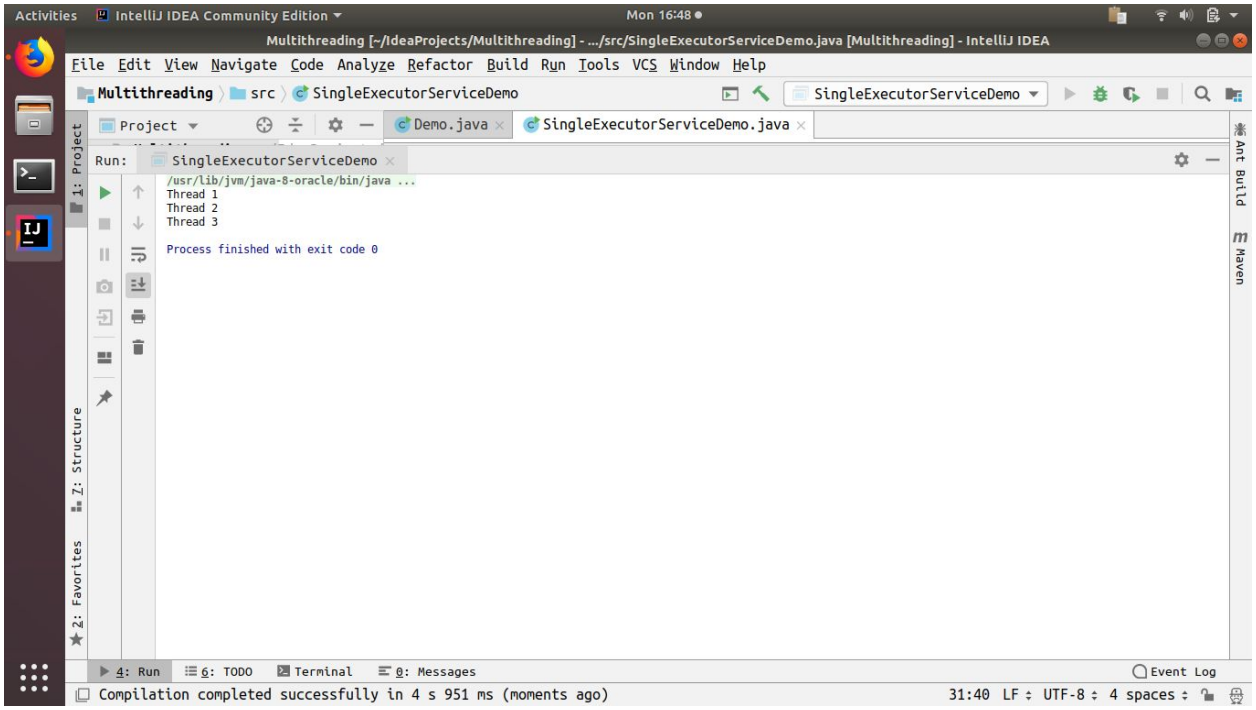


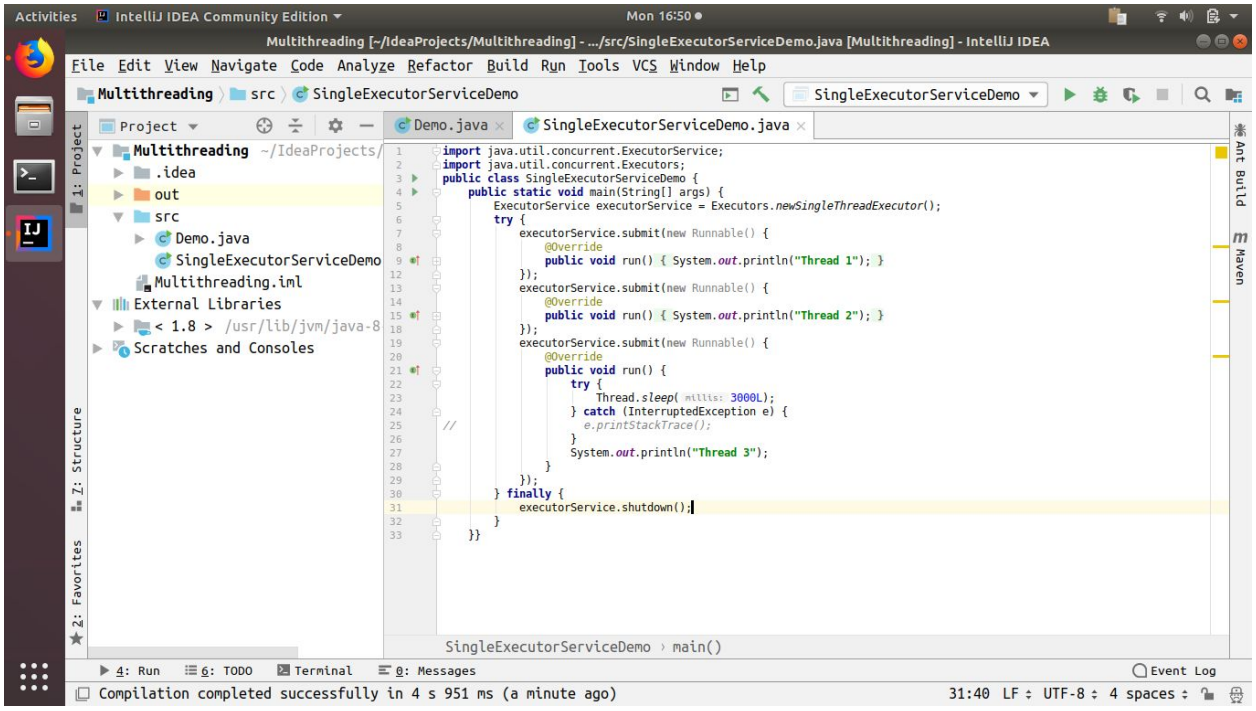


- Try shutdown() and shutdownNow() and observe the difference.

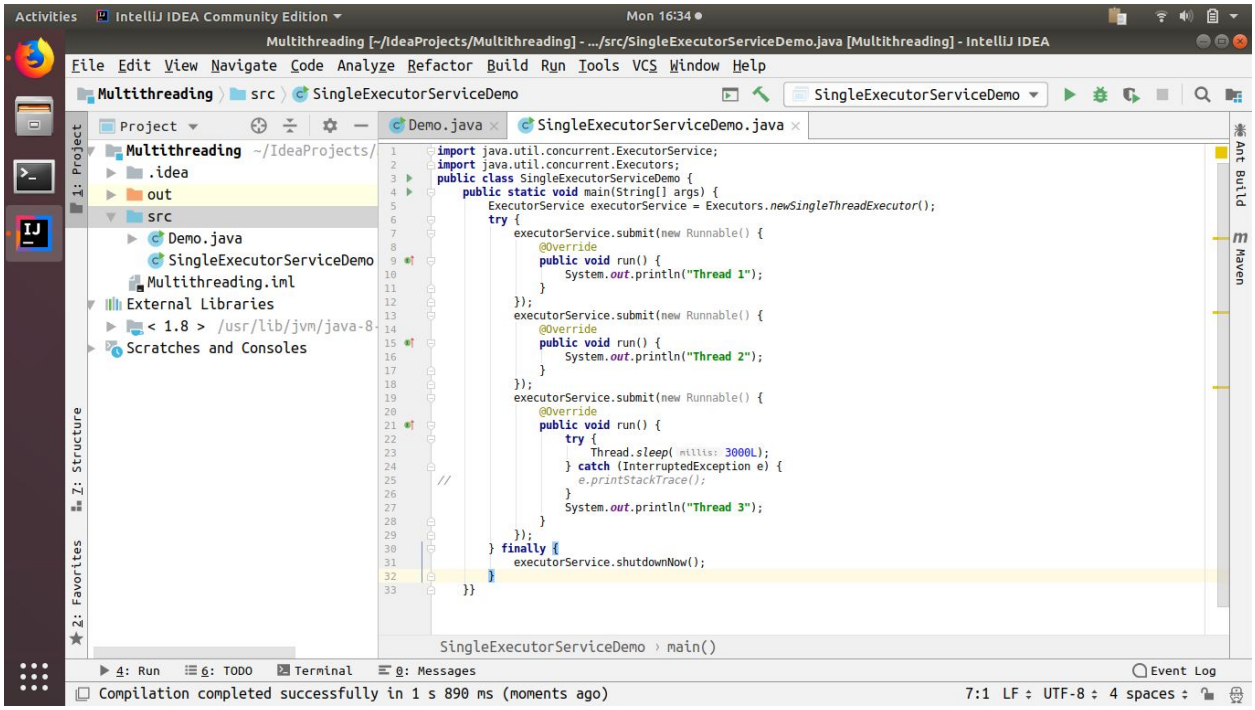
  In **shutdown -** is a safe way of shutdown it will wait for all threads to excute after that it will shutdown

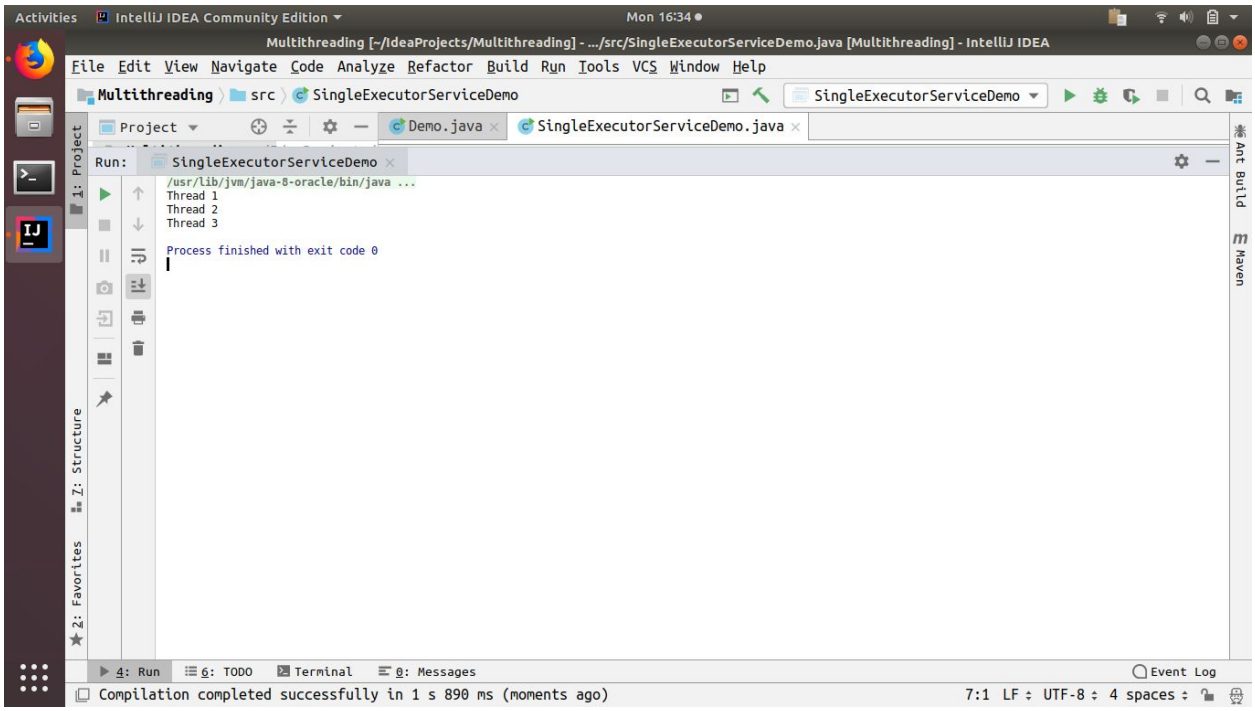




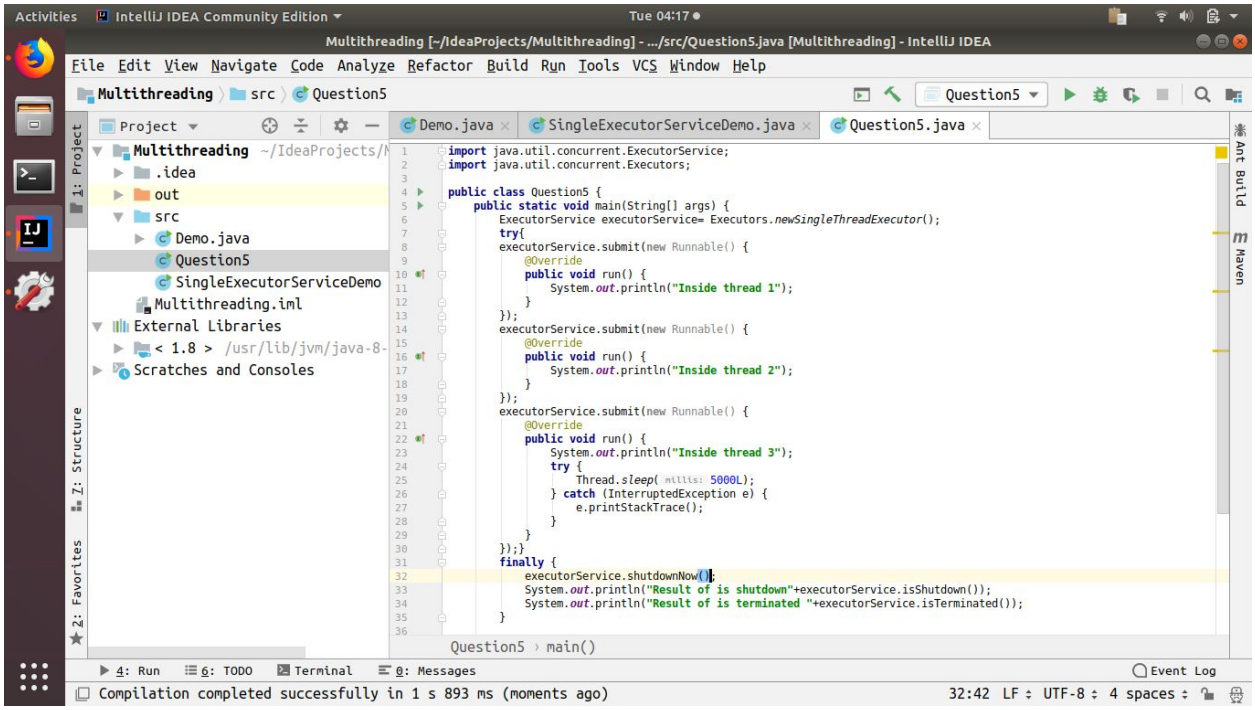


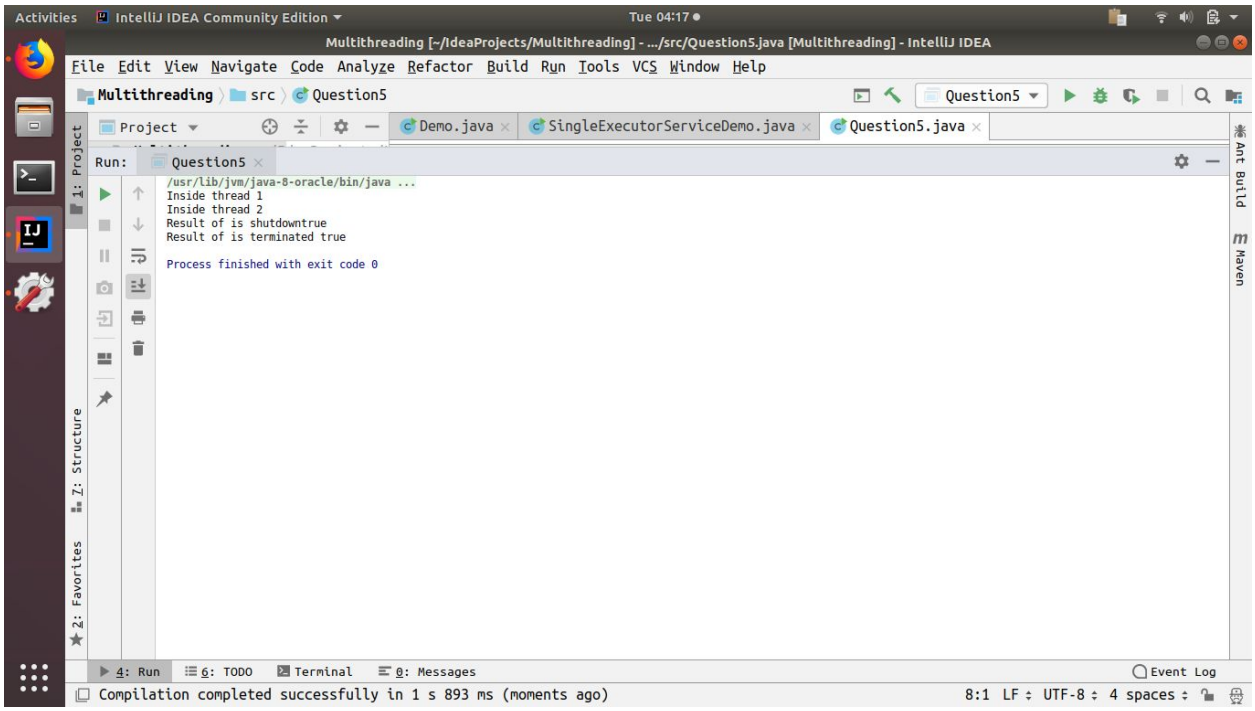
Whereas in **shutdownNow() -** it will terminate as soon as it is encountered without waiting for the thread to complete its execution



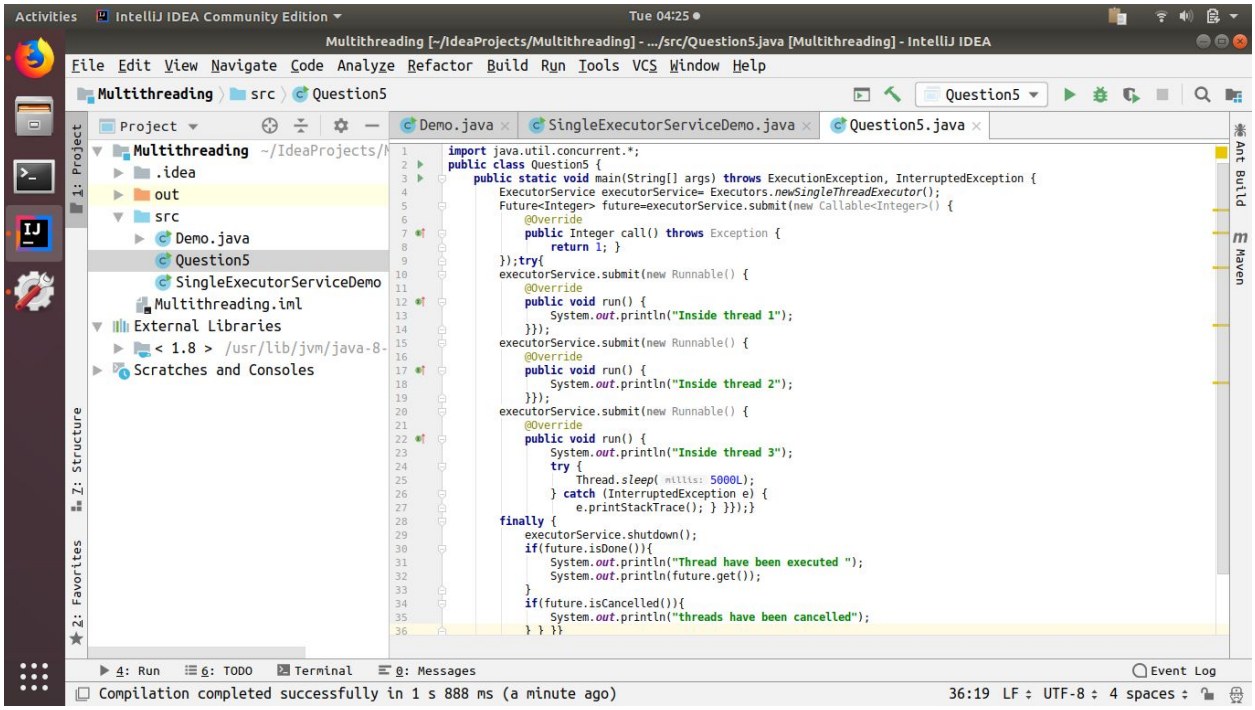


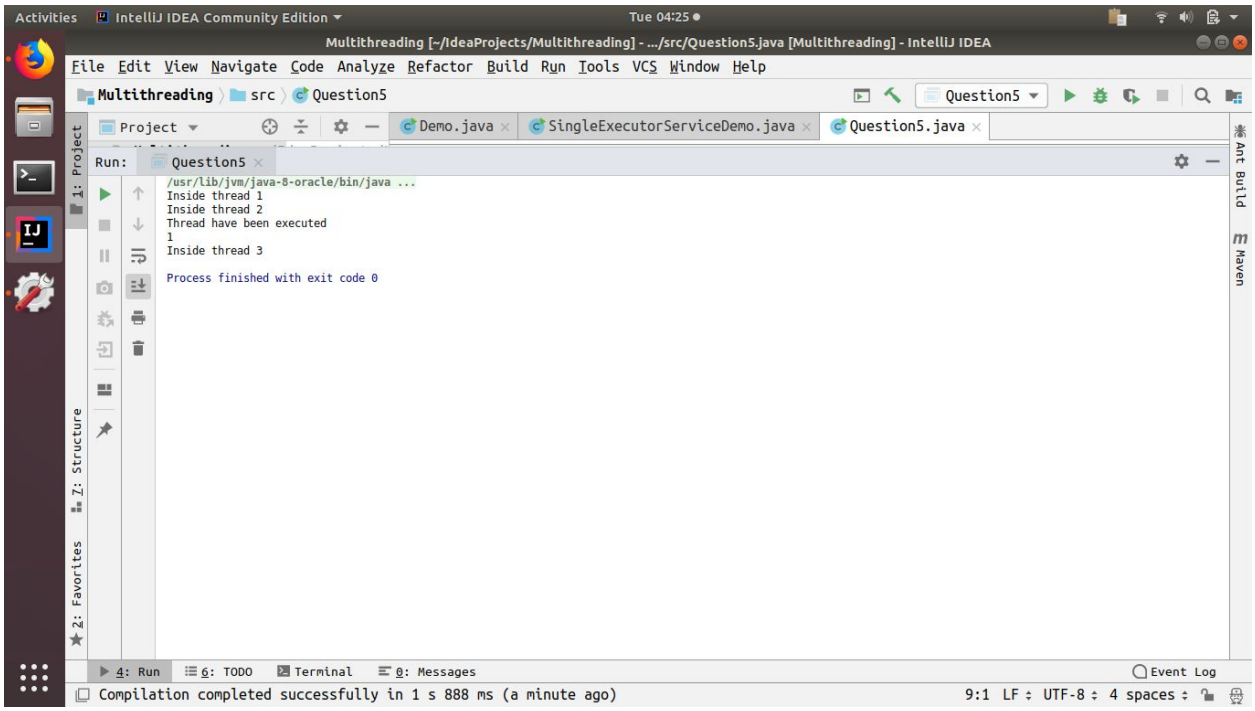
- Use isShutDown() and isTerminate() with ExecutorService.



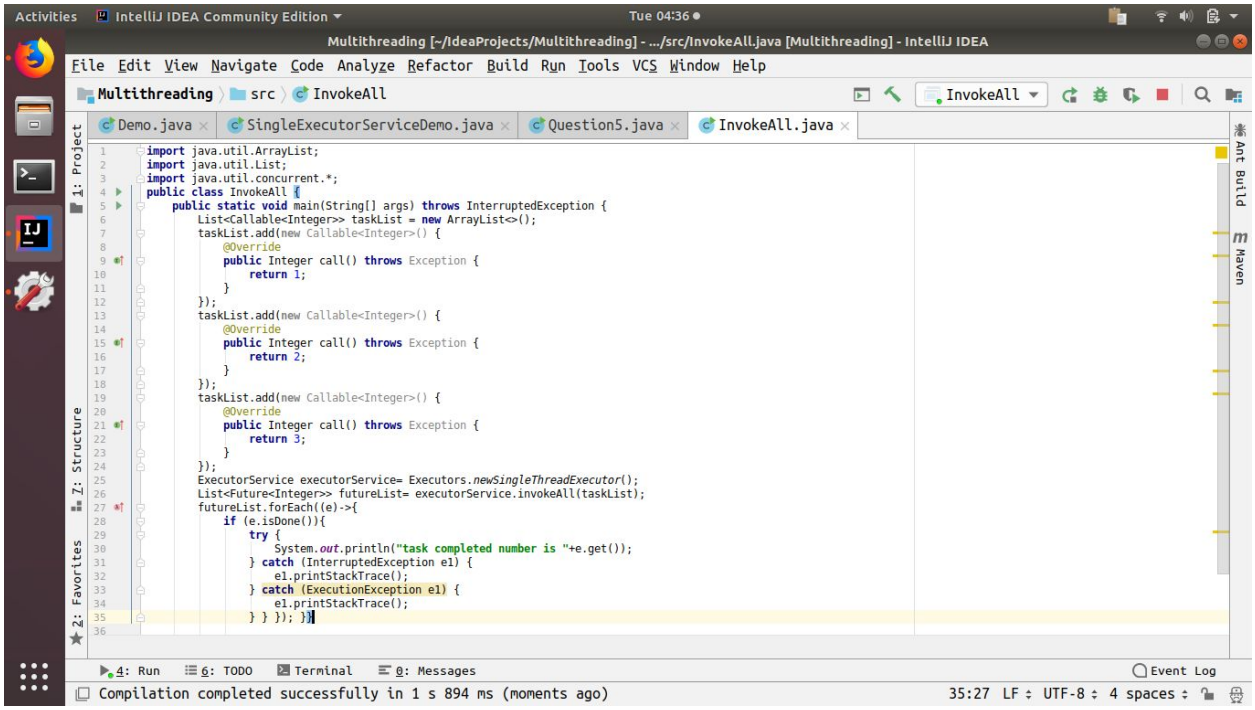


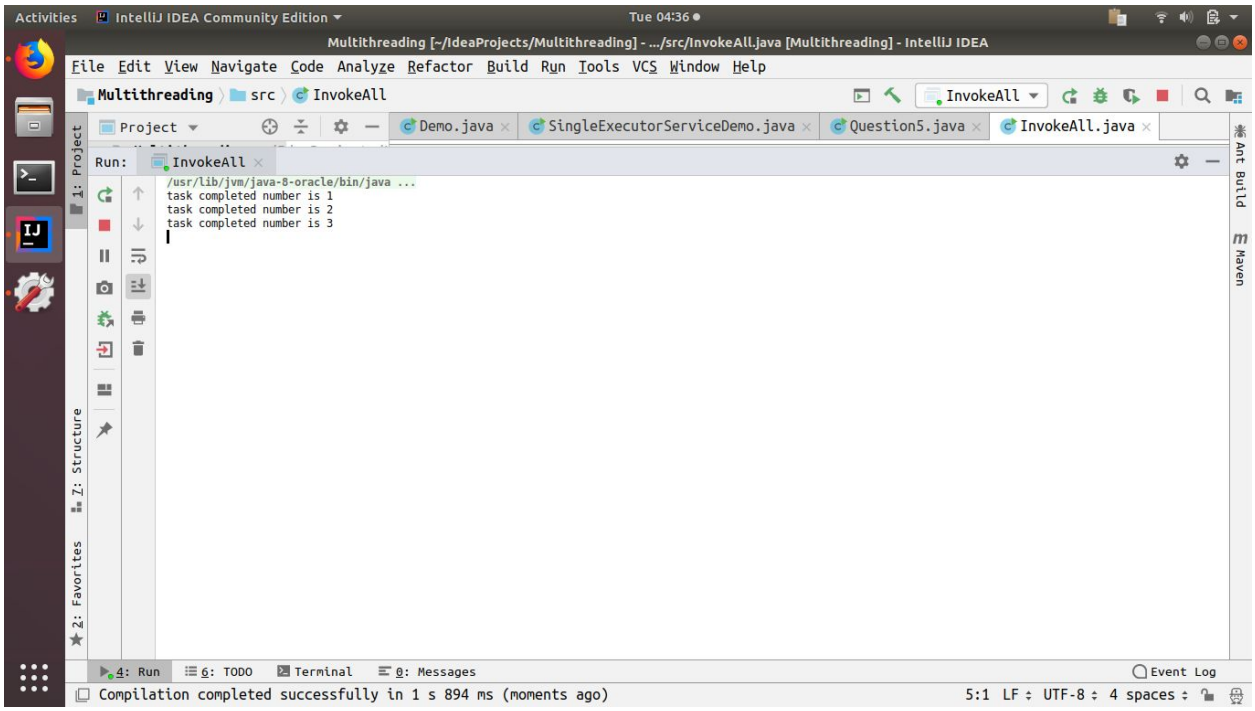
- Return a Future from ExecutorService by using callable and use get(), isDone(), isCancelled() with the Future object to know the status of task submitted.



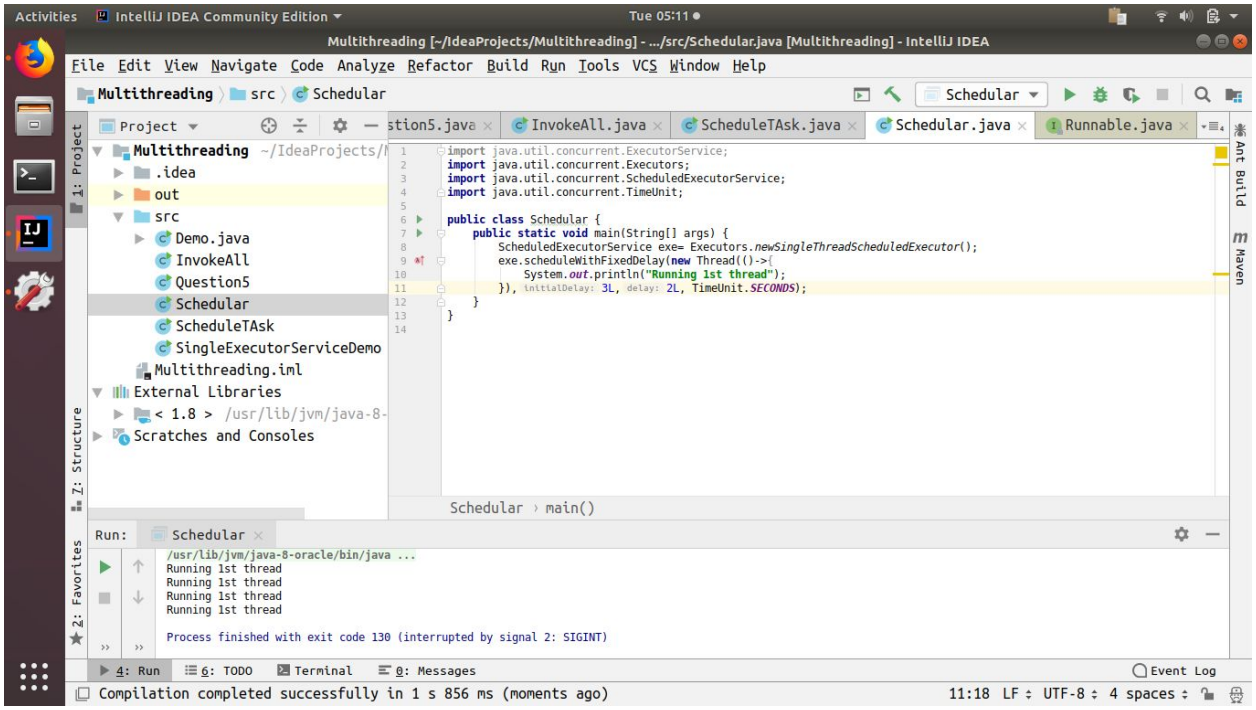


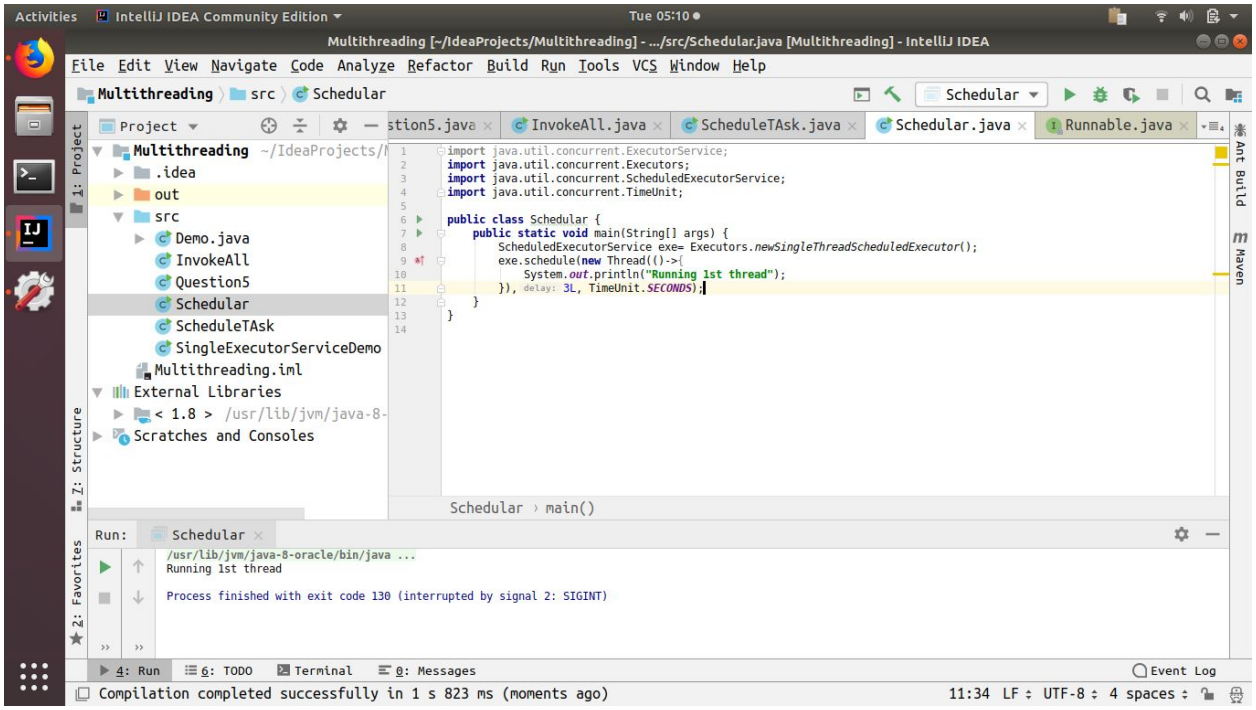
- Submit List of tasks to ExecutorService and wait for the completion of all the tasks.

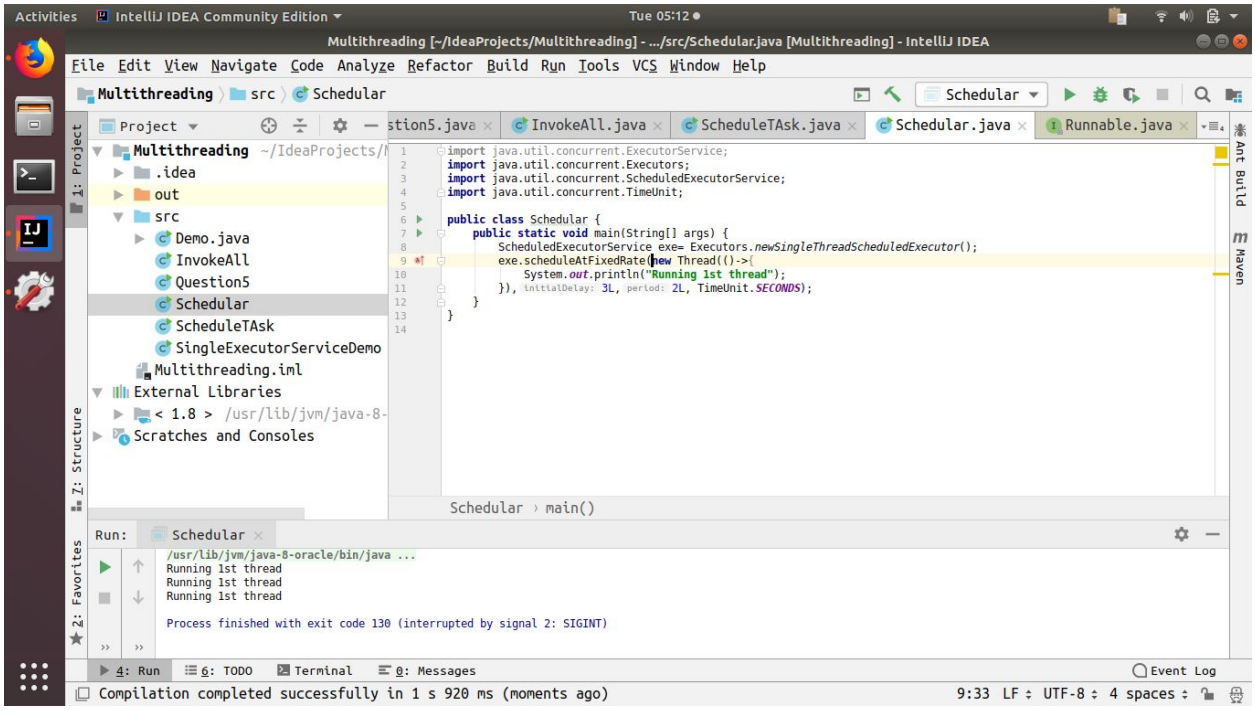




- Schedule task using schedule(), scheduleAtFixedRate() and scheduleAtFixedDelay()

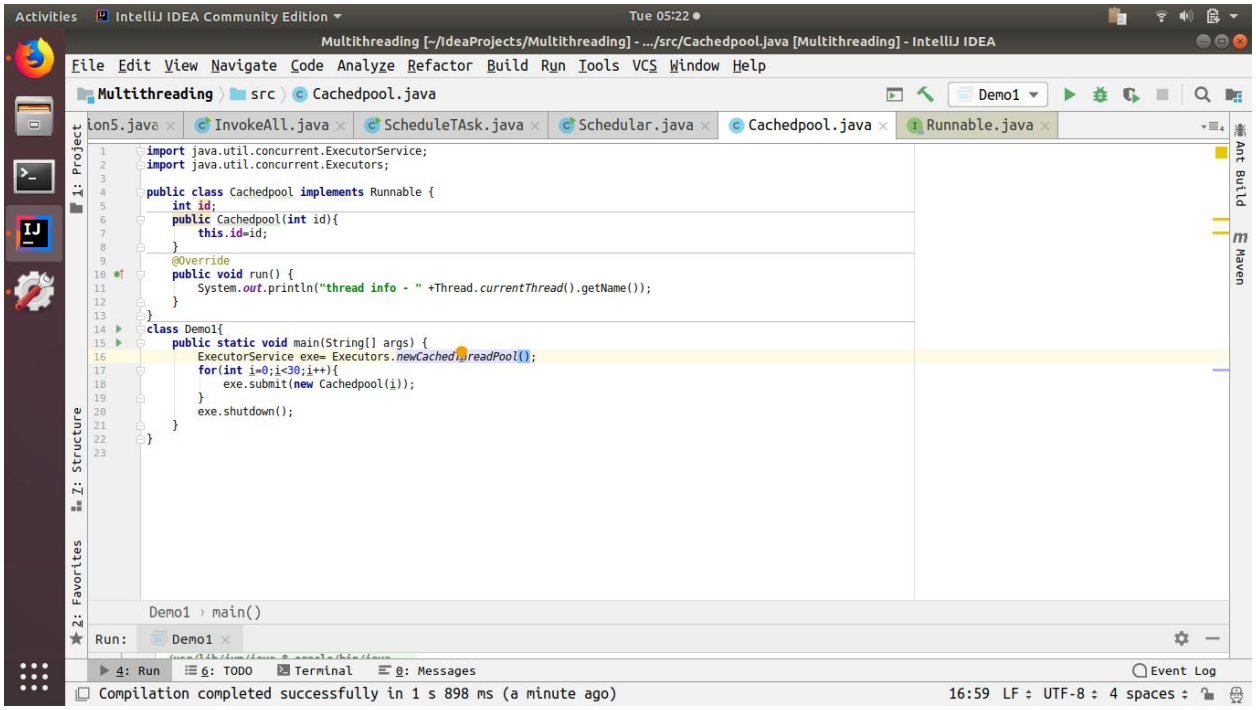

Activities
IntelliJ IDEA Community Edition
Tue 05:11
Multithreading [~/IdeaProjects/Multithreading] - .../src/Schedular.java [Multithreading] - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Multithreading src Schedular
Schedular
Project
InvokeAll.java
ScheduleTAsk.java
Schedular.java
Runnable.java
Multithreading ~/IdeaProjects/
.idea
out
src
Demo.java
InvokeAll
Question5
Schedular
ScheduleTAsk
SingleExecutorServiceDemo
Multithreading.iml
External Libraries
< 1.8 > /usr/lib/jvm/java-8-
Scratches and Consoles
import java.util.concurrent.ExecutorService;
import java.util.concurrent.Executors;
import java.util.concurrent.ScheduledExecutorService;
import java.util.concurrent.TimeUnit;
public class Schedular {
public static void main(String[] args) {
ScheduledExecutorService exe= Executors.newSingleThreadScheduledExecutor();
exe.scheduleWithFixedDelay(new Thread(()->{
System.out.println("Running 1st thread");
}), initialDelay: 3L, delay: 2L, TimeUnit.SECONDS);
}
}
Schedular › main()
Run: Schedular
/usr/lib/jvm/java-8-oracle/bin/java ...
Running 1st thread
Running 1st thread
Running 1st thread
Running 1st thread
Process finished with exit code 130 (interrupted by signal 2: SIGINT)
4: Run
6: TODO
Terminal
0: Messages
Event Log
Compilation completed successfully in 1 s 856 ms (moments ago)
11:18 LF UTF-8 4 spaces

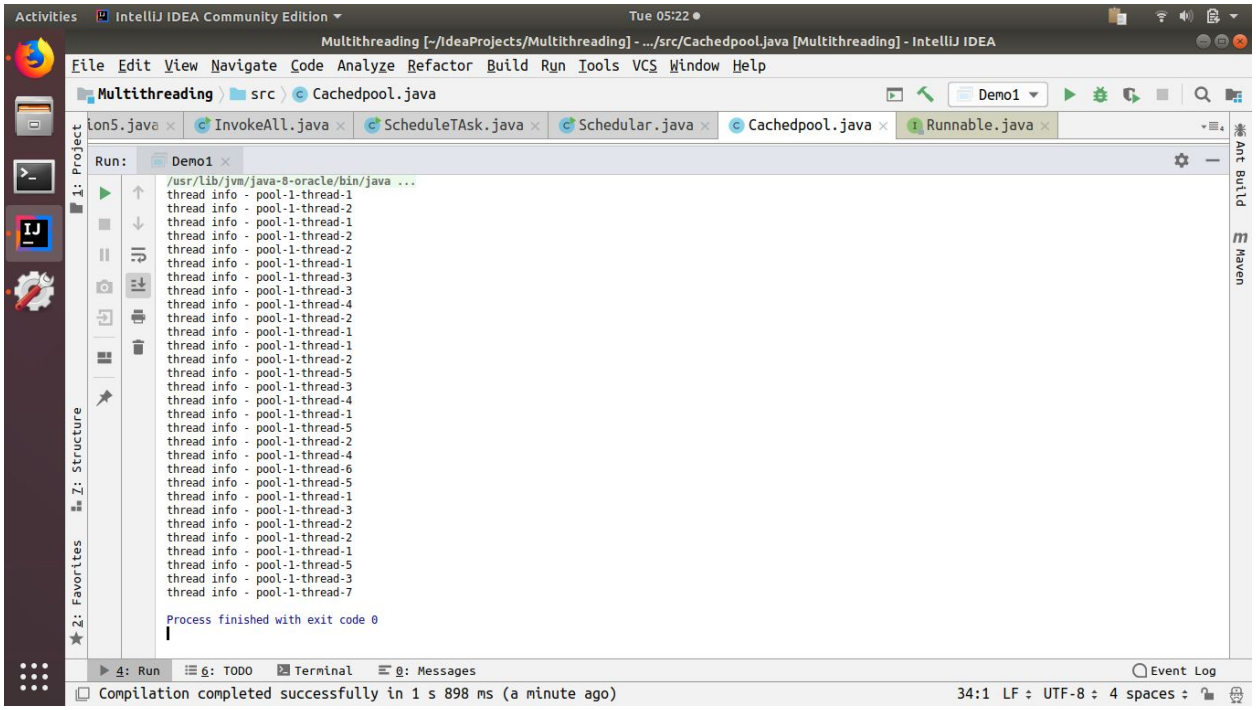

Activities
IntelliJ IDEA Community Edition
Tue 05:10
Multithreading [~/IdeaProjects/Multithreading] - .../src/Schedular.java [Multithreading] - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Multithreading src Schedular
Schedular
Project
InvokeAll.java
ScheduleTAsk.java
Schedular.java
Runnable.java
Multithreading ~/IdeaProjects/
.idea
out
src
Demo.java
InvokeAll
Question5
Schedular
ScheduleTAsk
SingleExecutorServiceDemo
Multithreading.iml
External Libraries
< 1.8 > /usr/lib/jvm/java-8-
Scratches and Consoles
import java.util.concurrent.ExecutorService;
import java.util.concurrent.Executors;
import java.util.concurrent.ScheduledExecutorService;
import java.util.concurrent.TimeUnit;
public class Schedular {
public static void main(String[] args) {
ScheduledExecutorService exe= Executors.newSingleThreadScheduledExecutor();
exe.schedule(new Thread(()->{
System.out.println("Running 1st thread");
}), delay: 3L, TimeUnit.SECONDS);
}
}
Schedular › main()
Run: Schedular
/usr/lib/jvm/java-8-oracle/bin/java ...
Running 1st thread
Process finished with exit code 130 (interrupted by signal 2: SIGINT)
4: Run
6: TODO
Terminal
0: Messages
Event Log
Compilation completed successfully in 1 s 823 ms (moments ago)
11:34 LF UTF-8 4 spaces



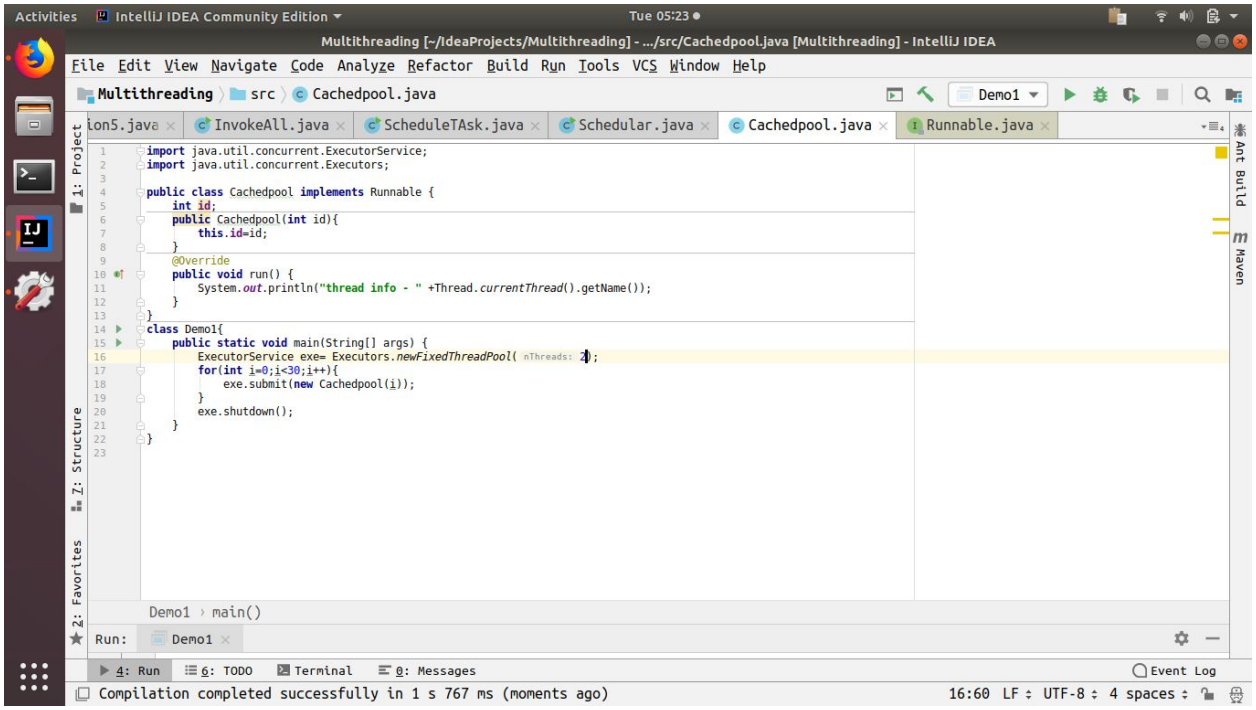
- Increase concurrency with Thread pools using newCachedThreadPool() and newFixedThreadPool().

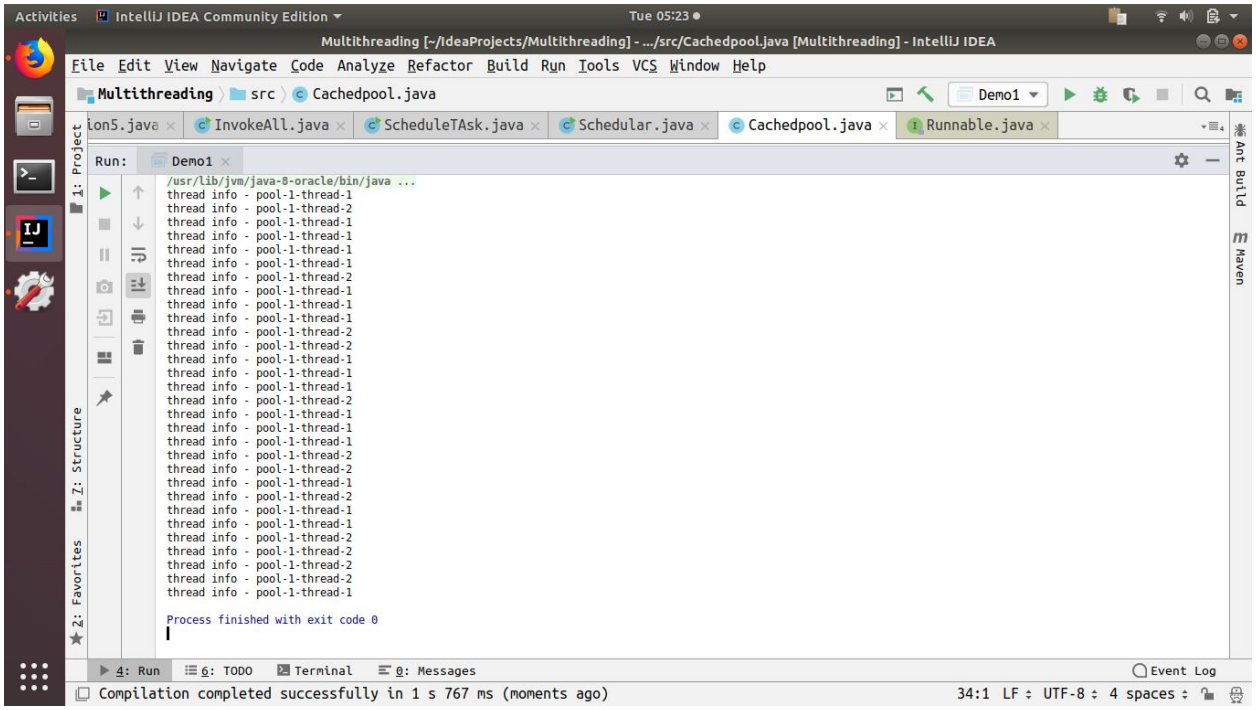
  Newcachedpool -



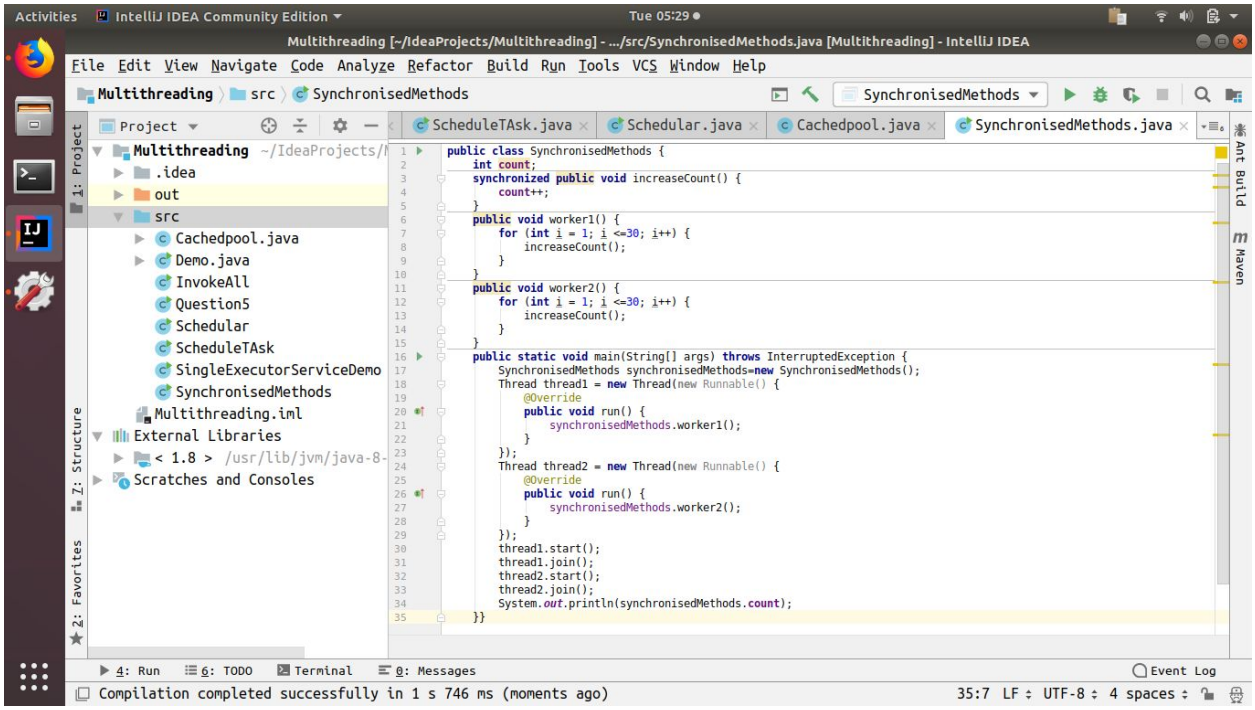


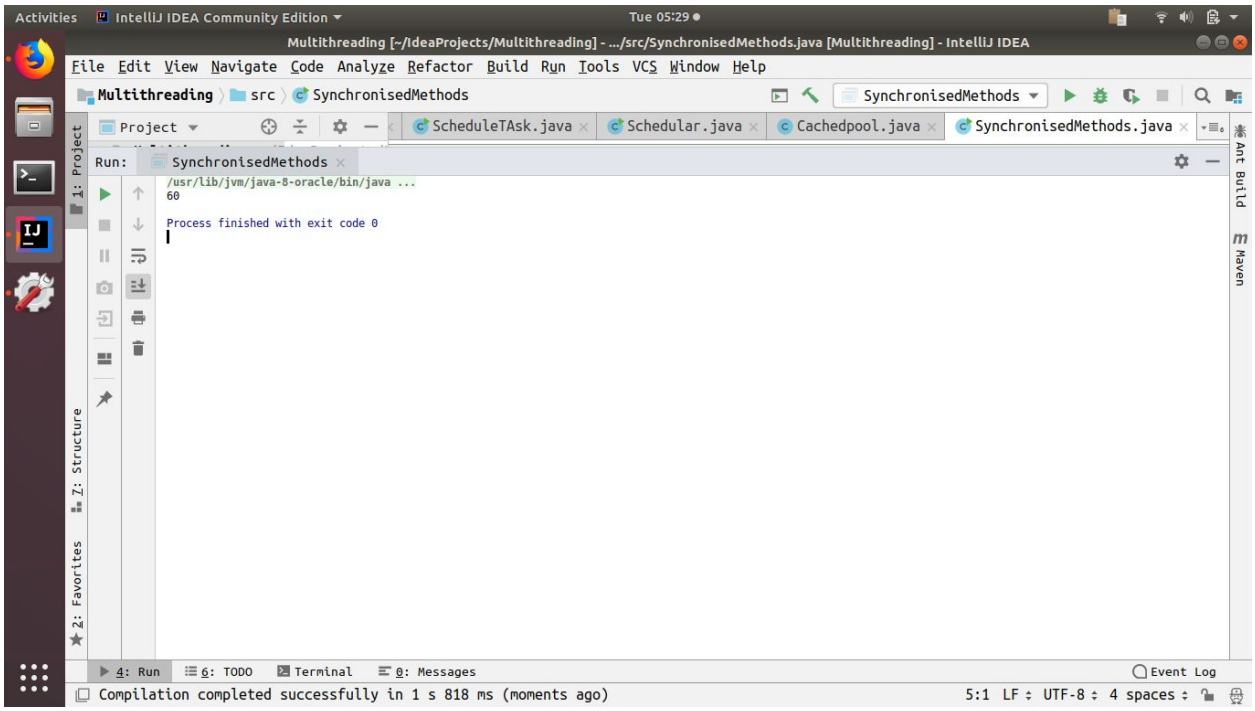
Fixedcachedpool -



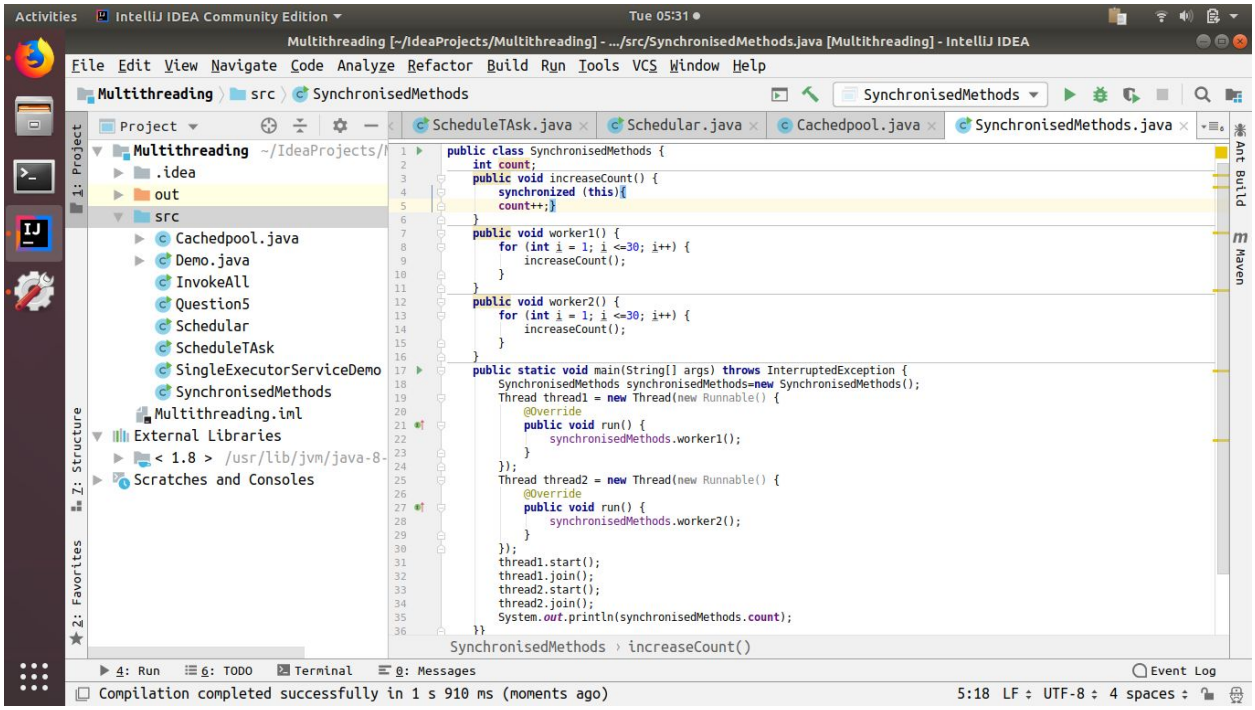


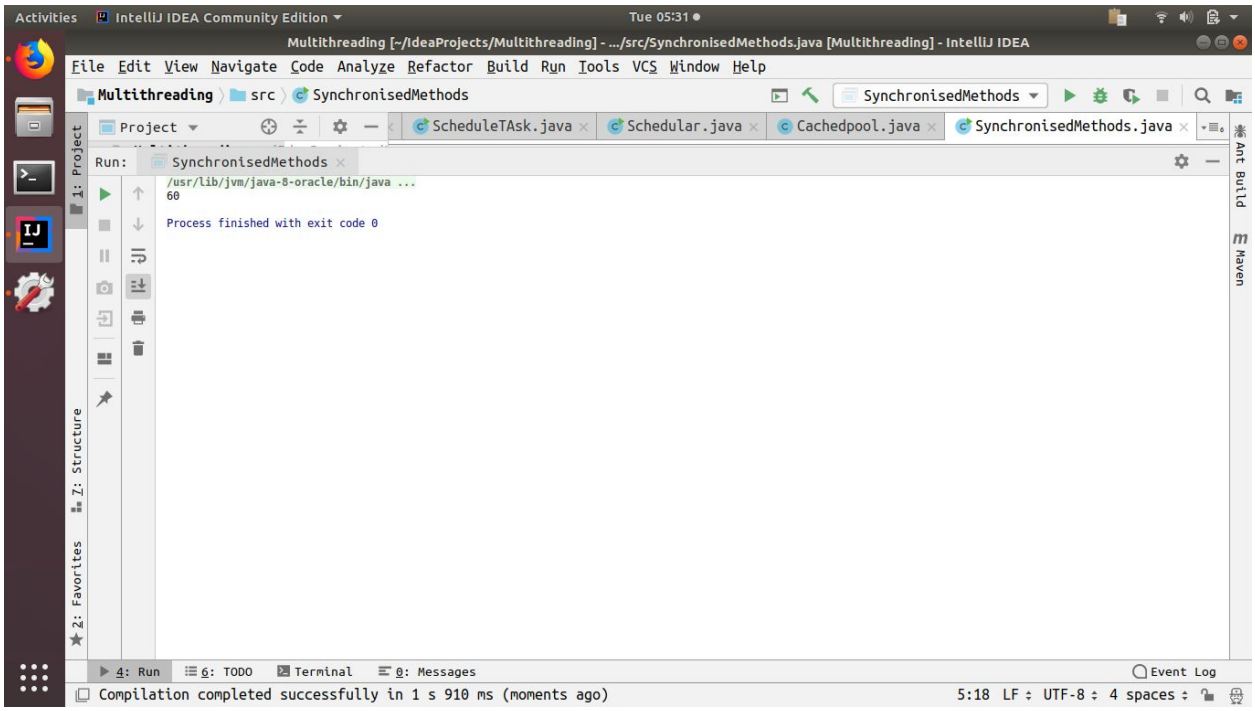
- Use Synchronize method to enable synchronization between multiple threads trying to access method at same time.



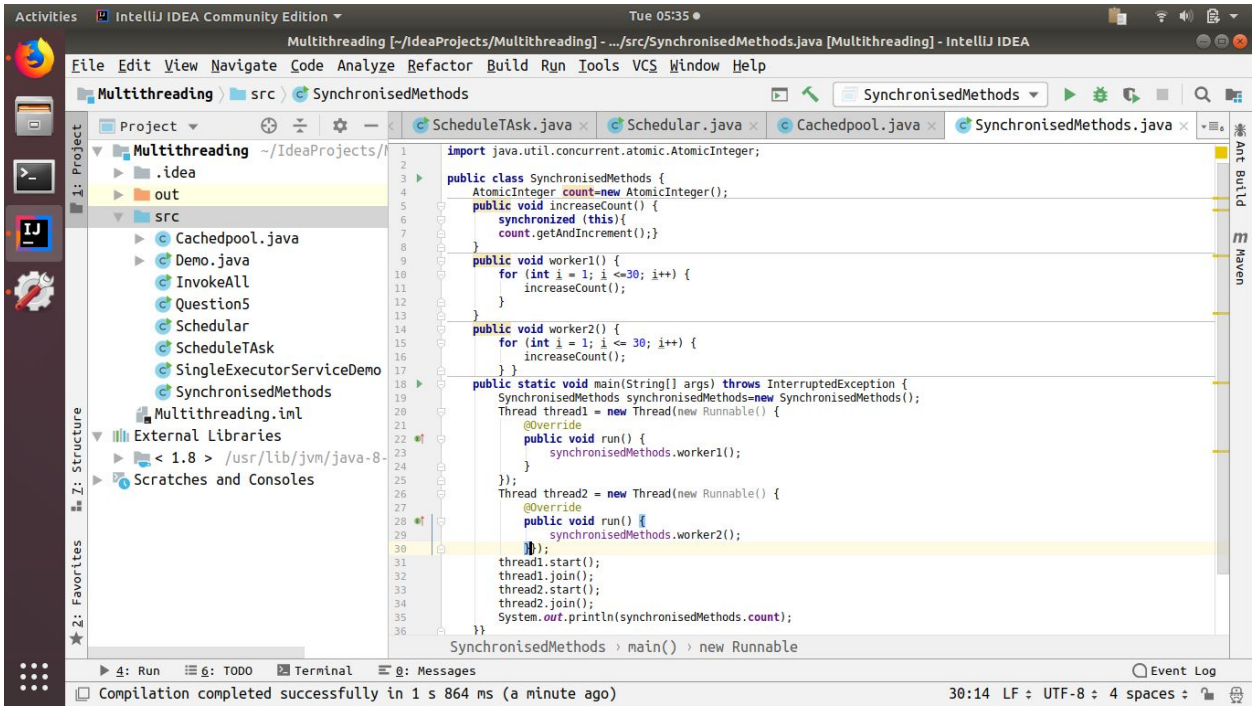


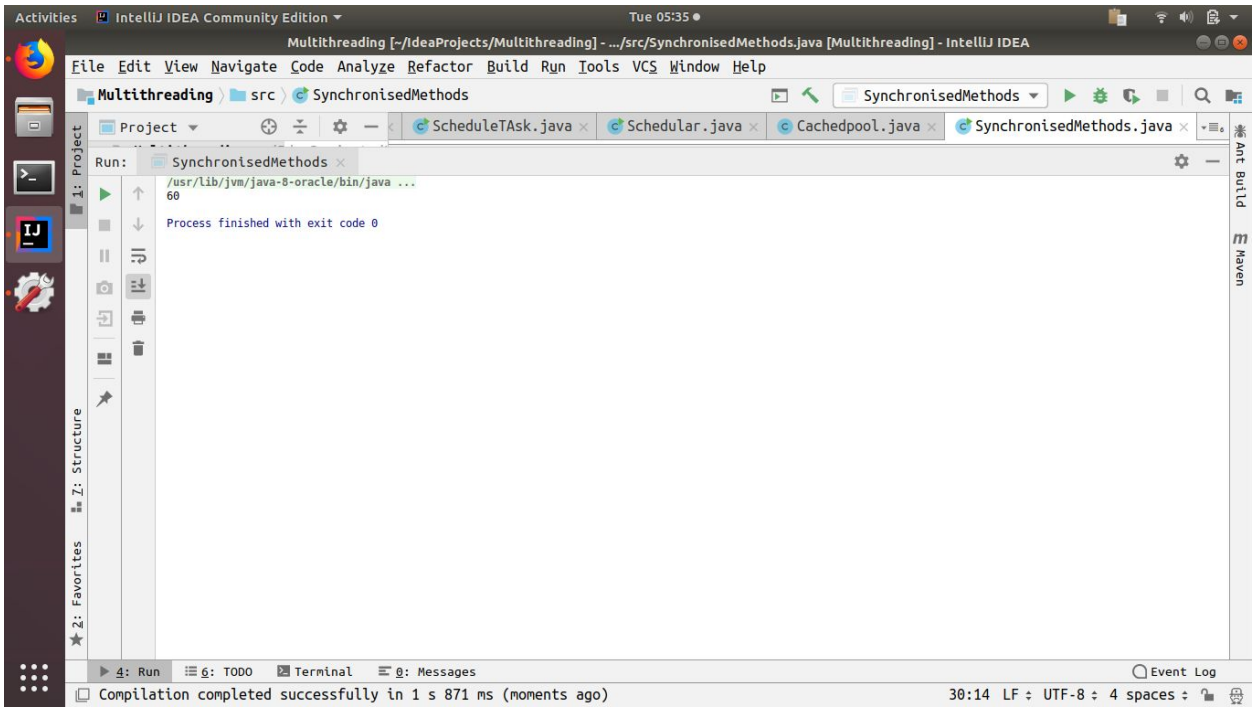
- Use Synchronize block to enable synchronization between multiple threads trying to access method at same time.



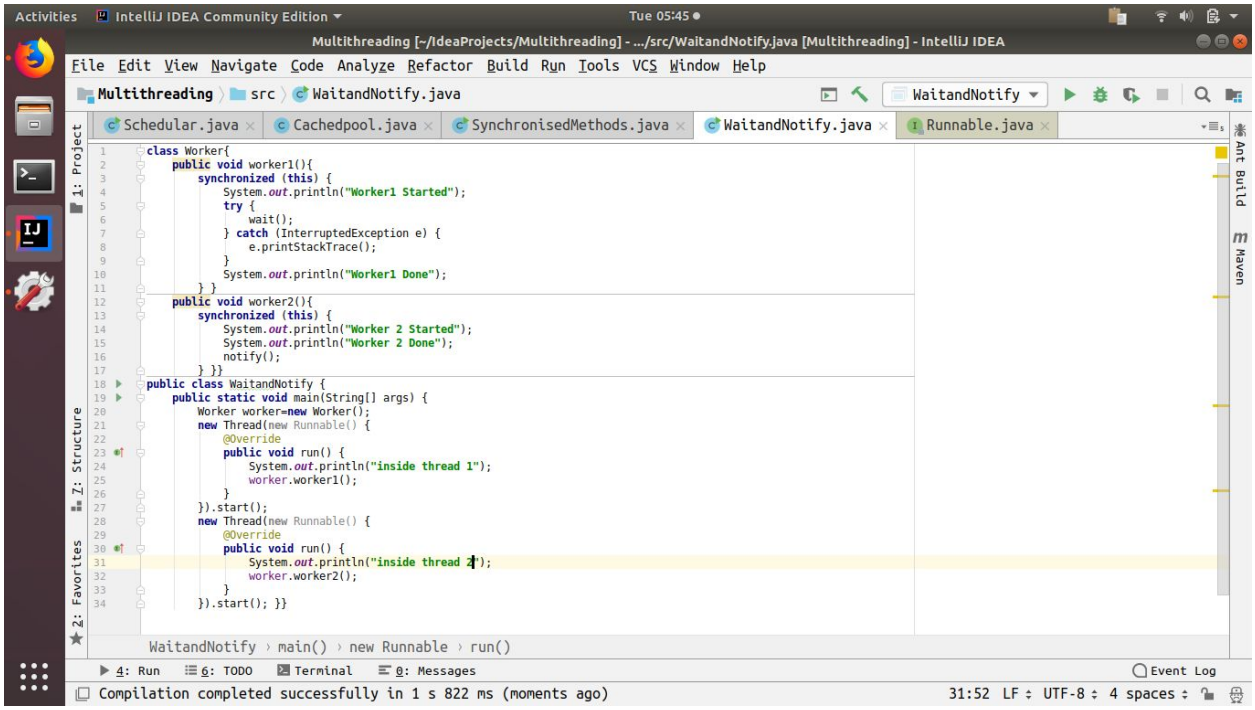


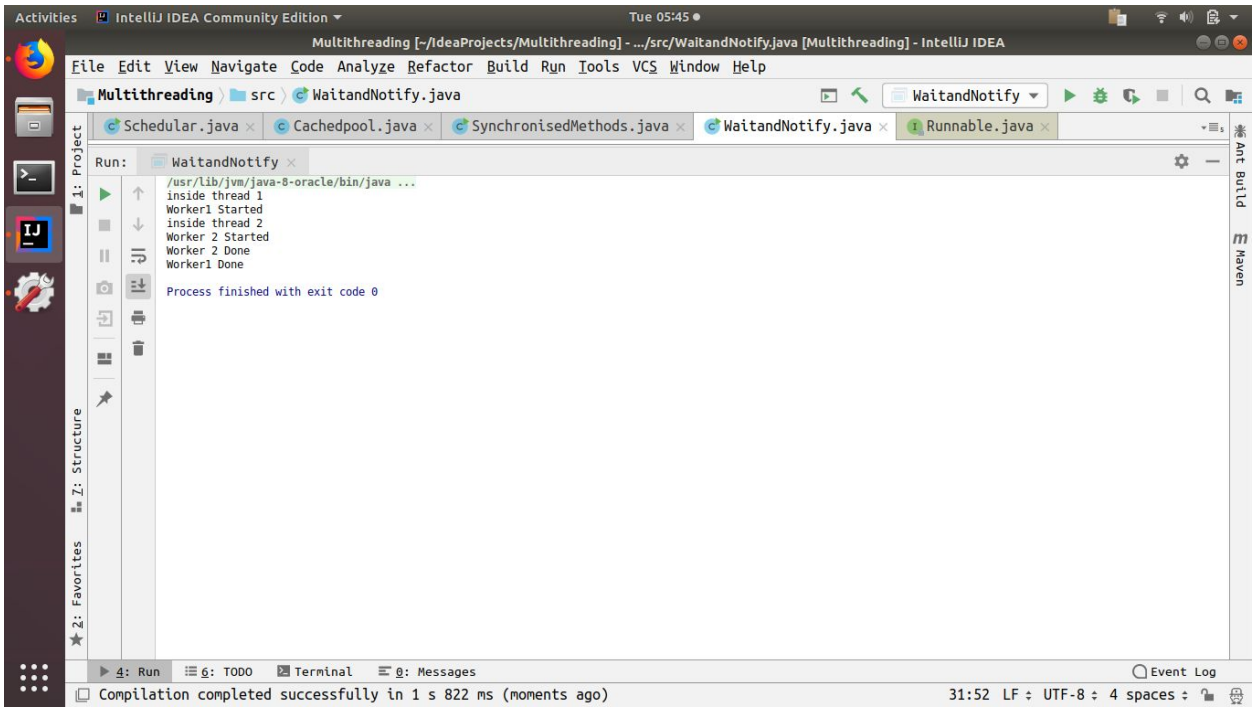
- Use Atomic Classes instead of Synchronize method and blocks.



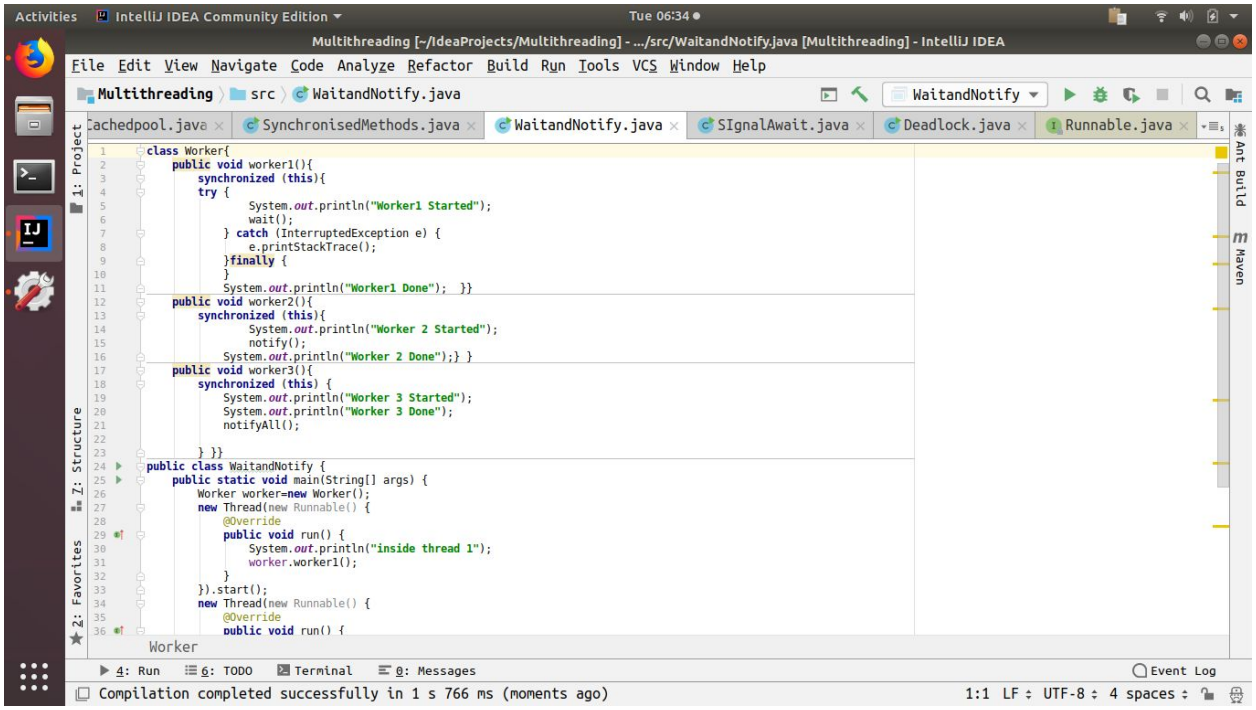


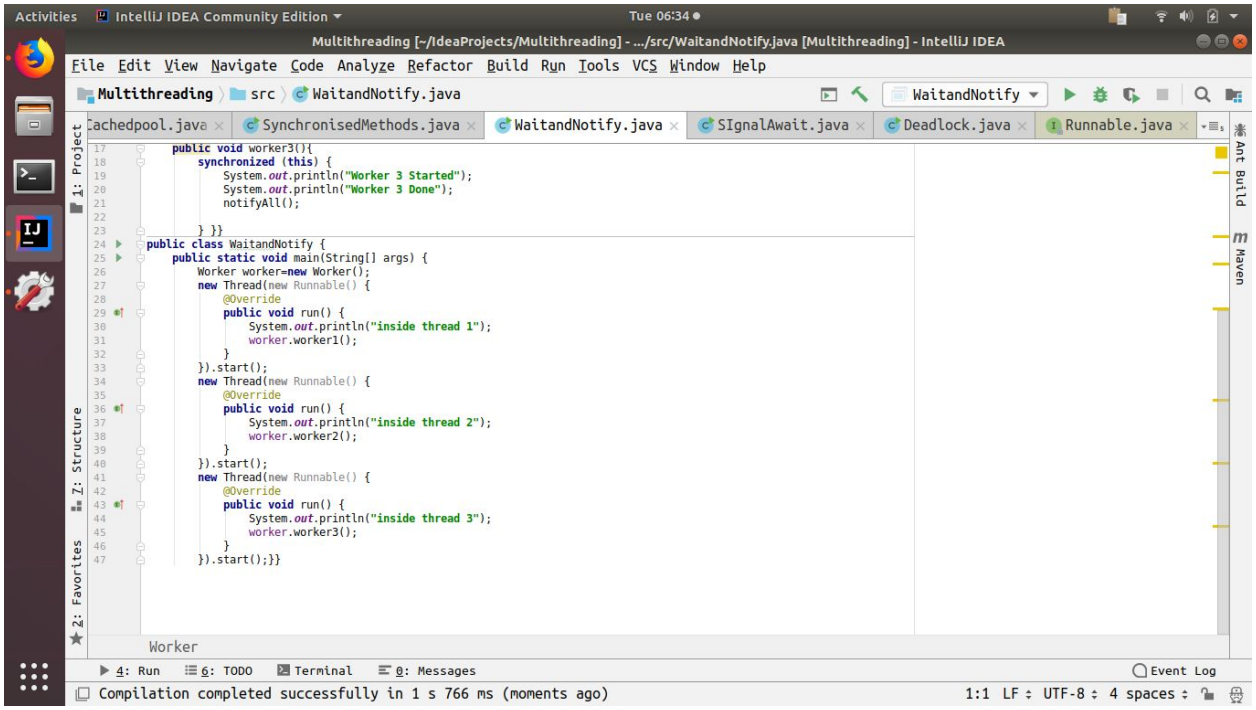
- Coordinate 2 threads using wait() and notify().

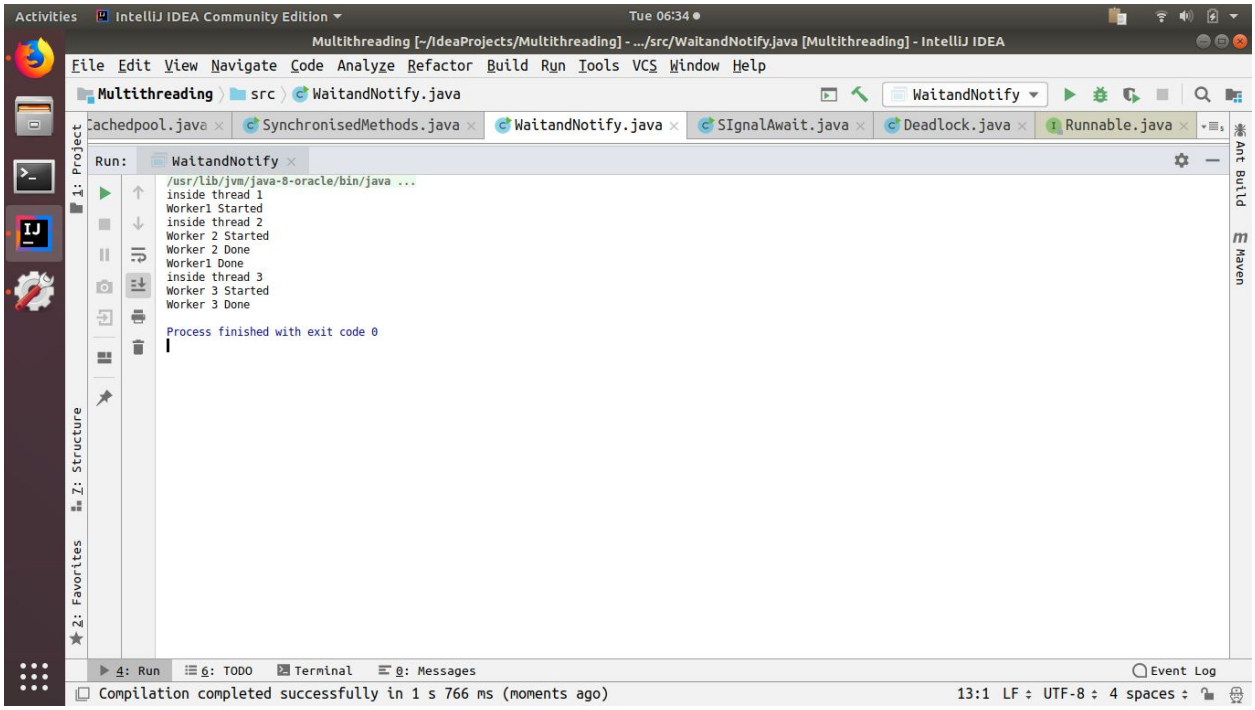




- Coordinate multiple threads using wait() and notifyAll()

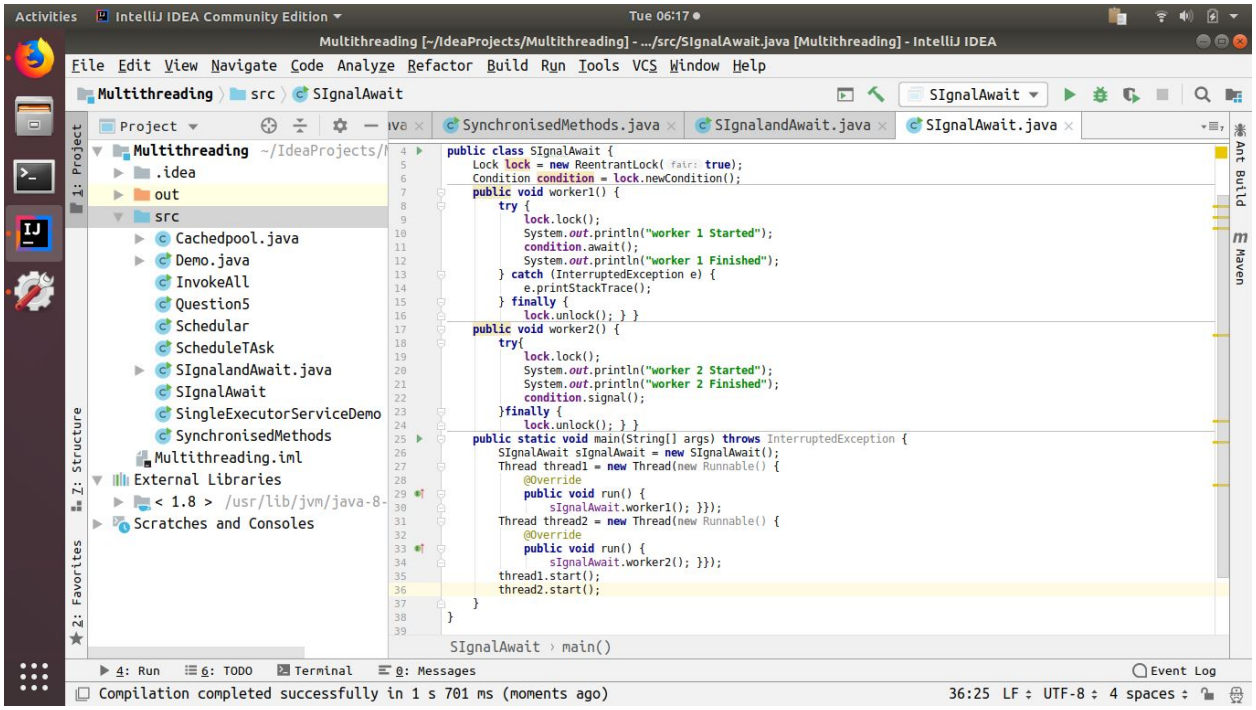

Activities
IntelliJ IDEA Community Edition
Tue 06:34
Multithreading [~/IdeaProjects/Multithreading] - .../src/WaitandNotify.java [Multithreading] - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Multithreading src WaitandNotify.java
WaitandNotify
Cachedpool.java
SynchronisedMethods.java
WaitandNotify.java
SIgnalAwait.java
Deadlock.java
Runnable.java
class Worker{
public void worker1(){
synchronized (this){
try {
System.out.println("Worker1 Started");
wait();
} catch (InterruptedException e) {
e.printStackTrace();
}finally {
}
System.out.println("Worker1 Done"); }}
public void worker2(){
synchronized (this){
System.out.println("Worker 2 Started");
notify();
System.out.println("Worker 2 Done");} }
public void worker3(){
synchronized (this) {
System.out.println("Worker 3 Started");
System.out.println("Worker 3 Done");
notifyAll();
} }}
public class WaitandNotify {
public static void main(String[] args) {
Worker worker=new Worker();
new Thread(new Runnable() {
@Override
public void run() {
System.out.println("inside thread 1");
worker.worker1();
}
}).start();
new Thread(new Runnable() {
@Override
public void run() {
Worker
4: Run
6: TODO
Terminal
0: Messages
Event Log
Compilation completed successfully in 1 s 766 ms (moments ago)
1:1 LF UTF-8 4 spaces
1: Project
7: Structure
2: Favorites
Ant Build
Maven

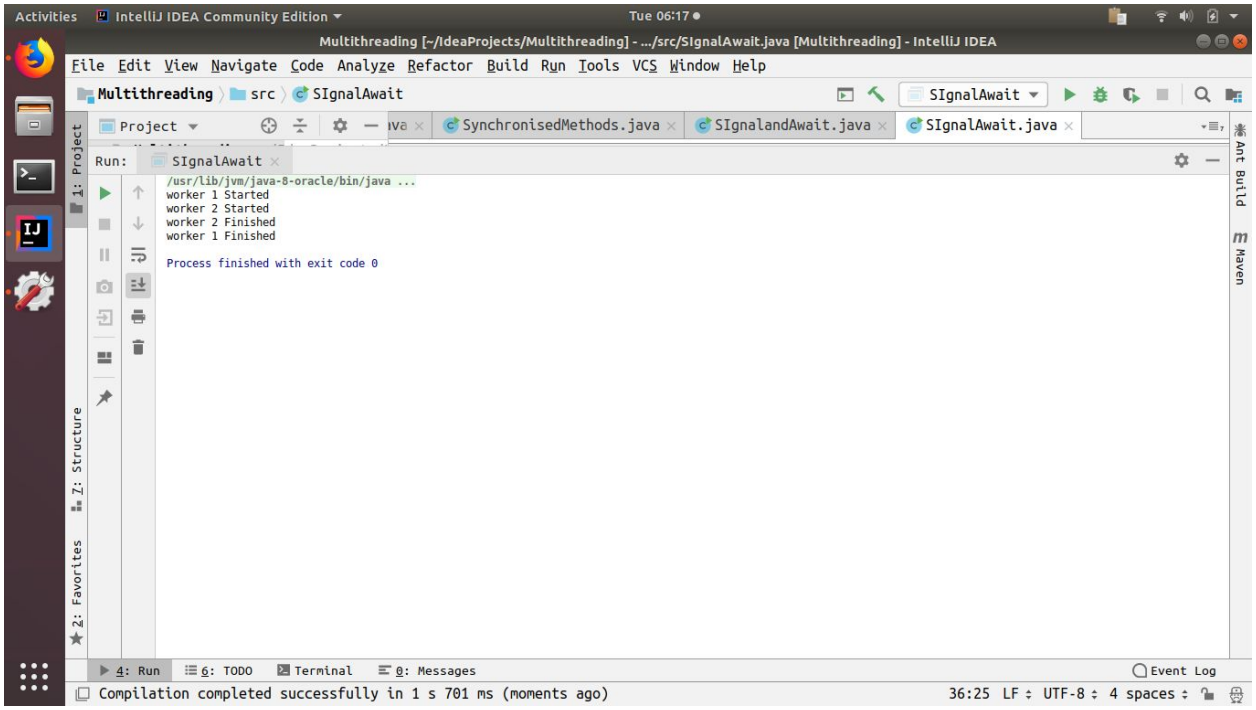

Activities
IntelliJ IDEA Community Edition
Tue 06:34
Multithreading [~/IdeaProjects/Multithreading] - .../src/WaitandNotify.java [Multithreading] - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Multithreading src WaitandNotify.java
WaitandNotify
Cachedpool.java
SynchronisedMethods.java
WaitandNotify.java
SIgnalAwait.java
Deadlock.java
Runnable.java
public void worker3(){
synchronized (this) {
System.out.println("Worker 3 Started");
System.out.println("Worker 3 Done");
notifyAll();
} }}
public class WaitandNotify {
public static void main(String[] args) {
Worker worker=new Worker();
new Thread(new Runnable() {
@Override
public void run() {
System.out.println("inside thread 1");
worker.worker1();
}
}).start();
new Thread(new Runnable() {
@Override
public void run() {
System.out.println("inside thread 2");
worker.worker2();
}
}).start();
new Thread(new Runnable() {
@Override
public void run() {
System.out.println("inside thread 3");
worker.worker3();
}
}).start();}}
Worker
4: Run
6: TODO
Terminal
0: Messages
Event Log
Compilation completed successfully in 1 s 766 ms (moments ago)
1:1 LF UTF-8 4 spaces
1: Project
7: Structure
2: Favorites
Ant Build
Maven



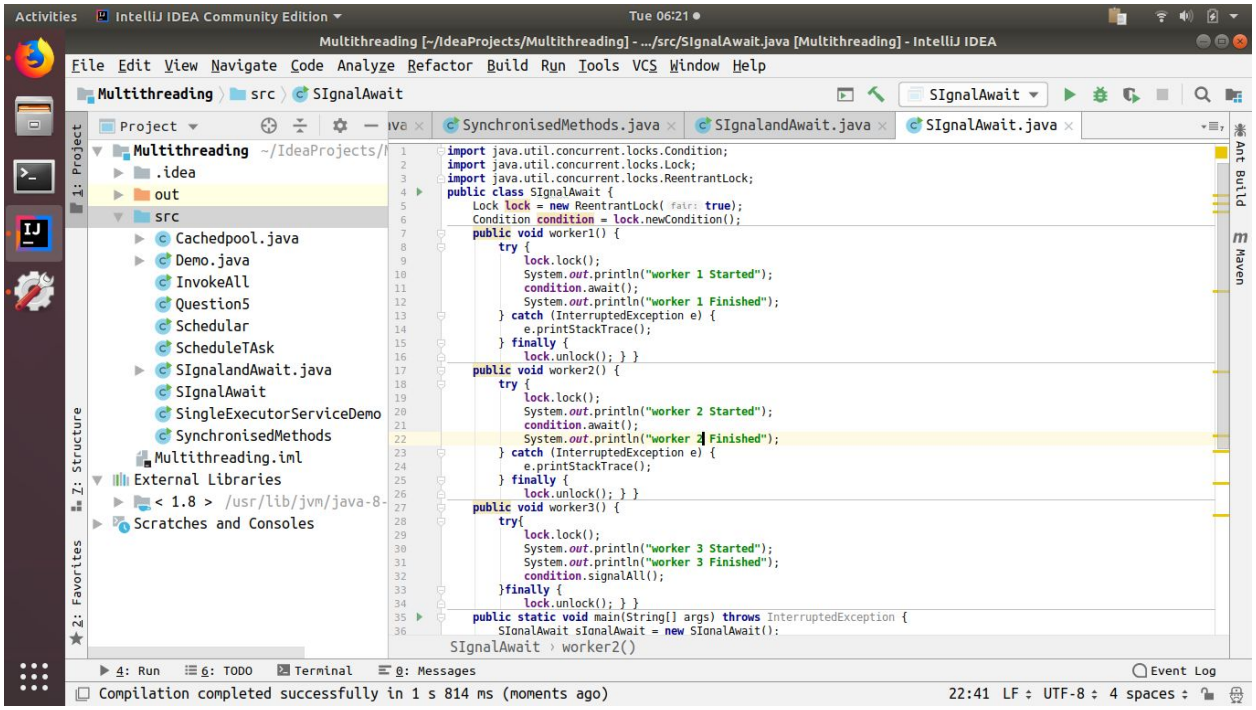
- Use Reentrantlock for coordinating 2 threads with signal(), signalAll() and wait().

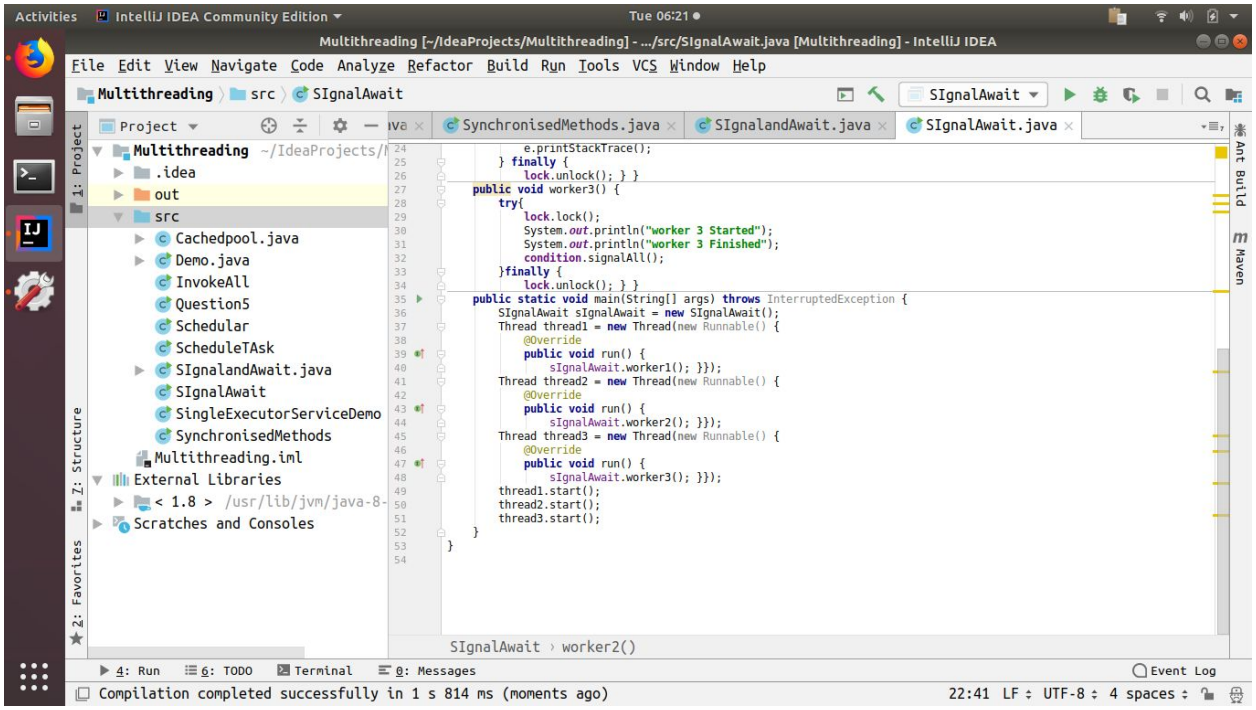
  SIGNAL AND AWAIT

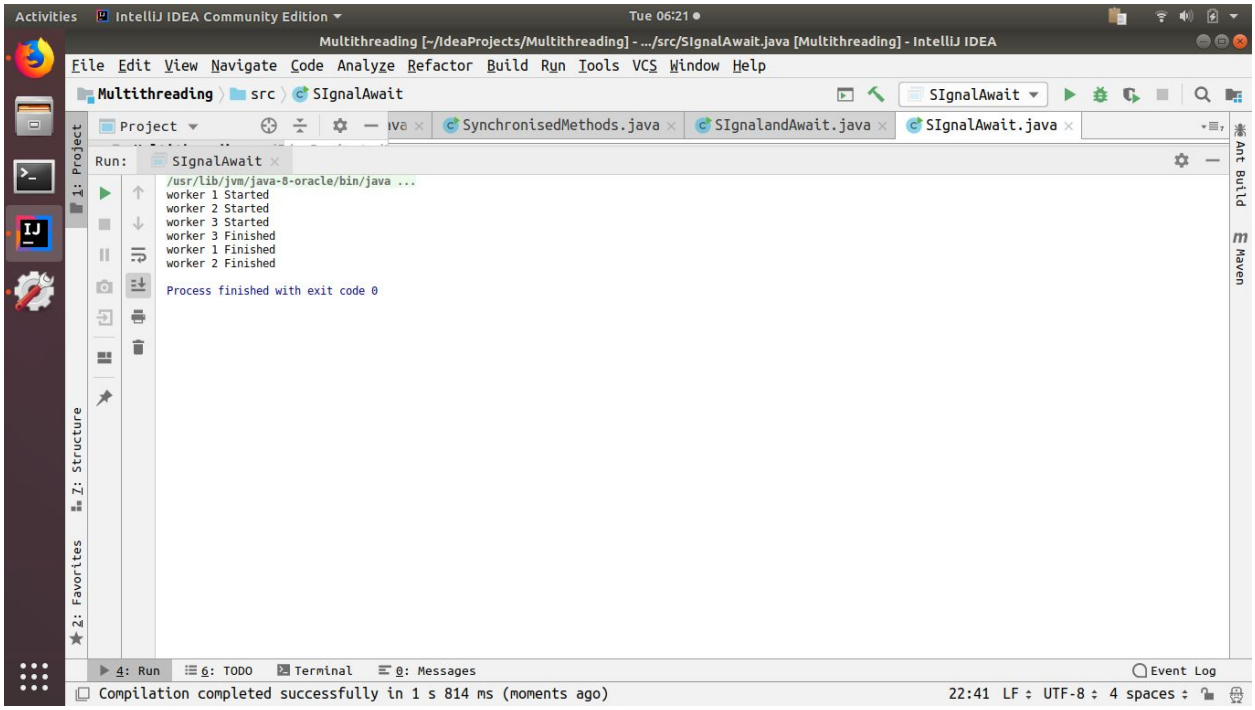




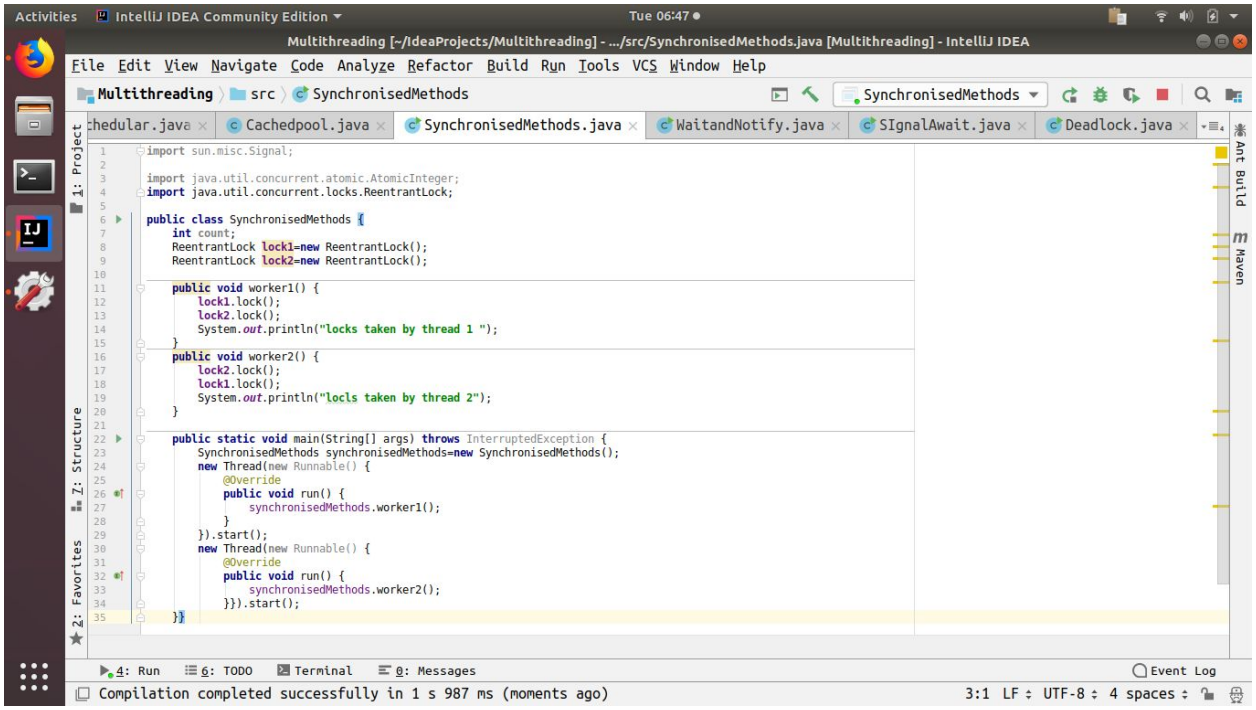
SIGNAL ALL AND AWAIT

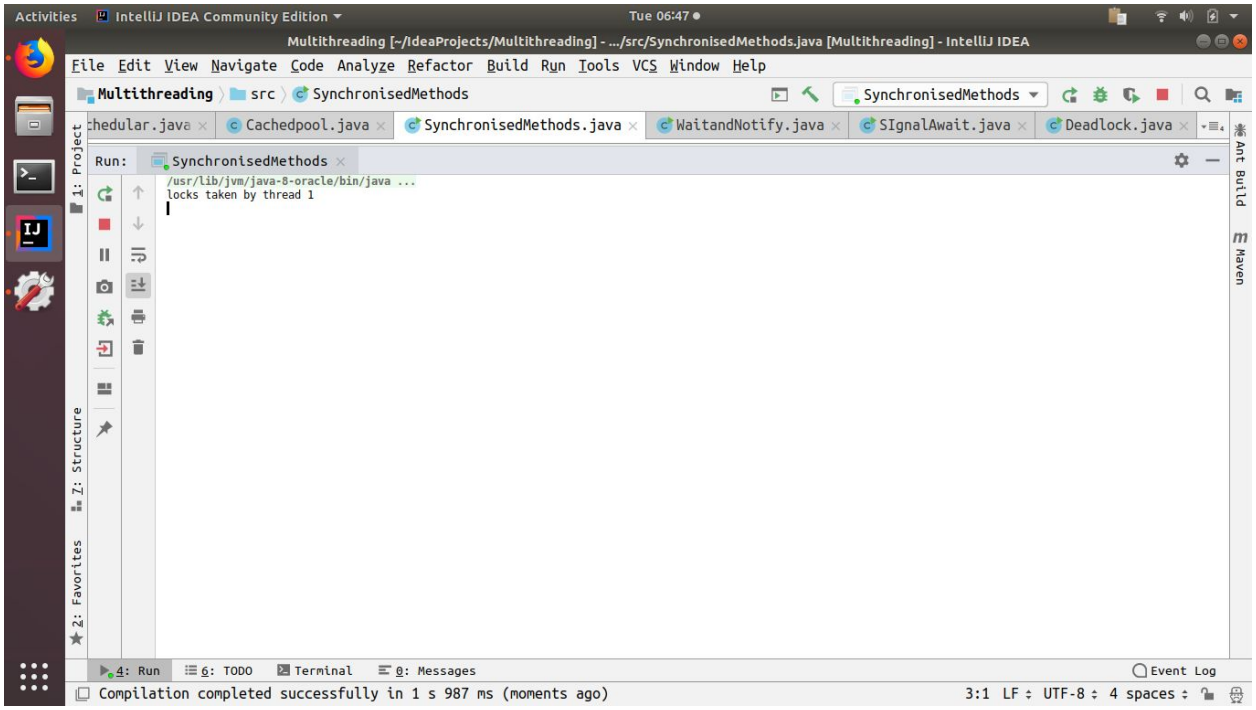

Activities
IntelliJ IDEA Community Edition
Tue 06:21
Multithreading [~/IdeaProjects/Multithreading] - .../src/SIgnalAwait.java [Multithreading] - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Multithreading src SIgnalAwait
SIgnalAwait
Project
SynchronisedMethods.java
SIgnalandAwait.java
SIgnalAwait.java
Multithreading ~/IdeaProjects/
.idea
out
src
Cachedpool.java
Demo.java
InvokeAll
QuestionS
Schedular
ScheduleTAsk
SIgnalandAwait.java
SIgnalAwait
SingleExecutorServiceDemo
SynchronisedMethods
Multithreading.iml
External Libraries
< 1.8 > /usr/lib/jvm/java-8-
Scratches and Consoles
import java.util.concurrent.locks.Condition;
import java.util.concurrent.locks.Lock;
import java.util.concurrent.locks.ReentrantLock;
public class SIgnalAwait {
Lock lock = new ReentrantLock(fair: true);
Condition condition = lock.newCondition();
public void worker1() {
try {
lock.lock();
System.out.println("worker 1 Started");
condition.await();
System.out.println("worker 1 Finished");
} catch (InterruptedException e) {
e.printStackTrace();
} finally {
lock.unlock(); } }
public void worker2() {
try {
lock.lock();
System.out.println("worker 2 Started");
condition.await();
System.out.println("worker 2 Finished");
} catch (InterruptedException e) {
e.printStackTrace();
} finally {
lock.unlock(); } }
public void worker3() {
try{
lock.lock();
System.out.println("worker 3 Started");
System.out.println("worker 3 Finished");
condition.signalAll();
}finally {
lock.unlock(); } }
public static void main(String[] args) throws InterruptedException {
SIgnalAwait sIgnalAwait = new SIgnalAwait();
SIgnalAwait › worker2()
4: Run
6: TODO
Terminal
0: Messages
Event Log
Compilation completed successfully in 1 s 814 ms (moments ago)
22:41 LF UTF-8 4 spaces
Ant Build
Maven
1: Project
7: Structure
2: Favorites


Activities
IntelliJ IDEA Community Edition
Tue 06:21
Multithreading [~/IdeaProjects/Multithreading] - .../src/SIgnalAwait.java [Multithreading] - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Multithreading src SIgnalAwait
SIgnalAwait
Project
SynchronisedMethods.java
SIgnalandAwait.java
SIgnalAwait.java
Multithreading ~/IdeaProjects/
.idea
out
src
Cachedpool.java
Demo.java
InvokeAll
QuestionS
Schedular
ScheduleTAsk
SIgnalandAwait.java
SIgnalAwait
SingleExecutorServiceDemo
SynchronisedMethods
Multithreading.iml
External Libraries
< 1.8 > /usr/lib/jvm/java-8-
Scratches and Consoles
e.printStackTrace();
} finally {
lock.unlock(); } }
public void worker3() {
try{
lock.lock();
System.out.println("worker 3 Started");
System.out.println("worker 3 Finished");
condition.signalAll();
}finally {
lock.unlock(); } }
public static void main(String[] args) throws InterruptedException {
SIgnalAwait sIgnalAwait = new SIgnalAwait();
Thread thread1 = new Thread(new Runnable() {
@Override
public void run() {
sIgnalAwait.worker1(); });
Thread thread2 = new Thread(new Runnable() {
@Override
public void run() {
sIgnalAwait.worker2(); });
Thread thread3 = new Thread(new Runnable() {
@Override
public void run() {
sIgnalAwait.worker3(); });
thread1.start();
thread2.start();
thread3.start();
}
}
SIgnalAwait › worker2()
4: Run
6: TODO
Terminal
0: Messages
Event Log
Compilation completed successfully in 1 s 814 ms (moments ago)
22:41 LF UTF-8 4 spaces
Ant Build
Maven
1: Project
7: Structure
2: Favorites

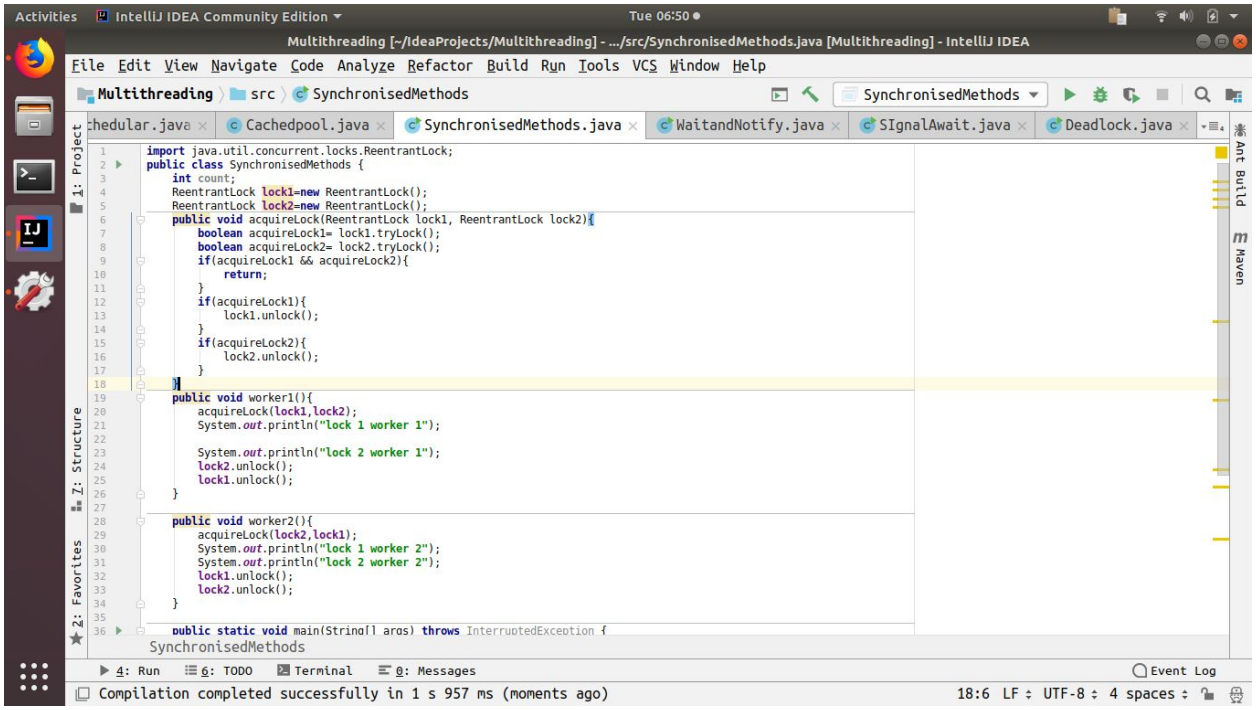


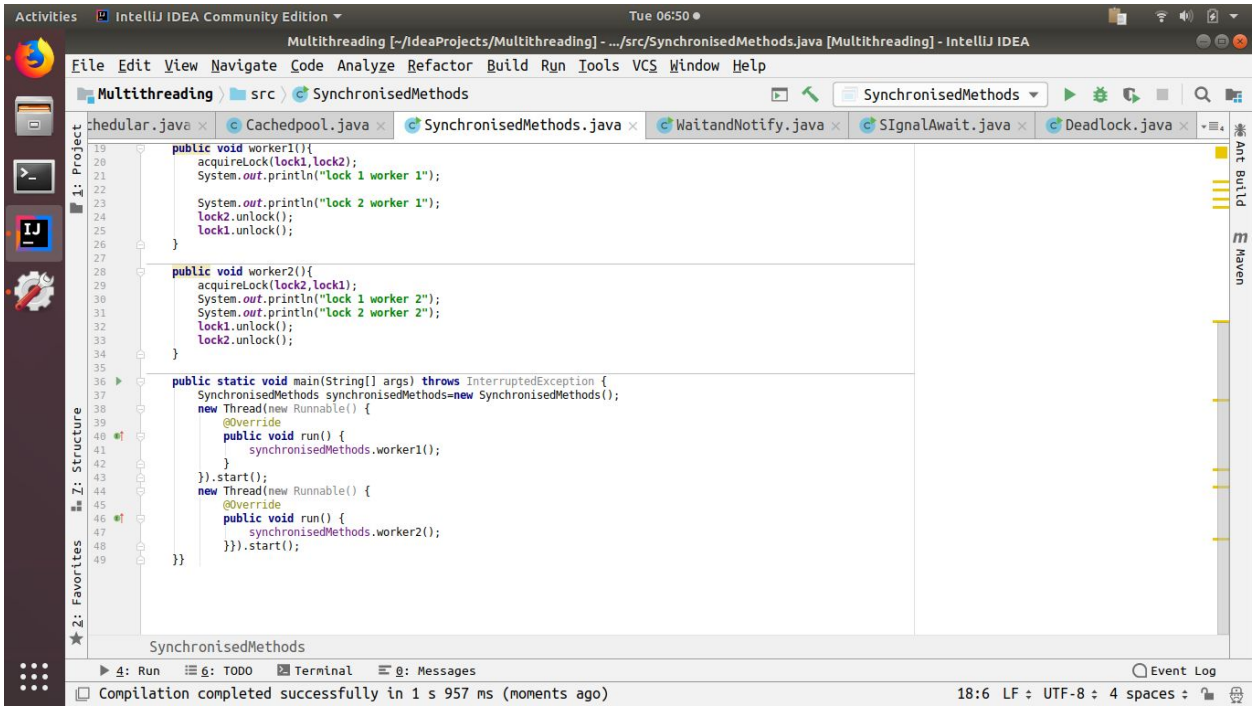
- Create a deadlock and Resolve it using tryLock().

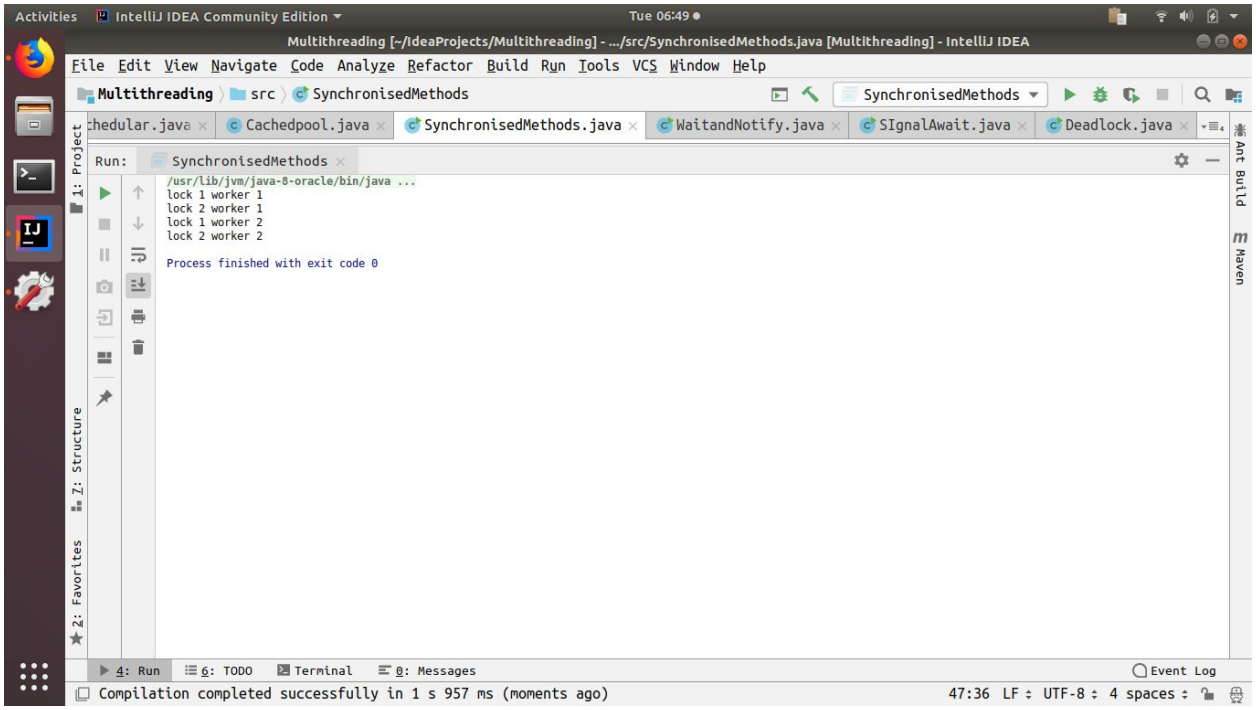




USING TRYLOCK -


Activities
IntelliJ IDEA Community Edition
Tue 06:50
Multithreading [~/IdeaProjects/Multithreading] - .../src/SynchronisedMethods.java [Multithreading] - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Multithreading src SynchronisedMethods
SynchronisedMethods
chedular.java
Cachedpool.java
SynchronisedMethods.java
WaitandNotify.java
SIgnalAwait.java
Deadlock.java
1: Project
7: Structure
2: Favorites
Ant Build
Maven
import java.util.concurrent.locks.ReentrantLock;
public class SynchronisedMethods {
int count;
ReentrantLock lock1=new ReentrantLock();
ReentrantLock lock2=new ReentrantLock();
public void acquireLock(ReentrantLock lock1, ReentrantLock lock2){
boolean acquireLock1= lock1.tryLock();
boolean acquireLock2= lock2.tryLock();
if(acquireLock1 && acquireLock2){
return;
}
if(acquireLock1){
lock1.unlock();
}
if(acquireLock2){
lock2.unlock();
}
}
public void worker1(){
acquireLock(lock1,lock2);
System.out.println("lock 1 worker 1");
System.out.println("lock 2 worker 1");
lock2.unlock();
lock1.unlock();
}
public void worker2(){
acquireLock(lock2,lock1);
System.out.println("lock 1 worker 2");
System.out.println("lock 2 worker 2");
lock1.unlock();
lock2.unlock();
}
public static void main(String[] args) throws InterruptedException {
SynchronisedMethods
4: Run
6: TODO
Terminal
0: Messages
Event Log
Compilation completed successfully in 1 s 957 ms (moments ago)
18:6 LF UTF-8 4 spaces


Activities
IntelliJ IDEA Community Edition
Tue 06:50
Multithreading [~/IdeaProjects/Multithreading] - .../src/SynchronisedMethods.java [Multithreading] - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Multithreading src SynchronisedMethods
SynchronisedMethods
chedular.java
Cachedpool.java
SynchronisedMethods.java
WaitandNotify.java
SIgnalAwait.java
Deadlock.java
1: Project
7: Structure
2: Favorites
Ant Build
Maven
public void worker1(){
acquireLock(lock1,lock2);
System.out.println("lock 1 worker 1");
System.out.println("lock 2 worker 1");
lock2.unlock();
lock1.unlock();
}
public void worker2(){
acquireLock(lock2,lock1);
System.out.println("lock 1 worker 2");
System.out.println("lock 2 worker 2");
lock1.unlock();
lock2.unlock();
}
public static void main(String[] args) throws InterruptedException {
SynchronisedMethods synchronisedMethods=new SynchronisedMethods();
new Thread(new Runnable() {
@Override
public void run() {
synchronisedMethods.worker1();
}
}).start();
new Thread(new Runnable() {
@Override
public void run() {
synchronisedMethods.worker2();
}}).start();
}}
SynchronisedMethods
4: Run
6: TODO
Terminal
0: Messages
Event Log
Compilation completed successfully in 1 s 957 ms (moments ago)
18:6 LF UTF-8 4 spaces


Activities
IntelliJ IDEA Community Edition
Tue 06:49
Multithreading [~/IdeaProjects/Multithreading] - .../src/SynchronisedMethods.java [Multithreading] - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Multithreading src SynchronisedMethods
SynchronisedMethods
chedular.java
Cachedpool.java
SynchronisedMethods.java
WaitandNotify.java
SIgnalAwait.java
Deadlock.java
Run: SynchronisedMethods
/usr/lib/jvm/java-8-oracle/bin/java ...
lock 1 worker 1
lock 2 worker 1
lock 1 worker 2
lock 2 worker 2
Process finished with exit code 0
1: Project
2: Favorites
7: Structure
Ant Build
Maven
4: Run
6: TODO
Terminal
0: Messages
Event Log
Compilation completed successfully in 1 s 957 ms (moments ago)
47:36 LF UTF-8 4 spaces