

## Q1: Create and Run a Thread using Runnable Interface and Thread class.

The screenshot shows the IntelliJ IDEA IDE interface. The top bar displays "Activities Jetbrains-idea-ce" and the date "Wed 12:54". The main window title is "multiThreading - Q1.java". The left sidebar shows a project structure for "multiThreading" with files like Q1.java, ThreadClass, UsingRunnable, Q2.java, etc. The right pane displays the Java code for Q1.java:

```
1 class ThreadClass extends Thread{
2     @Override
3     public void run() {
4         System.out.println("Thread is being executed using thread class");
5     }
6 }
7 class UsingRunnable implements Runnable{
8     @Override
9     public void run() { System.out.println("Thread is being executed using runnable interface"); }
10}
11
12 public class Q1 {
13     public static void main(String[] args) {
14         ThreadClass th = new ThreadClass();
15         UsingRunnable inter = new UsingRunnable();
16         Thread t1 = new Thread(inter);
17         th.start();
18         t1.start();
19     }
20 }
21
22
```

The "Run" tool window at the bottom shows the output of the program:

```
/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -javaagent:/home/ttn/Downloads/ideaIC-2020.3.2/idea-IC-203.7148.57/lib/idea_rt.jar=38291:/home/ttn/Downloads/ideaIC-2020.3.2/idea-IC-203.7148.57/lib/idea_rt.jar
Thread is being executed using thread class
Thread is being executed using runnable interface

Process finished with exit code 0
```

The status bar at the bottom right indicates "All files are up-to-date (a minute ago)" and the time "20:6".

## Q2: Use sleep and join methods with thread.

```
Activities JetBrains-Idea-ce Wed 12:55
multiThreading - Q2.java
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
multiThreading src Q2.java Q2
Project .idea out src Q1.java Q1 ThreadClass UsingRunnable Q2.java JoinAndSleep Q2
public class JoinAndSleep implements Runnable{
    @Override
    public void run() {
        Thread t1 = Thread.currentThread();
        for(int i=0;i<5;i++){
            System.out.println("current thread is: " + t1.getName());
        }
    }
}
public class Q2 {
    public static void main(String[] args) {
        JoinAndSleep t1 = new JoinAndSleep();
        JoinAndSleep t2 = new JoinAndSleep();
        JoinAndSleep t3 = new JoinAndSleep();
        Thread th1 = new Thread(t1);
        Thread th2 = new Thread(t2);
        Thread th3 = new Thread(t3);
    }
}
Run: Q2
current thread is: Thread-2
current thread is: Thread-0
current thread is: Thread-0
current thread is: Thread-0
current thread is: Thread-0
Process finished with exit code 0
Event Log
11:14 LF UTF-8 4 spaces
All files are up-to-date (moments ago)
```

```
Activities JetBrains-Idea-ce Wed 12:55
multiThreading - Q2.java
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
multiThreading src Q2.java Q2
Project .idea out src Q1.java Q1 ThreadClass UsingRunnable Q2.java JoinAndSleep Q2
public class Q2 {
    public static void main(String[] args) {
        JoinAndSleep t1 = new JoinAndSleep();
        JoinAndSleep t2 = new JoinAndSleep();
        JoinAndSleep t3 = new JoinAndSleep();
        Thread th1 = new Thread(t1);
        Thread th2 = new Thread(t2);
        Thread th3 = new Thread(t3);
        th1.start();
        th2.start();
        th3.start();
        try{
            th1.sleep(1000);
        } catch (Exception e){
            System.out.println(e);
        }
    }
}
Run: Q2
current thread is: Thread-2
current thread is: Thread-0
current thread is: Thread-0
current thread is: Thread-0
current thread is: Thread-0
Process finished with exit code 0
Event Log
11:14 LF UTF-8 4 spaces
All files are up-to-date (moments ago)
```

Activities Jetbrains-idea-ce ▾ Wed 12:55

multithreading - Q2.java

```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
multithreading > src > Q2.java > Q2
Project .idea out src
  Q1.java
  Q1
  ThreadClass
  UsingRunnable
  Q2.java
  JoinAndSleep
  Q2
  Q3.java
  Q4.java
  Q5.java
  Q6.java
  Q7.java
  Q8.java
  Q9.java
Q2.java
18     ...
19     ...
20     ...
21     ...
22     ...
23     ...
24     ...
25     ...
26     ...
27     ...
28     ...
29     ...
30     ...
31     ...
32     ...
33     ...
34     ...
35     ...
Thread th3 = new Thread(t3);
th1.start();
th2.start();
th3.start();
try{
    th1.sleep( 1000 );
}
catch (Exception e){
    System.out.println(e);
}
try{
    th3.join();
}
catch (Exception e){
    System.out.println(e);
}
Run: Q2
current thread is: Thread-2
current thread is: Thread-0
current thread is: Thread-0
current thread is: Thread-0
current thread is: Thread-0
Process finished with exit code 0
Run TODO Problems Terminal Build Event Log
All files are up-to-date (moments ago)
11:14 LF UTF-8 4 spaces
```

Activities Jetbrains-idea-ce ▾ Wed 12:55

multithreading - Q2.java

```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
multithreading > src > Q2.java > Q2
Project .idea out src
  Q1.java
  Q1
  ThreadClass
  UsingRunnable
  Q2.java
  JoinAndSleep
  Q2
  Q3.java
  Q4.java
  Q5.java
  Q6.java
  Q7.java
  Q8.java
  Q9.java
Q2.java
18     ...
19     ...
20     ...
21     ...
22     ...
23     ...
24     ...
25     ...
26     ...
27     ...
28     ...
29     ...
30     ...
31     ...
32     ...
33     ...
34     ...
35     ...
Thread th3 = new Thread(t3);
th1.start();
th2.start();
th3.start();
try{
    th1.sleep( 1000 );
}
catch (Exception e){
    System.out.println(e);
}
try{
    th3.join();
}
catch (Exception e){
    System.out.println(e);
}
Run: Q2
/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -javaagent:/home/ttn/Downloads/ideaIC-2020.3.2/idea-IC-203.7148.57/lib/idea_rt.jar=46303:/home/ttn/Downloads/ideaIC-2020.3.2/idea-IC-203.7148.57/lib/idea_rt.jar
current thread is: Thread-1
current thread is: Thread-2
current thread is: Thread-0
Process finished with exit code 0
Run TODO Problems Terminal Build Event Log
All files are up-to-date (moments ago)
11:14 LF UTF-8 4 spaces
```

### Q3: Use a singleThreadExecutor to submit multiple threads.

The screenshot shows the IntelliJ IDEA interface with the project 'multiThreading' open. The code editor displays `Q3.java` which contains the following code:

```
public class Q3 {
    public static void main(String[] args) {
        ExecutorService executor = Executors.newSingleThreadExecutor();
        RunThread r1 = new RunThread();
        RunThread r2 = new RunThread();
        executor.submit(r1);
        executor.submit(r2);
        executor.shutdown();
    }

    class RunThread implements Runnable {
        @Override
        public void run() {
            Thread t = new Thread();
            for (int i=0;i<5;i++){
                try {
                    Thread.sleep(1000);
                }catch (Exception e){
                    System.out.println(e);
                }
                System.out.println(t.getName());
            }
        }
    }
}
```

The 'Run' tool window shows the output of running `Q2`:

```
current thread is: Thread-0
```

The screenshot shows the IntelliJ IDEA interface with the project 'multiThreading' open. The code editor displays `Q3.java` which contains the following code:

```
import java.util.concurrent.ExecutorService;
import java.util.concurrent.Executors;

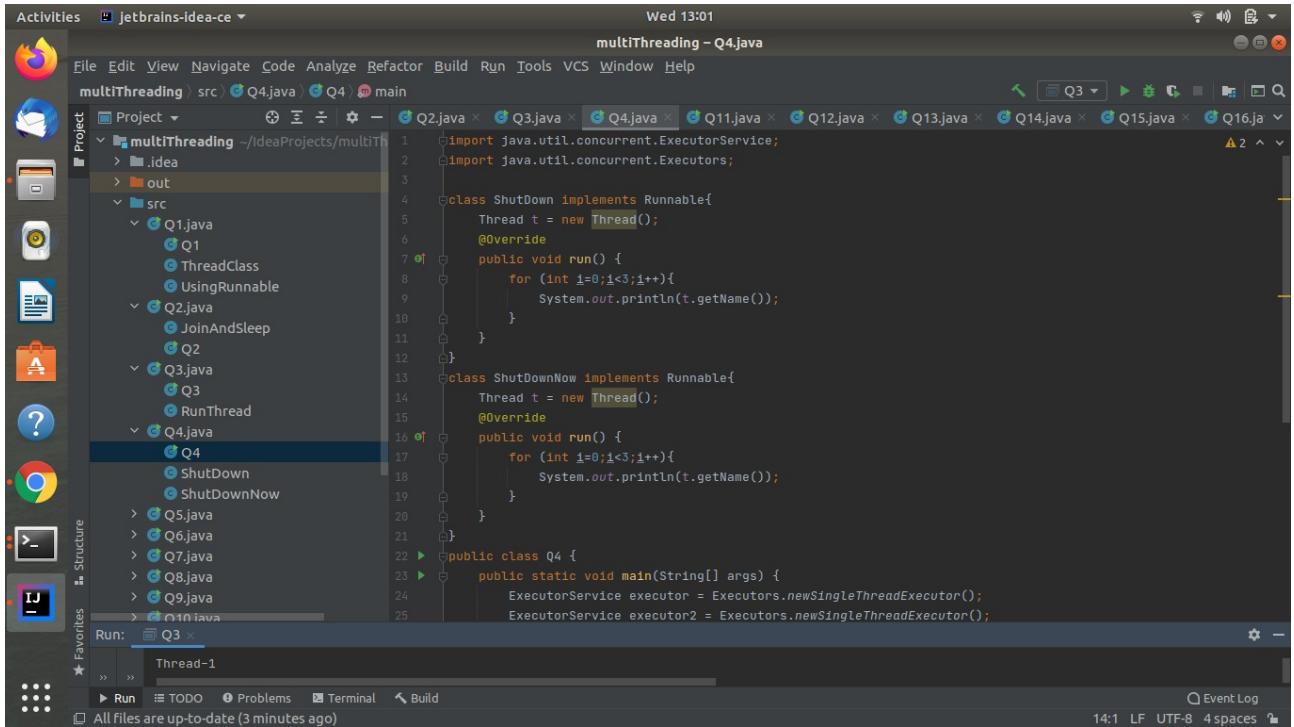
public class Q3 {
    public static void main(String[] args) {
        ExecutorService executor = Executors.newSingleThreadExecutor();
        RunThread r1 = new RunThread();
    }
}
```

The 'Run' tool window shows the output of running `Q3`:

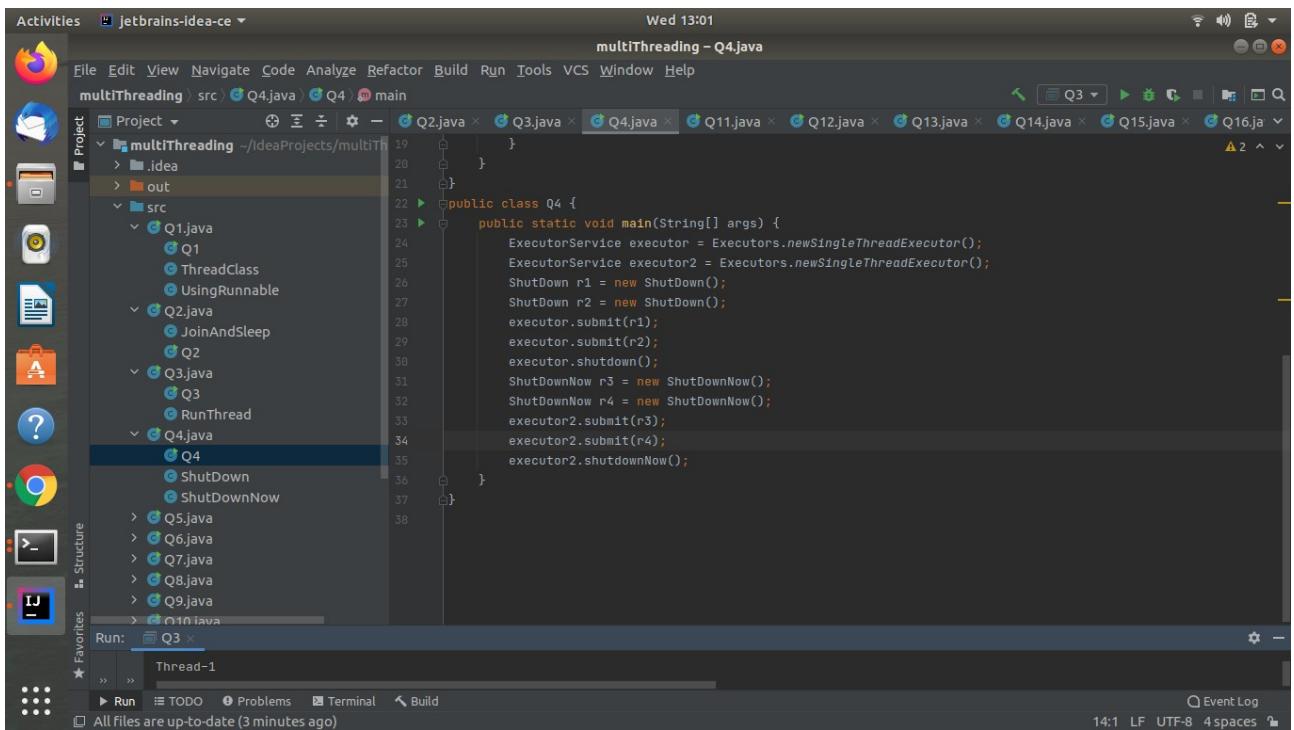
```
/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -javaagent:/home/ttn/Downloads/ideaIC-2020.3.2/idea-IC-203.7148.57/lib/idea_rt.jar=42279:/home/ttn/Downloads/idea-IC-2020.3.2/idea-IC-203.7148.57/lib/idea_rt.jar
Thread-0
Thread-0
Thread-0
Thread-0
Thread-0
Thread-1
Thread-1
Thread-1
Thread-1
Thread-1

Process finished with exit code 0
```

Q4 : Try shutdown() and shutdownNow() and observe the difference.



```
Activities JetBrains-Idea-CE Wed 13:01 multiThreading - Q4.java
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
multiThreading src Q4.java Q4 main
Project .idea out src
  Q1.java
  Q1
  ThreadClass
  UsingRunnable
  Q2.java
  JoinAndSleep
  Q2
  Q3.java
  Q3
  RunThread
  Q4.java
  Q4
  ShutDown
  ShutDownNow
  Q5.java
  Q6.java
  Q7.java
  Q8.java
  Q9.java
  Q10.java
Run: Q3
  Thread-1
  >> Run TODO Problems Terminal Build
  All files are up-to-date (3 minutes ago)
  1 import java.util.concurrent.ExecutorService;
  2 import java.util.concurrent.Executors;
  3
  4 class ShutDown implements Runnable{
  5     Thread t = new Thread();
  6     @Override
  7     public void run() {
  8         for (int i=0;i<3;i++){
  9             System.out.println(t.getName());
 10        }
 11    }
 12 }
 13 class ShutDownNow implements Runnable{
 14     Thread t = new Thread();
 15     @Override
 16     public void run() {
 17         for (int i=0;i<3;i++){
 18             System.out.println(t.getName());
 19        }
 20    }
 21 }
 22 public class Q4 {
 23     public static void main(String[] args) {
 24         ExecutorService executor = Executors.newSingleThreadExecutor();
 25         ExecutorService executor2 = Executors.newSingleThreadExecutor();
 26     }
 27 }
```



```
Activities JetBrains-Idea-CE Wed 13:01 multiThreading - Q4.java
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
multiThreading src Q4.java Q4 main
Project .idea out src
  Q1.java
  Q1
  ThreadClass
  UsingRunnable
  Q2.java
  JoinAndSleep
  Q2
  Q3.java
  Q3
  RunThread
  Q4.java
  Q4
  ShutDown
  ShutDownNow
  Q5.java
  Q6.java
  Q7.java
  Q8.java
  Q9.java
  Q10.java
Run: Q3
  Thread-1
  >> Run TODO Problems Terminal Build
  All files are up-to-date (3 minutes ago)
  1
  2
  3
  4
  5
  6
  7
  8
  9
  10
  11
  12
  13
  14
  15
  16
  17
  18
  19
  20
  21
  22 public class Q4 {
  23     public static void main(String[] args) {
  24         ExecutorService executor = Executors.newSingleThreadExecutor();
  25         ExecutorService executor2 = Executors.newSingleThreadExecutor();
  26         ShutDown r1 = new ShutDown();
  27         ShutDownNow r2 = new ShutDownNow();
  28         executor.submit(r1);
  29         executor.submit(r2);
  30         executor.shutdown();
  31         ShutDownNow r3 = new ShutDownNow();
  32         ShutDownNow r4 = new ShutDownNow();
  33         executor2.submit(r3);
  34         executor2.submit(r4);
  35         executor2.shutdownNow();
  36     }
  37 }
  38 }
```

Activities JetBrains-idea-ce

Wed 13:01

multithreading - Q4.java

The screenshot shows the IntelliJ IDEA interface with a Java project named "multithreading". The "src" directory contains several files: Q1.java, Q1.java, ThreadClass.java, UsingRunnable.java, and Q2.java. The Q2.java file is open in the editor, showing the following code:

```
public void run() {
    for (int i=0;i<3;i++){
        System.out.println(t.getName());
    }
}

class ShutDownNow implements Runnable{
    Thread t = new Thread();
    @Override
    public void run() {
```

In the "Run" tool window, the configuration "Q4" is selected, and the output shows:

```
/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -javaagent:/home/ttn/Downloads/ideaIC-2020.3.2/idea-IC-205.7148.57/lib/idea_rt.jar=42365:/home/ttn/Downloads/ideaIC-2020.3.2/idea-IC-205.7148.57/bin
Thread-2
Thread-2
Thread-0
Thread-0
Thread-0
Thread-1
Thread-1
Thread-1

Process finished with exit code 0
```

The status bar at the bottom indicates "All files are up-to-date (moments ago)" and the time "34:30".

## Q5 :Use isShutdown() and isTerminated() with ExecutorService.

The screenshot shows the IntelliJ IDEA interface with the project 'multiThreading' open. The code editor displays Q5.java, which contains the following code:

```
class IsTerminated implements Runnable {
    Thread t = new Thread();
    @Override
    public void run() {
        for (int i=0;i<3;i++){
            try{
                t.sleep( 300 );
            }catch ( Exception e){
                System.out.println(e);
            }
            System.out.println(t.getName());
        }
    }
}

public class Q5 {
    public static void main(String[] args) {
        ExecutorService executor = Executors.newSingleThreadExecutor();
        IsTerminated ist = new IsTerminated();
        executor.submit(ist);
        System.out.println(" Is Executor is Shutdown :"+ executor.isShutdown());
        executor.shutdown();
        System.out.println(" Is Executor is Terminated :"+ executor.isTerminated());
    }
}
```

The code uses an ExecutorService to submit a task that sleeps for 300ms and prints its thread name. It then checks the shutdown and terminated status of the executor.

The screenshot shows the IntelliJ IDEA interface after running Q5.java. The terminal output shows the following results:

```
Is Executor is Shutdown :false
Is Executor is Terminated :false
Thread-0
Thread-0
Thread-0
```

The output indicates that the executor is neither shutdown nor terminated, and it successfully printed the thread names three times.

Q6: Return a Future from ExecutorService by using callable and use get(), isDone(), isCancelled() with the Future object to know the status of task submitted.

The screenshot shows the IntelliJ IDEA interface with the project 'multithreading' open. The code editor displays Q6.java, which contains the following code:

```
import java.util.concurrent.ExecutionException;
import java.util.concurrent.ExecutorService;
import java.util.concurrent.Executors;
import java.util.concurrent.Future;

class FutureObj implements Runnable{
    Thread t = new Thread();
    @Override
    public void run() {
        for (int i=0;i<3;i++){
            try{
                t.sleep( 300 );
            }catch ( Exception e){
                System.out.println(e);
            }
            System.out.println(t.getName());
        }
    }
}

public class Q6 {
    public static void main(String[] args) throws ExecutionException, InterruptedException {
        ExecutorService executor = Executors.newSingleThreadExecutor();
        FutureObj fobj = new FutureObj();
        Future<String> obj = executor.submit(fobj, "Executing thread");
    }
}
```

The code implements a `FutureObj` class that extends `Runnable`. It contains a `run` method that sleeps for 300ms three times and prints the name of the thread each time. The `main` method creates a `FutureObj` instance and submits it to a single-threaded `ExecutorService`, passing a descriptive string for the task.

The screenshot shows the IntelliJ IDEA interface with the project 'multithreading' open. The code editor displays Q6.java, which now includes additional code at the top and bottom:

```
System.out.println("get is: " + obj.get());
System.out.println("isDone: " + obj.isDone());
System.out.println("isCancelled: " + obj.isCancelled());
```

This additional code demonstrates how to use the `get()`, `isDone()`, and `isCancelled()` methods on the `Future` object returned by the `submit` method. The `get` method retrieves the result of the task, while `isDone` and `isCancelled` check if the task has completed or been cancelled respectively.

Activities Jetbrains-idea-ce ▾ Wed 13:05

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

multiThreading - Q6.java

Project src Q2.java × Q3.java × Q4.java × Q5.java × Q6.java × Q13.java × Q14.java × Q15.java × Q16.java ×

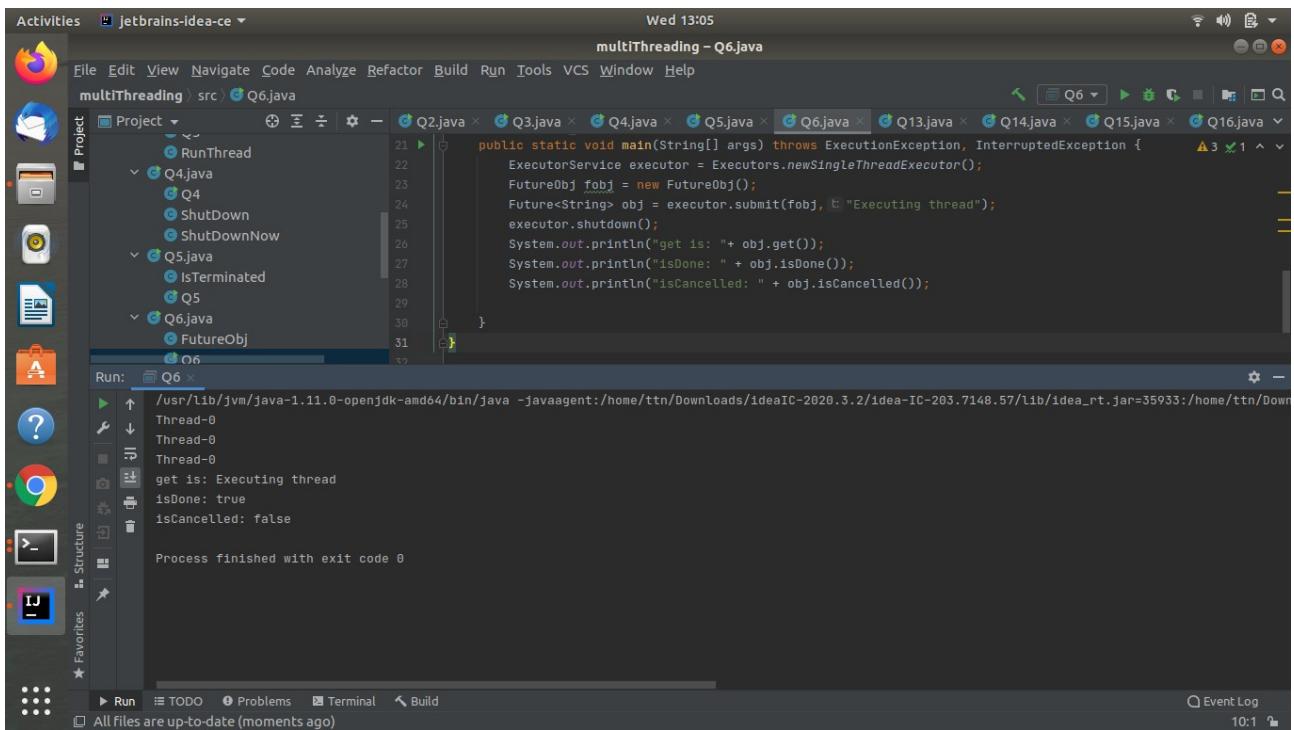
Run: Q6

```
public static void main(String[] args) throws ExecutionException, InterruptedException {
    ExecutorService executor = Executors.newSingleThreadExecutor();
    FutureObj fobj = new FutureObj();
    Future<String> obj = executor.submit(fobj, "Executing thread");
    executor.shutdown();
    System.out.println("get is: " + obj.get());
    System.out.println("isDone: " + obj.isDone());
    System.out.println("isCancelled: " + obj.isCancelled());
}
```

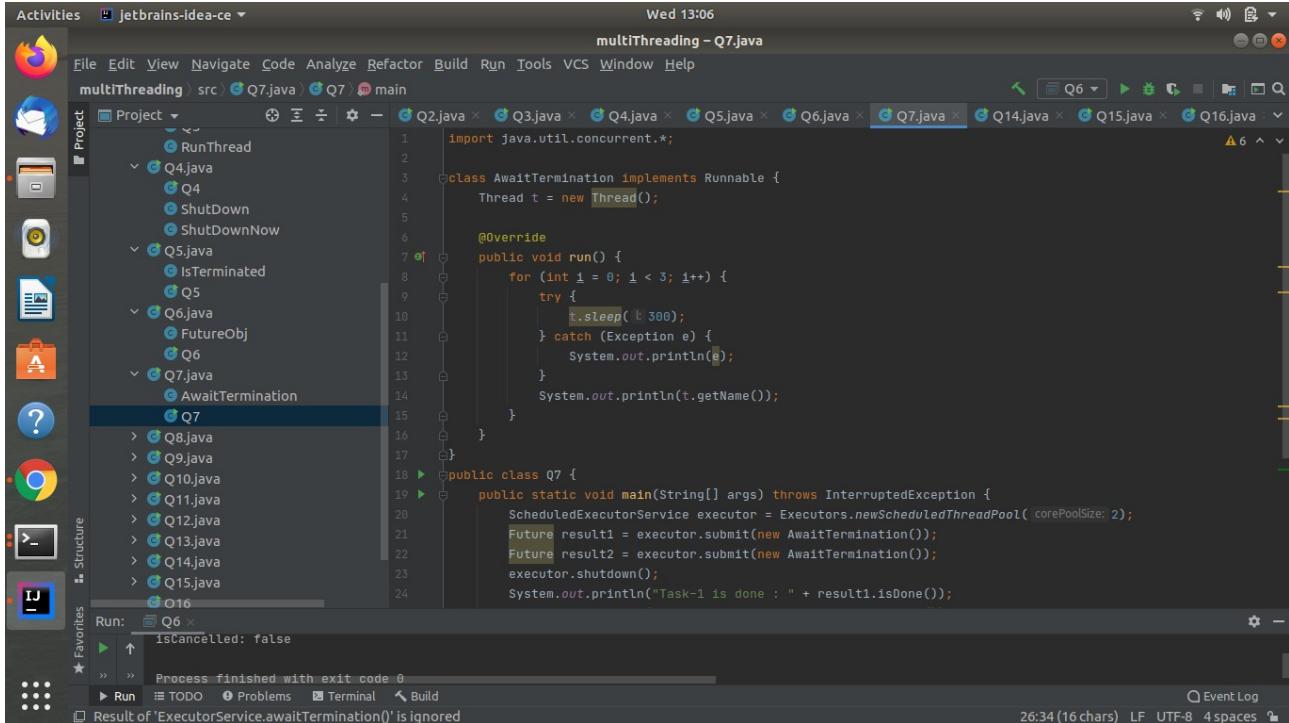
Process finished with exit code 0

All files are up-to-date (moments ago)

Event Log 10:1



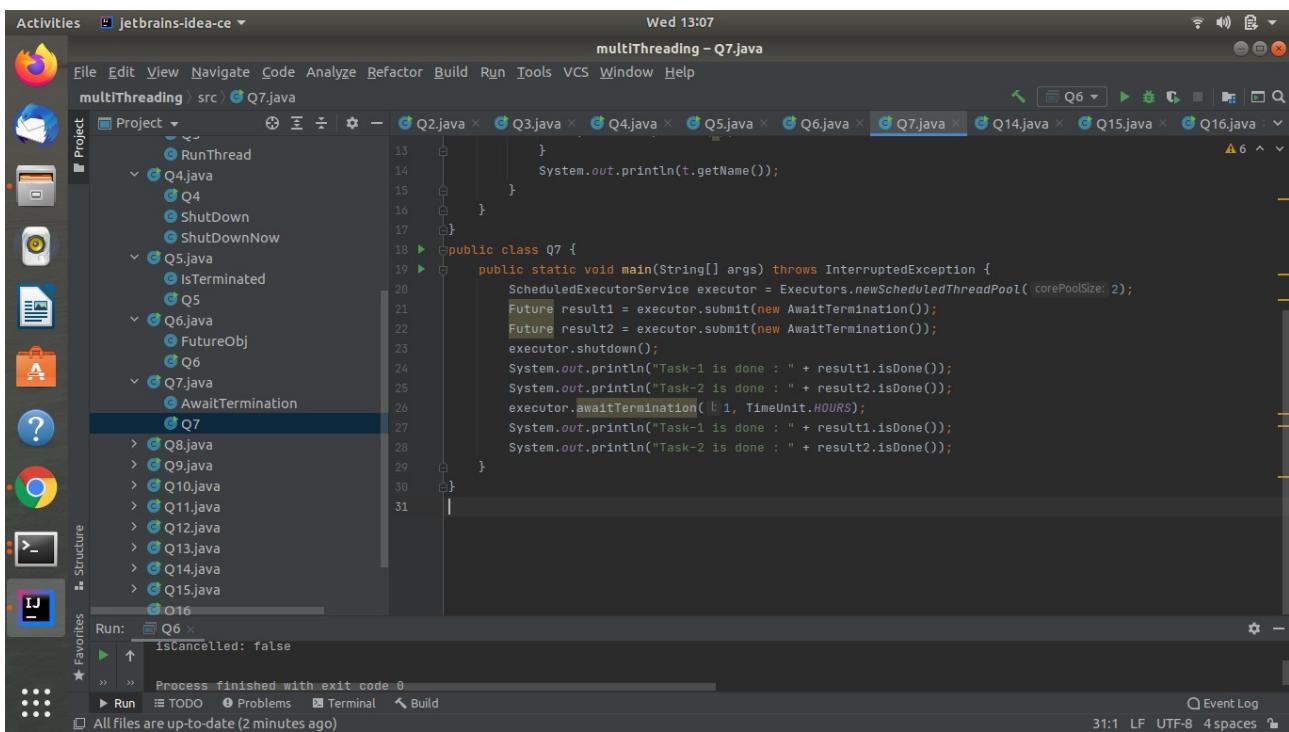
## Q7: Submit List of tasks to ExecutorService and wait for the completion of all the tasks.



```
Activities JetBrains-idea-ce Wed 13:06
multithreading - Q7.java
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
multithreading src Q7.java Q7 main
Project Q2.java x Q3.java x Q4.java x Q5.java x Q6.java x Q7.java x Q14.java x Q15.java x Q16.java x
Run Thread
  Q4.java
  Q4
  ShutDown
  ShutDownNow
  Q5.java
  IsTerminated
  Q5
  Q6.java
  FutureObj
  Q6
  Q7.java
  AwaitTermination
  Q7
  Q8.java
  Q9.java
  Q10.java
  Q11.java
  Q12.java
  Q13.java
  Q14.java
  Q15.java
  Q16
  Q6
  isCancelled: false
  Process finished with exit code 0
Run TODO Problems Terminal Build
Result of 'ExecutorService.awaitTermination()' is ignored
26:34 (16 chars) LF UTF-8 4 spaces

```

```
import java.util.concurrent.*;
class AwaitTermination implements Runnable {
    Thread t = new Thread();
    @Override
    public void run() {
        for (int i = 0; i < 3; i++) {
            try {
                t.sleep(300);
            } catch (Exception e) {
                System.out.println(e);
            }
        }
        System.out.println(t.getName());
    }
}
public class Q7 {
    public static void main(String[] args) throws InterruptedException {
        ScheduledExecutorService executor = Executors.newScheduledThreadPool( corePoolSize: 2 );
        Future result1 = executor.submit(new AwaitTermination());
        Future result2 = executor.submit(new AwaitTermination());
        executor.shutdown();
        System.out.println("Task-1 is done : " + result1.isDone());
    }
}
```



```
Activities JetBrains-idea-ce Wed 13:07
multithreading - Q7.java
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
multithreading src Q7.java
Project Q2.java x Q3.java x Q4.java x Q5.java x Q6.java x Q7.java x Q14.java x Q15.java x Q16.java x
Run Thread
  Q4.java
  Q4
  ShutDown
  ShutDownNow
  Q5.java
  IsTerminated
  Q5
  Q6.java
  FutureObj
  Q6
  Q7.java
  AwaitTermination
  Q7
  Q8.java
  Q9.java
  Q10.java
  Q11.java
  Q12.java
  Q13.java
  Q14.java
  Q15.java
  Q16
  Q6
  isCancelled: false
  Process finished with exit code 0
Run TODO Problems Terminal Build
All files are up-to-date (2 minutes ago)
31:1 LF UTF-8 4 spaces

```

```
System.out.println(t.getName());
}
public class Q7 {
    public static void main(String[] args) throws InterruptedException {
        ScheduledExecutorService executor = Executors.newScheduledThreadPool( corePoolSize: 2 );
        Future result1 = executor.submit(new AwaitTermination());
        Future result2 = executor.submit(new AwaitTermination());
        executor.shutdown();
        System.out.println("Task-1 is done : " + result1.isDone());
        System.out.println("Task-2 is done : " + result2.isDone());
        executor.awaitTermination( 1, TimeUnit.HOURS );
        System.out.println("Task-1 is done : " + result1.isDone());
        System.out.println("Task-2 is done : " + result2.isDone());
    }
}
```

Activities JetBrains-idea-ce Wed 13:07

multithreading - Q7.java

The screenshot shows the IntelliJ IDEA interface with the following details:

- Project View:** Shows a tree structure of files under "multiThreading/src". Files include Q2.java, Q3.java, Q4.java, Q5.java, Q6.java, and Q7.java.
- Code Editor:** The current file is Q7.java. The code contains several print statements and a loop that outputs thread names and task completion status.
- Run Tab:** The "Run" tab is selected, showing the run configuration "Q7". The output window displays the following text:

```
/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -javaagent:/home/ttn/Downloads/ideaIC-2020.3.2/idea-IC-203.7148.57/lib/idea_rt.jar=39545:/home/ttn/Downloads/ideaIC-2020.3.2/idea-IC-203.7148.57/bin/run.jar::
```

```
Task-1 is done : false
Task-2 is done : false
Thread-0
Thread-1
Thread-0
Thread-1
Thread-0
Thread-1
Task-1 is done : true
Task-2 is done : true

Process finished with exit code 0
```
- Bottom Status Bar:** Shows "All files are up-to-date (moments ago)" and "14:1 LF UTF-8 4 spaces".

## Q9: Schedule task using schedule(), scheduleAtFixedRate() and scheduleAtFixedDelay()

The screenshot shows the IntelliJ IDEA interface with the project 'multiThreading' open. The file 'Q8.java' is the active editor. The code implements a Runnable named Scheduling, which runs a loop three times, sleeping for 300ms each iteration. It also contains a main method that creates a ScheduledExecutorService and schedules the Scheduling task.

```
Activities JetBrains-idea-ce Wed 13:08
multiThreading - Q8.java
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
multiThreading src Q8.java Q8 main
Project Q2.java x Q3.java x Q4.java x Q5.java x Q6.java x Q7.java x Q8.java x Q15.java x Q16.java x
Run: Q7 x Thread-1 Event Log
All files are up-to-date (a minute ago) 35:54 (45 chars) LF UTF-8 4 spaces
1 import java.time.Duration;
2 import java.time.LocalDateTime;
3 import java.util.concurrent.Executors;
4 import java.util.concurrent.Future;
5 import java.util.concurrent.ScheduledExecutorService;
6 import java.util.concurrent.TimeUnit;
7
8 class Scheduling implements Runnable{
9     Thread t = new Thread();
10
11     @Override
12     public void run() {
13         for (int i = 0; i < 3; i++) {
14             try {
15                 t.sleep(300);
16             } catch (Exception e) {
17                 System.out.println(e);
18             }
19         }
20     }
21 }
22
23 public class Q8 {
24     public static void main(String[] args) throws InterruptedException {
25         ScheduledExecutorService executor = Executors.newScheduledThreadPool( corePoolSize: 3 );
26         LocalDateTime now = LocalDateTime.now();
27         Future result1 = executor.schedule(new Scheduling(), Duration.between(now, now.plusSeconds(2)).toMillis(),
28             TimeUnit.MILLISECONDS);
29         Future result2 = executor.scheduleAtFixedRate(new Scheduling(), 1, 1, TimeUnit.MILLISECONDS );
30         Future result3 = executor.scheduleAtFixedRate(new Scheduling(), 1, 3, TimeUnit.MILLISECONDS );
31         System.out.println(result1.isDone());
32         System.out.println(result2.isDone());
33         System.out.println(result3.isDone());
34         executor.shutdown();
35         executor.awaitTermination(1, TimeUnit.HOURS);
36         System.out.println(result1.isDone());
37         System.out.println(result2.isDone());
38         System.out.println(result3.isDone());
39     }
40 }
```

The screenshot shows the IntelliJ IDEA interface with the project 'multiThreading' open. The file 'Q8.java' is the active editor. The code implements a Runnable named Scheduling, which runs a loop three times, sleeping for 300ms each iteration. It also contains a main method that creates a ScheduledExecutorService and schedules the Scheduling task using scheduleAtFixedRate.

```
Activities JetBrains-idea-ce Wed 13:08
multiThreading - Q8.java
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
multiThreading src Q8.java
Project Q2.java x Q3.java x Q4.java x Q5.java x Q6.java x Q7.java x Q8.java x Q15.java x Q16.java x
Run: Q7 x Thread-1 Event Log
All files are up-to-date (a minute ago) 41:1 LF UTF-8 4 spaces
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23 public class Q8 {
24     public static void main(String[] args) throws InterruptedException {
25         ScheduledExecutorService executor = Executors.newScheduledThreadPool( corePoolSize: 3 );
26         LocalDateTime now = LocalDateTime.now();
27         Future result1 = executor.schedule(new Scheduling(), Duration.between(now, now.plusSeconds(2)).toMillis(),
28             TimeUnit.MILLISECONDS);
29         Future result2 = executor.scheduleAtFixedRate(new Scheduling(), 1, 1, TimeUnit.MILLISECONDS );
30         Future result3 = executor.scheduleAtFixedRate(new Scheduling(), 1, 3, TimeUnit.MILLISECONDS );
31         System.out.println(result1.isDone());
32         System.out.println(result2.isDone());
33         System.out.println(result3.isDone());
34         executor.shutdown();
35         executor.awaitTermination(1, TimeUnit.HOURS);
36         System.out.println(result1.isDone());
37         System.out.println(result2.isDone());
38         System.out.println(result3.isDone());
39     }
40 }
```

Activities Jetbrains-idea-ce ▾ Wed 13:09

multithreading - Q8.java

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

multiThreading > src > Q8.java

Project Run Thread Q2.java × Q3.java × Q4.java × Q5.java × Q6.java × Q7.java × Q8.java × Q15.java × Q16.java ×

Run: Q8

```
/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -javaagent:/home/ttn/Downloads/ideaIC-2020.3.2/idea-IC-203.7148.57/lib/idea_rt.jar=34449:/home/ttn/Downloads/ideaIC-2020.3.2/idea-IC-203.7148.57/lib/idea_rt.jar
```

false  
false  
false  
Thread-2  
Thread-1  
Thread-2  
Thread-1  
Thread-2  
Thread-1  
Thread-0  
Thread-0  
Thread-0  
true  
true  
true  
Process finished with exit code 0

Run TODO Problems Terminal Build Event Log

All files are up-to-date (moments ago) 19:1 LF UTF-8 4 spaces

The screenshot shows the IntelliJ IDEA Community Edition interface. A Java application named 'Q8.java' is running. The terminal output displays several lines of text, likely from multiple threads, showing alternating 'true' and 'false' values. This pattern suggests a race condition or non-deterministic execution of the code. The IntelliJ interface includes a project tree, toolbars, and status bars at the bottom.

## Q9: Increase concurrency with Thread pools using newCachedThreadPool() and newFixedThreadPool().

```
Activities Jetbrains-idea-ce Wed 13:10
multiThreading - Q9.java
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
multiThreading src Q9.java Q9
Project Q2.java x Q3.java x Q4.java x Q5.java x Q6.java x Q7.java x Q8.java x Q9.java x Q16.java x
  RunThread
  Q4.java
  ShutDown
  ShutDownNow
  Q5.java
    IsTerminated
    Q5
  Q6.java
    FutureObj
    Q6
  Q7.java
    AwaitTermination
    Q7
  Q8.java
    Q8
    Scheduling
  Q9.java
    Concurrency
      Q9
  Q10.java
  Q11.java
  Q12.java
  Q13.java
  Q14.java
Structure Run: Q8 x
Favorites Event Log
Run TODO Problems Terminal Build
All files are up-to-date (a minute ago) 19:14 LF UTF-8 4 spaces
public class Q9 {
    public static void main(String[] args){
        ExecutorService executor = Executors.newCachedThreadPool();
        ExecutorService executor2 = Executors.newFixedThreadPool( nThreads: 1 );
        ThreadPoolExecutor pool = (ThreadPoolExecutor) executor;
        ThreadPoolExecutor pool2 = (ThreadPoolExecutor) executor2;
        pool.setMaximumPoolSize();
        executor.submit(new Concurrency());
        executor.submit(new Concurrency());
    }
}
```

```
Activities Jetbrains-idea-ce Wed 13:10
multiThreading - Q9.java
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
multiThreading src Q9.java Q9
Project Q2.java x Q3.java x Q4.java x Q5.java x Q6.java x Q7.java x Q8.java x Q9.java x Q16.java x
  RunThread
  Q4.java
  ShutDown
  ShutDownNow
  Q5.java
    IsTerminated
    Q5
  Q6.java
    FutureObj
    Q6
  Q7.java
    AwaitTermination
    Q7
  Q8.java
    Q8
    Scheduling
  Q9.java
    Concurrency
      Q9
  Q10.java
  Q11.java
  Q12.java
  Q13.java
  Q14.java
Structure Run: Q8 x
Favorites Event Log
Run TODO Problems Terminal Build
All files are up-to-date (a minute ago) 19:14 LF UTF-8 4 spaces
public class Q9 {
    public static void main(String[] args){
        ExecutorService executor = Executors.newCachedThreadPool();
        ExecutorService executor2 = Executors.newFixedThreadPool( nThreads: 1 );
        ThreadPoolExecutor pool = (ThreadPoolExecutor) executor;
        ThreadPoolExecutor pool2 = (ThreadPoolExecutor) executor2;
        pool.setMaximumPoolSize();
        executor.submit(new Concurrency());
        executor.submit(new Concurrency());

        System.out.println("executor size:" + pool.getMaximumPoolSize());
        System.out.println("executor2 size: " + pool2.getMaximumPoolSize());
        executor2.submit(new Concurrency());
        executor2.submit(new Concurrency());
        executor2.submit(new Concurrency());
        executor.shutdown();
        executor2.shutdown();
    }
}
```

Activities Jetbrains-idea-ce ▾ Wed 13:10

multithreading - Q9.java

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

multiThreading > src > Q9.java

Project

Run: Q9

/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -javaagent:/home/ttn/Downloads/ideaIC-2020.3.2/idea-IC-203.7148.57/lib/idea\_rt.jar=40813:/home/ttn/Downloads/ideaIC-2020.3.2/idea-IC-203.7148.57/bin

executor size:2147483647

executor2 size: 1

Thread-0

Thread-1

Thread-2

Thread-0

Thread-1

Thread-2

Thread-3

Thread-3

Thread-4

Thread-4

Thread-4

Process finished with exit code 0

Run TODO Problems Terminal Build Event Log

All files are up-to-date (moments ago)

21:1 LF UTF-8 4 spaces

**Q10: Use Synchronize method to enable synchronization between multiple threads trying to access method at same time.**

Activities JetBrains-idea-ce Wed 13:11  
multThreading - Q10.java

```
1 class Synchronize {
2     Thread t = new Thread();
3     synchronized public void show(int n){
4         int count = 0;
5         for (int i = 0; i < n; i++) {
6             try {
7                 t.sleep(300);
8             } catch (Exception e) {
9                 System.out.println(e);
10            }
11            count++;
12        }
13        System.out.println("count is: " + count);
14    }
15 }
16 class CreateThread implements Runnable {
17     Synchronize s1;
18     CreateThread(Synchronize s1) { this.s1 = s1; }
19     @Override
20     public void run() {
21         s1.show(2);
22     }
23 }
24 class CreateThread2 implements Runnable {
25     Synchronize s1;
26     CreateThread2(Synchronize s1) { this.s1 = s1; }
27     @Override
28     public void run() {
29         s1.show(5);
30     }
31 }
32 public class Q10 {
33     public static void main(String[] args) {
34         Synchronize sy = new Synchronize();
35         CreateThread ct = new CreateThread(sy);
36         CreateThread2 ct2 = new CreateThread2(sy);
37         Thread t1 = new Thread(ct);
38         Thread t2 = new Thread(ct2);
39         t1.start();
40         t2.start();
41     }
42 }
```

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help  
multThreading src Q10.java Q10

Project Run Thread Q4.java Q5.java Q6.java Q7.java Q8.java Q9.java Q10.java Q2.java Q3.java Q4.java Q5.java Q6.java Q7.java Q8.java Q9.java Q10.java Q2.java Q3.java Q4.java Q5.java Q6.java Q7.java Q8.java Q9.java Q10.java

Run: Q9 Event Log  
Run TODO Problems Terminal Build  
All files are up-to-date (a minute ago) 36:14

Activities JetBrains-idea-ce Wed 13:11  
multThreading - Q10.java

```
16 class CreateThread implements Runnable {
17     Synchronize s1;
18     CreateThread(Synchronize s1) { this.s1 = s1; }
19     @Override
20     public void run() {
21         s1.show(2);
22     }
23 }
24 class CreateThread2 implements Runnable {
25     Synchronize s1;
26     CreateThread2(Synchronize s1) { this.s1 = s1; }
27     @Override
28     public void run() { s1.show(5); }
29 }
30 public class Q10 {
31     public static void main(String[] args) {
32         Synchronize sy = new Synchronize();
33         CreateThread ct = new CreateThread(sy);
34         CreateThread2 ct2 = new CreateThread2(sy);
35         Thread t1 = new Thread(ct);
36         Thread t2 = new Thread(ct2);
37         t1.start();
38         t2.start();
39     }
40 }
```

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help  
multThreading src Q10.java Q10

Project Run Thread Q4.java Q5.java Q6.java Q7.java Q8.java Q9.java Q10.java Q2.java Q3.java Q4.java Q5.java Q6.java Q7.java Q8.java Q9.java Q10.java Q2.java Q3.java Q4.java Q5.java Q6.java Q7.java Q8.java Q9.java Q10.java

Run: Q9 Event Log  
Run TODO Problems Terminal Build  
All files are up-to-date (2 minutes ago) 36:14

Activities Jetbrains-idea-ce

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

multiThreading > src > Q10.java > Q10

Project Run Thread Q2.java Q3.java Q4.java Q5.java Q6.java Q7.java Q8.java Q9.java Q10.java 3

Q4.java  
Q4  
ShutDown  
ShutDownNow  
Q5.java  
IsTerminated  
Q5

31     @Override  
32        public void run() { s1.show( m: 5 ); }  
33  
34  
35     }  
36     public class Q10 {  
37        public static void main(String[] args) {  
38            Synchronize sy = new Synchronize();  
39            CreateThread ct = new CreateThread(sy);  
40            CreateThread2 ct2 = new CreateThread2(sy);  
41            Thread t1 = new Thread(ct);

Run: Q10

/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -javaagent:/home/ttn/Downloads/ideaIC-2020.3.2/ideaIC-203.7148.57/lib/idea\_rt.jar=46095:/home/ttn/Downloads/ideaIC-2020.3.2/ideaIC-203.7148.57/lib/idea\_rt.jar

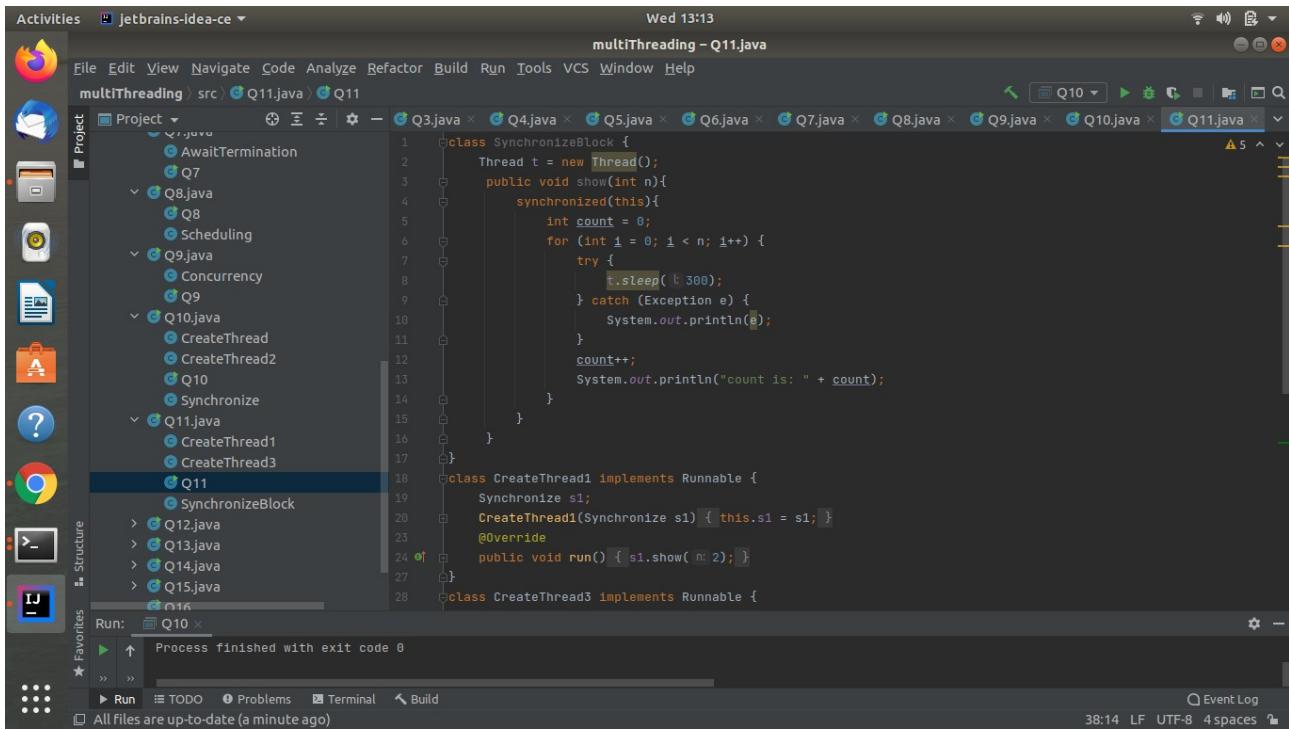
count is: 1  
count is: 2  
count is: 1  
count is: 2  
count is: 3  
count is: 4  
count is: 5

Process finished with exit code 0

Run TODO Problems Terminal Build Event Log

All files are up-to-date (moments ago)

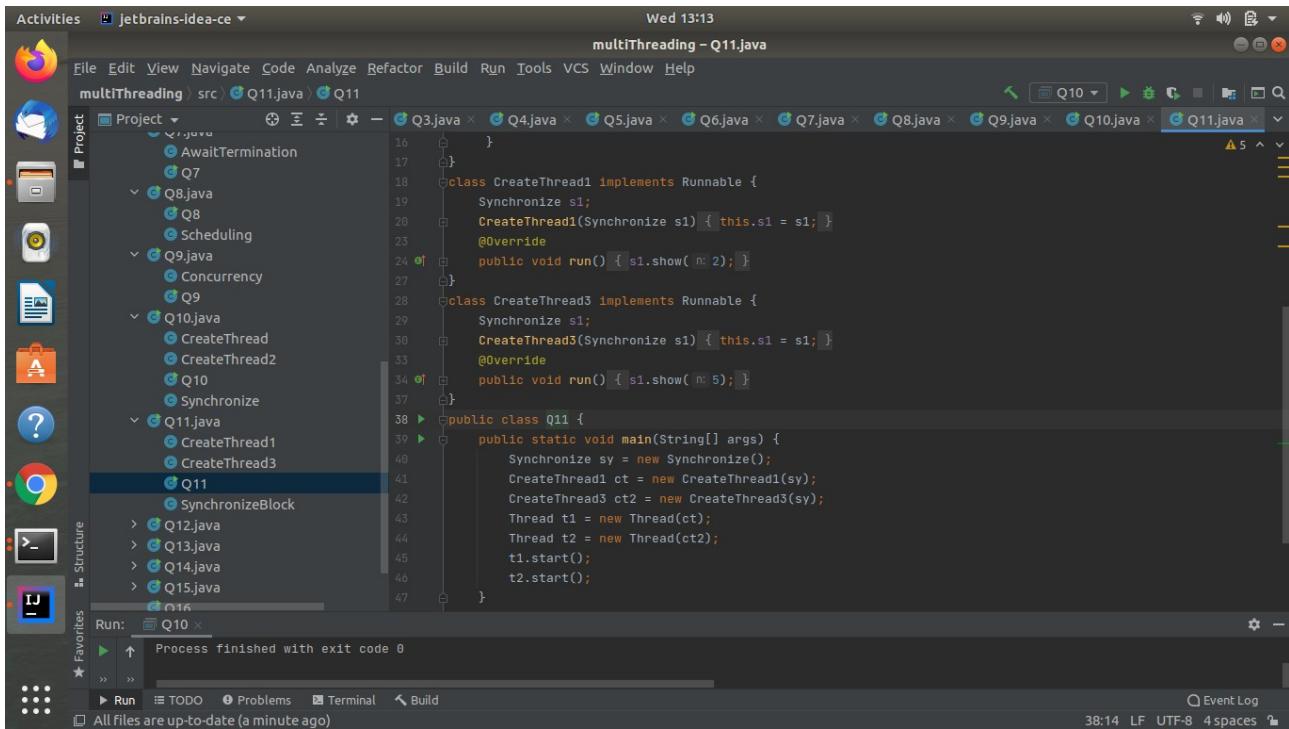
**Q11: Use Synchronize block to enable synchronization between multiple threads trying to access method at same time.**



```
Activities Jetbrains-idea-ce Wed 13:13
multThreading - Q11.java
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
multThreading src Q11.java
Project Q1.java
  - Q3.java
  - Q4.java
  - Q5.java
  - Q6.java
  - Q7.java
  - Q8.java
    - Q8
    - Scheduling
  - Q9.java
    - Concurrency
    - Q9
  - Q10.java
    - CreateThread
    - CreateThread2
    - Q10
    - Synchronize
  - Q11.java
    - CreateThread1
    - CreateThread3
    - Q11
    - SynchronizeBlock
  - Q12.java
  - Q13.java
  - Q14.java
  - Q15.java
  - Q16
Run: Q10
Process finished with exit code 0
Event Log
38:14 LF UTF-8 4 spaces
All files are up-to-date (a minute ago)
```

The code in `Q11.java` is as follows:

```
1 class SynchronizeBlock {
2     Thread t = new Thread();
3     public void show(int n){
4         synchronized(this){
5             int count = 0;
6             for (int i = 0; i < n; i++) {
7                 try {
8                     t.sleep(100);
9                 } catch (Exception e) {
10                     System.out.println(e);
11                 }
12                 count++;
13             }
14             System.out.println("count is: " + count);
15         }
16     }
17
18 class CreateThread1 implements Runnable {
19     Synchronize s1;
20     CreateThread1(Synchronize s1) { this.s1 = s1; }
21     @Override
22     public void run() { s1.show( n, 2); }
23
24 class CreateThread3 implements Runnable {
25     Synchronize s1;
26     CreateThread3(Synchronize s1) { this.s1 = s1; }
27     @Override
28     public void run() { s1.show( n, 5); }
29
30 public class Q11 {
31     public static void main(String[] args) {
32         Synchronize sy = new Synchronize();
33         CreateThread1 ct = new CreateThread1(sy);
34         CreateThread3 ct2 = new CreateThread3(sy);
35         Thread t1 = new Thread(ct);
36         Thread t2 = new Thread(ct2);
37         t1.start();
38         t2.start();
39     }
40 }
```



```
Activities Jetbrains-idea-ce Wed 13:13
multThreading - Q11.java
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
multThreading src Q11.java
Project Q1.java
  - Q3.java
  - Q4.java
  - Q5.java
  - Q6.java
  - Q7.java
  - Q8.java
    - Q8
    - Scheduling
  - Q9.java
    - Concurrency
    - Q9
  - Q10.java
    - CreateThread
    - CreateThread2
    - Q10
    - Synchronize
  - Q11.java
    - CreateThread1
    - CreateThread3
    - Q11
    - SynchronizeBlock
  - Q12.java
  - Q13.java
  - Q14.java
  - Q15.java
  - Q16
Run: Q10
Process finished with exit code 0
Event Log
38:14 LF UTF-8 4 spaces
All files are up-to-date (a minute ago)
```

The code in `Q11.java` is as follows:

```
17
18 class CreateThread1 implements Runnable {
19     Synchronize s1;
20     CreateThread1(Synchronize s1) { this.s1 = s1; }
21     @Override
22     public void run() { s1.show( n, 2); }
23
24 class CreateThread3 implements Runnable {
25     Synchronize s1;
26     CreateThread3(Synchronize s1) { this.s1 = s1; }
27     @Override
28     public void run() { s1.show( n, 5); }
29
30 public class Q11 {
31     public static void main(String[] args) {
32         Synchronize sy = new Synchronize();
33         CreateThread1 ct = new CreateThread1(sy);
34         CreateThread3 ct2 = new CreateThread3(sy);
35         Thread t1 = new Thread(ct);
36         Thread t2 = new Thread(ct2);
37         t1.start();
38         t2.start();
39     }
40 }
```

Activities Jetbrains-idea-ce ▾ Wed 13:14

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

multiThreading - Q11.java

multiThreading src Q11.java

Project

Q11.java

AwaitTermination

Q7

Q8.java

Q8

Scheduling

Run: Q11 x

/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -javaagent:/home/ttn/Downloads/ideaIC-2020.3.2/ideaIC-203.7148.57/lib/idea\_rt.jar=46191:/home/ttn/Downloads/ideaIC-2020.3.2/ideaIC-203.7148.57/lib/idea\_rt.jar

count is: 1  
count is: 2  
count is: 1  
count is: 2  
count is: 3  
count is: 4  
count is: 5

Process finished with exit code 0

Run TODO Problems Terminal Build

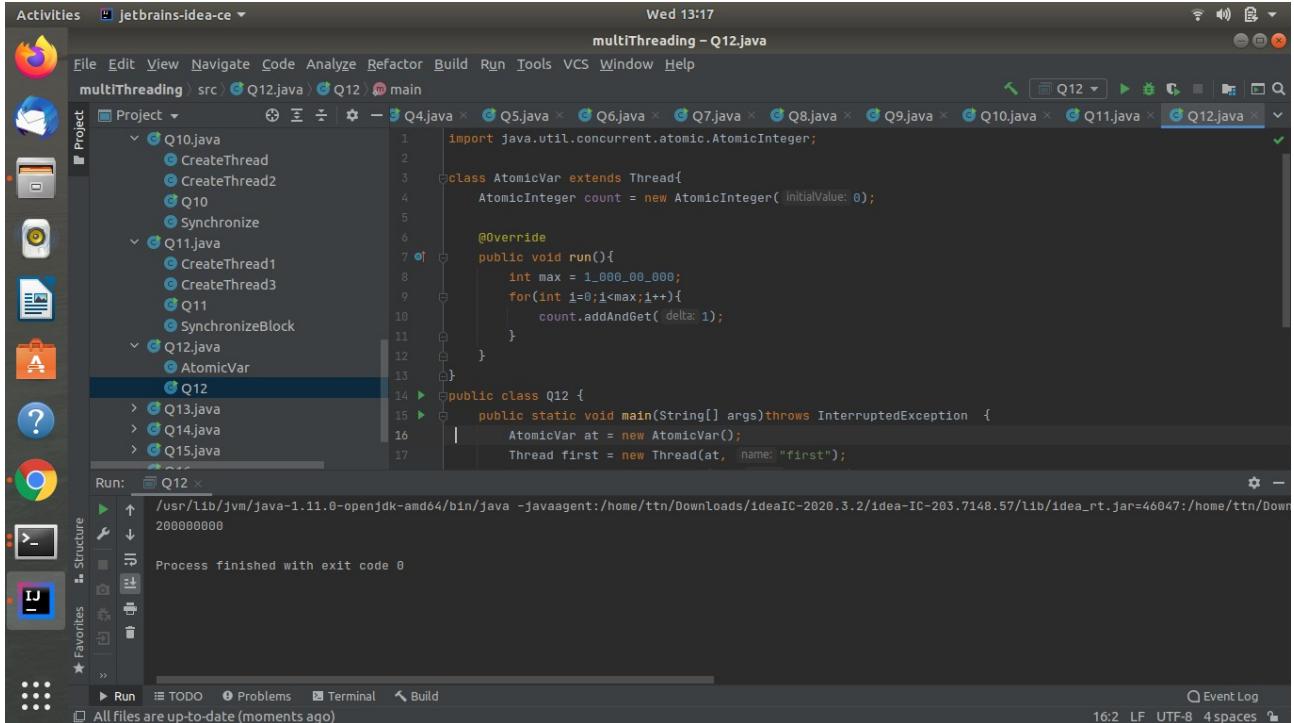
All files are up-to-date (moments ago)

Event Log

11:1 LF UTF-8 4 spaces

The screenshot shows the IntelliJ IDEA interface with a Java project named 'multiThreading'. The 'src' directory contains files Q11.java, Q3.java, Q4.java, Q5.java, Q6.java, Q7.java, Q8.java, and Q9.java. The 'Q11.java' file is open, displaying a class Q11 with a main method that creates a Synchronize object and calls its show method 5 times. The 'Run' tool window shows the output of the run command, which prints 'count is:' followed by values 1 through 5, indicating the execution of the threads. The status bar at the bottom right shows the current time as 11:1, file encoding as LF, and a 4-space indent setting.

## Q12: Use Atomic Classes instead of Synchronize method and blocks.



The screenshot shows the IntelliJ IDEA interface with the project 'multithreading' open. The code editor displays `Q12.java` which contains the following code:

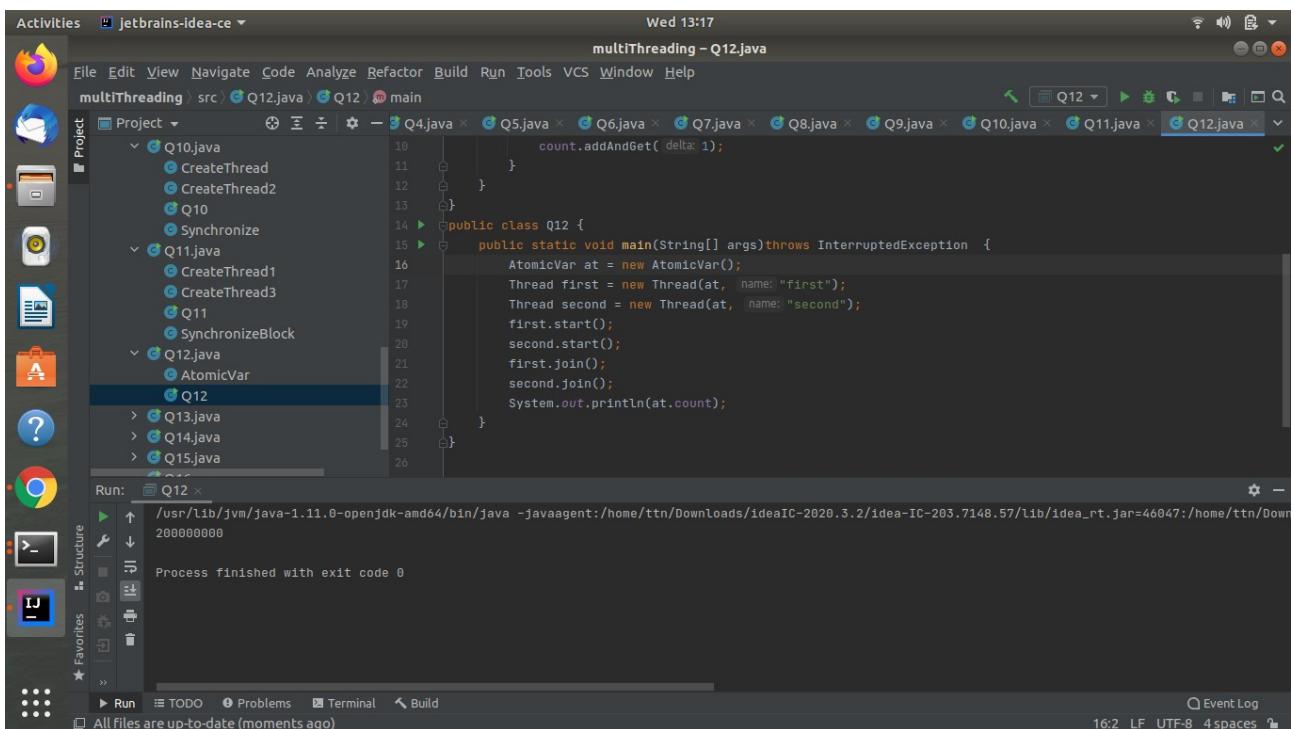
```
import java.util.concurrent.atomic.AtomicInteger;

class AtomicVar extends Thread{
    AtomicInteger count = new AtomicInteger( initialValue: 0);

    @Override
    public void run(){
        int max = 1_000_000;
        for(int i=0;i<max;i++){
            count.addAndGet( delta: 1);
        }
    }
}

public class Q12 {
    public static void main(String[] args) throws InterruptedException {
        AtomicVar at = new AtomicVar();
        Thread first = new Thread(at, name: "first");
    }
}
```

The code uses `AtomicInteger` from `java.util.concurrent.atomic` instead of the `synchronized` keyword or `LOCK-BEGIN`/`LOCK-END` blocks.



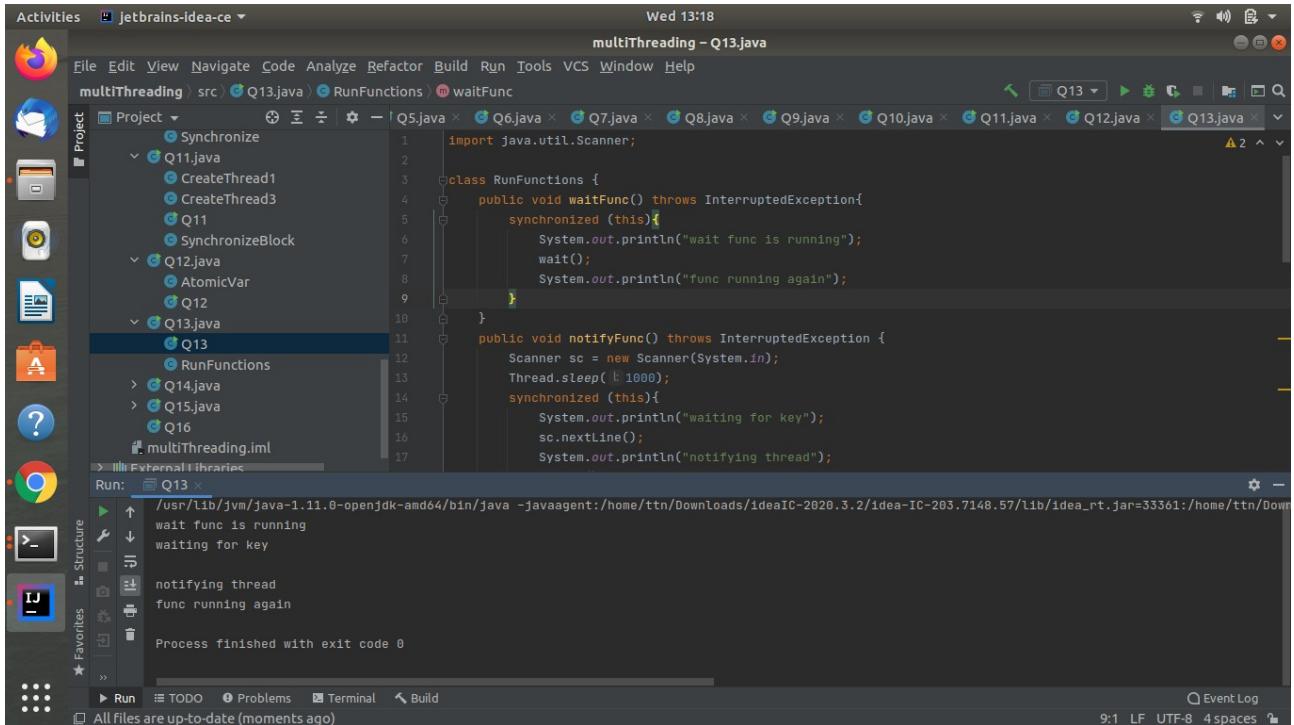
The screenshot shows the IntelliJ IDEA interface with the project 'multithreading' open. The code editor displays `Q12.java` which contains the following code:

```
count.addAndGet( delta: 1);

public class Q12 {
    public static void main(String[] args) throws InterruptedException {
        AtomicVar at = new AtomicVar();
        Thread first = new Thread(at, name: "first");
        Thread second = new Thread(at, name: "second");
        first.start();
        second.start();
        first.join();
        second.join();
        System.out.println(at.count);
    }
}
```

This version of the code uses two separate threads to increment the `AtomicVar`, demonstrating a different approach to thread synchronization.

### Q13: Coordinate 2 threads using wait() and notify().



```
import java.util.Scanner;

class RunFunctions {
    public void waitFunc() throws InterruptedException{
        synchronized (this){
            System.out.println("wait func is running");
            wait();
            System.out.println("func running again");
        }
    }

    public void notifyFunc() throws InterruptedException {
        Scanner sc = new Scanner(System.in);
        Thread.sleep( 1000 );
        synchronized (this){
            System.out.println("waiting for key");
            sc.nextLine();
            System.out.println("notifying thread");
        }
    }
}

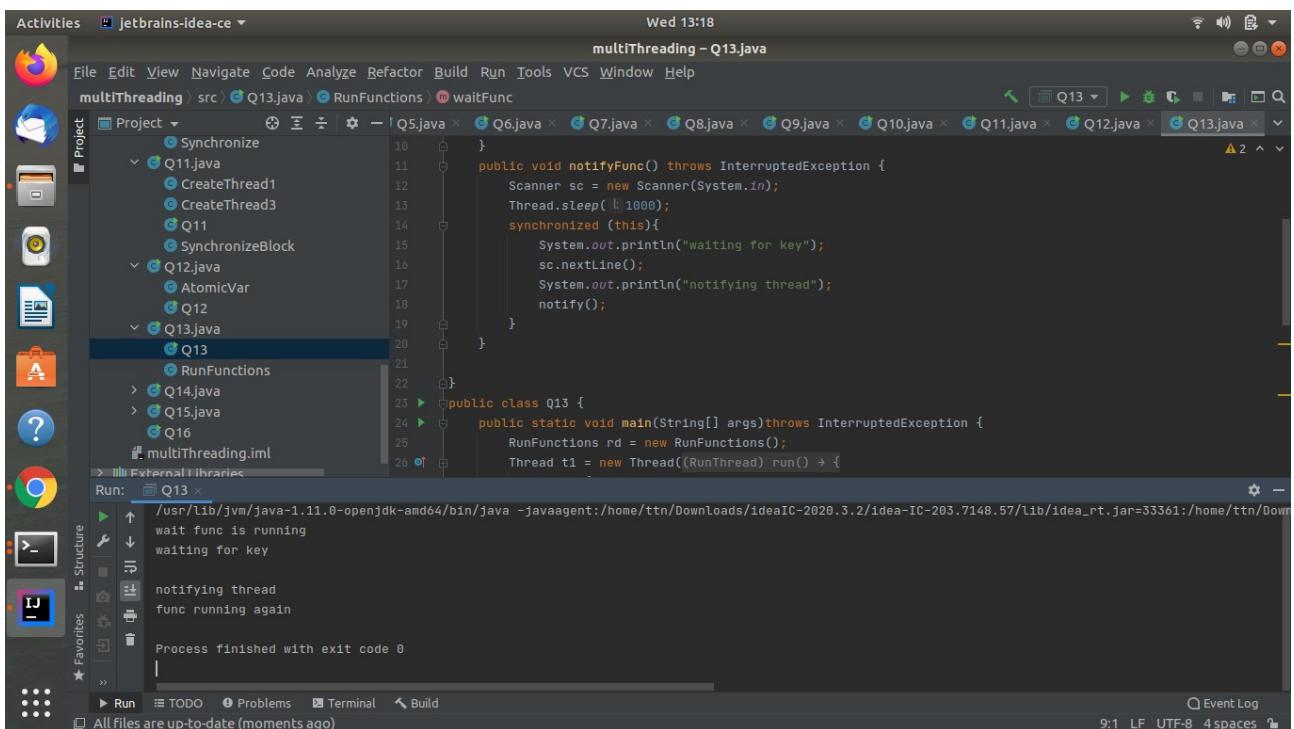
public class Q13 {
    public static void main(String[] args) throws InterruptedException {
        RunFunctions rd = new RunFunctions();
        Thread t1 = new Thread((RunThread) run() -> {
            rd.waitFunc();
        });
        Thread t2 = new Thread((RunThread) run() -> {
            rd.notifyFunc();
        });
        t1.start();
        t2.start();
    }
}
```

The output window shows:

```
/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -javaagent:/home/ttn/Downloads/ideaIC-2020.3.2/idea-IC-203.7148.57/lib/idea_rt.jar=33361:/home/ttn/Downloads/ideaIC-2020.3.2/idea-IC-203.7148.57/lib/idea_rt.jar
wait func is running
waiting for key

notifying thread
func running again

Process finished with exit code 0
```



```
public void notifyFunc() throws InterruptedException {
    Scanner sc = new Scanner(System.in);
    Thread.sleep( 1000 );
    synchronized (this){
        System.out.println("waiting for key");
        sc.nextLine();
        System.out.println("notifying thread");
        notify();
    }
}

public class Q13 {
    public static void main(String[] args) throws InterruptedException {
        RunFunctions rd = new RunFunctions();
        Thread t1 = new Thread((RunThread) run() -> {
            rd.waitFunc();
        });
        Thread t2 = new Thread((RunThread) run() -> {
            rd.notifyFunc();
        });
        t1.start();
        t2.start();
    }
}
```

The output window shows:

```
/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -javaagent:/home/ttn/Downloads/ideaIC-2020.3.2/idea-IC-203.7148.57/lib/idea_rt.jar=33361:/home/ttn/Downloads/ideaIC-2020.3.2/idea-IC-203.7148.57/lib/idea_rt.jar
wait func is running
waiting for key

notifying thread
func running again

Process finished with exit code 0
```

Activities Jetbrains-idea-ce ▾ Wed 13:18

multithreading - Q13.java

```
File Edit Navigate Code Analyze Refactor Build Run Tools VCS Window Help
multithreading > src > Q13.java > RunFunctions > waitFunc
Project Synchronize
  Q11.java
  CreateThread1
  CreateThread3
  Q11
  SynchronizeBlock
  Q12.java
  AtomicVar
  Q12
  Q13.java
    Q13
    RunFunctions
  Q14.java
  Q15.java
  Q16
  multithreading.iml
External Libraries
Run: Q13 x
/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -javaagent:/home/ttn/Downloads/ideaIC-2020.3.2/idea-IC-203.7148.57/lib/idea_rt.jar=33361:/home/ttn/Downloads/ideaIC-2020.3.2/idea-IC-203.7148.57/lib/idea_rt.jar
wait func is running
waiting for key

notifying thread
func running again

Process finished with exit code 0
Event Log
9:1 LF UTF-8 4 spaces
All files are up-to-date (moments ago)
```

Activities Jetbrains-idea-ce ▾ Wed 13:18

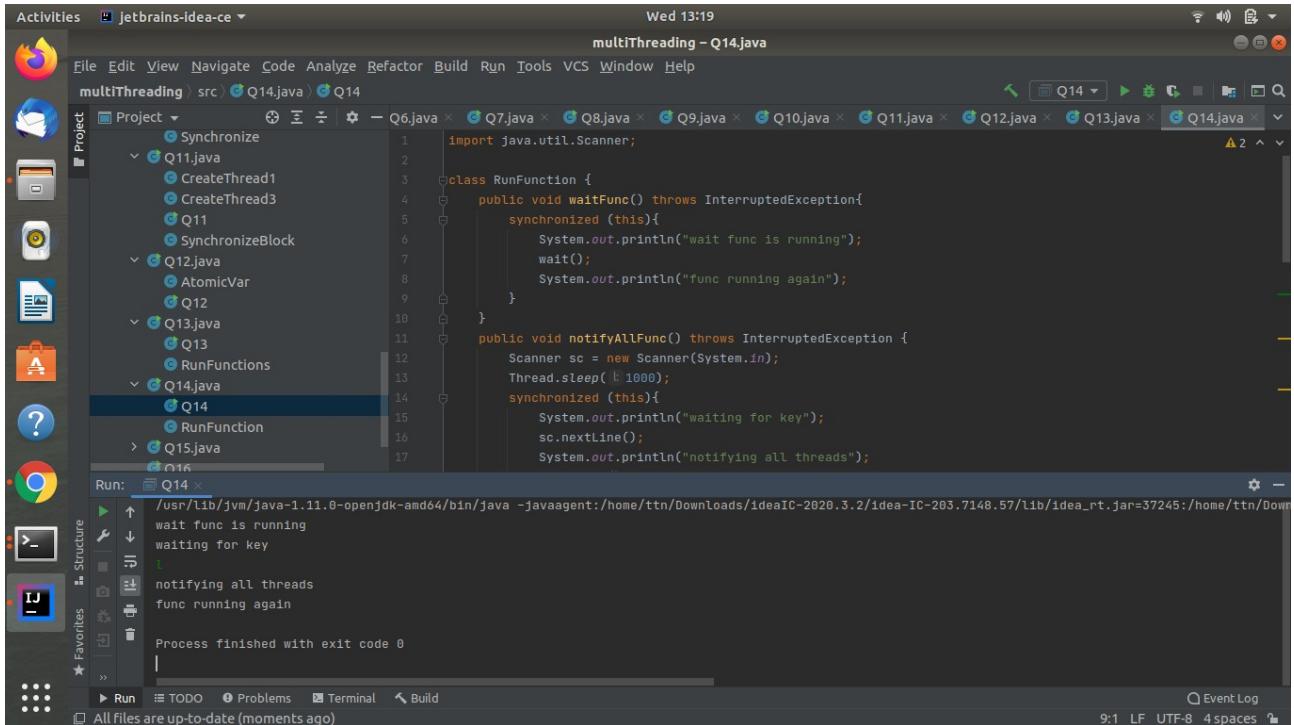
multithreading - Q13.java

```
File Edit Navigate Code Analyze Refactor Build Run Tools VCS Window Help
multithreading > src > Q13.java > RunFunctions > waitFunc
Project Synchronize
  Q11.java
  CreateThread1
  CreateThread3
  Q11
  SynchronizeBlock
  Q12.java
  AtomicVar
  Q12
  Q13.java
    Q13
    RunFunctions
  Q14.java
  Q15.java
  Q16
  multithreading.iml
External Libraries
Run: Q13 x
/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -javaagent:/home/ttn/Downloads/ideaIC-2020.3.2/idea-IC-203.7148.57/lib/idea_rt.jar=33361:/home/ttn/Downloads/ideaIC-2020.3.2/idea-IC-203.7148.57/lib/idea_rt.jar
wait func is running
waiting for key

notifying thread
func running again

Process finished with exit code 0
Event Log
9:1 LF UTF-8 4 spaces
All files are up-to-date (moments ago)
```

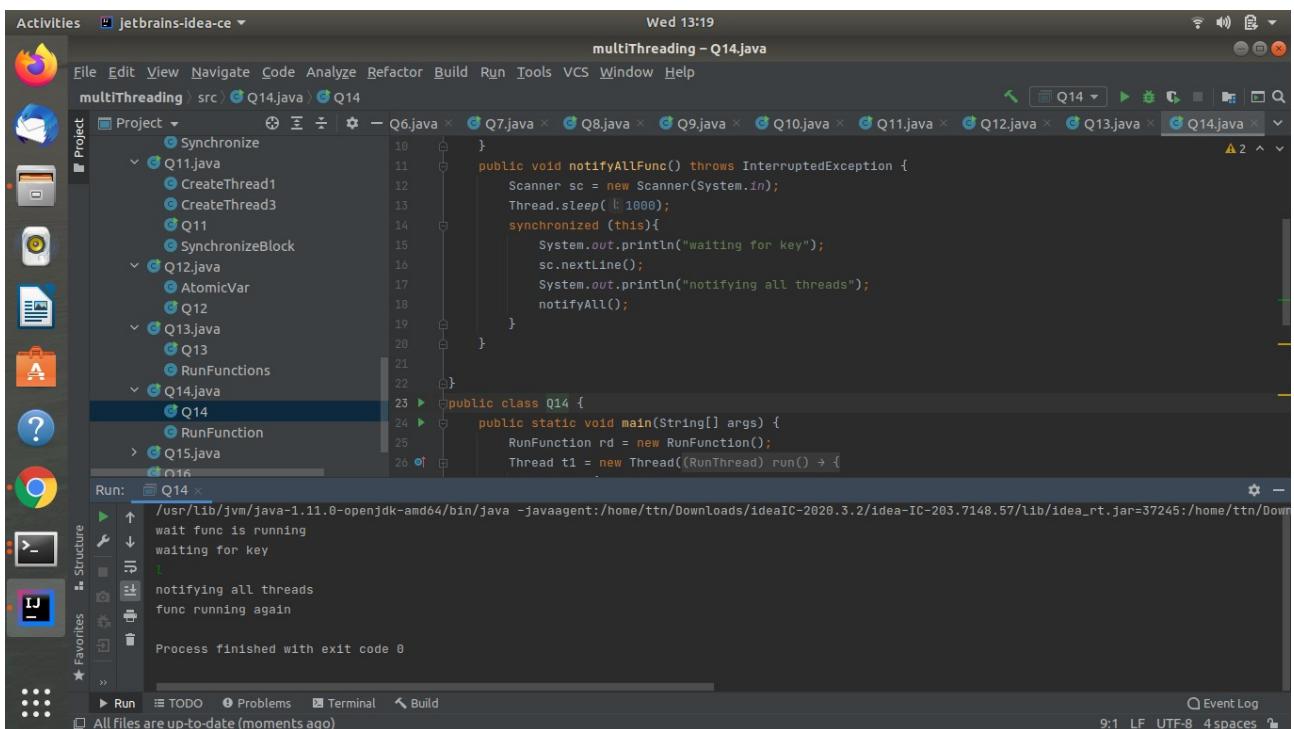
## Q14: Coordinate multiple threads using wait() and notifyAll()



```
import java.util.Scanner;

public class RunFunc {
    public void waitFunc() throws InterruptedException{
        synchronized (this){
            System.out.println("wait func is running");
            wait();
            System.out.println("func running again");
        }
    }

    public void notifyAllFunc() throws InterruptedException {
        Scanner sc = new Scanner(System.in);
        Thread.sleep( 1000 );
        synchronized (this){
            System.out.println("waiting for key");
            sc.nextLine();
            System.out.println("notifying all threads");
        }
    }
}
```



```
public void notifyAllFunc() throws InterruptedException {
    Scanner sc = new Scanner(System.in);
    Thread.sleep( 1000 );
    synchronized (this){
        System.out.println("waiting for key");
        sc.nextLine();
        System.out.println("notifying all threads");
        notifyAll();
    }
}

public class Q14 {
    public static void main(String[] args) {
        RunFunction rd = new RunFunction();
        Thread t1 = new Thread((RunThread) run() > {

```

Activities Jetbrains-idea-ce ▾ Wed 13:19

multithreading - Q14.java

```
public class Q14 {
    public static void main(String[] args) {
        RunFunction rd = new RunFunction();
        Thread t1 = new Thread((RunThread) run() -> {
            try{
                rd.waitFunc();
            }catch (InterruptedException e){
                System.out.println(e);
            }
        });
        Thread t2 = new Thread((RunThread) run() -> {
            try{
                rd.notifyAllFunc();
            }catch (InterruptedException e){
                System.out.println(e);
            }
        });
    }
}
```

Project: multiThreading src: Q14.java

Run: Q14 ×

/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -javaagent:/home/ttn/Downloads/ideaIC-2020.3.2/idea-IC-203.7148.57/lib/idea\_rt.jar=37245:/home/ttn/Downloads/ideaIC-2020.3.2/idea-IC-203.7148.57/lib/idea\_rt.jar

wait func is running  
waiting for key  
t  
notifying all threads  
func running again  
Process finished with exit code 0

Event Log

Activities Jetbrains-idea-ce ▾ Wed 13:19

multithreading - Q14.java

```
public class Q14 {
    public static void main(String[] args) {
        RunFunction rd = new RunFunction();
        Thread t1 = new Thread((RunThread) run() -> {
            try{
                rd.waitFunc();
            }catch (InterruptedException e){
                System.out.println(e);
            }
        });
        Thread t2 = new Thread((RunThread) run() -> {
            try{
                rd.notifyAllFunc();
            }catch (InterruptedException e){
                System.out.println(e);
            }
        });
        t1.start();
        t2.start();
    }
}
```

Project: multiThreading src: Q14.java

Run: Q14 ×

/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -javaagent:/home/ttn/Downloads/ideaIC-2020.3.2/idea-IC-203.7148.57/lib/idea\_rt.jar=37245:/home/ttn/Downloads/ideaIC-2020.3.2/idea-IC-203.7148.57/lib/idea\_rt.jar

wait func is running  
waiting for key  
t  
notifying all threads  
func running again  
Process finished with exit code 0

Event Log

## Q15: Use Reentrant lock for coordinating 2 threads with signal(), signalAll() and wait().

```
Activities JetBrains-idea-ce Wed 13:21
multiThreading - Q15.java
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
multiThreading > src > Q15.java > Q15 > main
Project Q10 Synchronize
Q11.java CreateThread1 CreateThread3 Q11 SynchronizeBlock
Q12.java AtomicVar Q12 Q13.java Q13 RunFunctions
Q14.java Q14 RunFunction Q15.java Q15 Runner Q16
multithreading.iml External Libraries
Run: Q15 x
waiting.... press key
Run TODO Problems Terminal Build
All files are up-to-date (moments ago) 4:10 LF UTF-8 4 spaces Event Log
```

```
1 import java.util.Scanner;
2 import java.util.concurrent.locks.Condition;
3 import java.util.concurrent.locks.Lock;
4 import java.util.concurrent.locks.ReentrantLock;
5
6 class Runner{
7     int count =0;
8     Lock lock = new ReentrantLock();
9     Condition cond = lock.newCondition();
10
11     public void increase(){
12         for(int i=0;i<10000;i++){
13             count++;
14         }
15     }
16     public void firstThread() throws InterruptedException {
17         lock.lock();
18         System.out.println("waiting");
19         cond.await();
20         System.out.println("Woken up!!!");
21         try{
22             increase();
23         }finally {
24             lock.unlock();
25         }
26     }
27     public void secondThread() throws InterruptedException {
28         lock.lock();
29         System.out.println("waiting.....");
30         cond.await();
31         System.out.println("Woken up!!!");
32         try{
33             increase();
34         }finally {
35             lock.unlock();
36         }
37     }
38 }
```

```
Activities JetBrains-idea-ce Wed 13:21
multiThreading - Q15.java
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
multiThreading > src > Q15.java > Q15 > main
Project Q10 Synchronize
Q11.java CreateThread1 CreateThread3 Q11 SynchronizeBlock
Q12.java AtomicVar Q12 Q13.java Q13 RunFunctions
Q14.java Q14 RunFunction Q15.java Q15 Runner Q16
multithreading.iml External Libraries
Run: Q15 x
waiting.... press key
Run TODO Problems Terminal Build
All files are up-to-date (a minute ago) 4:10 LF UTF-8 4 spaces Event Log
```

```
16     public void firstThread() throws InterruptedException {
17         lock.lock();
18         System.out.println("waiting");
19         cond.await();
20         System.out.println("Woken up!!!");
21         try{
22             increase();
23         }finally {
24             lock.unlock();
25         }
26     }
27     public void secondThread() throws InterruptedException {
28         lock.lock();
29         System.out.println("waiting.....");
30         cond.await();
31         System.out.println("Woken up!!!");
32         try{
33             increase();
34         }finally {
35             lock.unlock();
36         }
37     }
38     public void thirdThread() throws InterruptedException {
```

Activities Jetbrains-idea-ce ▾ Wed 13:21

multithreading - Q15.java

```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
multithreading > src > Q15.java > Q15 > main
Project 7.java × Q8.java × Q9.java × Q10.java × Q11.java × Q12.java × Q13.java × Q14.java × Q15.java ×
Q10
  Synchronize
  Q11.java
    CreateThread1
    CreateThread3
    Q11
    SynchronizeBlock
  Q12.java
    AtomicVar
    Q12
  Q13.java
    Q13
    RunFunctions
  Q14.java
    Q14
    RunFunction
  Q15.java
    Q15
    Runner
    Q16
    multithreading.iml
External Libraries
Run: Q15 ×
waiting..... press key
Run TODO Problems Terminal Build
All files are up-to-date (a minute ago)
```

4:10 LF UTF-8 4 spaces Event Log

Activities Jetbrains-idea-ce ▾ Wed 13:21

multithreading - Q15.java

```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
multithreading > src > Q15.java > Q15 > main
Project 7.java × Q8.java × Q9.java × Q10.java × Q11.java × Q12.java × Q13.java × Q14.java × Q15.java ×
Q10
  Synchronize
  Q11.java
    CreateThread1
    CreateThread3
    Q11
    SynchronizeBlock
  Q12.java
    AtomicVar
    Q12
  Q13.java
    Q13
    RunFunctions
  Q14.java
    Q14
    RunFunction
  Q15.java
    Q15
    Runner
    Q16
    multithreading.iml
External Libraries
Run: Q15 ×
waiting..... press key
Run TODO Problems Terminal Build
All files are up-to-date (a minute ago)
```

4:10 LF UTF-8 4 spaces Event Log

Activities Jetbrains-idea-ce ▾ Wed 13:21

multithreading - Q15.java

```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
multiThreading > src > Q15.java > Q15 > main
Project 7.java × Q8.java × Q9.java × Q10.java × Q11.java × Q12.java × Q13.java × Q14.java × Q15.java ×
Q10
Synchronize
Q11.java
CreateThread1
CreateThread3
Q11
SynchronizeBlock
Q12.java
AtomicVar
Q12
Q13.java
Q13
RunFunctions
Q14.java
Q14
RunFunction
Q15.java
Q15
Runner
Q16
multithreading.iml
External Libraries
Run: Q15 ×
waiting..... press key
Run TODO Problems Terminal Build
All files are up-to-date (a minute ago)
4:10 LF UTF-8 4 spaces Event Log
```

The code implements a multi-threaded application. It starts with a static main method that creates a Runner object and two threads, t1 and t2. Each thread overrides the run method to print the first and second threads respectively. The Runner class has a firstThread and secondThread method.

```
public class Q15 {
    public static void main(String[] args) throws Exception {
        Runner run = new Runner();
        Thread t1 = new Thread(new Runnable() {
            @Override
            public void run() {
                try {
                    run.firstThread();
                } catch (InterruptedException e) {
                    System.out.println(e);
                }
            }
        });
        Thread t2 = new Thread(new Runnable() {
            @Override
            public void run() {
                try {
                    run.secondThread();
                } catch (InterruptedException e) {
                    System.out.println(e);
                }
            }
        });
    }
}

class Runner {
    public void firstThread() {
        System.out.println("firstThread");
    }

    public void secondThread() {
        System.out.println("secondThread");
    }
}
```

Activities Jetbrains-idea-ce ▾ Wed 13:21

multithreading - Q15.java

```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
multiThreading > src > Q15.java > Q15 > main
Project 7.java × Q8.java × Q9.java × Q10.java × Q11.java × Q12.java × Q13.java × Q14.java × Q15.java ×
Q10
Synchronize
Q11.java
CreateThread1
CreateThread3
Q11
SynchronizeBlock
Q12.java
AtomicVar
Q12
Q13.java
Q13
RunFunctions
Q14.java
Q14
RunFunction
Q15.java
Q15
Runner
Q16
multithreading.iml
External Libraries
Run: Q15 ×
waiting..... press key
Run TODO Problems Terminal Build
All files are up-to-date (a minute ago)
4:10 LF UTF-8 4 spaces Event Log
```

The code is identical to the previous one, but the Runner class now contains three methods: firstThread, secondThread, and thirdThread. The thirdThread method is currently empty.

```
public class Q15 {
    public static void main(String[] args) throws Exception {
        Runner run = new Runner();
        Thread t1 = new Thread(new Runnable() {
            @Override
            public void run() {
                try {
                    run.secondThread();
                } catch (InterruptedException e) {
                    System.out.println(e);
                }
            }
        });
        Thread t2 = new Thread(new Runnable() {
            @Override
            public void run() {
                try {
                    run.thirdThread();
                } catch (InterruptedException e) {
                    System.out.println(e);
                }
            }
        });
        Thread t3 = new Thread(new Runnable() {
            @Override
            public void run() {
                try {
                    run.thirdThread();
                } catch (InterruptedException e) {
                    System.out.println(e);
                }
            }
        });
        Thread t4 = new Thread(new Runnable() {
            @Override
            public void run() {
                try {
                    run.thirdThread();
                } catch (InterruptedException e) {
                    System.out.println(e);
                }
            }
        });
    }
}

class Runner {
    public void firstThread() {
        System.out.println("firstThread");
    }

    public void secondThread() {
        System.out.println("secondThread");
    }

    public void thirdThread() {
        System.out.println("thirdThread");
    }
}
```

Activities Jetbrains-idea-ce ▾ Wed 13:21

multithreading - Q15.java

```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
multithreading > src > Q15.java > Q15 > main
Project 7.java × Q8.java × Q9.java × Q10.java × Q11.java × Q12.java × Q13.java × Q14.java × Q15.java ×
Q10
  Synchronize
  Q11.java
    CreateThread1
    CreateThread3
    Q11
    SynchronizeBlock
  Q12.java
    AtomicVar
    Q12
  Q13.java
    Q13
    RunFunctions
  Q14.java
    Q14
    RunFunction
  Q15.java
    Q15
    Runner
    Q16
      multithreading.iml
External Libraries
Run: Q15 ×
waiting..... press key
Run TODO Problems Terminal Build
All files are up-to-date (a minute ago)
4:10 LF UTF-8 4 spaces Event Log
```

```
90      });
91      Thread t3 = new Thread(new Runnable() {
92          @Override
93          public void run() {
94              try{
95                  run.thridThread();
96              }catch (InterruptedException e){
97                  System.out.println(e);
98              }
99      });
100     Thread t4 = new Thread(new Runnable() {
101         @Override
102         public void run() {
103             try{
104                 run.fourthThread();
105             }catch (InterruptedException e){
106                 System.out.println(e);
107             }
108         });
109     t1.start();
110     t2.start();
111 }
```

Activities Jetbrains-idea-ce ▾ Wed 13:21

multithreading - Q15.java

```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
multithreading > src > Q15.java > Q15 > main
Project 7.java × Q8.java × Q9.java × Q10.java × Q11.java × Q12.java × Q13.java × Q14.java × Q15.java ×
Q10
  Synchronize
  Q11.java
    CreateThread1
    CreateThread3
    Q11
    SynchronizeBlock
  Q12.java
    AtomicVar
    Q12
  Q13.java
    Q13
    RunFunctions
  Q14.java
    Q14
    RunFunction
  Q15.java
    Q15
    Runner
    Q16
      multithreading.iml
External Libraries
Run: Q15 ×
waiting..... press key
Run TODO Problems Terminal Build
All files are up-to-date (a minute ago)
4:10 LF UTF-8 4 spaces Event Log
```

```
99      });
100     Thread t4 = new Thread(new Runnable() {
101         @Override
102         public void run() {
103             try{
104                 run.fourthThread();
105             }catch (InterruptedException e){
106                 System.out.println(e);
107             }
108         });
109     t1.start();
110     t2.start();
111     t3.start();
112     t4.start();
113     t1.join();
114     t2.join();
115     t3.join();
116     t4.join();
117 }
```

Activities JetBrains-idea-ce Wed 13:21

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

multithreading - Q15.java

Project

- Q10
- Synchronize
- Q11.java
  - CreateThread1
  - CreateThread3
  - Q11
  - SynchronizeBlock
- Q12.java
  - AtomicVar
  - Q12
- Q13.java
  - Q13
  - RunFunctions

Q15.java

```
110    t1.start();
111    t2.start();
112    t3.start();
113    t4.start();
114    t1.join();
115    t2.join();
116    t3.join();
117    t4.join();
118}
119}
120}
121}
122}
```

Run: Q15

```
/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -javaagent:/home/ttn/Downloads/ideaIC-2020.3.2/ideaIC-203.7148.57/lib/idea_rt.jar=/home/ttn/Down
waiting
waiting.....
press key
press key

Woken up!!!
Woken up2!!!

Process finished with exit code 0
```

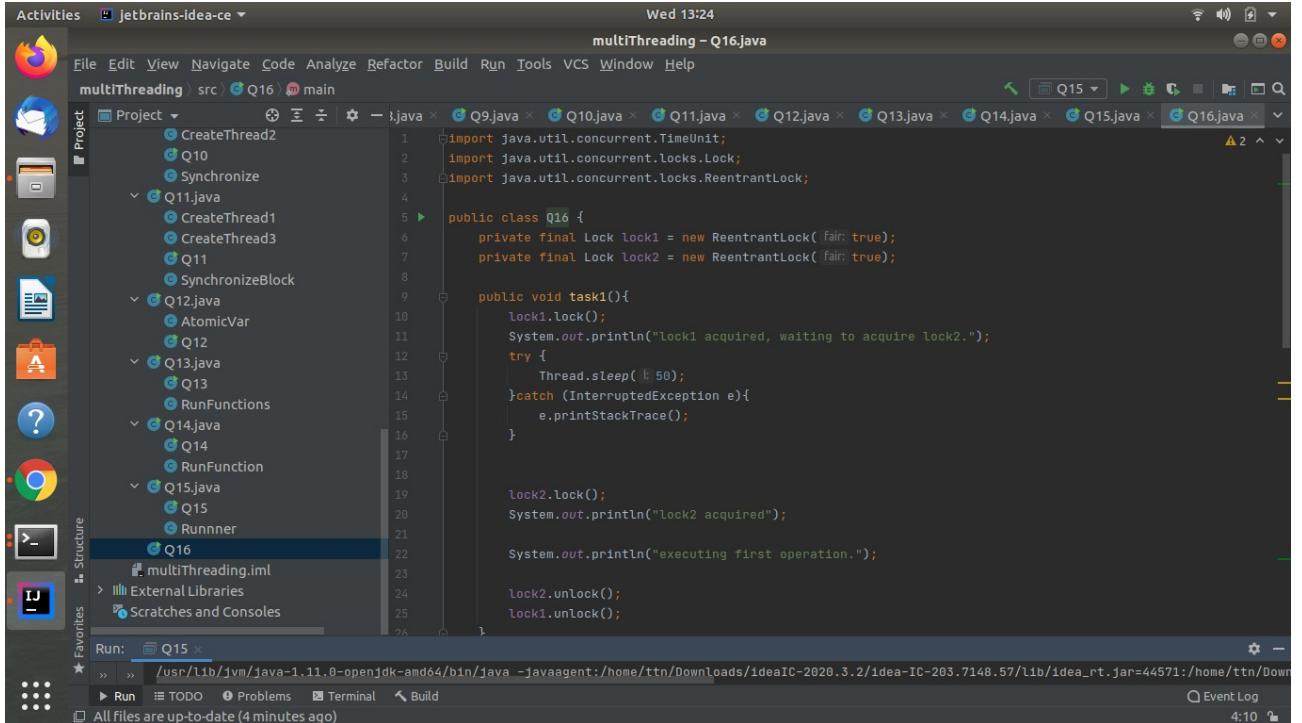
All files are up-to-date (a minute ago)

Event Log

117:19 LF UTF-8 4 spaces

The screenshot shows the IntelliJ IDEA interface with a Java project named 'multithreading'. The 'Q15.java' file is open, displaying code related to thread creation and joining. The 'Run' tool window shows the application was run with the command '/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -javaagent:/home/ttn/Downloads/ideaIC-2020.3.2/ideaIC-203.7148.57/lib/idea\_rt.jar=/home/ttn/Down'. The output indicates the program is waiting for input ('press key') and has woken up twice ('Woken up!!!' and 'Woken up2!!!'). The status bar at the bottom right shows the time as 117:19, file encoding as LF, and a 4-space indent.

## Q16: Create a deadlock and Resolve it using tryLock().



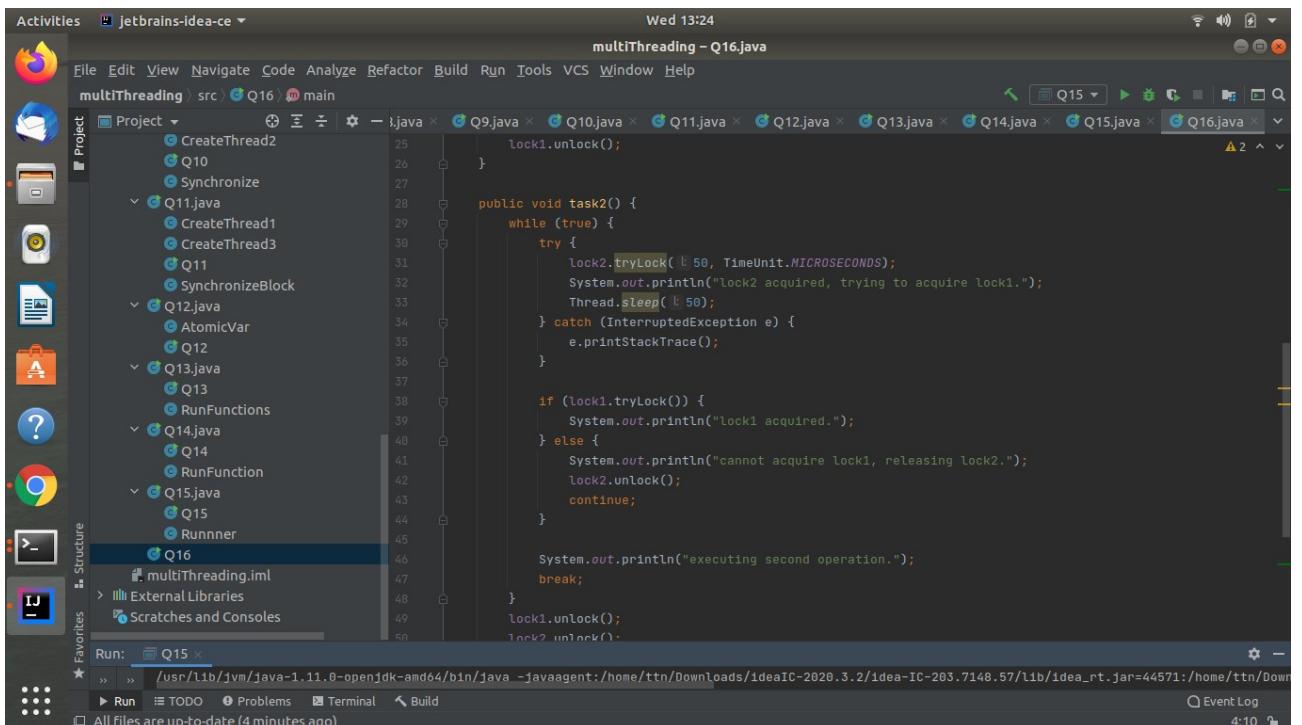
```
Activities JetBrains-idea-ce Wed 13:24
multiThreading - Q16.java

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
multiThreading src Q16 main

Project Q16.java Q9.java Q10.java Q11.java Q12.java Q13.java Q14.java Q15.java Q16.java
  CreateThread2
  Q10
  Synchronize
  Q11.java
  CreateThread1
  CreateThread3
  Q11
  SynchronizeBlock
  Q12.java
  AtomicVar
  Q12
  Q13.java
  Q13
  RunFunctions
  Q14.java
  Q14
  RunFunction
  Q15.java
  Q15
  Runner
  Q16
    multiThreading.iml
  External Libraries
  Scratches and Consoles

1 import java.util.concurrent.TimeUnit;
2 import java.util.concurrent.locks.Lock;
3 import java.util.concurrent.locks.ReentrantLock;
4
5 public class Q16 {
6     private final Lock lock1 = new ReentrantLock( fair: true );
7     private final Lock lock2 = new ReentrantLock( fair: true );
8
9     public void task1(){
10         lock1.lock();
11         System.out.println("lock1 acquired, waiting to acquire lock2.");
12         try {
13             Thread.sleep( 50 );
14         } catch (InterruptedException e){
15             e.printStackTrace();
16         }
17
18         lock2.lock();
19         System.out.println("lock2 acquired");
20
21         System.out.println("executing first operation.");
22
23         lock2.unlock();
24         lock1.unlock();
25     }
26
27     public void task2(){
28         while ( true ) {
29             try {
30                 lock2.tryLock( 50, TimeUnit.MICROSECONDS );
31                 System.out.println("lock2 acquired, trying to acquire lock1.");
32                 Thread.sleep( 50 );
33             } catch (InterruptedException e) {
34                 e.printStackTrace();
35             }
36
37             if ( lock1.tryLock() ) {
38                 System.out.println("lock1 acquired.");
39             } else {
40                 System.out.println("cannot acquire lock1, releasing lock2.");
41                 lock2.unlock();
42                 continue;
43             }
44
45             System.out.println("executing second operation.");
46             break;
47         }
48         lock1.unlock();
49         lock2.unlock();
50     }
51 }

Run: Q15 > /usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -javaagent:/home/ttn/Downloads/ideaIC-2020.3.2/idea-IC-203.7148.57/lib/idea_rt.jar=44571:/home/ttn/Down Event Log
Run TODO Problems Terminal Build
All files are up-to-date (4 minutes ago)
```



```
Activities JetBrains-idea-ce Wed 13:24
multiThreading - Q16.java

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
multiThreading src Q16 main

Project Q16.java Q9.java Q10.java Q11.java Q12.java Q13.java Q14.java Q15.java Q16.java
  CreateThread2
  Q10
  Synchronize
  Q11.java
  CreateThread1
  CreateThread3
  Q11
  SynchronizeBlock
  Q12.java
  AtomicVar
  Q12
  Q13.java
  Q13
  RunFunctions
  Q14.java
  Q14
  RunFunction
  Q15.java
  Q15
  Runner
  Q16
    multiThreading.iml
  External Libraries
  Scratches and Consoles

25
26         lock1.unlock();
27
28     public void task2(){
29         while ( true ) {
30             try {
31                 lock2.tryLock( 50, TimeUnit.MICROSECONDS );
32                 System.out.println("lock2 acquired, trying to acquire lock1.");
33                 Thread.sleep( 50 );
34             } catch (InterruptedException e) {
35                 e.printStackTrace();
36             }
37
38             if ( lock1.tryLock() ) {
39                 System.out.println("lock1 acquired.");
40             } else {
41                 System.out.println("cannot acquire lock1, releasing lock2.");
42                 lock2.unlock();
43                 continue;
44             }
45
46             System.out.println("executing second operation.");
47             break;
48         }
49         lock1.unlock();
50         lock2.unlock();
51     }
52 }

Run: Q15 > /usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -javaagent:/home/ttn/Downloads/ideaIC-2020.3.2/idea-IC-203.7148.57/lib/idea_rt.jar=44571:/home/ttn/Down Event Log
Run TODO Problems Terminal Build
All files are up-to-date (4 minutes ago)
```

Activities Jetbrains-idea-ce ▾ Wed 13:24

multithreading - Q16.java

The screenshot shows the IntelliJ IDEA interface with the project 'multithreading' open. The 'src' directory contains several Java files: CreateThread2, Q10, Synchronize, Q11, Q12, Q13, Q14, Q15, and Runner. The 'Q16.java' file is the active editor, showing the following code:

```
1T (LOCK1.TRYLOCK()) {
    System.out.println("lock1 acquired.");
} else {
    System.out.println("cannot acquire lock1, releasing lock2.");
    lock2.unlock();
    continue;
}

System.out.println("executing second operation.");
break;
lock1.unlock();
lock2.unlock();

public static void main(String[] args) {
    Q16 deadlock = new Q16();
    new Thread(deadlock::task1, name: "T1").start();
    new Thread(deadlock::task2, name: "T2").start();
}
```

The 'Run' tool bar at the bottom has 'Q15' selected. The terminal output shows:

```
lock1 acquired, waiting to acquire lock2.
lock2 acquired, trying to acquire lock1.
cannot acquire lock1, releasing lock2.
lock2 acquired
executing first operation.
lock2 acquired, trying to acquire lock1.
lock1 acquired.
executing second operation.
```

Activities Jetbrains-idea-ce ▾ Wed 13:25

multithreading - Q16.java

The screenshot shows the IntelliJ IDEA interface with the project 'multithreading' open. The 'src' directory contains several Java files: CreateThread2, Q10, Synchronize, Q11, Q12, Q13, Q14, Q15, and Runner. The 'Q16.java' file is the active editor, showing the same Java code as the previous screenshot. The 'Run' tool bar at the bottom has 'Q16' selected. The terminal output shows:

```
lock1 acquired, waiting to acquire lock2.
lock2 acquired, trying to acquire lock1.
cannot acquire lock1, releasing lock2.
lock2 acquired
executing first operation.
lock2 acquired, trying to acquire lock1.
lock1 acquired.
executing second operation.

Process finished with exit code 0
```