SimpleThread

```
polymorpnism.java
                                    4 4 */
 > 🛅 Test Packages
                                          package multithreadapp;
 > 🝙 Libraries
 > 🝙 Test Libraries
                                    7 - /**
 MultiThreadApp
 Source Packages
                                           * @author USER
    ∨ ■ multithreadapp
                                   10
        Counter.java
                                   11
                                          public class SimpleThread extends Thread{
        MultiThreadApp.java
                                   12
                                              @Override
        RunnableTask.java
                                    0
                                              public void run(){
        SimpleThread.java
                                                  System.out.println(Thread.currentThread() .getId() +"is executing the thread");
                                   14
         SynchronizedExample.java
                                   15
 > iii Test Packages
                                   16
                                              public static void main(String[]args) {
 > 🝙 Libraries
                                   17
                                                   SimpleThread thread1=new SimpleThread();
 > 廜 Test Libraries
                                                   SimpleThread thread2=new SimpleThread();
                                   18
 paySlip
                                   19
 Source Packages
                                                   thread1.start();
                                   20
    v 🖺 payslip
                                   21
                                                   thread2.start();
        Atom.java
                                   22
        PaySlip.java
                                   23
  > 廜 Libraries
                                   24
/ 🐡 Practis
Output - MultiThreadApp (run)
     run:
     21is executing the thread
     22is executing the thread
     BUILD SUCCESSFUL (total time: 0 seconds)
```

RunnableTask

```
> 🔓 Test Packages
                                       package multithreadapp;
> 廜 Libraries
> 🝙 Test Libraries
                                  7 - /**
MultiThreadApp
                                  8
9
                                        * @author USER

∨ 

■ multithreadapp

                                 10
       Counter.java
                                 11
                                       public class RunnableTask implements Runnable{
       MultiThreadApp.java
                                 12
                                            @Override
       RunnableTask.java
                                 1
                                            public void run() {
       SimpleThread.java
                                 14
                                               System.out.println(Thread.currentThread() .getId() +"is executing the runnable task.");
       SynchronizedExample.java
                                 15
> a Test Packages
                                 16
                                            public static void main(String[]args) {
> 📦 Libraries
                                 17
                                                 RunnableTask task1=new RunnableTask();
> 廜 Test Libraries
                                                 RunnableTask task2=new RunnableTask();
                                 18
paySlip
                                 19
Thread thread1=new Thread(task: task1);
                                 20
   ∨ 🖺 payslip
                                                Thread thread2=new Thread(task: task2);
                                 21
       Atom.java
                                 22
       PaySlip.java
                                 23
                                                 thread1.start();
> 廜 Libraries
                                 24
                                                 thread2.start();
Practis
                                 25
utput - MultiThreadApp (run)
   22is executing the runnable task.
   21is executing the runnable task.
   BUILD SUCCESSFUL (total time: 0 seconds)
```

Synchronizing Thread

Final counter value: 2000

BUILD SUCCESSFUL (total time: 0 seconds)

```
Vehicle.java
                                 L */
 d child.java
                             11
                                    public class SynchronizedExample extends Thread {
    parent.java
                              9
                                        private Counter counter;
 polymorphism.java
                             13
est Packages
                             14
                                        public SynchronizedExample(Counter counter) {
ibraries
                             15
                                             this.counter = counter;
est Libraries
                             16
tiThreadApp
                             17
Source Packages
                             18
                                        @Override
multithreadapp
                              0
                                        public void run() {
 Counter.java
                             20
                                            for (int i = 0; i < 1000; i++) {
 MultiThreadApp.java
                                                 counter.increment();
                             21
 RunnableTask.java
                             22
 SimpleThread.java
                             23
SynchronizedExample.java
                             24
est Packages
                             25
                                        public static void main(String[] args) throws InterruptedException {
.ibraries
                             26
                                            Counter counter = new Counter();
est Libraries
                             27
Slip
                             28
                                             Thread thread1 = new SynchronizedExample(counter);
Source Packages
                             29
                                             Thread thread2 = new SynchronizedExample(counter);
payslip
                             30
 Atom.java
                             31
                                             thread1.start();
 PaySlip.java
                                             thread2.start();
                             32
.ibraries
                             33
                             34
                                             thread1.join();
SynchronizedExample.java...
                             35
                                             thread2.join();
                             36
abases
                             37
                                             System.out.println("Final counter value: " + counter.getCount());
ers
                             38
en Repositories
                             39
                             40
son Builders
```

```
Vehicle.java
                                10
     dild.java
                                11
                                       public class SynchronizedExample extends Thread {
     parent.java
                                 <u>Q.</u>
                                           private Counter counter;
     polymorphism.java
                                13
Test Packages
                                           public SynchronizedExample(Counter counter) {
                                14
Libraries
                                15
                                               this.counter = counter;
Test Libraries
                                16
MultiThreadApp
                                17
Source Packages
                                18
                                           @Override
v 🖺 multithreadapp
                                 0
                                           public void run() {
     Counter.iava
                                20
                                               for (int i = 0; i < 1000; i++) {
     MultiThreadApp.java
                                21
                                                    counter.increment();
     RunnableTask.java
                                22
     SimpleThread.java
                                23
    SynchronizedExample.java
                                24
Test Packages
                                           public static void main(String[] args) throws InterruptedException {
                                25
Libraries
                                26
                                               Counter counter = new Counter();
■ Test Libraries
                                27
paySlip
                                28
                                               Thread thread1 = new SynchronizedExample(counter);
Source Packages
                                29
                                               Thread thread2 = new SynchronizedExample(counter);

√ III payslip

                                30
     🐴 Atom.java
                                31
                                                thread1.start();
     PaySlip.java
                                32
                                                thread2.start();
■ Libraries
                                33
Practis
                                                thread1.ioin():
ıt - MultiThreadApp (run)
```

Thread Pooling

```
GPG.java
                              11 import java.util.concurrent.ExecutorService;
    LinearSearch.java
                                    import java.util.concurrent.Executors;
                              12
    LinkedQueue.java
                              13
    🐴 Linklist1.java
                              14
                                    public class ThreadPoolExample {
    Push.java
                              15 📮
                                         static class Task implements Runnable {
    Queue.java
                               Q.
                                             private int taskId;
    Stack.java
                              17
    StackLinkList.java
                              18
                                             public Task(int taskId) { // Fixed constructor assignment
    Vehicle.java
                              19
                                                 this.taskId = taskId;
    d child.java
                              20
    parent.java
                              21
    polymorphism.java
                              ₩.
                                             public void run() {
Test Packages
                              23
                                                 System.out.println("Task " + taskId + " is being processed by " + Thread.currentThread().getName());
Libraries
                              24
Test Libraries
                              25
MultiThreadApp
                              26
Source Packages
                              27
                                         public static void main(String[] args) { // Fixed method signature
✓ 
■ multithreadapp
                              28
                                             ExecutorService executorService = Executors.newFixedThreadPool(nThreads: 3); // Fixed typo
    Counter.java
                              29
    MultiThreadApp.java
                              30
                                             for (int i = 1; i \le 5; i++) {
    RunnableTask.java
                              31
                                                 executorService.submit(new Task(taskId:i)); // Correct instance creation
    SimpleThread.java
                              32
    SynchronizedExample.java
                              33
    ThreadPoolExample.java
                              34
                                             executorService.shutdown();

■ Test Packages
```

- MultiThreadApp (run)

```
Task 1 is being processed by pool-1-thread-1
Task 2 is being processed by pool-1-thread-2
Task 4 is being processed by pool-1-thread-1
Task 3 is being processed by pool-1-thread-3
Task 5 is being processed by pool-1-thread-2
BUILD SUCCESSFUL (total time: 0 seconds)
```

Thread Lifecycle Example

```
parent.java
                                        * @author USER
        polymorphism.java
 > 📔 Test Packages
                                 11
                                       public class ThreadLifecycleExample extends Thread {
 > 廜 Libraries
                                 12
                                            @Override
   Test Libraries
                                  6 E
                                           public void run() {
 MultiThreadApp
                                 14
                                                System.out.println(Thread.currentThread().getName() + " - state: " + Thread.currentThread().getState());
 15
   ∨ 

multithreadapp
                                                    Thread. sleep (millis: 2000);
                                 16
        Counter.java
                                 17
                                                } catch (InterruptedException e) {
        MultiThreadApp.java
                                                    e.printStackTrace();
        RunnableTask.java
                                 19
        SimpleThread.java
                                                System.out.println(Thread.currentThread().getName() + " - state after sleep: " + Thread.currentThread().getState());
                                  20
        SynchronizedExample.java
                                  21
        ThreadLifecycleExample.java
                                  22
        ThreadPoolExample.java
                                  23 🖃
                                            public static void main(String[] args) {
 > 🔓 Test Packages
                                                ThreadLifecycleExample thread = new ThreadLifecycleExample();
                                 24
 > 廜 Libraries
                                 25
                                                System.out.println(thread.getName() + " - state before start: " + thread.getState());
 > 🍙 Test Libraries
                                 26
 paySlip
                                 27
                                                thread.start();
 28
   ∨ 🖺 payslip
                                 29
                                                System.out.println(thread.getName() + " - state after start: " + thread.getState());
        Atom.java
                                  30
        PaySlip.java
                                 31
                                  32
Output - MultiThreadApp (run)
    run:
    Thread-0 - state before start: NEW
    Thread-0 - state after start: RUNNABLE
    Thread-0 - state: RUNNABLE
    Thread-0 - state after sleep: RUNNABLE
     BUILD SUCCESSFUL (total time: 2 seconds)
```