JAVA.UTIL.TIMER.SCHEDULEATFIXEDRATE METHOD

Description

The **scheduleAtFixedRate***TimerTasktask, longdelay, longperiod* method is used to schedule the specified task for repeated fixed-rate execution, beginning after the specified delay.

Declaration

Following is the declaration for **java.util.Timer.scheduleAtFixedRate** method.

```
public void scheduleAtFixedRate(TimerTask task,long delay,long period)
```

Parameters

- **task** -- This is the task to be scheduled.

- **delay** -- This is the delay in milliseconds before task is to be executed.

- **period** -- This is the time in milliseconds between successive task executions.

Return Value

- **NA**

Exception

- **IllegalArgumentException** -- This exception is thrown if time.getTime is negative.

- **IllegalStateException** -- This is thrown if task was already scheduled or cancelled, timer was cancelled, or timer thread terminated.

Example

The following example shows the usage of java.util.Timer.scheduleAtFixedRate

```java
package com.tutorialspoint;

import java.util.*;

public class TimerDemo {
   public static void main(String[] args) {
   // creating timer task, timer
   TimerTask tasknew = new TimerScheduleFixedRateDelay();
   Timer timer = new Timer();

   // scheduling the task at fixed rate delay
   timer.scheduleAtFixedRate(tasknew,500,1000);
   }
   // this method performs the task
   public void run() {
   System.out.println("working at fixed rate delay");
   }
}
```

Let us compile and run the above program, this will produce the following result.

```
working at fixed rate delay
working at fixed rate delay
working at fixed rate delay
working at fixed rate delay and so on ....
```

Loading [MathJax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js