**Hands On Lab 6**

## **[Timer and TimerTask](https://personales.unican.es/corcuerp/java/Labs/LAB_23.htm" \l "Exercise_1)**

In this exercise, you will learn how to use Timer and TimerTask to schedule a single or repeating task.

* 1. **[Schedule one-time task](https://personales.unican.es/corcuerp/java/Labs/LAB_23.htm" \l "6.1)**
  2. **[Schedule repeating task](https://personales.unican.es/corcuerp/java/Labs/LAB_23.htm" \l "6.2)**

### **(6.1) Schedule one-time task**

1. Create a new Java project

* Select **File->New Java Project (Alt+Shift+N)**. The **New Java Project** dialog box appears.
* For the Project Name field, type in **TimerReminder**as project name.
* Choose ‘Use default JRE (11….). Make sure you have set your default JRE set to version 11 and the appropriate compliance settings as well. Deselect ‘Create module-info.java file’ if already selected. Click Next.Click Finish.
* Observe that **TimerReminder**project appears
* Right click on the project and select **File**->**New Class.**
* Enter in a package name of your choosing or use the default. Type in **TimerReminder**as the class name. The main method stub should also be created.
* The IDE generated **TimerReminder .java** is displayed in the source editor window of STS IDE.

2. Modify the IDE generated **TimerReminder.java**as shown in Code-6.11 below.  Study the code by paying special attention to the bold fonted parts.

|  |
| --- |
| import java.util.Timer; import java.util.TimerTask;  /\*\*  \* Simple demo that uses java.util.Timer to schedule a task to execute  \* once 5 seconds have passed.  \*/  public class TimerReminder {         Timer timer;         public TimerReminder(int seconds) {         **timer = new Timer();**         **timer.schedule(new RemindTask(), seconds\*1000);**     }       **class RemindTask extends TimerTask {**         **public void run() {**             System.out.println("Time's up!");            **timer.cancel(); //Terminate the timer thread**         }     }         public static void main(String args[]) {         System.out.println("About to schedule Reminder task in 5 seconds");         new TimerReminder(5);         System.out.println("Task scheduled.");     } } |

Code-6.11: TimerReminder.java  
  
3. Build and run the project

* Right click **TimerReminder**project and select **Run**.
* Observe the result in the **Output**window. (Figure-6.12 below)

|  |
| --- |
| About to schedule Reminder task in 5 seconds Task scheduled. Time's up! |

Figure-6.12: Result of running UnTimerReminder application

### **(6.2) Schedule a repeating task**

1. Create a new Java project

* Select **File->New Java Project (Alt+Shift+N)**. The **New Java Project** dialog box appears.
* For the Project Name field, type in **AnnoyingBeep**as project name.
* Choose ‘Use default JRE (11….). Make sure you have set your default JRE set to version 11 and the appropriate compliance settings as well. Deselect ‘Create module-info.java file’ if already selected. Click Next.Click Finish.
* Observe that **AnnoyingBeep** project appears
* Right click on the project and select **File**->**New Class.**
* Enter in a package name of your choosing or use the default. Type in **AnnoyingBeep**as the class name. The main method stub should also be created.

The IDE generated **AnnoyingBeep.java** is displayed in the source editor window of STS IDE.

2. Modify the IDE generated **AnnoyingBeep.java**as shown in Code-6.21 below.  Study the code by paying special attention to the bold fonted parts.

|  |
| --- |
| import java.util.Timer; import java.util.TimerTask; import java.awt.Toolkit;  /\*\*  \* Schedule a task that executes once every second.  \* Beep every second.  \*/  public class AnnoyingBeep {     Toolkit toolkit;     Timer timer;         public AnnoyingBeep() {         toolkit = Toolkit.getDefaultToolkit();        **timer = new Timer();         timer.schedule(new RemindTask(),                 0,        //initial delay                 1\*1000);  //subsequent rate**     }        **class RemindTask extends TimerTask {**         int numWarningBeeps = 3;              **public void run() {**             if (numWarningBeeps > 0) {                 toolkit.beep();                 System.out.format("Beep!%n");                 numWarningBeeps--;             } else {                 toolkit.beep();                 System.out.format("Time's up!%n");                 //timer.cancel(); //Not necessary because we call System.exit                 System.exit(0);   //Stops the AWT thread (and everything else)             }         }     }         public static void main(String args[]) {         System.out.format("About to schedule task.%n");         new AnnoyingBeep();         System.out.format("Task scheduled.%n");     } } |

Code-6.21: AnnoyingBeep.java

3 Build and run the project

* Right click **AnnoyingBeep**project and select **Run**.
* Observe the result in the **Output**window. (Figure-6.22 below)
* Your machine should beep thrice.

|  |
| --- |
| About to schedule task. Task scheduled. Beep! Beep! Beep! Time's up! |

Figure-6.22: Result

### **Summary**

In this exercise, you have learned how to use Timer and TimerTask classes to schedule one-time or repeating tasks.

## **Homework**

1. The homework is to create a new Java project called **MyRunnableProject** as follows.  

* Create a class called MyCurrentDate that implements Runnable interface.
* The MyCurrentDate class displays the current date and time 10 times, with 100 milli seconds interval - use sleep() method for this interval.
* Create a class called MyMain, which contans main() method, in which 3 instances of MyCurrentDate threads are being run.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*