

MINISTRY OF EDUCATION AND RESEARCH



---

**TECHNICAL UNIVERSITY**  
OF CLUJ-NAPOCA, ROMANIA

---

# **FUNDAMENTAL PROGRAMMING TECHNIQUES**

## **ASSIGNMENT 5**

### **PROCESSING SENSOR DATA OF DAILY LIVING ACTIVITIES**

# 1. Requirements

Consider designing, implementing and testing an application for analysing the behaviour of a person recorded by a set of sensors installed in its house. The historical log of the person's activity is stored as tuples (*start\_time*, *end\_time*, *activity\_label*), where *start\_time* and *end\_time* represent the date and time when each activity has started and ended while the activity label represents the type of activity performed by the person: Leaving, Toileting, Showering, Sleeping, Breakfast, Lunch, Dinner, Snack, Spare\_Time/TV, Grooming. The data is spread over several days as many entries in the log *Activities.txt*, taken from [1-2] and downloadable at [http://coned.utcluj.ro/~salomie/PT\\_Lic/4\\_Lab/Assignment\\_5/](http://coned.utcluj.ro/~salomie/PT_Lic/4_Lab/Assignment_5/)

Write a program that uses functional programming in Java with lambda expressions and stream processing to perform the tasks listed in the table below. The results of each task must be written in a separate .txt file (each .txt file must be named according to the following template *task\_number.txt*, for example Task\_1.txt).

Task	Task Description
TASK_1	Define a class <i>MonitoredData</i> with 3 fields: start time, end time and activity as string. Read the data from the file <i>Activity.txt</i> using streams and split each line in 3 parts: <i>start_time</i> , <i>end_time</i> and <i>activity_label</i> , and create a list of objects of type <i>MonitoredData</i> .
TASK_2	Count the distinct days that appear in the monitoring data.
TASK_3	Count how many times each activity has appeared over the entire monitoring period. <ul style="list-style-type: none"><li>Return a structure of type Map&lt;String, Integer&gt; representing the mapping of each distinct activity to the number of occurrences in the log; therefore the key of the Map will represent a String object corresponding to the activity name, and the value will represent an Integer object corresponding to the number of times the activity has appeared over the monitoring period.</li></ul>
TASK_4	Count for how many times each activity has appeared for each day over the monitoring period. <ul style="list-style-type: none"><li>Return a structure of type Map&lt;Integer, Map&lt;String, Integer&gt;&gt; that contains the activity count for each day of the log; therefore the key of the Map will represent an Integer object corresponding to the number of the monitored day, and the value will represent a Map&lt;String, Integer&gt; (in this map the key which is a String object corresponds to the name of the activity, and the value which is an Integer object corresponds to the number of times that activity has appeared within the day)</li></ul>
TASK_5	For each activity compute the entire duration over the monitoring period. <ul style="list-style-type: none"><li>Return a structure of type Map&lt;String, LocalTime&gt; in which the key of the Map will represent a String object corresponding to the activity name, and the value will represent a LocalTime object corresponding to the entire duration of the activity over the monitoring period.</li></ul>
TASK_6	Filter the activities that have more than 90% of the monitoring records with duration less than 5 minutes, collect the results in a List<String> containing only the distinct activity names and return the list.

## 2. Deliverables

- A **solution description document** (minimum 2000 words, Times New Roman, 10pt, Single Spacing) with the structure specified in the **Lab Description** document.
- **Source files**
- **jar file** required for executing the application
- The required task files in .txt format in which the results obtained after executing the tasks have been written

The deliverables will be **submitted** as follows:

- Create a repository on **gitlab** with the name:  
***PT2020\_Group\_LastName\_FirstName\_Assignment\_5***
- Push the following: **source code files (push the code not an archive with the code), jar file, .txt files, documentation**
- Share the repository with the user **utcn\_dsrl**.

## 3. Evaluation

The assignment will be graded as follows:

Requirement	Grading
<b>Minimum to pass</b> <ul style="list-style-type: none"><li>• Object-oriented programming design</li><li>• Classes with maximum 300 lines</li><li>• Methods with maximum 30 lines</li><li>• Java naming conventions</li><li>• Basic documentation</li><li>• Implementation of TASK1, TASK 2, and TASK 3</li><li>• jar file - the application should permit to be run with the following command: <b>java -jar PT2020_Group_LastName_FirstName_Assignment_5.jar</b></li></ul>	5 points
TASK 4	1 point
TASK 5	2 points
TASK 6	1 point
Quality of the Documentation	1 point

## 4. Bibliography

[1] Ordóñez, F.J.; de Toledo, P.; Sanchis, A. Activity Recognition Using Hybrid Generative/ Discriminative Models on Home Environments Using Binary Sensors. Sensors 2013, 13, 5460-5477.

[2] Available online at

[https://archive.ics.uci.edu/ml/datasets/Activities+of+Daily+Living+\(ADLs\)+Recognition+Using+Binary+Sensors](https://archive.ics.uci.edu/ml/datasets/Activities+of+Daily+Living+(ADLs)+Recognition+Using+Binary+Sensors)

[3] <https://docs.oracle.com/javase/tutorial/java/javaOO/lambdaexpressions.html>

[4] <https://docs.oracle.com/javase/tutorial/java/javaOO/methodreferences.html>

[5] <https://www.oracle.com/technical-resources/articles/java/ma14-java-se-8-streams.html>

[6] <https://winterbe.com/posts/2014/07/31/java8-stream-tutorial-examples/>