

### **Lab task 3 - LISTS**

#### **Michelin Guide**

##### **Problem description**

The Michelin Guide is preparing the top of the most visited restaurant sites (cafes, restaurants, and bars) in Castilla-La Mancha. To do this, it has contacted the Castilla-La Mancha Statistics Service (SECLM) which has provided statistics on the most visited restaurant sites during 2021.

Based on the open data provided by the SECLM in the `datos.gob` portal, the Michelin Guide has preprocessed the data to keep certain information related to the establishments. This file with the preprocessed information called "*restaurants.csv*" is available in the Campus Virtual, and the information is saved as follows:

`<EstablishmentType>;<EstablishmentName>;<EstablishmentPostalCode>;<Town>;<Province>;<TotalVisits>`

José Luis Michelin, the technician in charge of obtaining the top, has requested the help of the Data Structures students to obtain the different rankings<sup>1</sup> of the restaurant sites through the following lists:

- A list called "Personal Top" that will be customizable and that will include the establishments of a province (ALBACETE, CIUDAD REAL, CUENCA, GUADALAJARA, TOLEDO) and of a specific type (Restaurants, Bars, Cafes) selected by the user through the keyboard. This list will be of variable length depending on the province and establishment type indicated by the user through the keyboard.
- A list called "Top 10 CR Awards" that will provide the top 10 of the most visited sites in the city of Ciudad Real, ordered from the most visited to the least visited.
- A list called "Best Cafe Awards" of the 10 most visited cafes in Castilla-La Mancha. For the latter case, the list does not have to follow a specific order.

##### **To do**

In this scenario, José Luis Michelin asks for a Java program that does the following:

1. Select the ranking to consult, what will be done by entering an option by keyboard in a menu.
2. Show the list with each of the tops, with information on the name and type of the establishment, as well as the province in which it is located and the number of visits.

##### **Technical requirements**

- The original data file will not be modified.
- An *Establishment* class will be used, which will correspond to each restaurant site.
- It is recommended to use the *Iterator* class to iterate through the list of restaurants.
- Every class that will be generated should be in a different file, including the possible classes that define exceptions. Try to modularize the code properly.
- Each class/method in the program should include appropriate internal documentation to make it fully understandable. In "Campus Virtual" you can find information to write internal doc (see Lab section).
- In order to use anti-plagiarism and semantic similarity detection systems, source code files may not contain personal data about students such as name, email... Instead, the authors and the group will be identified in the comments of the code with the initials of the students and the assigned group code.

---

<sup>1</sup> For the ordering implementation, it is recommended to use the `Java Collection.sort()` function.

**Additional improvements**

- Generation of a .jar file with all classes of the program
- Generation of a batch process file .bat that executes the program using the mentioned library as .jar file.

**Running and delivery rules**

- The project must be carried out by each of the work groups that have been previously formed in practical classes. It cannot be done individually.
- The delivery and evaluation will be made on the day previously indicated in the Campus Virtual. One of the members of the group will send a compressed file with all the classes that compose the program. Individually, you must also submit the participation template, which will be available in "Campus Virtual".
- For the practice to be evaluated, the program must compile and run correctly.
- Remember that this activity is mandatory to pass the subject.