



SURNAME, NAME:

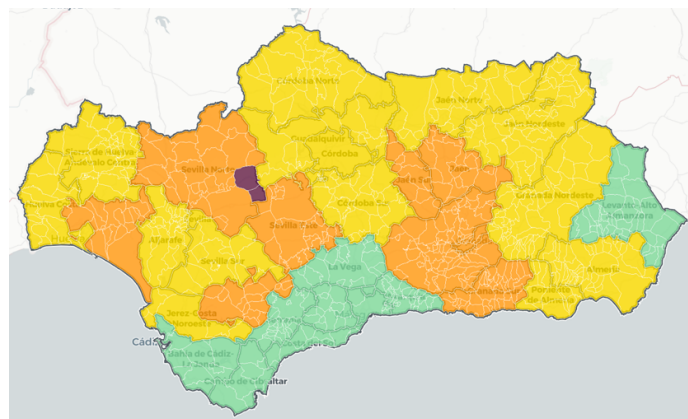
DEGREE

MACHINE

## NOTES FOR THE EXERCISE:

- If you do not meet the requirements to take advantage of the continuous assessment mode, in addition to this practical exercise, you must take an additional theoretical exam. If you have not done it yet, please, contact the teaching staff.
- Add a comment with your name (surname followed by first name) and degree at the beginning of each file to be submitted.
- **You are allowed to:**
  - Consult the class notes (CV), the API (local network server), and the quick guide to the API (CV).
  - Add private methods to classes.
- **You are not allowed to:**
  - Exchange documentation.
  - Add non-private methods to classes.
  - Add variables or constants to classes.
  - Modify the visibility of the variables, constants, and methods.
  - Modify the supplied code.
- Once the exercise is completed, compress the `src` folder of your project and upload it to the corresponding task in the virtual campus.
- In the evaluation of your exercise, the clarity of the code and the correct choice of data structures, as well as the design decisions, will be taken into account.

We want to develop a project to maintain information on the situation of the COVID-19 epidemic in a certain region, so that each province in it will be organized according to its health districts. For instance, Andalusia is a region with eight provinces (Málaga, Granada, Sevilla, ...), and each of them has several health districts (Axarquía, Valle del Guadalhorce and Costa del Sol are some of the health districts in the province of Málaga). With such goal in mind, we will develop a project `prCOVIDMap`, with classes `COVIDException`, `HealthDistrict`, `COVIDMap`, `PerimeterClosureInfo`, and `PopulationInfo`, and interface `COVIDInfo` in the `covid` package. Additional test classes will be defined in the default package. A test class `TestCOVIDMap` and a text file `COVIDData-dist.csv` are provided with information on the health districts of the Andalusian provinces.



- 1) **(0.25 pts.)** The `COVIDException` class represents *unchecked* exceptions to handle the exceptional situations that may occur in the following classes.
- 2) **(2.50 pts.)** The `HealthDistrict` class has a name (`String`), the population it serves (`int`), and the number of positive COVID cases accumulated in the last 14 days (`int`). The definition of the class will include:
  - a. A constructor that has as arguments the initial values of the previous attributes (in that same order). You must validate all the attributes, generating a `COVIDException` if the population is zero or negative, or if the diagnosed cases are less than zero.

- b. Three methods that allow obtaining the values stored in the attributes (getters), as well as an additional method that allows modifying the number of accumulated cases in the last 14 days (setter). In this case, if the provided value is a negative number, a `COVIDException` will be thrown. Moreover, the class will have a method
 

```
public int accumulatedIncidence()
```

 that returns the number of positive cases for every 100.000 inhabitants.
- c. Two health districts are considered equal when they have the same name, case insensitive.
- d. The class will provide a natural order, so that a sanitary district is considered smaller than another when the name (without distinguishing between upper and lower case) is lexicographically smaller.
- e. The textual representation of a sanitary district will be given with the format:
 

```
District(district-name, number-of-COVID-cases-in-the-last-14-days)
```

 For example,
 

```
District(Axarquía, 184)
```

- 3) **(1,00 pts.)** The `TestHealthDistrict` class will test the previous two classes by creating health districts with the following information:

```
La Vega: 110176, args[0]
Axarquía: 170141, args[1]
Valle del Guadalhorce: 156298, args[2]
Costa del Sol: 560785, args[3]
Málaga Distrito: 633521, args[4]
Serranía: 54999, args[5]
```

Where `args[ ]` is the argument of the main function. That is, it will create health districts with the numbers of cases provided as argument of the main function.

The erroneous situations that may arise must be addressed by issuing differentiated error messages (`System.err`). These situations will include:

- Not enough values are provided as arguments to the main function.
- Some of the entered values do not correspond to an integer number.
- Some of the entered values are negative.

- 4) **(3.75 pts.)** The `COVIDMap` class will store the information of the COVID map of a region in a structure in which each of its provinces (`String`) is associated to the set of its health districts (`SortedMap<String, SortedSet<HealthDistrict>>`). The `COVIDMap` class will also include an attribute with the name of the region (`String`). Both the provinces and the districts must be kept in order, according to their natural order. The class will also provide:

- a. A constructor that receives as arguments the name of the region and the name of a file with information about provinces and health districts (see exercise 6.b below).
- b. Methods

```
public void readData(String filename) throws FileNotFoundException
public void readData(Scanner sc)
```

that add the information stored in the file named `filename` and from the scanner `sc`, respectively. This information will be organized by lines with the following format:

```
Health-district(Province):population:number-of-cases
```

For example,

```
Axarquía(Málaga):170141:184
```

Note that the delimiters will respond to the expression "`[ ( ) : ]`". If a health district appears more than once, the information corresponding to the first appearance will be added (see exercise 6.c below). If a line does not have the correct format, it will be ignored, and the processing will continue with the next line.

- c. The method
 

```
public void addDistrict(String prov, HealthDistrict hd)
```

 adds the health `hd` to the province `prov`. In case the district was already associated with the province, nothing is added. In other words, if there was already a district equal to the new one, the new one will not be added.
- d. Methods
 

```
public String getRegion()
public Set<String> getProvinces()
public Set<HealthDistrict> getDistricts()
```

that return, respectively, the name of the region, the set of provinces, and the health districts stored on the map.

e. The method

```
public int incidenceOfTheProvincia(String prov)
```

returns the accumulated incidence per 100,000 inhabitants of the province prov. To calculate the incidence of a province, the total population and accumulated cases of all districts of the province must be obtained. And, with both values, then calculate the incidence corresponding to 100,000 inhabitants. The incidence of a province will be 0 when there is no population or the province is not in the map.

f. Methods

```
public void printMap(String filename) throws FileNotFoundException
public void printMap(PrintWriter pw)
```

that allow showing the information of the COVID map, either on a file or through a PrintWriter, respectively. The information will include an initial line with the name of the region, in capital letters, and lines for each province and its health districts, indented with a tab. For example:

ANDALUCIA:

Almeria

```
District(Almeria Distrito, 246)
District(Levante-Alto Almanzora, 119)
District(Poniente de Almeria, 157)
```

Cadiz

```
District(Bahia de Cadiz-La Janda, 542)
District(Campo de Gibraltar Este, 93)
```

...

5) **(2,5 pts.)** To be able to extract diverse information from a COVID map, we will define the following interface and classes:

a. The COVIDInfo interface with a method

```
Set<String> getInfo(COVIDMap mapa)
```

that will return a set of names (provinces or districts) extracted from the map that is passed as argument, fulfilling certain conditions to be set in the classes that implement the interface.

b. The PopulationInfo class that implements the COVIDInfo interface, and provides:

- Two instance variables that establish a population range.
- A constructor to which the minimum and maximum values of that range are provided.
- An implementation of the getInfo(COVIDMap) method, which returns the set of names of districts in the map whose population is between the established limits.

c. The PerimeterClosureInfo class that implements the COVIDInfo interface, and provides:

- An implementation of the getInfo(COVIDMap) method, which returns the set of provinces whose accumulated incidence exceed 500 cases.

d. Add a method

```
public Set<String> getCOVIDInfo(COVIDInfo info)
```

to the COVIDMap class that returns the set of province or district names determined by the criteria defined in the info object that is passed as an argument.

