|  |
| --- |
|  |
| Assignment 1 documentation |
| Advanced Java |

## Contents

[**1.Introduction** 2](#_Toc93591208)

[**2.Solution** 3](#_Toc93591209)

[**3.Other information:** 4](#_Toc93591210)

# **1.Introduction**

The reason for making this documentation is to provide more information on the solution of the project problem in assignment 1. The problem to be solved in the assignment is described as : “Your application should read this data into an appropriate data structure and then process it according to command line parameters.” Another thing is that the command line should be given in a proper format by the user and that format should be : “-file pathToFile -param1 value1 -param2 value2 ... -paramN valueN.”, where the “param” fields from param1 to paramN are stat, limit, by and display.

After “-file” should be the path where the file is stored and then the values from val1-valN should be as follow: “after the first parameter stat the values should be either min(for minimum) or max(for maximum)”, “after the second parameter limit should be the number of data to be retrieved which is a number from 1-100”, “after the third parameter by should be the value by which the items should be compared and the possibilities are: NC(stands for new cases), NCS(new cases smoothed), ND(new deaths), NDS(new deaths smoothed), NT(new tests), NDPC(new deaths per case or ND/NC).”, and lastly “the forth parameter display is which data should be displayed after they are sorted based on the by parameter and how many data are selected based on the limit parameter and the possible values are: DATE, COUNTRY, CONTINENT.”. As we will mention this later on as well the command line that the user will provide should be in the correct format. The data will be retrieved through streams and lambdas. Lastly as a data structure we should use classes/modules for the data to be stored and as I will also mention below I have created 2 such classes and then in the main class I have instantiated a list of both of them of type arraylist which will store the data to be retrieved from the file.

# **2.Solution**

The project is composed of 3 classes. For the data structure part of the project I have created 2 classes, one called “Country” which holds the fields for the Iso Code, continent, country (which is denoted as “location” there) which are of type String and a list of “CovidData” which is denoted as “datas” and the other one called “CovidData” which has values about the covid data like new cases, total cases, new cases smoothed, new deaths, new deaths smoothed, total deaths, new tests, total tests and stringency index, some general information like median age, population and reproduction rate all of which are of type double, has a field for the date and lastly an object of type country. Both of them have their specific constructors and the getters and setters for their fields (as for the getters and setters they have those names because of how the variables were called before being changed). Then there is the main class(the class where the main method is and where the logic part happens) which is called “retrievingFileData”. In the main class I ask the user for the command and after I get the command from the user I extract the necessary information like the path of the file, the stat file (so either min or max), the numbers of rows to be gotten which is denoted by “limit”, the field from which the data have to be sorted by and the data to be displayed from that command. If the command is not written in the right format an error will occur. If the command is in the right format and the necessary information are extracted, I have created a list of type arraylist of both “CovidData” and “Country” objects and then populated them by taking the values from the file. After this the method “retrievingData” is called which inside also calls the method “sort”. As the name suggests the method “sort” sorts the data inside the list based on the field they have to be sorted (so they are sorted through the value of the “by” parameter mentioned above”. After the data in the list are sorted the method “retrievingData” retrieves the necessary data depending on the fields supplied (like the number of data or the data to be shown, as well whether they should be the top or bottom data from the sorted list). Besides them there are 2 more methods called “tryParseDouble” which tries to parse the strings retrieved to type double and if it cannot it returns the value 0 (like the case when there is an empty field), and “tryParseDate” which parses the string as a date.

# **3.Other information:**

When the data are retrieved into the list the rows which are missing the “continent” field are filtered out. As the case when we try to get a number of data and see at least 1 repetition of the data we are trying to retrieve, I have decided to add another data to show that it is not the same occurrence:

When I try to retrieve the date and there is a repetition of at least 1 of them, I also display the country field for all the dates to show that it is not the same occurrence.

When I try to retrieve the country and there is a repetition of at least 1 of them, I also display the date field for all the countries to show that it is not the same occurrence.

Lastly when I try to retrieve the continent and there is a repetition of at least 1 of them, I also display the country field and the date field for all the countries to show that it is not the same occurrence.

Besides that for the negative values on the “new cases” field that we discussed I saw there were more than 1 such negative values and that they also reflected on the “total cases” field so decided not to filter them out like I did with the data that had an empty “continent” field and I decided not to return 0 in their case either but to return their true value(as this could easily be done in the tryParseDouble method with only an if-else statement the way I returned 0 for those empty fields).

Something else very important to be mentioned is that there was another method in the main class called “tryParseInteger” which did the same thing as the “tryParseDouble” and had the same structure, just that instead of returning a double value it returned an int value, but the problem was that sometimes when the string was an integer number which could be parsed it returned 0 and that is why I made the number fields of type double and only have the “tryParseDouble” method there. One way how this could be fixed is by making another such method which in this cases took as a parameter a double number instead of a String and returned an int value and everywhere we wanted an int value we could use this method with the value parsed by the “tryParseDouble” method as parameter, but I felt like it was not really necessary as we already had the “tryParseDouble” method which basically got the number that we needed.

Above I have mentioned that the command has to be written right in the right format otherwise an error will occur. Another way this could be done to make it even easier and less prone to errors would have been to ask and get all the values about our parameters one by one, after each other (this way the user would know what he has to input and I wouldn’t have to use substrings to get the necessary string). The reason why I did it with one full command like this that has to be of the given format in order for it to work is because that is how I understood that it had to be done.

**Worked by:** Enis Sejdini