**SOFTWARE CONSTRUCTION 2**

**LAB REPORT 3**

**BY**

**EMMANUEL DURU**

**MATRIC NUMBER: 2502665**

**GROUP 32**

**PROFESSOR WOLFGANG RENZ**

**(B - list by country)** Provide a method ***public void listByCountry()*** that iterates through your table and first prints the country before printing the full information for each power plant. Therefore, you have to add a ***public String getCountry()*** method to your ***PowerPlant***class.

**SOLUTION:** I created the method of ***public void listByCountry()*** of which the result is stated in the class.

**(B - power by country)** Provide a method ***public HashMap<String, Double> powerByCountry()*** that stores for each country (key) the total power of power plants in your table as the value. Therefore, it iterates through all power plants and adds for each country all the power plant capacities to the double value. Thus, you have to check for each power plant, if the country is already contained as a key in the map. If yes, you just add the power plant capacity to the power that is already in the value for that key. Otherwise, you just put the key-value pair for that power plant to the map. (For help, you refer to the implementation of the WordCounter.)

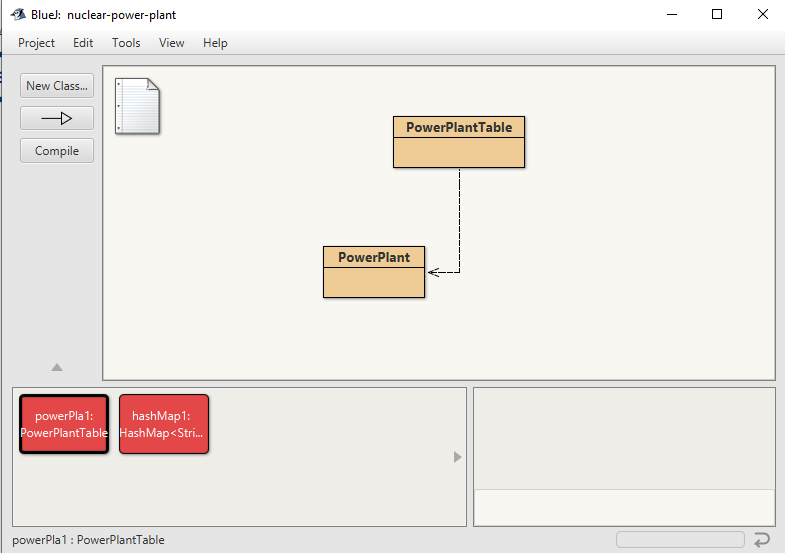
**SOLUTION:**

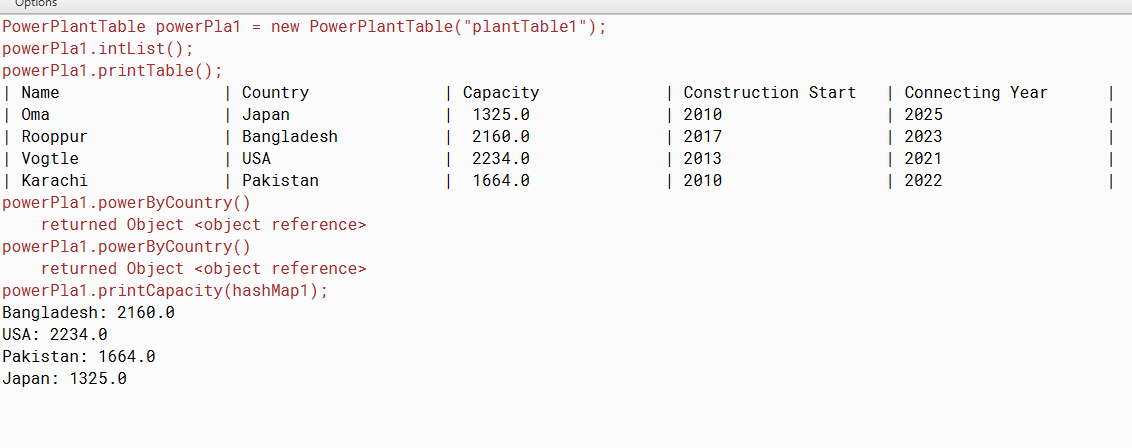
I generated a method of the ***public HashMap<String, Double> powerByCountry()*** which endeavored to iterate through all power plants capacity to double its value as given. Furthermore I checked for each power plant of which the country is contained as a key in the map.

**(B - printCapacities)** Provide a method ***public void printCapacity(HashMap<String, Double>)***that prints the table of countries together with the total power of power plants of your list in each country. Test the combination of this and the former method and report that test. (You might keep the HashMap as an instance variable, but then you need to consider Topic 2 of Lab 2 PostProcessing and Exercises.)

**SOLUTION:**

I instantiated the powerPlant and called the method Hashmap <String, Double> powerByCountry which returned the HashMap< String, Double>. Thus I called the print capacity method to give the results stated in the screenshot below.





**(B - map by country, challenging)** Recall from the lecture: The table of contents of a book corresponds to a map with a keyword as a key and the set of page numbers as values, i.e. ***HashMap<String, HashSet<Integer>>***Thus it can contain e.g. an entry like ("instance variable",{15, 36, 157}). Now, provide a method ***public HashMap<String, HashSet<PowerPlant> mapByCountry()*** that adds all the power plants to the map. The keys are the countries and the value the set of power plants in that country  Therefore, you have to check for each power plant, if the country is already contained as a key in the map. If yes, you just add the power plant to the set that is already in the value for that key. Otherwise, you have to instantiate a new set, add the power plant to it and put the key-value pair to the map.

**SOLUTION:** i checked for each power plant, if the country is already contained as a key in the map. If yes, you just add the power plant to the set that is already in the value for that key the result of the given challenge is stated below and well-illustrated in the code .

