**SOFTWARE CONSTRUCTION 2**

**LAB PREPARATION**

**BY**

**EMMANUEL DURU**

**MATRIC NUMBER: 2502665**

**GROUP 3**

**PROFESSOR WOLFGANG RENZ**

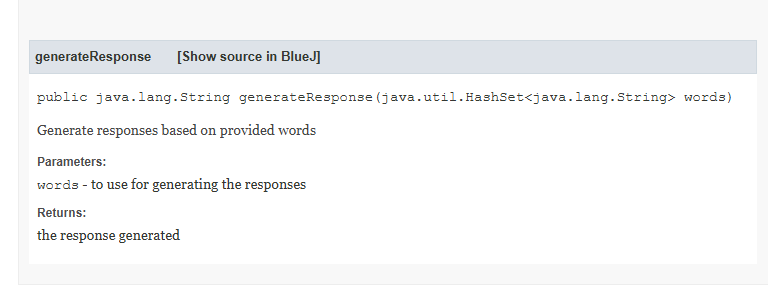
* + **(B - getInput)**  Compare the class ***InputReader***in the two bluej projects *tech-support2* and *tech-support-complete* and answer the following questions:
  + On what type of Objects are the methods trim() and toLowerCase() applied? Read their description in the JavaAPI and put it in your report.
  + What are the differences between the HashSet that is returned by the getInput method and an ArrayList that could have been used as an alternative

**SOLUTION:**

* + 1. The methods, **trim()** and  **toLowerCase()** are applied to String objects. The trim method eliminates all leading and trailing spaces while the toLowerCase method converts the string to a lower case.
    2. The HashSet is a set or collection of distinct objects, which means an object cannot exist more once, while an ArrayList can contain an object multiple times.

**(B - generateAllResponses)** Compare the class ***Responder***in the two bluej projects *tech-support2* and *tech-support-complete*, write a comment for the method generateResponse and generate the javadoc for it by switching in the editor from "source code" to "documentation". Program a new method ***public String generateAllResponses(HashSet<String> words)***that returns the concatenated String for all responses to keyword matches found. Test this function by calling it from the ***SupportSystem*** and add this test to your report.

SOLUTION:



**(A and B - findAllMatching)** Program, in the class ***MusicOrganizer***of the bluej projects *music-organizer-v4*, a new method ***public HashSet<String> findAllMatching (String searchString)*** that returns the set of filenames for all matches found. Write a comment for that method and generate the javadoc for it. Test and document this function similarly as you did for ***findMatching***.

SOLUTION:

MusicOrganizer musicOrg1 = new MusicOrganizer();

musicOrg1.addFile("Food");

musicOrg1.addFile("Good Food");

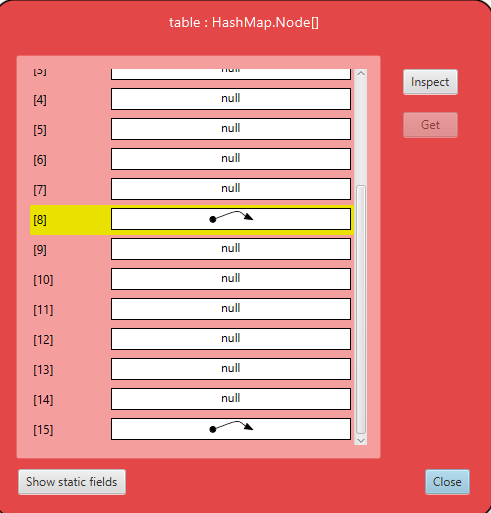
musicOrg1.addFile("Food is Life");

musicOrg1.addFile("Sister");

musicOrg1.addFile("Sweet mother");

musicOrg1.findAllMatching("Food")

returned Object <object reference>

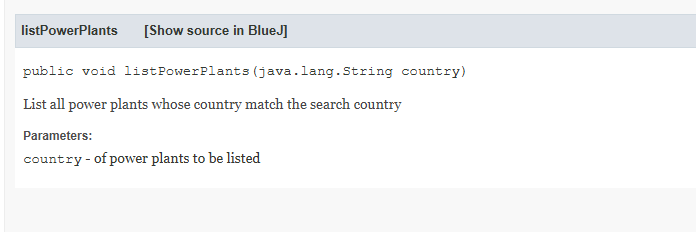


**(A and B - word counting)** Study the class ***WordCounter***of the bluej project tech-support-analysis (see section 5.10 in the book), look up the ***getOrDefault*** method of the class ***HashMap*** in the java API and explain the way it works in short words.

SOLUTION:

The getOrDefault method takes 2 arguments, one is the key with which the value is to be searched for, the second is a default value that would be returned if the key doesn’t match any value

**(B - list power plants)**  Add, in your initList() method (Lab2) more power plants from several countries taken from the Wikipedia list. Provide a method ***public void listPowerPlants(String country)*** that prints all ***PowerPlant*** instances located in the country to the terminal. Write a comment for that method and generate the javadoc for it. Test this function for your filled table with different countries and report the tests.



powerPla1.listPowerPlants("USA");

| Vogtle | USA | 2234.0 | 2013 | 2021 |

powerPla1.listPowerPlants("Pakistan");

| Karachi | Pakistan | 1664.0 | 2010 | 2022 |

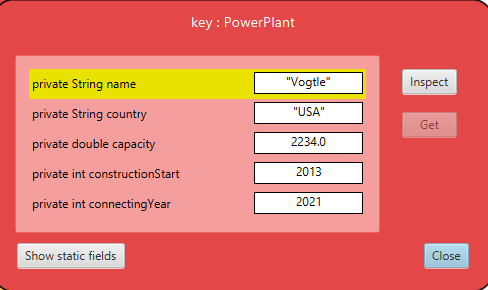
* + **(B - find power plants)** Provide a method ***public HashSet<PowerPlant> findPowerPlants(String country)***that adds all ***PowerPlant*** instances located in the country to a set that it returns. Write a comment for that method and generate the javadoc for it. Test this function for your filled table with different countries by using the CodePad and dragging the returned HashSet to the ObjectBench so that you can use the inspector for the returned HashSet. Add this test to your report.

SOLUTION:

PowerPlantTable powerPla1 = new PowerPlantTable("Plant1");

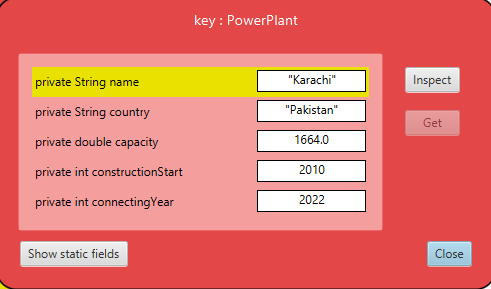
powerPla1.intList();

powerPla1.findPowerPlants("USA")

returned Object <object reference>

powerPla1.findPowerPlants("Pakistan")

returned Object <object reference>



powerPla1.findPowerPlants("Japan")

returned Object <object reference>

