

LAB ASSIGNMENT 5

DUE TO NEXT LAB

1. Open the basic **tech-Support1** project that we used in the lecture.
Expand it to give random responses as we did in class.
 - Which fields and methods are needed?
 - Write a method that uses an `ArrayList` of `Strings` to generate random responses.
 - The method should return a random response from the `ArrayList` when called.
2. Advance the `Responder` class using a `HashMap` to store possible key terms as keys (to be later identified in user input) and corresponding system responses as values.
 - Implement a method to fill the `HashMap` with key terms and responses of your choice.
 - Enhance the `generateResponse()` method, so that it takes a term as a parameter and returns the associated system response using your `HashMap`.

E-Learning

Objective:

Learn about the Java `HashSet` class, understand its characteristics, and compare it with the `ArrayList` class.

Learning goals:

- a. Gain proficiency in using `HashSet` to store and manage collections of unique elements.
- b. Understand how `HashSet` differs in behavior from `ArrayList`, particularly in terms of element uniqueness and order.

Practical application:

In the current version of the tech-support project the user input is returned by the `getInput()` method as a *single* string ("why is my internet so slow?"). In the following you should change this to a new version in which the `getInput()` method of the `Reader` class returns the user's input as a set of words ("why","is","my","internet"...). Towards this goal work on tasks 3-7:

3. Research how to cut a single string containing a whole sentence into a set of word.

- Read about the method `split()` from class `String`. Pay attention to its return type!
 - https://www.youtube.com/watch?v=NY33-ErTvjw&list=PL9HfA4ZKbzintNeji09vJxll_RiOn9eB&index=5&t=2948s
4. Find out how to store those words in a HashSet.
- Read about the Java `HashSet` class.
 - Identify similarities and differences between `HashSet` and `ArrayList`.
 - https://www.youtube.com/watch?v=NY33-ErTvjw&list=PL9HfA4ZKbzintNeji09vJxll_RiOn9eB&index=5&t=2846s
5. Based on tasks 3 and 4: Improve the `getInput()` method to return a `HashSet` containing the terms from the user's input.
6. Revise your `generateResponse()` method in the `Responder` class to accept a `HashSet` parameter.
- The method should iterate through each term in the `HashSet` and verify if any of these terms are present as keys in the previously created `HashMap` (see 2).
 - If found, return the associated response.
 - If no specific response is found, generate a random response from your `ArrayList` (see 1).
7. Update your while loop in the `start()` method of your `SupportSystem` class, to align with the recent changes made in the other classes.
8. Watch the video sequence about Java documentation. Apply `JavaDoc` to document the methods you have worked on.
- https://www.youtube.com/watch?v=NY33-ErTvjw&list=PL9HfA4ZKbzintNeji09vJxll_RiOn9eB&index=5&t=3420s
9. Add on: How can you deal with punctuation marks? What if there is more than one space between words? Can your tech support system deal with this?
10. Add on: Ensure that the same default response is never repeated twice in a row.

11.Add on: When no word is recognized, use other words from the user's input to pick a well-fitting default response: for example, words such as "why", "how", and "who" might be useful.

12.Add on: (For the bored) There are often synonyms that should provoke the same response. Can you figure out how to store only one response even if there are a number of different words that trigger this response?