

Teacher Guide: Mental Health Project

Project Goal

The goal of this project is to create a real-world scenario where students develop an app to match people with appropriate mental health resources.

Project Overview

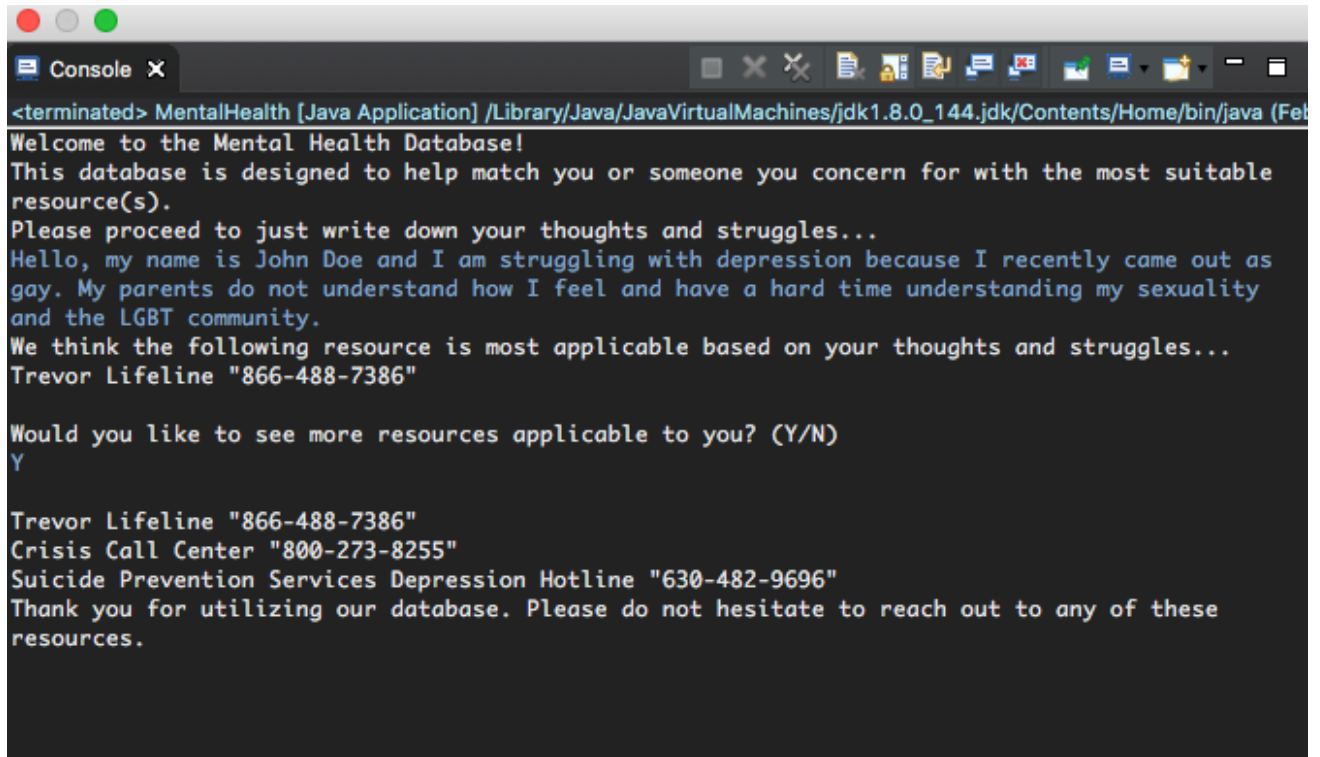
This project is meant to be a relaxed project intended to let students review topics they have learned throughout the year, in addition to exposing them to file readings. The task is to read a provided .txt file and load an arrayList of MentalHealth objects. After reading the file, the students will write the main class for user interaction and the methods for the object to manipulate each line of the file. This project reviews topics such as classes, objects, public vs. private methods, for-loops, indexes, substrings, parameters, and much more.

Teaching this Project

This is a 3-5 day project. Students can allocate their time as they wish, and should work in groups individually or with a partner. Begin by briefly introducing the project on the first day (15 minutes, using the intro slides), and then let students explore. This project is not intended to be graded for accuracy -- as long as students put in effort into some aspect of the project, they should receive full credit. Complete directions that students can follow are given in the student guide, which should be clear.

Student Guide: Mental Health Database

Your goal will be to create a personalized database for users to input message and be given personalized resources specific to their needs as indicated through their text.



```
<terminated> MentalHealth [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_144.jdk/Contents/Home/bin/java (Feb
Welcome to the Mental Health Database!
This database is designed to help match you or someone you concern for with the most suitable
resource(s).
Please proceed to just write down your thoughts and struggles...
Hello, my name is John Doe and I am struggling with depression because I recently came out as
gay. My parents do not understand how I feel and have a hard time understanding my sexuality
and the LGBT community.
We think the following resource is most applicable based on your thoughts and struggles...
Trevor Lifeline "866-488-7386"

Would you like to see more resources applicable to you? (Y/N)
Y

Trevor Lifeline "866-488-7386"
Crisis Call Center "800-273-8255"
Suicide Prevention Services Depression Hotline "630-482-9696"
Thank you for utilizing our database. Please do not hesitate to reach out to any of these
resources.
```

Project Guidelines

This project's goal is to allow you the creative freedom to explore any aspect of the project you find interesting. For a base implementation, you only have to implement the given methods. This is to ensure that the program will run correctly. However, we encourage you to explore and go beyond the given methods to increase user interaction and customization.

Grading

This project should be **stress free**. The goal of this is to let you explore aspects of the project you find interesting. Of course, you should implement at least the basic methods, but after that, you can spend your time on any of the following:

- Optimizing the .txt file to allow for chains of words (i.e. "teen pregnancy" vs. just "teen" and "pregnancy" as two separate keywords)
- Creating a larger, more diverse database of resources to allow for optimum user accessibility.
- Prevent printing the top resource twice in both the top resource and additional resource section.

- Print the additional resources in order of occurrences to allow the user to see the best resources all in order.
- Print multiple resources as the best, if more than one resource occurs the same maximum times.

This project will be graded on the effort you put into it. As long as you're working and exploring aspects of the project you're interested in, you'll receive full credit. If you go above and beyond, there might even be an opportunity for extra credit.

Implementation

Before you start writing code, we recommend that you explore the .txt file and the implementation of both classes to get a better feel of how the two classes interact with the .txt file.

We will be giving you some of the starter code to begin with. This will include the method names and parameters needed in both the main and object class. In the object class, however, you will need to declare a few instance variables. In both programs, you will be writing the actual methods with any other local variables you think you will need.

We're not going to give you any answers to how to exactly implement things, as there is no one right algorithm! That's what makes this project exciting. To build your methods, some tips:

Going about MentalHealth.java (main class)

This main class is the class that controls the user input and interaction. It will be calling methods with the object class.

- readFile()
 - The file reading method is given to you. However, it is your job to read in the file and convert each line to a MentalHealth object. Hint: String line is one line of the file. Use line to load the objects.
 - Think about when the break statement should occur. When line equals what?
- inputMessage()
 - This is the method to allow the user to actually type in the message. It should return the message the user wrote.
- readMessage(String message)
 - This method takes in the user message as the parameter and will work through each word in the message to look for keywords. Notice that we introduce to you a new variable called a StringTokenizer. You probably are not familiar with this and thus we will provide some useful methods and explanations. First, a StringTokenizer takes a string and a delimiter as its two parameters. The string is what the tokenizer is looking through and the delimiter is the character the tokenizer will break the string by. This will allow you to look through each word of the message.

- Two useful methods will be `.hasMoreTokens()` that will return a boolean indicating if any tokens are left and `.nextToken()` that will return the string of the next token.
- This method as a whole should look through each word, check if that word is a keyword of any of MentalHealth Object, and if it is, increment the occurrence of that object. Ignore punctuations and be case-insensitive.
- `printResources()` and `askForMoreResources()`
 - `printResources()` should look for the MentalHealth object with the most occurrences and print out the **first** resource with the most occurrence. You may extend this and make it print with all the resources with the max occurrences.
 - `askForMoreResources()` should ask the user if they would like to view other resources applicable to them. Depending on their answer, you must print the rest of the resources or not.

Going about MentalHealthObject.java (object class)

This is the object class where the object gets its state and behavior.

- Declare any instance variables needed.
- `MentalHealthObject(String fileLine)`
 - The constructor that sets all variables to its initial declaration.
- `containsKeywords(String inputWord)`
 - This method checks if the given word is a keyword of the object. This method should ensure to ignore any punctuations and be case-insensitive.
- `getOutputLine()`
 - This method should return a string that is printable and friendly for the user to see.
- `incrementOccurrences()` and `getOccurrences()`
 - Setter and getter methods.
- `loadKeywords(String line)`
 - This method takes a line of the file and loads its keywords to the current MentalHealth Object.

Going about mentalhealthdatabase.txt (text file/database)

This is the text file containing the database of mental health resources.

- Observe and analyze the file. Take note of where certain indexes of characters are and how the file is formatted in terms of the name of the resource, its hotline, and its keywords.