IT-145

Final Project

Process Documentation

David Gameiro

21 April, 2019

Problem Statement/Scenario:

I have been contracted by the Atlanta Zoo to build a system to help monitor their animals and habitats. This is used to help know the activities of the animals and to monitor their living habitats. There are a few requirements that were asked for it to contain. It should have a way for the user to choose whether to monitor an animal, a habitat, or to exit. Then, it should be able to read a provided file and display a list of animals or habitats that are available to monitor. With the they user is then able to define more specifically what they would like to do. Displays all of the information for what the user is asking for. In addition, some notes are urgent and require a dialog box to inform the user and at the same time removing the symbol that allows the system to know what is urgent. Lastly, there should be a way for the user to look at more than one area.

Overall Process:

First, I made a way for the user to leave the system before looking up anything. Next, I created a way that the user may repeatedly look up information without having to exit and start all over. With that in place I was able to allow for the user to pick if they would like to monitor an animal or a habitat. In order for me to ask them more specifics I had to display a list of all of the animals or all of the habitats that were available to monitor. I created a separated class that would handle all of the digging for the necessary information. With that I made it so that it would go through the file to look for what the user was looking for and all of the information that corresponded with it. Some of the information had some urgent message that needed to me fixed and so I had to make a dialog box pop up when the system noticed \*\*\*\*\* in the beginning of the line. This dialog made it so that the user had to read the message and press OK to close the message and to return to the system. Lastly, the user had a choice to look for more information or to exit the system completely.

Pseudocode:

function ZooMonitoringSystem():

Gather user input

Standardize user input

If user would like to exit

Else

While user input not exit

If user input animal

Open animal file

While to read every line in file

Read all lines starting with Details

Print out list of all lines with Details

Ask user which animal they would like to monitor

Gather from Animal class and animalMont method

Standardize user input

While to read every line in file

Read all lines for “Animal – “ and the animal they chose

If line contains “Animal –“ and the animal they chose

Print that line

If line contains \*\*\*\*\* display dialog box

Else print line as is

If line contains \*\*\*\*\* display dialog box

Else print line as is

If line contains \*\*\*\*\* display dialog box

Else print line as is

If line contains \*\*\*\*\* display dialog box

Else print line as is

Return Information to main class

Else user input habitat

Open habitat file

While to read every line in file

Read all lines starting with Details

Print out list of all lines with Details

Ask user which habitat they would like to monitor

Gather from Habitat class and habitatMont method

Standardize user input

While to read every line in file

Read all lines for “Habitat – “ and the animal they chose

If line contains “Habitat –“ and the animal they chose

Print that line

If line contains \*\*\*\*\* display dialog box

Else print line as is

If line contains \*\*\*\*\* display dialog box

Else print line as is

If line contains \*\*\*\*\* display dialog box

Else print line as is

Return Information to main class

Ask user if they would like to monitor an animal, a habitat or exit

Display exit statement

This Pseudocode differs greatly from the one that was submitted earlier because the other one was not as specific and narrow as this one is. This Pseudocode follow greatly to the flow of how the code is written. It is also indented to show a work flow and also to show information that are at the same level.

Methods and Classes:

The Animal class was used with the animalMont method to gather the information about each animal from the file. And Habitat class was used with the habitatMont method to gather the information about each habitat from the file.

Error Documentation:

The biggest problem that I was having was being able to read the files for the exact information that I was looking for. It took me some time to be able to read the list of details at the beginning of the file and to print them out for the user. Also, being able to pull up the necessary information about a specific animal or habitat gave me some issues. But I was able to get it to pull up, but I believe that my code can be simplified even more to make if run smoother and for the code to be read easier.

Solution Documentation:

I found that there are ways to be able to read lines in a file. The first way I found was to use .nextLine() and then use a print statement for that line. With this method it was enclosed inside of a while loop that repeatedly ran .nextLine() until it found the line that it was looking for. The second way that I found was to use a while loop that would increment i every time. The i was used to represent a specific line in the file. I used Files.readAllLines(Path.get(“animal.text”)).get(i) to be able to read each line. Once it found the line that contained the animal or habitat I wanted it would enter into an if statement that would gather the other lines of information for it. This way also allowed for the system to easily read if there were \*\*\*\*\* in the beginning of the line to create a dialog box.