INFO 1531 – Java II Programming – Assignment 2 – Exceptions

Assignment

Overview

The "Varmint Cong" Golf Tournament needs the ability to track golfers from all their previous games to now. They have player's names and scores from previous games all saved in a text file. The problem is we need to make sure our program doesn't crash having it be a well rounded working application. To do this some exception handling/throwing is required. A bunch of the code is written even with some exceptions in it, BUT this is proprietary so we don't want others to know how our application works.

You are tasked with making a quick custom exception to help make the program user friendly and reworking the code.

Directions

To Get Started:

1. Create a new project in IntelliJ(or your preferred IDE) called YOUR_MCC_Name-Assignment2

2. Download and extract the files needed for the assignment:

-
    - The java files Golfer.java and GolferSystem.java should go in your **src** folder
    - The GolfStats.txt file is our data and should go in your project folder

3. If you look at the Golfer.java file that is our object for holding the name and an Arraylist of their scores. Most everything is coded for that, you will have to visit a part of it during the final part of this assignment. Now the GolferSystem.java is where most of the work is at. This class has our main running with every thing split up into different methods. The code for reading and writing is already done for you BUT we need to rework it with our own exception and propagate the exceptions to the main.

Lastly if you open up the GolfStats.txt file you will see each golfer's name separated by a "-" then the list of scores separated by a ",". Again the system class already has this set up just reviewing each file with you.

INFO 1531 – Java II Programming – Assignment 2 – Exceptions

Part 1: Custom Exception

First we would like to have our own custom exception that we can throw to really control what is being put out to the user of our program. So you need to create a new java file for the exception. You can name it what you would like, some suggestions could be GolfSystemException, GolfStatException, SystemException, etc. Usually we just need to make sure and have the word "Exception" at the end.

The class should:

- extend from the Exceptions class
- have an empty default constructor
- have a constructor that takes on a String as parameter and sends it to the super class

Part 2: Integrate Custom Exception and Propagate

I was going to try and split this part up but then you would be doing double the work so we are going to do them together. Right now we have all the exception handling in each method but we want to handle everything in the main() method AND use our new custom exception.

First you need to modify the catch statements for both the readInGolfers() and writeOutGolfers() methods so that instead of printing out the exception(e) that you throw your new custom exception from part one. You can have it say whatever message you would like. Here are some suggestions:

- Data file on golfers does not exist/Stat file to read in doesn't exist
- Error on reading in data from file
- Error on writing the stats to the file

Note on that first bullet point I'm talking about the file not existing. Currently the the readInGolfers() method we are only catching IOExceptions. You should adjust this and add in a FileNotFoundException because that is a very specific exception that we should always catch to know if we have the file or not. Again you still throw your custom exception but with a different message.

Lastly, you now need to modify your method definition at the top so that they "throws NEWEXCEPTIONHERE". Then you will need to go to the main and adjust the switch statement so that you use a try/catch and simply print out the getMessage() of each of your custom exceptions.

To test and see if your exception throwing and catching/printing works take out the GolfStats.txt from your project and you should see something like this where the exception message is printed before the menu.

```
Stat file to read in doesn't exist.
 **** Golfer System Menu ****
1.View Golfers
2.Add Golfers
3.Delete Golfer
4.Save and Exit
Selection:
```

Part 3: Arithmetic Exception Handling

If you haven't already noticed when running the system, if you view the golfers you will get an error right after "Carl Spackler" gets printed out.

```
Exception in thread "main" java.lang.ArithmeticException: / by zero
    at Golfer.getAverage(Golfer.java:88)
    at Golfer.toString(Golfer.java:115)
    at java.base/java.lang.String.valueOf(String.java:4225)
    at GolferSystem.printGolfers(GolferSystem.java:92)
    at GolferSystem.main(GolferSystem.java:32)Exception in thread "main" java.lang.Ar
ithmeticException: / by zero
    at Golfer.getAverage(Golfer.java:88)
    at Golfer.toString(Golfer.java:115)
    at java.base/java.lang.String.valueOf(String.java:4225)
    at GolferSystem.printGolfers(GolferSystem.java:92)
    at GolferSystem.main(GolferSystem.java:32)
```

If you look in the text file you will see that Steve Rodgers has no golf scores. Look at the toString() method in the Golfer.java class. You will see we are printing the golfer, scores and also using the getAverage() method. If you look at our GolfStats.txt file you will see Steve Rodgers has no scores so here we are getting a dividing by zero problem since the size of the scores Arraylist is 0.

You need to fix this in the Golfer.java file however you see fit. I have example code where his average just prints out as '0' and no message prints. This is one of those do we print an error message or just have the program ignore it and still catch it but continue on since a user(non-programmer) doesn't care as long as it doesn't crash.

Some options include:

- Using a try/catch
- Using and if statement

Check the output (User input bolded):

For this run I don't have a basic "checker" file but rather showing you a run where I enter in some information and make sure it appears. So view the golfers, add one, then view again to make sure it was added, delete a golfer, then view to make sure they were deleted and lastly save and exit, then check the text file. No errors should pop up as we have them all handled correctly.

```
 **** Golfer System Menu ****
1.View Golfers
2.Add Golfers
3.Delete Golfer
4.Save and Exit
Selection: 1

 *** Golfers in System ***
Name: Clint Barton
Scores: 18  18  18  18
Average: 18

Name: Tiger Woods
Scores: 65  72  74  72  88  68
Average: 73

Name: Happy Gilmore
Scores: 77  77  88  79  85  47
Average: 75

Name: Carl Spackler
Scores: 90  102  99
Average: 97

Name: Steve Rodgers
Scores:
Average: 0

Name: Shooter McGavin
Scores: 84  83  78  92
Average: 84


 **** Golfer System Menu ****
1.View Golfers
2.Add Golfers
3.Delete Golfer
4.Save and Exit
Selection: 2

Enter name: Phil Mickelson
Enter scores:
74
85
```

```
83

 **** Golfer System Menu ****
1.View Golfers
2.Add Golfers
3.Delete Golfer
4.Save and Exit
Selection: 1

 *** Golfers in System ***
Name: Clint Barton
Scores: 18  18  18  18
Average: 18

Name: Tiger Woods
Scores: 65  72  74  72  88  68
Average: 73

Name: Happy Gilmore
Scores: 77  77  88  79  85  47
Average: 75

Name: Carl Spackler
Scores: 90  102  99
Average: 97

Name: Steve Rodgers
Scores:
Average: 0

Name: Shooter McGavin
Scores: 84  83  78  92
Average: 84

Name: Phil Mickelson
Scores: 74  85  83
Average: 80


 **** Golfer System Menu ****
1.View Golfers
2.Add Golfers
3.Delete Golfer
4.Save and Exit
Selection: 3

 *** Golfers in System ***
1. Name: Clint Barton
Scores: 18  18  18  18
Average: 18

2. Name: Tiger Woods
Scores: 65  72  74  72  88  68
Average: 73
```

```
3. Name: Happy Gilmore
Scores: 77  77  88  79  85  47
Average: 75

4. Name: Carl Spackler
Scores: 90  102  99
Average: 97

5. Name: Steve Rodgers
Scores:
Average: 0

6. Name: Shooter McGavin
Scores: 84  83  78  92
Average: 84

7. Name: Phil Mickelson
Scores: 74  85  83
Average: 80


Enter golfer to delete: 1

 **** Golfer System Menu ****
1.View Golfers
2.Add Golfers
3.Delete Golfer
4.Save and Exit
Selection: 1

 *** Golfers in System ***
Name: Tiger Woods
Scores: 65  72  74  72  88  68
Average: 73

Name: Happy Gilmore
Scores: 77  77  88  79  85  47
Average: 75

Name: Carl Spackler
Scores: 90  102  99
Average: 97

Name: Steve Rodgers
Scores:
Average: 0

Name: Shooter McGavin
Scores: 84  83  78  92
Average: 84

Name: Phil Mickelson
Scores: 74  85  83
```

```
Average: 80


 **** Golfer System Menu ****
1.View Golfers
2.Add Golfers
3.Delete Golfer
4.Save and Exit
Selection: 4
```

Submission

1. Compress the IntelliJ project folder and submit it to this assignment.

to compress: On Windows, right click -> send to -> compressed .zip file

on Mac, right-click -> Compress

Hints/Tips (Before Submitting):

- *Yes there are other exceptions we could check/handle but this is just practice.*
- *Don't forget to have a header at the top of your file and include a **Resource statement**.*
- *Use comments throughout for full points.*
- *Follow all Java Styling Guides as covered*