Lesson 2

Mark Eric Dieckmann

January 24, 2023

Lesson 2 addresses dynamic arrays, Input/Output (IO), and exception handling. Revise for this purpose lectures 4 and 5. We implement code that loads text from a file and saves it to a second file. Loading and saving are done through an instance of the class **Database**. Its methods are called from the method *main(..)* in the class **Lesson2**.

1 Task 1: The class Database

Database
-words: ArrayList<String>
+Database()
+load(String): String
+save(String): String

The instance variable *words* has a global scope and is initialized in the constructor.

load(arg) loads in all the lines of text from the file *arg*. It should subdivide the lines at one or more empty spaces and add each individual string to *words*. *load(arg)* should catch IOExceptions. If an exception occurs, it should return "Something went wrong". Otherwise, it should return a formatted string with "Text loaded:" followed by the number of words (4 digits) followed by " words". Close the file reader after you loaded all lines from the file.

save(arg) should save the content of *words* to an output file named *arg*. It should use in the file one line per element of *words*. This method should throw Exceptions. It should return the string "Wrote" followed by the number of elements of *words* on 4 digits followed by " words to the file" followed by the name of the output file (on 15 characters). It should close the reader after you wrote all lines to the file.

2 Task 2: The class Lesson2

Your class **Lesson2** should contain the method main(..) and a class variable that can read text from the console.

Create an instance variable of type **Database** in your method *main(..)*. Implement a while loop. At every loop execution, the loop should ask you for an input string. If you respond "load", your code should call the method *load(arg)* of **Database** with the argument "inputFile.txt". If you respond "save", the method should call **Database**'s

save(arg) method with the argument "outputFile.txt". Write the return values of both methods to the console. The while loop should stop when you type in "end".

Place the input file "inputFile.txt" in your project directory. Running the code should give you the output

```
Your command: load
Text loaded:  13 words
Your command: save
Wrote  13 words to the file  outputFile.txt
```