CS1 — Homework Module 3: Conditionals

September, 2022

Exercise 1

Use the methods `equals`, `equalsIgnoreCase` and `compareTo` to:

1. Compare two Strings retrieved from the keyboard and inform the user about whether they are equal or not
2. Compare two Strings retrieved from the keyboard and inform the user about whether they are equal or not regardless the (upper or lower) case used
3. Sort, in lexicographic order, three Strings (you can ask for them or hard-code them). Then, check Strings such as (maintaining the proposed upper/lower cases):
   - Arrested development, Big Bang Theory, Modern family
   - arrested development, BIG BANG THEORY, modern family
   - Parks and recreation, The office, Scrubs
   - Parks and recreation, the office, scrubs
   - AAA, aaa, Aaa

For an extra challenge, read in 4 or even 5 words. (Don't worry, when you learn more about programming, this will be come much much easier than it is now.)

Exercise 2

Check how the following methods on Strings work:

- `startsWith(String a)` - returns boolean
- `endsWith(String a)` - returns boolean
- `length()` - returns int
- `charAt(int index)` - returns char; starts counting from 0 to length()-1!
- `substring(int begin, int end)` - returns String; again counting from 0!

More information:

- https://docs.oracle.com/javase/7/docs/api/java/lang/String.html
- https://www.w3schools.com/java/java_strings.asp

You can also practice some of them in CodingBat (*warmup2*).

Exercise 3

Write a Java program that reads an integer number from the command line and prints out whether the number is even or odd. (You can employ the `Absolute Number` method described here: https://www.w3schools.com/java/ref_math_abs.asp)

Exercise 4

Write a Java program that does the same as in Exercise 3, but lies about the result 10% of the time. Not more not less!

Exercise 5 — Things are getting a bit tougher ...

Some more recent (and geeky) board games use dice with different numbers of sides, 4, 6, 8, 10, 12, 20, ... Write a Java program called Dice that first asks the user for the number of sides on the dice s/he wants to roll, and then simulates a random roll with that dice and prints the result. Your program should roll a fair dice, with equal probabilities for all possible outcomes and work for all whole positive numbers, not just for the 6 values listed above.

You can use the example shown in the lecture (for 6-sided dice) as a starting point.