CS 145 Homework 3
Trutilities

1 Overview

Your objective in this homework is to reason about data using logical and relational operators. Expressions built out of these operators yield yes/no answers, which can ultimately be used to steer our code one way or another. You will do this in the context of solving several disconnected problems that have no overarching story. Sorry.

2 Requirements

Complete the two classes described below. Place all classes in package `hw3`. Make all methods `static`.

2.1 Trutilities

Write a class `Trutilities`, which contains a number of methods that report the truth or falsehood of various questions one might ask in a piece of software. All methods return `boolean`s.

1. Write a method `inBox` that accepts six `double` parameters. The first pair is an xy-coordinate pair of a rectangle's lower left corner. The second pair is the upper right corner. The third pair is some other 2-D point, which may or may not be inside the rectangle. Return true if the point is inside the rectangle. Consider points on the perimeter inside.

2. Write a method `inCircle` that accepts five `double` parameters. The first pair is an xy-coordinate pair of a circle's center. The third parameter is the circle's radius. The last two parameters are an xy-coordinate pair for some point, which may or may not be inside the circle. Return true if the point is include the circle. Consider points on the perimeter inside.

3. Write a method `isGrayscale` that accepts a `String` hexadecimal color code, which has the form `#RRGGBB`. `RR`, `GG`, and `BB` may be any two hexadecimal digits, which are numbers 0–9 or letters A–F or letters a–f. Return true if the color is grayscale. A grayscale color has equal red, green, and blue intensities.

4. Write a method `isNorth` that accepts a `double` decimal latitude in $[-90, 90]$ and a `String` degrees-minutes-seconds latitude of the form `D°M'S"`. `D`, `M`, and `S` are all positive integers. (Note that there's no way to represent latitudes in the southern hemisphere with this format. Supporting southern latitudes requires conditional statements, which we haven't met yet.) In Java's Unicode `String`s, the degree symbol is represented as `\u00B0`. There are 60 minutes in a degree, and 60 seconds in a minute. Return true if the latitude represented by the `double` is north of the `String` latitude. (By the way, never write software that mixes units. See `http://en.wikipedia.org/wiki/Mars_Climate_Orbiter#Cause_of_failure`.)

5. Write a method `isBefore` that accepts six `int`s. Each triplet is a date in the following order: a year, a month in $[1, 12]$, and a day in $[1, 31]$. Return true if the first date happens strictly before the second date.

6. Write a method `isEqualEnough` that accepts three `double` parameters. Let's call the third `threshold`. Return true if the first two are within `threshold` units of each other.

7. Write a method `isPowerOfTen` that accepts an `int` parameter. Return true if the parameter is a power of 10. `Math.log` tells you to which power you need to raise 10 to yield the parameter you send it. Powers of 10 are those numbers which can be reached through an integral exponent. There are a number of ways to determine if a particular `double` is an integer. One way is to see if the `double` is the same as its floor.

8. Write a method `isGameOver` that accepts a `String` command parameter. It returns true if the command is exactly one of `quit`, `exit`, `done`. Case is not significant.

2.2 Main

Write a class `Main`. It has a `main` method, which you are encouraged to use to test your code. Nothing in particular is required of it, but it must exist.

3 Submission

To submit your work for grading:

1. Put the SpecChecker for this homework in your Build Path. Run the SpecChecker as a Java Application and fix problems until all tests pass.

2. Commit and push your work to your repository. Verify that your solution is on Bitbucket.

A passing SpecChecker does not guarantee you credit. Your grade is conditioned on a few things:

- You must meet the requirements described above. The SpecChecker checks some of them, but not all.

- You must not plagiarize. Write your own code. Talk about code with your classmates. Ask questions of your instructor or TA. Do not look at others' code. Do not ask questions specific to your homework anywhere online but Piazza. Your instructor employs a vast repertoire of tools to sniff out academic dishonesty, including: drones, CS 145 moles, and a piece of software called MOSS that rigorously compares your code to every other submission. You don't want to live in a world serviced by those who squeaked by through questionable means. For your future self, career, and family, do your own work.

- Your code must be submitted correctly and on time. Most excuses devolve into, "I started too late." The fix for this problem is not an extension.