Delta College

CST 283
Programming Assignment 2

Winter 2024
Instructor: T. Klingler

Objective

This program provides an opportunity to building a Java class to search for a target position on Earth. It also includes construction of a basic JavaFX user interface as the application *front-end*.

Overview & Instructions

Whether it is a military operation or simple geocaching, many activities involve searching for a target at a precise geographic position on Earth from a given starting position. Write a Java program that utilizes the basic required formulas to search for a target.

Build a class to manage a **`Target`**. Details for the class are provided in the following UML specification:

```
Target
------------------------------------------------
  - targetLatitude : double
  - targetLongitude : double
  - myLatitude : double
  - myLongitude : double
  - movingSpeed : double
------------------------------------------------
  + Target()
  + Target(double,double,double,double,double)
  + { set/get methods }
  + double distanceFrom()
  + double bearingToDegrees()
  + String bearingToOrdinal()
  + double timeToTarget()
  + boolean dataValid()
  + String getDateTime()
  + toString()
```

The class should also include the following elements:

- Include both a no-argument constructor and at least one parameterized constructor (even though you might not need them for this assignment).
- Be sure also to include all "set" and "get" methods associated with each of the primary data members of the class.
- Include a **`toString()`** method as part of the class that will return a String object that includes the current values stored in a class object for the primary data elements.
- Comment your **`Target`** class using ***javadoc***

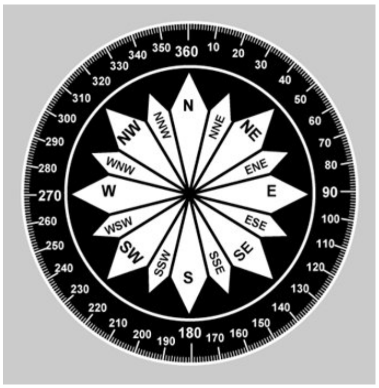
To assist with the required mathematics, the following methods are offered in the demo program: GeoDistBearing.java

- Great circle distance
- Bearing to target

Note that these define *line-of-sight* targeting and are not related to road paths, etc.

Note the reference to directional bearings at the right. Traditional bearings are returned in degrees 0.0...360.0 but can also be defined using ordinal descriptions (N, NNE, NE, …)

Note that the general bearing calculation returns an angle measurement, but the *ordinal* bearing should return one of the 16 compass points as a String. You will need to define a translation from degrees to a given compass point based on ranges of degrees (and for grading there is some discretion in these ranges).



For the current date/time, consider use of the Java `GregorianCalendar` or `LocalDateTime` classes.

Next, build a simple **JavaFX** front-end user interface for your solution. Your front-end interface will communicate with one object of the back-end `Target` object. The basic operation of your interface will include:

- Text field designating name of target
- Text fields to enter latitude and longitude of target
- Text field for speed you will be moving toward target
- A "Go" or "Calculate" button to launch navigation calculation actions
- A "Clear" button to clear out all text fields and a "Quit" button to terminate application

Include data validation to first insure that the numerical values are valid numbers (using exception-handling) and that the target identification is not blank. Furthermore, validate that the latitude is between (–90.0 and 90.0) and the longitude is –180.0 to 180.0. (Note that North American longitudes are negative (west of the *Greenwich* line in England). Also be sure the speed is not negative.

Feel free to hard-code the latitude and longitude of your current position as a constant. Feel free to use coordinates of Delta College or any other point of your choice (Delta College: latitude: 43.559; longitude: –83.986). Note that Google Maps can provide you with very precise geographical coordinates of any place with one or two clicks.

If the input data are valid, perform necessary method calls to generate an output report using a JavaFX **`Alert`** action (Please avoid use of **`JOptionPane`** dialogs). The report could be similar to what you see below:

```
Target:  Coffee Shop
Distance: 2.3 km
Bearing 68 degrees (ENE)
Currently:  13 JUL 2021 14:52
Time to target: 0.22 hours
```

Deliverables

Demonstrate the development steps of your program with at least two version commits to the assignment Git repository.

Deliver the following to the eLearning system **Assignment Dropbox** as your final product:

- **Upload** your **source code** (.java) file(s); preferably zipped in more than one file