CSC116 Spring 2019 - Project 1

Distance Calculator

Distance Calculator Overview

Tasks

1. Read the Software Requirements

Review the formulas for calculating the planar and spherical distance between two points given their latitude and longitude.

2. Design Your Software

Your software must include the `calculatePlanarDistance` and `calculateSphericalDistance` methods.

3. Implement Your Software

Implement the software that produces the distance table.

4. Test Your Software

Execute the provided test cases against your software to ensure the expected results match the actual results.

5. Submit Your Software

Follow the submission instructions for your instructor.

Read the Software Requirements

The shape of the earth is closely approximated by a sphere, but there are still some people who believe the earth is flat! For this project, you will calculate both the planar and spherical distances from Raleigh, NC, to other points in North America given their latitude and longitude. Latitude is the angular distance of a place north or south of the equator while longitude is the angular distance of a place east or west of the meridian at Greenwich, England. The latitude and longitude values of Raleigh are 35.78 degrees and -78.64 degrees, respectively. You will produce a table that gives both the planar and spherical distances from Raleigh to locations with latitudes ranging from 30 (southern United States) to 50 degrees (southern Candada) and longitudes ranging from -120 degrees (western United States) to -70 degrees (eastern United States).

Planar Distance Formula

This formula gives the distance between two points assuming the spherical earth has been projected to a plane.

$$D = R\sqrt{(\Delta\phi)^2 + (\cos(\phi_m)\Delta\lambda)^2}$$

R - radius of earth (3959 miles)
ΔΦ - difference between the two latitudes (in radians)
Δλ - difference between the two longitudes (in radians)
Φm - mean (average) of the two latitudes (in radians)
D - planar distance between the two points (in miles)

Source: https://en.wikipedia.org/wiki/Geographical_distance

▶ Click to Expand Planar Distance Calculation Example

Spherical Distance Formula

This formula gives the distance between two points assuming the earth is a sphere.

$$D = R\arccos(\sin\phi_1 \sin\phi_2 + \cos\phi_1 \cos\phi_2 \cos\Delta\lambda)$$

R - radius of earth (3959 miles)
Φ1 - latitude of first point (in radians)
Φ2 - latitude of second point (in radians)
Δλ - difference between the two longitudes (in radians)
D - spherical distance between the two points (in miles)

Source: https://en.wikipedia.org/wiki/Great-circle_distance

▶ Click to Expand Spherical Distance Calculation Example

Distance Calcultor

Write a program named **DistanceCalculator** that calculates and outputs the planar and spherical distances in miles from Raleigh to points located at latitudes of 30, 40, and 50 degrees and longitudes of -120, -115, -110, -105, -100, -95, -90, -85, -80, -75, and -70 degrees using the formulas given above. Use 35.78 and -78.64 as the latitude and longitude in degrees for Raleigh and 3959 miles as the radius of the earth.

You will display your output in a table, with each row representing a different longitude and the columns used to show the planar and spherical distances from Raleigh for the three different latitudes. Your output format should be similar to that shown below. Your values should exactly match the values given below, but include distances for longitudes of -120, -115, -110, -105, -100, -95, -90, -85, -80, -75, and -70 degrees.

➤TERMINAL

```
          Distance (miles) from Raleigh (lat 35.78 long -78.64)
                lat 30              lat 40              lat 50
 long      Planar Spherical    Planar Spherical    Planar Spherical
 ----      ------ ---------    ------ ---------    ------ ---------
 -120        2433      2415      2274      2254      2313      2276
 -115        2147      2135      2004      1990      2087      2059
    .           .         .         .         .         .         .
    .           .         .         .         .         .         .
    .           .         .         .         .         .         .
  -70         641       640       554       554      1076      1074
```

Reminder for Dr. King's and Dr. Pala's Sections

With the expected project structure, you will use the following commands to compile and run code from the `Project1` directory:

➤TERMINAL

```
csc$ java -cp bin DistanceCalculator
```

Design Your Software

The `main` method of your program must call the methods, `calculatePlanarDistance()` and `calculateSphericalDistance()`, which must be defined and documented as shown below:

➤JAVA

```java
/**
 * Calculates the distance between two points on the earth's surface assuming
 * the spherical earth has been projected to a plane
 *
 * @param latitude1 latitude of first point in degrees
 * @param longitude1 longitude of first point in degrees
```

```
     * @param latitude2 latitude of second point in degrees
     * @param longitude2 longitude of second point in degrees
     * @return planar distance in miles between the two points
     */
    public static double calculatePlanarDistance(double latitude1, double longitude1,
                                                 double latitude2, double longitude2) {

    }
```

➤JAVA

```
    /**
     * Calculates the spherical distance between two points on the earth's surface
     *
     * @param latitude1 latitude of first point in degrees
     * @param longitude1 longitude of first point in degrees
     * @param latitude2 latitude of second point in degrees
     * @param longitude2 longitude of second point in degrees
     * @return spherical distance in miles between the two points
     */
    public static double calculateSphericalDistance(double latitude1, double longitude1,
                                                    double latitude2, double longitude2) {

    }
```

The `calculatePlanarDistance()` and `calculateSphericalDistance()` methods must use the appropriate formula given above to calculate the distance in miles between two points given their latitude and longitude in degrees. The method will return the distance to the `main` method, which will output a table listing the planar and spherical distances from Raleigh to points at latitudes of 30, 40, and 50 degrees, and longitudes of of -120, -115, -110, -105, -100, -95, -90, -85, -80, -75, and -70 degrees, as shown above.

The `main` method must call the `calculatePlanarDistance()` and `calculateSphericalDistance()` methods from within a **nested for loop**. The outer loop will vary the longitude from -120 to -70. The inner loop will vary the latitudes from 30 to 50 and output the distance returned from each method. All output should be done in the `main` method; there should not be any output in the `calculatePlanarDistance()` and `calculateSphericalDistance()` methods.

Implement Your Software

See the middle of page 158 in the textbook for an example of how to get the value returned from a method and print it out.

Your `calculatePlanarDistance()` and `calculateSphericalDistance()` methods must return the correct distance value as a double for all reasonable input values, not just the ones used to create the table above.

Output the distance values with no decimal places using a `printf` statement to format the value. For

Degree to Radian Conversion

NOTE that these methods must convert the latitudes and longitudes from degrees to radians in order to use the given formulas and to use the Math class trigonometric methods. You may want to check the Math API for available methods that may help with the conversion to radians.

example, the following statement formats a distance value so that there are no places after the decimal point, it is right-justified, and takes up total of 9 spaces:

```java
System.out.printf("%9.0f", distance);
```

The following statement may be used to output an integer value so that it is right-justified and takes up a total of 5 spaces:

```java
System.out.printf("%5d", value);
```

You can read more about the printf statement on pages 269 - 273 of the textbook.

Use the following named constants (which you will need to javadoc) in your program rather than magic numbers - the latitude and longitude values are given as `int`'s to make them easier to use in a nested for loop:

```java
    public static final double EARTH_RADIUS = 3959; //miles

    public static final double RALEIGH_LATITUDE = 35.78; //degrees

    public static final double RALEIGH_LONGITUDE = -78.64; //degrees

    public static final int MIN_LATITUDE = 30; //degrees

    public static final int MAX_LATITUDE = 50; //degrees

    public static final int MIN_LONGITUDE = -120; //degrees

    public static final int MAX_LONGITUDE = -70; //degrees

    public static final int LATITUDE_INCREMENT = 10; //degrees

    public static final int LONGITUDE_INCREMENT = 5; //degrees

```

You may use numbers for formatting in your `printf` statements as well as the numbers 0, 1, 2, and 100 without defining them as a class constants.

Test Your Software

After you submit your solution, we are going to use a combination of code inspection, manual testing, and automated testing to evaluate your work. For the automated testing to work, it's important that your methods exactly match the method headers given above. To help you make sure you've used the correct header for the methods, we're providing providing a test program, `DistanceCalculatorTest.java`.

Choose the appropriate set of instructions below based on your section of the course:

▶ Expand Instructions for Dr. Balik's & Ms. Glatz's Students

▶ Expand Instructions for Dr. King's & Dr. Pala's Students

Submit Your Software

To view your specific submission instructions, please select the section of the course in which you are enrolled:

Course Section	Instructor	Submission Instructions Link
001	King	Submission Instructions
002	Balik	Submission Instructions
003	Pala	Submission Instructions
004	Balik	Submission Instructions
006	Balik	Submission Instructions
007	Glatz	Submission Instructions
008	Glatz	Submission Instructions

Reminders

Before you submit your work, make sure the program:

- behaves as specified in this document. Be sure to consider what results should be displayed for each major scenario.
- is thoroughly tested.
- satisfies the rubric for this assignment.

601	Glatz	Submission Instructions

◀ Animal ASCII Art

Grading Rubric ▶