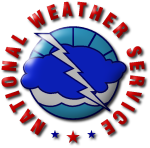
 ****

**Homework #5**

Java Course

National Weather Service

Use Classes written in the previous homework to complete this assignment.

1. Apply private access restriction to instance variables in the following classes: GeoPoint, BoundingBox, NamedPoint, NamedLine, NamedPolygon. Once this access restriction has been applied those objects will become immutable. Therefore there will be no way to change the data within them once they are created. Once you have applied this changes verify to yourself that you cannot change the latitude in a GeoPoint object:

GeoPoint point = new GeoPoint(-87.5,34.0);

point.latitude = 35.6;

This should cause an error, because it is now protected. Also you will note that the variable does not show up in the code completion.

1. To be able to display our new NamedPoint, NamedLine and NamedPolygon in a window we need to develop a relationship between geographic coordinates, and window coordinates. We will use a simple linear relationship using our BoundingBox class. One interesting note is in Screen Coordinates Y is positive in the down direction and 0,0 is at the upper left corner.

50, 80

-97.1, 34.5

0,0

Geographic Coordinates

Screen Coordinates

x

upperLeft

y

lowerRight

width, height

Screen Coordinates vary between (0,0) to (width, height). To translate from Geographic Coordinates to Screen Coordinates we need to define the Geographic coordinates that correspond with the (0,0) (upper left corner of the Screen Coordinates) and with the (width, height) Screen coordinates. We will do this by defining a new class called Coverage. Coverage will be constructed by giving the BoundingBox for the geographic area covered, and a width and height of the Screen coordinates. The BoundingBox is upper left corner (-95.0, 34.0) and the lower right corner is (-90.0, 32.0). Then -95.0, 34.0 would translate to 0, 0 in screen coordinates, and -90.0, 32.0 should correspond to the width and height of the Screen Coordinates.

The Coverage class should contain instance variables to contain BoundingBox and width and height, where width and height are of type “int”. Screen coordinates will be described using an existing java object java.awt.Point, below is referred as Point. This will have to be imported, and the IDE will help you import the proper namespace.

Create two methods to provide the translation.

geographicToScreenCoordinates(GeoPoint geoPoint) which given a point in geographic coorindates it will return a Point in screen coordinates.

geographicToScreenCoordinates(GeoPoint[] geoPoints) which given an array of geographic points will return an array of Screen coordinate points for the array of geographic points.

Here is the equation to calculate the transformation from geographic to screen coordinates.

degreesLatPerPixel = (upper\_left\_latitude - lower\_right\_latitude)height;

degreesLonPerPixel = (upper\_left\_longitude - lower\_right\_longitude)/width;

x = (upper\_left\_longitude-geoPoint\_longitude)/degreesLonPerPixel;

y = (upper\_left\_latitude-geoPoint\_latitude)/degreesLatPerPixel;

where upper\_left and lower\_right type variables come from BoundingBox, and x and y are screen coordinates. And width and height refer to the size of the window, which are passed into the Coverage Object.

Add these methods to your Coverage class. To test your new Coverage Class, create a Coverage object inside your main method and perform transformations on values you know, like the upperLeftCorner of the BoundingBox and make sure you get the value 0,0. Do the same for lower right corner and the middle point. See if you can provide a reverse transformation from screen to geographic coordinates, e.g. screenToGeographicCoordinates(Point point).

Once we have coverage created if we need to draw a feature such as a polygon or line or point in a window we will just need to use it to calculate where to draw it on the screen.