COMP 1409 – Assignment #3 (90 Points)

Due: 11:59 p.m. the night before session 11

Fix any errors and omissions from assignments 1 and 2. Your final assignment is built on those requirements. In assignment 1 you wrote ParkedCar and ParkingMeter classes. In assignment 2 you added PoliceOfficer and ParkingTicket. For assignment 3 you will add some functionality to the PoliceOfficer class and you will add a PoliceDepartment class.

Add the following attribute to the **PoliceOfficer** class

- An ArrayList of the issued parking tickets. The ArrayList is created in every constructor of the class.
- An accessor for the newly added field.

Add the following functionality to the PoliceOfficer class:

- Modify the method that issues the parking ticket in a way that the method will add each issued parking ticket to the collection. This modified method will not return anything.

- A method to calculate and return the sum of fines of all the parking tickets in the collection.

- A method that takes the license number of a car as parameter, and counts and returns the number of tickets issued to that specific car. The license number must be case-insensitive.

PoliceDepartment class

This class has the following attributes:

- A collection of PoliceOfficers, i.e. ArrayList<PoliceOfficer>
- Department location

PoliceDepartment class has the following:

- A default constructor that initializes department location to the string “unknown”.
- A non-default constructor that accepts a parameter to initialize the location field.

Both constructors call the set method to initialize the location field. Both constructors create a new ArrayList to initialize the collection.

- An accessor method for the location field.

- A method that takes a PoliceOfficer as a parameter and adds it to the ArrayList only if it’s not null.
- A mutator method for the location field. This mutator checks the passed parameter and uses it only if it’s not null or an empty String (“”). If the passed parameter is null or an empty String the method does not use it, the value of the instance variable will be set to “unknown”, and the following message is displayed:
  “The passed parameter cannot be null or an empty string”.

- A method that takes PoliceOfficer object as a parameter and adds it to the ArrayList only if it's not null.

- A method that accepts an officer name as a parameter, validates that the passed parameter is not null, searches the collection and returns an ArrayList of the parking tickets issued by that officer or returns null if there were no parking tickets issued by the specified officer. The officer name is case-insensitive. This method uses a while loop and an iterator to implement the required functionality.
  Here is the method signature:
  public ArrayList<ParkingTicket> getAllTicketsByOfficer(String officerName)

- A method that calculates and returns the total amount of fines of all the parking tickets issued by all the officers in the collection.

- A method that accepts a license number of a car as a parameter; searches the collection and returns the total number of parking tickets issued to that specific car by any of the police officers in that department. The license number should be case insensitive.

Marks will be given for:

- Comments – appropriate and complete, including Javadoc tags @author, @version, @return and @param.
- Style – see the style guide Appendix J of your textbook. Style includes following the Java naming convention for classes, variables and methods. It also includes correct indentation.
- Correctness and completeness – code meets the requirements listed above.

Create a .zip file containing your entire BlueJ project (zip the folder, not the individual files). Name the .zip file with your name and the assignment number, e.g. "JoeZhangAssign3.zip". Upload the file to D2L before the cutoff time.