COMP 1409 – Assignment #2 (70 points)

Due 11:59 p.m. the night before session 8

Assignment 2 builds on assignment 1. You will add the following classes:

PoliceOfficer
Here are the relevant attributes

- officer name
- badge number
- parked car of type ParkedCar
- parking meter of type ParkingMeter

Provide the following:

- A default constructor that uses the set methods to initialize the fields. To initialize the String fields an empty String (“”) is passed to the corresponding set method. To initialize other fields a new object created by the default constructor is passed to the corresponding set method like this:
  ```
  setParkedCar( new ParkedCar( ) );
  setParkingMeter( new ParkingMeter( ) );
  ```

- A non-default constructor that expects parameters and uses the set methods to initialize the fields.

- An accessor (get) method for each field.

- A mutator (set) method for each field. The mutator methods of parkedCar and parkingMeter must test their parameter. If the parameter is null, set the field by creating a new object using the default constructor of that class. The mutator methods of the String type fields test their parameter. If null, set the field to an empty string (“”).

- A method to check if the parkedCar time has expired. The method signature is: public boolean isParkingTimeExpired()

- A method to calculate the fine. The fine is $25 for the first hour or part of it and $20 for every additional hour of part of it. Be sure to use symbolic constants, not “magic” numbers.

- A method to issue a parking ticket. If the parking time for the car has expired, create a new ParkingTicket object by passing all required information to the ParkingTicket constructor, and then display the ticket's details on the screen by calling the appropriate method from ParkingTicket. Return the new object, or return null if no ticket issued.

ParkingTicket
Here are the relevant attributes:

- ticketnumber of type String
- the name of the police officer who issued the ticket
- the police officer's badge number
- the car's license number
- the car's make
- the car's model
- amount of the fine

Note that the ticket number must be unique, created by a private method createTicketNumber() that is called from the constructor. The ticketNumber must be set when the object is created and not allowed to change. For example the first ticket will have the number "V1001", the second ticket will have the number "V1002" etc. (Hint: use a static variable to hold the number part of the ticket number and increment it in the method that creates the ticket number).

Provide the following:

- A default constructor that sets the fields to the default value, and calls the method to create a ticket number.

- A non-default constructor that accepts parameters to set the fields, and calls the method to create a ticket number.
  Note that both constructors use the set methods to initialize the fields.

- An accessor (get) method for each field.

- A mutator (set) method for each field except ticket number. The mutator methods of the String type fields test their parameter. If null, set the field to an empty string (""). The mutator for the fine must not let the fine be a negative number.

- a method to display the details of the ticket on the screen in this format:

```
Ticket number: V1006
Officer Name: John Smith
Badge number: VAN25852
Car license number: 345 UKL
Cars make: Nissan
Car model: Pathfinder
Fine: $65.0
```

Use BlueJ to interactively test your methods as you write them. Write them one at a time, and test each immediately to be sure it is correct. Testing requires creating an object and invoking the method. Be sure to test with both valid and invalid data, i.e. test the "set" methods with positive values to be sure the field is changed correctly, and test with negative values to be sure the field is set to the default value.

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. “JoeZhangAssign2.zip”. Upload the file to D2L before the cutoff time.