Assignment 3 is due: 11:59 p.m. the night before session 12

**Key Rules:**

* The project must compile and run to be marked otherwise it will receive a grade of 0
* The Assignment is **due 11:59 PM the night before session #12**. Late submissions are not permitted
* Class PoliceOfficerTest.java and PoliceDepartmentTest.java will be used to mark the majority of your assignment
* This assignment is meant to be done individually, group work will be considered plagiarism.
* All classes must have full Java documentation comments for all public elements( public methods and constructors) using @author , @version, @param, @return tags when applicable and a brief description of the class**. 1 mark will be deducted for each incomplete of missing Java** **documentation tag**
* All classes **must not have magic numbers**, **1 mark will be deducted for each use of a magic number.** Any number other than 0 is considered a magic number

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 to **PoliceOfficer class**

|  |
| --- |
| Attributes:  An ArrayList of the issued parking tickets. The ArrayList is created in the constructor of the class. |
| Appropriately named accessor for the newly added field |
| Modify the method that issues the parking ticket in a way that the method will add each issued parking ticket to the ArrayList. This modified method will not return anything. |
| Provide a method to calculate and return the sum of fines of all the parkingTickets in the collection |
| Provide a method that takes a license plate number as a parameter , counts and returns the total amount of parking tickets issued to the car with the given plate number. The license plate number examination must be case-insensitive. |
| Provide a method that takes a car license number as a parameter and returns an array of all the parking ticket objects issued to the car with the given license number. The license plate number examination must be case-insensitive. |
| Provide a method that takes the license plate number of a car as parameter, calculates and returns the total amount of fines issued to that specific car. The license plate number examination must be case-insensitive. |
| Provide a method that displays the details of all the parking ticket objects in the ArrayList |
| Provide a method that takes a car license plate number as a parameter and deletes all the parking tickets issued to the car with the provided plate number. The method returns the number of deleted Parking Ticket objects. The license plate number examination must be case-insensitive. The method will use a while loop and an iterator. |

Create a class called **PoliceDepartment**. The class has the following:

|  |
| --- |
| Attributes:  Address e.g. “Vancouver”  An ArrayList of PoliceOfficer objects. The ArrayList is created in the constructor of the class. |
| Provide a constructor, the constructor will create the ArrayList and it will take one parameter to set the address field. the following restrictions will be implemented in the constructor **using appropriate method calls**:   * address should not be null or an empty String. If the provided value was null the value will be rejected and the error message "address cannot be null" will be produced. If the provided value was an empty String the value will be rejected and the error message "address cannot be an empty String" will be produced |
| Provide appropriately named accessors for the fields  Address  officerList |
| Provide appropriately named mutator for the field  Address  The mutator will apply the same validations mentioned in the constructor section |
| Provide a method that takes PoliceOfficer object as a parameter and adds it to the arrayList only if it’s not null |
| Provide a method that accepts an officer name as a parameter, validates that the passed parameter. If it was not null, the method will search the collection and displays a list of the parking tickets issued by that officer |
| Provide a method that calculates and returns the total amount of fines of all the parking tickets issued by all the officers in the collection. |
| Provide a method that accepts a license plate 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 examination should be case insensitive. |

Add the following to the main method in the Driver class:

|  |
| --- |
| * Create two Car objects. Pass 150 as the numberOfMinutesParked value for the first one and 140 for the second one * Create two PoliceOfficer objects. Invoke method issueParkingTicket() with the meter object and one of the cars from the first police officer object and invoke the same method on the other car object and the meter object form the second police officer object * Create a PoliceDepartment object and add all three officer objects to the department. * Call method showAllTticketsByOfficer() on the second police officer object * Call the method that deletes the parkingTickets on one of the cars that had tickets issued by that officer * Call method showAllTticketsByOfficer() on the same police officer object form the above requirement |

**Instance variable, constructor and method names should match the following names in for the test classes to work properly. Here is a list of the used names**

|  |  |
| --- | --- |
| Class PoliceOfficer | |
| Symbolic constants | ONE\_HOUR\_FINE\_AMOUNT  MINUTES\_IN\_HOUR |
| Fields | officerName  officerBadgeNumber  ticketList |
| Constructor | PoliceOfficer(name, badgeNumber) |
| Methods | getOfficerName  setOfficerName  getOfficerBadgeNumber  setOfficerBadgeNumber  isParkingTimeExpired  calculateFine  issueParkingTicket  getTicketList  sumAllfines  getParkingTicketsCountForACar  getTicketArrayByLicenseNumber  getSumOfFinesByCar  deleteTicketsByCarLicense  displayticketsDetails |
| Class PoliceDepartment | |
| Fields | address  officersList |
| Constructor | PoliceDepartment(address) |
| Methods | getAddress  setAddress  getOfficerList  addPoliceOfficer  displayTicketsByOfficer  calculateSumOfAllTicketsOfAllOfficers  totalParkingTicketCountOfACar |

**Grading**

The classes will be marked as follows:

|  |  |
| --- | --- |
| Class PoliceOfficer | 17 marks |
| Class PoliceDepartment | 13 marks |
| Style and Driver class implementation | 10 marks |
| Total | 40 marks |

Marks will be given for:

* Comments – appropriate and complete, including Javadoc tags @author, @version, @return and @param.
* Style – In addition to comments style includes following the Java naming convention for classes, variables and methods. It also includes correct indentation and the use of symbolic constants instead of magic numbers whenever needed
* 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 the assignment number, e.g. “Assign3.zip”. Upload the file to D2L before the cutoff time.