COMP 1409 Assignment #1 (60 Points)

Due: 11:59 p.m. the night before session 5

Assignment 1 consists of a project called “Comp 1409 Assignment” that has two classes. We will build on this in future assignments.

ParkedCar
This class has the following attributes:

- car make
- car model
- car colour
- license number
- number of minutes parked
- owner name

Provide the following constructors and methods:

- A default constructor that initializes all fields of type String to an empty String (“”), and initializes the rest of the fields to their default values according to their data types.

- A non-default constructor that accepts parameters to initialize all the fields. The constructor validates each passed parameter and uses it if it’s valid. If not, the instance variable will be set to the default value except that the string will be set to an empty string (“”). The number of minutes parked must not be negative.

- An appropriately named accessor method and an appropriately named mutator method for each field. The mutator methods check their parameter to ensure it’s valid. If not valid, the parameter is not used and an error message is displayed on the screen. Note that the number of minutes parked should always have a positive value.

- A method to display the details of the car on the screen.

ParkingMeter
This class has the following attributes:

- The constant MAXIMUM_AMOUNT_OF_PURCHASED_MINUTES = 180
- number of purchased minutes
- meter type for example “single parking”, “multi parking” ...
- meter location
- price of one minute
- is there a camera

Provide the following constructors and methods:

- A default constructor that initializes all the fields of type String to an empty string (“”) instead of null, and initializes the rest of the fields to their default values according to their data types.

- A non-default constructor that accepts parameters to initialize all the fields. The constructor validates each passed parameter and uses it if it’s valid. For the number of minutes purchased, if the passed parameter is greater than MAXIMUM_AMOUNT_OF_PURCHASED_MINUTES, the instance variable is set to that maximum value. If the number of minutes purchased is negative, the instance variable is set to zero. For the other parameters, if invalid the instance variable is set to the default value except that strings will be set to empty strings (“”).

- An appropriately named accessor method and an appropriately named mutator method for each field. The mutator methods check their parameter to ensure it’s valid. For the number of minutes purchased, if the passed parameter is greater than MAXIMUM_AMOUNT_OF_PURCHASED_MINUTES, the instance variable is set to that maximum value and a message to that effect is displayed on the screen. If the number of minutes purchased is negative, the instance variable is not changed and an error message is displayed. For the other mutators, if the passed parameters is invalid it is not used and an error message is displayed. Note that the valid values of number of purchased minutes should be between 0 and 180 inclusive and the price of one minute should never be a negative value.

- A method to display the details of the parking meter on the screen.

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. “JoeZhangAssign1.zip”. Upload the file to D2L before the cutoff time.