Note: This assignment is worth 7pts out of your course total.

For this assignment, you will design a set of classes that work together to simulate a police officer issuing a parking ticket. You should design the following classes:

- The **ParkedCar** Class: This class should simulate a parked car. The class's responsibilities are as follows:
  - To know the car's make, model, color, license number, and the number of minutes that the car has been parked.

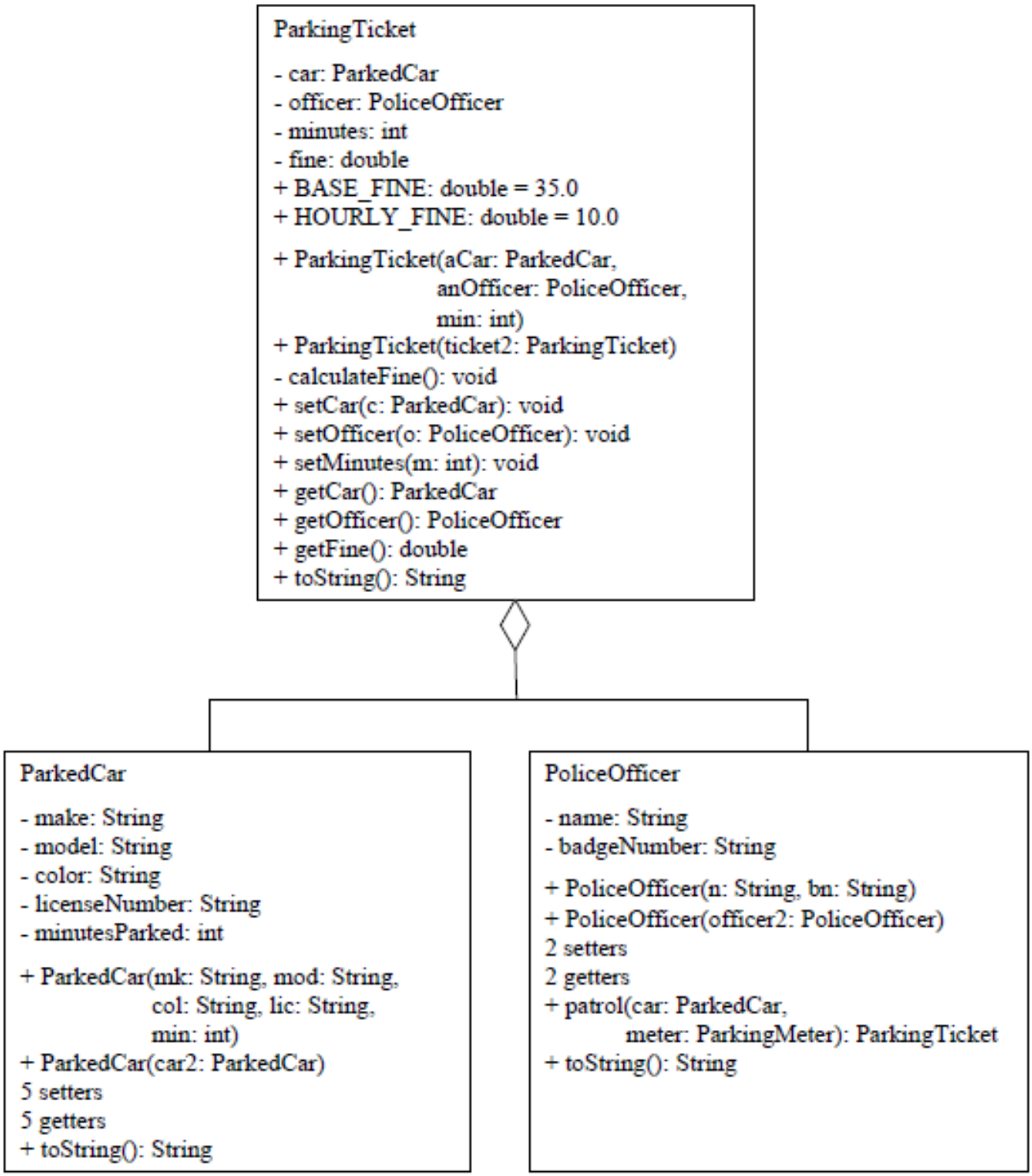
- The **ParkingMeter** Class: This class should simulate a parking meter. The class's only responsibility is as follows:
  - To know the number of minutes of parking time that has been purchased.

- The **ParkingTicket** Class: This class should simulate a parking ticket. The class's responsibilities are as follows:
  - To report the make, model, color, and license number of the illegally parked car.
  - To report the amount of the fine, which is $35 for the first hour or part of an hour that the car is illegally parked, plus $10 for every additional hour or part of an hour that the car is illegally parked. Use a public static constant to represent the base fine and hourly fine.
  - To report the name and badge number of the police officer issuing the ticket.

- The **PoliceOfficer** Class: This class should simulate a police officer inspecting parked cars. The class's responsibilities are as follows:
  - To know the police officer's name and badge number.
  - To examine a ParkedCar object and a ParkingMeter object, and determine whether the car's time has expired. If so, issue a parking ticket (generate a ParkingTicket object).

For each of the above classes, also provide appropriate constructors, setters, getters, and toString() methods. Make sure your implementation will not create security holes that can allow code outside the class to modify private data inside the class. So, when you are dealing with reference variables in constructors, setters, and getters, make sure you perform deep copies.

The following is the UML diagram for ParkingTicket, ParkedCar, and PoliceOfficer. Note that fine is calculated based on minutes. In our design, the constructor of ParkingTicket shall call the method calculateFine() in order to initialize fine.


ParkingTicket
- car: ParkedCar
- officer: PoliceOfficer
- minutes: int
- fine: double
+ BASE_FINE: double = 35.0
+ HOURLY_FINE: double = 10.0
+ ParkingTicket(aCar: ParkedCar, anOfficer: PoliceOfficer, min: int)
+ ParkingTicket(ticket2: ParkingTicket)
- calculateFine(): void
+ setCar(c: ParkedCar): void
+ setOfficer(o: PoliceOfficer): void
+ setMinutes(m: int): void
+ getCar(): ParkedCar
+ getOfficer(): PoliceOfficer
+ getFine(): double
+ toString(): String
ParkedCar
- make: String
- model: String
- color: String
- licenseNumber: String
- minutesParked: int
+ ParkedCar(mk: String, mod: String, col: String, lic: String, min: int)
+ ParkedCar(car2: ParkedCar)
5 setters
5 getters
+ toString(): String
PoliceOfficer
- name: String
- badgeNumber: String
+ PoliceOfficer(n: String, bn: String)
+ PoliceOfficer(officer2: PoliceOfficer)
2 setters
2 getters
+ patrol(car: ParkedCar, meter: ParkingMeter): ParkingTicket
+ toString(): String

Partial demo class code:

```
public class ParkingTicketDemo
{
	public static void main(String[] args)
	{
		System.out.println("\n... set minutes purchased to 60, and");
		System.out.println(" parked for 125 minutes...");

		// Create a ParkedCar object.
		// The car was parked for 125 minutes.
		ParkedCar car = new ParkedCar("Volkswagen", "2002", "Red", "3RHZ147", 125);

		// Create a ParkingMeter object. 60 minutes were purchased.
		ParkingMeter meter = new ParkingMeter(60);

		// Create a PoliceOfficer object.
		PoliceOfficer officer = new PoliceOfficer("Joe Friday", "4788");

		// Let the officer patrol.
		ParkingTicket ticket = officer.patrol(car, meter);

		// Did the officer issue a ticket?
		if (ticket != null)
			System.out.println(ticket);
		else
			System.out.println("No crimes committed!");
		// Test case 2
		…
	}
}
```

Submission: To be submitted as a one zip file **FirstnameLastname-141-A5.zip** on Blackboard (under Assignments). Include all the .java files used in your program in the zip file and the output as a text file.

- ParkedCar.java
- ParkingMeter.java
- ParkingTicket.java
- PoliceOfficer.java
- ParkingTicketDemo.java
- Assignment5Output.txt

In addition, each .java file must contain the following information at the top:

//Your name
//CS141
//Assignment 5
//Date

Due: Friday June 1, 11.59 PM

DEMO OUTPUT:

```
... set minutes purchased to 60, and
        parked for 125 minutes...
Car Data:
Make: Volkswagen
Model: 2002
Color: Red
License Number: 3RHZ147
Minutes Parked: 125

Officer Data:
Name: Joe Friday
BadgeNumber: 4788

Minutes Illegally Parked: 65
Fine: $45.00

... change minutes parked to 60...
No crimes committed!

... change minutes parked to 61...
Car Data:
Make: Volkswagen
Model: 2002
Color: Red
License Number: 3RHZ147
Minutes Parked: 61

Officer Data:
Name: Joe Friday
BadgeNumber: 4788

Minutes Illegally Parked: 1
Fine: $35.00

... change car model to 2010...
Car Data:
Make: Volkswagen
Model: 2010
Color: Red
License Number: 3RHZ147
Minutes Parked: 61

Officer Data:
Name: Joe Friday
BadgeNumber: 4788

Minutes Illegally Parked: 1
Fine: $35.00

... change minutes purchased to 70...
No crimes committed!
```