CSE102L Computer Programming Exercise 2

Due: March 24th, 2022 until 12pm

Exercise Definition

In this exercise you are required to write following classes.

1. Engine
    a. Attributes
        i. gearCount: int
        ii. currentGear: int
    b. Methods
        i. Constructor that takes number of gears and defaults currentGear to 0
        ii. getGearCount(): int
        iii. getCurrentGear(): int
        iv. shiftUp(): None
        v. shiftDown(): None
        vi. toString(): returns a String that gives information about current gear and number of gears

(Bonus point if you figure out a way to make every method of Engine class accessible or useable by Vehicle class but not accessible or useable from Main method) (Main method should be under the public class {yourStudentNumber})

2. Vehicle
    a. Attributes
        i. engine: Engine
        ii. speed: int
        iii. maxSpeed: int
        iv. changeInSpeed: int
        v. gasTankCapacity: int
        vi. currentGasInTank: double
    b. Methods
        i. Constructor that takes number of gears, maximum speed of the vehicle, change in speed, and gas tank capacity that sets corresponding attributes and creates an engine instance with given number of gears. Current gas in tank attribute will be set randomly, between 5 and 15 less from gas tank capacity.
        ii. setGear(): None; this method **is not overridable.** (Tip: use shiftUp and shiftDown according to speed)
        iii. accelerate(): None; changes speed according to changeInSpeed attribute. (Bonus point if you write this method according to **method chaining**)

iv. decelerate(): None; changes speed according to changeInSpeed attribute. (Bonus point if you write this method according to **method chaining**)
v. setter and getter methods for changeInSpeed attribute
vi. getGasPercentage(): double
vii. consumeGas(): None; this method is **not overridable.** Every Car instance will consume 2, every Motorcycle instance will consume 3, every Bus instance will consume 5 times current gear. (Do not ask any question about this method. Write like what you understand from this description. Tip: use **instanceof).**
viii. Getter method for speed.
ix. refuel(): None
x. toString(): String; returns a string of class name(Bonus point if you **do not write** class name manually), speed, change in speed, maximum speed, gas percentage information and engine information

(Bonus point if user cannot instantiate an object from Vehicle class)

3. Car – a child of Vehicle
    a. Attributes
        i. NUMBER_OF_GEARS: int, constant, class's attribute (can be 6)
        ii. GAS_TANK_CAPCITY: int, constant, class's attribute (can be 50),
        iii. doorCount: int = Assign a default value
    b. Methods
        i. No-argument constructor that sets max speed and changeInSpeed to any default value (Be reasonable).
        ii. Constructor that takes maximum speed and change in speed and uses parent's constructor
        iii. Constructor that takes maximum speed, change in speed and door count.
        iv. accelerate(int changeInSpeed): None; changes speed according to changeInSpeed attribute. Print a warning message if fuel is low (Bonus point if you write this method according to **method chaining**)
        v. decelerate(int changeInSpeed): None; changes speed according to changeInSpeed attribute. (Bonus point if you write this method according to **method chaining**)
        vi. refuel(): double; always refuels for 50₺ worth of fuel. Returns amount of fuel taken
        vii. toString(): String; add doorCount to parent class's toString()

4. Motorcycle – same with Car except:
    a. Attributes
        i. Instead of doorCount, will have color: String
    b. Methods
        i. toString(): String; according to attribute change

ii. refuel(): double, always refuels to full. Returns amount of fuel taken

5. Bus – same with Car except:
    a. Attributes
        i. Instead of doorCount, will have standingPassengerCount: int
    b. Methods
        i. toString(): String; according to attribute change
        ii. refuel(): double, always refuels to 80% of fuel capacity. Returns amount of fuel taken
6. GasStation
    a. Attributes
        i. vehicles: ArrayList<Vehicle>
        ii. costPerLiter: double, constant, class's attribute = 20.40
        iii. vehicleLimit: int
    b. Methods
        i. Constructor that takes vehicleLimit and sets it
        ii. No-argument constructor that sets a default value to vehicleLimit
        iii. fuelUpAll(): None, When the limit is reached, fuels vehicles. For each vehicle calculates gas price and prints it with toString method of vehicle's
        iv. fuelUpAll(ArrayList<Vehicle>): None, fuels vehicles. For each vehicle calculates gas price and prints it with toString method of vehicle's

Write your **own** code. Your file name should be **Ex02_YourStudentNumber.java** (e.g. Ex01_202051056016.java)**. Remember, public class's name must be the same with your file name.** You may add any additional code, methods, attributes, or classes that you think would be necessary to complete the assignment.