Assignment 1 - Object Oriented Programming

Daniel Hannon (19484286)

October 2020

1 Overview

Within this assignment we have to model a car. There's three components in particular that we look at: Car, Engine, and Wheel.
For the car, I imagined it as the body of the car, holding the spedometer and keeping track of distance. As we were presented with the decision as to the location in which we store the fuel within the object of the car, I decided to store it in the Engine as it is more vital for the Engine to have direct access to the fuel than it is the car.
As the engine is a solid object, I felt that you didn't require to modify the basic parameters of the engine for they would not change in a solid block engine, the same applied for the wheel. At Initilaisation of Wheel, I felt it would make sense to calculate circumference as it would remain a constant for the entire lifecycle of the wheel.
I wrote error handling code for anything that requires there to be an object in a certain location within the parent object to prevent the program from crashing as much as possible. Additionally, there were a few logic checks within the code to prevent the presence of negative values in locations like TPL, Distance, and Fuel Level.

2 Code

2.1 Car.java

```
1  public class Car {
2          /*Private variables*/
3          private String name;
4          private double distance;
5          private double totalKm;
6          private Engine engine;
7
8          /*Initialization*/
9          public Car(String cname) {
10                 this.name = cname;
11                 this.distance = 0;
12                 this.totalKm = 0;
13                 this.engine = null;
14         }
15
16         /*Functions of the Car*/
17         public void add(Engine nengine) {
18                 this.engine = nengine;
19         }
20         /* Overloading is used to avoid typechecking */
21         public void add(Wheel nwheel) {
22                 try {
23                         engine.setWheel(nwheel);
24                 } catch(Exception e) {
25                         System.out.println("ERROR: No Engine");
26                 }
27         }
28
29         public void drive() {
30                 try {
31                         this.distance = engine.move();
32                         this.totalKm+=this.distance;
33                 } catch(Exception e) {
```

```
                        System.out.print("ERROR: No Engine");
                }
        }

        public void printState() {
                /*copies instance of wheel from engine to reduce total amount of lines of code*/
                try {
                        Wheel tempWheel = engine.getWheel();
                        System.out.println("Configuration: " + this.name);
                        System.out.println("Engine name: " + engine.getEnginemodel());
                        System.out.printf("Engine turns per liter: %.2f\n",
                         ↪  engine.getTpl());
                        System.out.printf("Engine Total turn count: %.2f\n",
                         ↪  engine.getTotalturns());
                        System.out.println("Wheel name: " + tempWheel.getBrand());
                        System.out.printf("Wheel radius: %.2f\n", tempWheel.getRadius());
                        System.out.printf("Wheel circumference (distance per turn):
                         ↪  %.2f\n", tempWheel.turn());
                        System.out.printf("Distance this trip: %.2f\n", this.distance);
                        System.out.printf("Total distance Travelled: %.2f\n",
                         ↪  this.totalKm);
                        System.out.printf("Current fuel Status: %.2f\n\n",
                         ↪  engine.getFuel());
                } catch(Exception e) {
                        System.out.println("ERROR: No Engine");
                }
        }

        /*Setters*/
        public void setDistance(double dist) {
                if(dist > 0) {
                        this.distance = dist;
                } else {
                        System.out.println("You cannot set a negative distance!");
                }
        }
        public void setFuel(double nfuel) {
                try {
                        if (nfuel > 0) {
                                engine.setFuel(nfuel);
                        } else {
                                System.out.println("You Cannot take Fuel Out!");
                        }
                } catch (Exception e) {
                        System.out.println("ERROR: No Engine");
                }
        }
        public double getFuel() {
                return engine.getFuel();
        }
}
```

2.2 Engine.java

```java
public class Engine {
	/*Private variables*/
	private Wheel wheel;
	private String engineModel;
	private double tpl;
	private double totalTurns;
	private double fuelLevel;

	/*Initialization*/
	public Engine(String nenginemodel, double ntpl) {
		if (ntpl < 0) {
			ntpl *= -1;
		}
		this.engineModel = nenginemodel;
		this.tpl = ntpl;
		this.totalTurns = 0;
		this.fuelLevel = 0;
		this.wheel = null;
	}

	/*Methods*/
	public double move() {
		try {
			double distance = (tpl * fuelLevel) * wheel.turn();
			this.totalTurns += tpl * fuelLevel;
			this.fuelLevel = 0;
			return distance;
		} catch(Exception e) {
			System.out.println("ERROR: No Wheels!");
			return 0;
		}
	}

	/*Getters*/
	public String getEnginemodel() {
		return this.engineModel;
	}
	public Wheel getWheel() {
		return this.wheel;
	}
	public double getTpl() {
		return this.tpl;
	}
	public double getFuel() {
		return this.fuelLevel;
	}

	public double getTotalturns() {
		return this.totalTurns;
	}

	/*Setters, you cannot individually change tpl
	or engine model as that would not make sense*/
	public void setWheel(Wheel nwheel) {
		this.wheel = nwheel;
	}
	public void setFuel(double nfuel) {
		/* This is done to simulate a more realisitc
```

```
                model of a car where you can add fuel more than once */
                this.fuelLevel += nfuel;
        }
}
```

2.3 Wheel.java

```
public class Wheel {
        /*Private variables*/
        private String brand;
        private double radius;
        private double circumference;

        /*Initilization*/
        public Wheel(String nbrand, double nradius) {
                if (nradius < 0) {
                        nradius *= -1;
                }
                this.brand = nbrand;
                this.radius = nradius;
                this.circumference = 2 * Math.PI * radius;
        }

        /*Getters*/
        public double getRadius() {
                return this.radius;
        }
        public String getBrand() {
                return this.brand;
        }
        /*As engine calls turn(), I called getCircumference() turn() instead*/
        public double turn() {
                return this.circumference;
        }

        /*Realisitcally this would not have any
        setters beyond the initilisation as it is a tyre*/
}
```

2.4 TestCar.java

```
public class TestCar {
        public static void main(String[] args) {
                Car car = new Car("Toyota Corolla");
                Engine engine = new Engine("2ZZ-GE",57);
                car.add(engine);
                Wheel wheel =  new Wheel("Michelin Pilot",21);
                car.add(wheel);
                car.setFuel(500);
                System.out.println("Current Fuel: " + car.getFuel());
                car.drive();
                car.printState();
                car.setFuel(125);
                System.out.println("Current Fuel: " + car.getFuel());
```

```
14                  car.drive();
15                  car.printState();
16                  /*Change tyres for the craic*/
17                  wheel = new Wheel("Pirelli P1",22);
18                  car.add(wheel);
19                  car.setFuel(250);
20                  System.out.println("Current Fuel: " + car.getFuel());
21                  car.drive();
22                  car.printState();
23          }
24  }
```

3 Output

```
TERMINAL   PROBLEMS   OUTPUT   DEBUG CONSOLE

[daniel@Void3 Assignment 1]$ java TestCar
Current Fuel: 500.0
Configuration: Toyota Corolla
Engine name: 2ZZ-GE
Engine turns per liter: 57.00
Engine Total turn count: 28500.00
Wheel name: Michelin Pilot
Wheel radius: 21.00
Wheel circumference (distance per turn): 131.95
Distance this trip: 3760486.41
Total distance Travelled: 3760486.41
Current fuel Status: 0.00

Current Fuel: 125.0
Configuration: Toyota Corolla
Engine name: 2ZZ-GE
Engine turns per liter: 57.00
Engine Total turn count: 35625.00
Wheel name: Michelin Pilot
Wheel radius: 21.00
Wheel circumference (distance per turn): 131.95
Distance this trip: 940121.60
Total distance Travelled: 4700608.01
Current fuel Status: 0.00

Current Fuel: 250.0
Configuration: Toyota Corolla
Engine name: 2ZZ-GE
Engine turns per liter: 57.00
Engine Total turn count: 49875.00
Wheel name: Pirelli P1
Wheel radius: 22.00
Wheel circumference (distance per turn): 138.23
Distance this trip: 1969778.59
Total distance Travelled: 6670386.60
Current fuel Status: 0.00

[daniel@Void3 Assignment 1]$
```