Interfaces Lab 01: Defining Interfaces and Abstract Classes

Objective

- To practice implementing interfaces and extending abstract classes.

Unit tests

You are expected to write unit tests for each class you implement.

Instructions

class One implements Part:

Create an abstract class called `Car` and an interface called `Driveable`. Driveable objects must have a drive() method, `Car` objects must have `turnOn()`, `turnOff()` and `refuel()` methods. Cars should also have a fuel level that can be checked.

Create at least three models of Car, each with their own class. Some of your cars can be fueled by gas, electricity, or something else (feel free to get creative). Cars should only drive when they are on, and must be turned off to refuel.

In a main method, declare an array of Car objects and populate it with at least one instance of each of your subclasses. Loop through the array to call all of the methods defined in the interface.

class Two implements Part:

Create a `FuelStation` abstract class or interface and use it in two subclasses, `GasStation` and `PowerStation`. Your FuelStations should have a finite amount of fuel and should include appropriate methods to refuel cars.

Update your main to drive each car to the appropriate FuelStation, fuel up, and drive back.