CICS-160 Fall 2023

Assignment 3: Linked Lists Operations

Due on Sunday November 5 2023

Learning Goals:

This assignment is designed to for you to demonstrate the ability to implement linked lists, use lists of any type, and engage with inheritance. We will do that by implementing the system similar but not identical to the one you designed during assignment #2.

Overview

For this project, you will be implementing many parts of a system similar to the one you designed in assignment 2. Assignment 2 dealt only with objects of class Car; the system you will implement in this part will need handle objects of type Car, type ElectricCar, and type GasolineCar, all in the same list. Class ElectricCar and class GasolineCar inherit from class Car. If you want, you can use, as a starting point, any idea/concept/code we have used or talked about in class/labs for those classes. Contrary to assignment 2, instead of keeping objects inside an object of class FleetOfCars, you will keep them in an object of class LinkedList, which you will develop. Any point in this document where lists are mentioned, you should understand that to mean linked lists, of the type we have to declare ourselves, as opposed to the Python built-in lists. Y**our implementation of LinkedList should allow for objects of type LinkedList to store ANY TYPE OF ELEMENT, including, for example, integers, as covered in class. This will require you to implement a class called Node.**

Your tasks

Implement the following ADTs for classes Car, ElectricCar, GasolineCar, and LinkedList.

Car:

- __init__(makeAndModel="none assigned", numberOfDoors=4, maximumNumberOfPassengers=5) 5 points
  - initializes a Car object
- setMakeAndModel(string) 1 point
  - sets the makeAndModel of the object to be equal to the string parameter provided
- getMakeAndModel()→ string 1 point
  - returns the makeAndModel of the Car as a string
- getMaximumNumberOfPasengers→ Integer 1 point
  - returns the maximum number of passengers the car can carry, an integer

ElectricCar:

- __init__(makeAndModel="none assigned", numberOfDoors=4, maximumNumberOfPassengers=5, batterySize = -1) 5 pts
  - initializes an ElectricCar object
- getBatterySize()→ int 1 point
  - returns the battery size the ElectricCar, as an integer
- setBatterySize(int) 1 point
  - sets the batterySize of the ElectricCar to be equal to the integer parameter provided

GasolineCar:

- __init__(makeAndModel="none assigned", numberOfDoors=4, maximumNumberOfPassengers=5, gasTankSize = -1)
  - initializes a GasolineCar object 5 points
- getGasTankSize()→ int 1 point
  - returns the GasolineCar, as an integer
- setGasTankSize(int) 1 point
  - sets the gasTankSize of the GasolineCar object to be equal to the int parameter provided

LinkedList:

- __init__() 2 point
  - initializes an empty linked list
- add(object) 3 points
  - appends the object provided as parameter to the list
- insert(int, object) 8 points
  - places the provided object at index equal to the provided integer
- length() → int 2 points
  - returns the length of the list as an integer
- __getitem__(int) → Object 5 points
  - returns the object stored at the index of the list equal to the integer provided as parameter
- delete(int) 10 points
  - deletes from the list the element at the index provided as parameter

Things that will be checked manually:

No Python’s built-in lists are used.	20 points
Class ElectricCar and class GasolineCar inherit from Car.	5 points
ElectricCar and GasolineCar do not implement things they can inherit from Car.	5 points.
Code includes unittests for LinkedList.delete()	5 points.
Code includes unittests for LinkedList.insert()	5 points
__str__ methods of each of the classes display atribute data	8 points

Submit your work in at least five files: Car.py, ElectricCar.py, GasolineCar.py, Node.py, and LinkedList.py. If you want, you can also upload a main program that implements the menu system, but it is not necessary. You can have a separate file that includes tests, or you can have unittests as a separate file.

NB: for the Gradescope autograder to function properly, it needs to find four different files: Car.py, ElectricCar.py, GasolineCar.py, Node.py, and LinkedList.py. If you want to test things as you go along, before completing all four classes, create empty files and upload them to with your submissions.