CICS-160 Spring 2023

Assignment 3: Linked Lists Operations and Inheritance

Due on Sunday April 16 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 Person; the system you will implement in this part will need handle objects of type Person, type Student, and type Employee, all in the same list. Class Employee and class Student inherit from class Person. If you want, you can use, as a starting point, any idea/concept/code we have used or talked about in class for those classes. Contrary to assignment 2, instead of keeping objects inside an object of class Persons, you will keep them in an object of class LinkedList. 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 Person, Student, Employee, and LinkedList.

Person:

- __init__(newName="none", address="none", phone="999-999-9999") 5 points
  - initializes a Person object
- setName(string) 1 point
  - sets the name of the object to be equal to the string parameter provided
- getName()→ string 1 point
  - returns the name of the Person as a string
- getPhone()→ string 1 point
  - returns the phone number of the Person as a string

Student:

- __init__(newName="none", address="none", phone="999-999-9999", year=9999) 5 pts
  - initializes a Student object
- getGraduationYear()→ int 1 point
  - returns the graduation year of the Student, as an integer
- setGraduationYear(int) 1 point
  - sets the graduation year of the Student to be equal to the integer parameter provided

Employee:

- __init__(newName="none", address="none", phone="999-999-9999", department=”not assigned”)
  - initializes an Employee object 5 points
- getDepartment()→ string 1 point
  - returns the Employee department, as a string
- setDepartment(string) 1 point
  - sets the department of the Employee object to be equal to the string parameter provided

LinkedList:

- __init__() 1 point
  - initializes an empty linked list
- search(string)→LinkedList 5 points
  - returns a linked list of objects with a name attribute equal to the provided string
- add(object) 2 points
  - appends the object provided as parameter to the list
- insert(int, object) 5 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) → Person 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 Employee and class Student inherit from Person.	5 points
Employee and Person do not implement things they can inherit from Person.	5 points.
Code includes tests for LinkedList.delete()	5 points.
Code includes tests for LinkedList.insert()	5 points
__str__ methods of each of the classes display required data	8 points

Submit your work in five files: Person.py, Student.py, Employee.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.

NB: for the Gradescope autograder to function properly, it needs to find four different files: Person.py, Student.py, Employee.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.