**License and Code Details**

Code written by: Yousef Osama

Student at the Faculty of Computers and Information, Egyptian Chinese University

The code took **17** hours to write after completing a full course in Python.

Date: 25-June-2024

# Code Explanation:

**\_\_init\_\_(self, root):** This is the constructor method for the LinkList class. It initializes the properties when creating an instance of the class, including the 'root' which is the Tkinter window.

**heading\_with\_label\_subheading(self):** This method creates the header and subheading for the linked list interface.

**make\_start\_with\_other(self):** This method is likely used to initiate other configurations or start operations related to the linked list.

**make\_btn(self):** This method creates buttons on the Tkinter window for the user interface.

**set\_of\_input\_method(self):** This method defines the input methods available to the user, such as text boxes or buttons.

**checking\_of\_existence(self):** This method checks for the existence of a specific element or node in the linked list.

**make\_node\_with\_label(self, take\_notation):** This method creates a new node in the linked list with a label or notation provided as input.

**input\_take(self, take\_notation):** This method captures the user's input for adding a new node in the linked list.

**make\_main\_container\_with\_node\_value\_set\_and\_next\_arrow\_creation(self, take\_notation):** This method sets up the main container and creates nodes with values, linking them with arrows to represent the linked list.

**calculate\_average(self):** This method calculates the average of values in the linked list, likely used to compute the mean of node data.

**update\_node\_input(self):** This method updates the input for a specific node in the list.

**update\_node(self):** This method performs the update operation on a specific node in the linked list.

**perform\_update(self, position):** This method updates the node at the given position in the list.

**search\_node\_input(self):** This method accepts input to search for a specific node in the linked list.

**search\_node(self):** This method searches for a node in the linked list based on user input.

**insert\_node(self, take\_notation):** This method inserts a new node at the appropriate position in the linked list.

**reset\_with\_store(self, take\_notation):** This method resets the list with stored values, possibly to clear or reinitialize the list.

**delete\_last\_node(self, locator):** This method deletes the last node in the linked list.

**delete\_first\_node(self):** This method deletes the first node in the linked list.

**delete\_single\_node\_infrastructure(self):** This method sets up the infrastructure for deleting a single node in the linked list.

**delete\_single\_node(self):** This method deletes a specific node based on the user input.