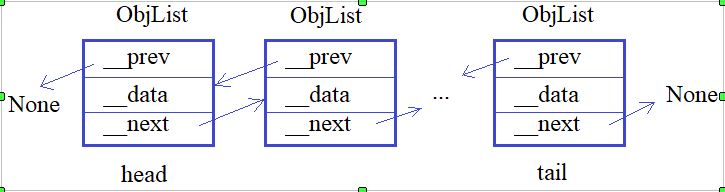
Видео-разбор подвига (решение смотреть только после своей попытки): <https://youtu.be/6-xKuQp9b7Y>

Подвиг 5. Объявите класс LinkedList (связный список) для работы со следующей структурой данных:



Здесь создается список из связанных между собой объектов класса ObjList. Объекты этого класса создаются командой:

obj = ObjList(data)

где data - строка с некоторой информацией. Также в каждом объекте obj класса ObjList должны создаваться следующие локальные атрибуты:

\_\_data - ссылка на строку с данными;  
\_\_prev - ссылка на предыдущий объект связного списка (если объекта нет, то \_\_prev = None);  
\_\_next - ссылка на следующий объект связного списка (если объекта нет, то \_\_next = None).

В свою очередь, объекты класса LinkedList должны создаваться командой:

linked\_lst = LinkedList()

и содержать локальные атрибуты:

head - ссылка на первый объект связного списка (если список пуст, то head = None);  
tail - ссылка на последний объект связного списка (если список пуст, то tail = None).

А сам класс содержать следующие методы:

add\_obj(obj) - добавление нового объекта obj класса ObjList в конец связного списка;  
remove\_obj(indx) - удаление объекта класса ObjList из связного списка по его порядковому номеру (индексу); индекс отсчитывается с нуля.

Также с объектами класса LinkedList должны поддерживаться следующие операции:

len(linked\_lst) - возвращает число объектов в связном списке;  
linked\_lst(indx) - возвращает строку \_\_data, хранящуюся в объекте класса ObjList, расположенного под индексом indx (в связном списке).

Пример использования классов (эти строчки в программе писать не нужно):

linked\_lst = LinkedList()

linked\_lst.add\_obj(ObjList("Sergey"))

linked\_lst.add\_obj(ObjList("Balakirev"))

linked\_lst.add\_obj(ObjList("Python"))

linked\_lst.remove\_obj(2)

linked\_lst.add\_obj(ObjList("Python ООП"))

n = len(linked\_lst)  # n = 3

s = linked\_lst(1) # s = Balakirev

P.S. На экран в программе ничего выводить не нужно.