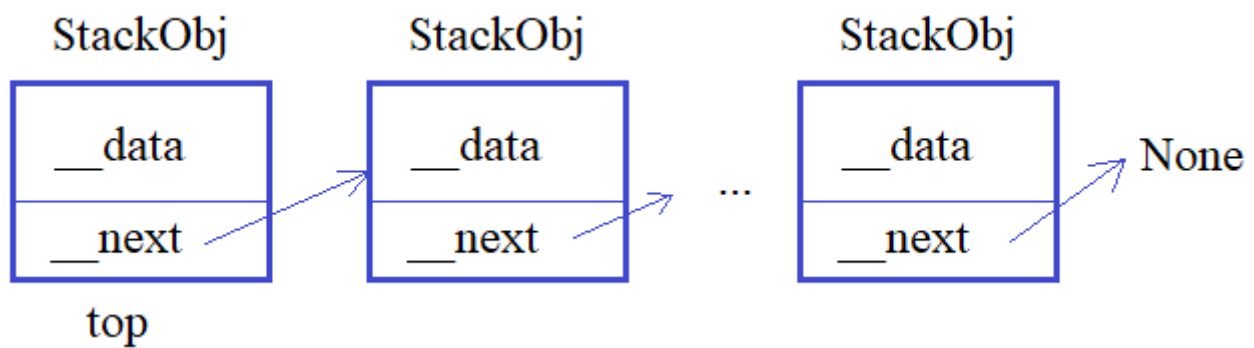


Видео-разбор подвига (решение смотреть только после своей попытки):

<https://youtu.be/mg4b8nhVDKY> (<https://youtu.be/mg4b8nhVDKY>)

Подвиг 6. Реализуйте односвязный список (**не список Python**), когда один объект ссылается на следующий и так по цепочке до последнего:



Для этого объявите в программе два класса:

StackObj - для описания объектов односвязного списка;

Stack - для управления односвязным списком.

Объекты класса StackObj предполагается создавать командой:

```
obj = StackObj(данные)
```

Здесь данные - это строка с некоторым содержимым. Каждый объект класса StackObj должен иметь следующие локальные приватные атрибуты:

___data - ссылка на строку с данными, указанными при создании объекта;

___next - ссылка на следующий объект класса StackObj (при создании объекта принимает значение None).

Также в классе StackObj должны быть объявлены объекты-свойства:

next - для записи и считывания информации из локального приватного свойства ___next;

data - для записи и считывания информации из локального приватного свойства ___data.

При записи необходимо реализовать проверку, что ___next будет ссылаться на объект класса StackObj или значение None. Если проверка не проходит, то ___next остается без изменений.

Класс Stack предполагается использовать следующим образом:

```
st = Stack() # создание объекта односвязного списка
```

В объектах класса Stack должен быть локальный публичный атрибут:

top - ссылка на первый объект односвязного списка (если список пуст, то top = None).

А в самом классе Stack следующие методы:

push(self, obj) - добавление объекта класса StackObj в конец односвязного списка;

pop(self) - извлечение последнего объекта с его удалением из односвязного списка;

`get_data(self)` - получение списка из объектов односвязного списка (список из строк локального атрибута `__data` каждого объекта в порядке их добавления).

Пример использования классов `Stack` и `StackObj` (эти строчки в программе писать не нужно):

```
st = Stack()
st.push(StackObj("obj1"))
st.push(StackObj("obj2"))
st.push(StackObj("obj3"))
st.pop()
res = st.get_data()      # ['obj1', 'obj2']
```

P.S. В программе требуется объявить только классы. На экран ничего выводить не нужно.

To solve this problem please visit

<https://stepik.org/lesson/701984/step/7>