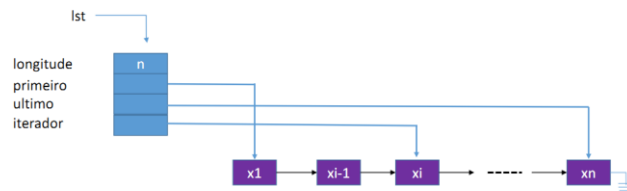


lst = < x1,...,[xi],...xn >



ListNode



data nextNode (guarda o endereço/referência/valor do proximo Node)

```
class ListNode:
# atributos, propriedades ou estados
self.data
self.nextNode
# métodos ou funções que manipulam o comportamento do objeto
def __init__(self, data, nextNode=None)
def getData(self):
def setData(self, val):
def getNextNode(self):
def setNextNode(self, val):

class SinglyLinkedListIterator:
# atributos, propriedades ou estados
self.firstNode (guarda o endereço/referência/valor do primeiro Node)
self.lastNode (guarda o endereço/referência/valor do ultimo Node)
self.iterator (guarda o endereço/referência do Node sob o iterador)
self.size (guarda o número de elementos da Lista)
# métodos ou funções que manipulam o comportamento do objeto
def __init__(self, firstNode=None)
def getSize(self):
def get_firstNode(self):
def get_lastNode(self):
def get_iterator(self):
def setSize(self, size):
def set_firstNode(self, firstNode=None):
def set_lastNode(self, lastNode=None):
def set_iterator(self, iterator = None):
def addNode(self, data):add Node depois do it e it fica neste Node
def insNode(self, data):insere Node antes do it e it fica neste Node
def elimNode(self): # elimina Node sob it e it avanca p/ prox Node
def first_Node(self): # coloca o it sobre o primeiro Node da Lista
def last_Node(self): # coloca o iterador sobre o último Node da Lista
def nextNode(self): # avança it uma pos. Se it no ult Node, it=None
def posNode(self, position): poe it em <=1 pos <=size, senao it=None
def undefinedIterator(self): # True se o it =None e False o contrario
```

Criando uma Lista vazia: myList = SinglyLinkedListIterator()

Criando um Node: newNode = ListNode(data, None)

Criando uma Lista com um Node: myList2 = SinglyLinkedListIterator(newNode)