

Node

Deze opgave lijkt er op de link-opgave voor C++. Echter Python en C++ zijn verschillende programmeertalen, dus de C++ oplossing is niet gelijk aan de Python oplossing.

Deze opgave gaat over het manipuleren van lijsten van Objecten. Natuurlijk kun hiervoor de standaard `list` class van Python gebruiken, maar het doel van deze opgave is te zien hoe de `list` klasse kan worden geïmplementeerd.

We definiëren een class `List` en een class `Node`. De class `node` is een hulp-klasse en wordt daarom binnen de class `List` gedefinieerd

```
class List:
    class Node:
        """ Een node bevat een value (val) en een next-ptr (lnk) """
        def __init__(self, val, lnk=None):
            self.val = val
            self.lnk = lnk

    def __init__(self):
        """ Initialeer de root van de lijst op None """
        self.root = None
```

De class `List` heeft een aantal methoden.

```

def toon(self):
    """ returns a string containing de elements-value of list.
        The order is de order in the list, values are seperated by ","
        and enclosed by "[" and "]".
        An example "[een,twee,zes]".
    """

def append(self, val):
    """ Appends a node with value `val` to the end of the list.
    """

def insert(self, val):
    """ Inserts a node with value `val` to the beginning of the list.
    """

def addSorted(self, val):
    """ Add a node with value `val` to list.
        It's place is based on value `val`.
        All the items in de list are sorted on their value.
        An example: We have "[een,zes]"
        Adding "twee" will give "[een,twee,zes]"
        To work well the list must be in order.
    """

def delete(self, val):
    """ Remove the node with value `val` from the list.
    """

```

Programma-files

Bovenstaande code is te vinden in `SMPpy23node.py`.

Inleveren

Stuur de file `SMPpy_23_node.py` op, met daarin de class `List` en `Node` met de uitwerking van de methoden

Punten

Dit programma is **vijf** punten waard.