CS240

Nathan Sprague

September 19, 2012

Three Confusingly Related Concepts

- Abstract Data Type - Programmer-defined data type that specifies a set of operations.
    - ADT is defined independently of its implementation.
- Data Structure - Actual data organization that underlies the implementation of an ADT.
- Python Type (or Class) - Complete implementation of an ADT using some data structure.

We Will Be Talking About “Lists”

- Don’t be confused!
- Abstract Data Type - “List ADT” specification for container types that hold a set of elements in linear order and can grow or shrink to hold an arbitary number of items.
- Data Structure - “Linked List” refers to a data structure that organizes a collection of elements using linked nodes. *We won’t talk about these for a while.*
- Python Type (or Class) - “Python List” is a Python type that implements a List ADT.

PA#2

- Let's look at the Vector (List) ADT for PA#2

PA#2 Data Structure

- Let’s look at t_array.Array

The Long Way to Code a For Loop

```
it = iter(myList)

while True:
    try:
        item = it.next()
        doBodyOfLoop(item)
    except StopIteration:
        break
```

Example Iterator Class

```python
class ReverseIterator(object):
    def __init__(self, theList):
        self._list = theList
        self._index = len(theList)

    def __iter__(self):
        return self

    def next(self):
        self._index -= 1
        if self._index < 0:
            raise StopIteration()
        return self._list[self._index]
```