CSE 341, Winter 2005, Assignment 6
Due: Sunday, March 6, 9:00 PM

Last updated: 3-2-05

For this assignment you will write a simple class implementing a linked list. This should have been covered in CSE 143, so the purpose is mainly to help you get used to working with SmallTalk and the Squeak environment. The exact details of the implementation are left up to you. You must write a class with all the methods listed below that work as specified, but you may use additional helper methods and classes. You may not use any of the collection/array/list classes provided by SmallTalk.

Problems

Create the category HW6 in the Squeak class browser. Within this category, create a subclass of `Object` called `MyList` and give it the following methods:

1. `add:`
   Takes a value **`v`** and appends it to the end of the list, returning the appended element.

2. `length`
   Returns the length of the list.

3. `at:`
   Takes an index **`i`** and returns the element at that index. The first element of the list is at index 0. If the index is out of the bounds of the list, signal a meaningful error ("index out of bounds" is meaningful, but "message not understood" is not).

4. `at:put:`
   Takes an index **`i`** and a value **`v`** and changes the element at that index to be the new value. Returns the value previously stored at that index. If the index is out of bounds, signal an error. For the empty list, the index 0 *is* out of bounds.

5. `removeAt:`
   Takes an index **`i`** and removes the corresponding element from the list, returning the value stored there. If the index is out of bounds, signal an error.

6. `at:insert:`
   Takes an index **`i`** and a value **`v`** and inserts that value into the specified index of the list, pushing the value previously stored there and everything following it to a higher index and returning the value inserted. Keep in mind that inserting at index 0 is equivalent to adding the value to the beginning of the list, and inserting at *n*, where *n* is the length of the list, is equivalent to adding the value to the end of the list. This means that inserting just past the end of the list or at index 0 of the empty list is legal and well-defined. If the index is otherwise out of bounds, signal an error.

7. `forEach:`
   Takes a code block `b` of a single argument and invokes the block once with each value in the list, starting with the first element and going in order. Take notice that this essentially gives you a looping construct to iterate over a list, though the semantics and syntax are no different from the rest of SmallTalk. In Java, you would use an

iterator object with a while loop to accomplish the same task. Assuming some `MyList` **`lst`** is a list of numbers, the following code will compute the sum and leave it in the variable **`x`**:

```
x := 0.
lst forEach: [:v |
    x := x + v
].
```

Creating an instance of your class with `MyList new` should yield the empty list. Depending on your implementation, it may be necessary to override the class method `new` to call an initialization method on the new list. If this is the case, remember to invoke new on the superclass in order to actually create the new instance, and make sure the new instance actually gets returned. Unlike constructors in Java, the new method is just like any other method on a class, and it must explicitly ensure that the instance gets created and returned.

Hints

- It will be much easier to define new methods if you use the browser to create new categories (HW6) and methods, rather than typing in `Object subclass:` and other method calls by hand in a workspace window. See GUZDIAL sec. 2.4 for more information and examples if you want them.
- Remember that you may add additional methods for internal use, such as helper functions or getter/setter methods. You may define additional classes as well if you wish.
- To save your work to a text file that you can turn in, select the category containing your work, right-click with the mouse, and select fileOut. This will save the category in a file named Category-Name.st, probably in the same directory containing squeak.

Assessment

- Your solution should contain (at a minimum) the class MyList in the category HW6 and (at a minimum) respond to the messages listed above. It should give correct results and signal errors when appropriate.
- You must not use any collection/array/list classes provided with SmallTalk.

Turn-in Instructions

Use the turn-in form linked from the course website.