CSE 341 Assignment 8

February 28, 2006

Due Thursday, March 9, 2006 by 11pm. No late assignments will be accepted.

As an introduction to Smalltalk and Squeak, implement a class `MySet` that is a subclass of `Object` and is part of the category `Homework8` (which you should define). This class should implement a set, which is an unordered collection of objects with no duplicate elements. Your class should contain the following methods.

1. Method `add:` takes a value `v` and stores it in the set if it is not already a member of the set. If the item is already in the set, `add:` does nothing.
2. Method `isEmpty:` returns true if the set is empty and false otherwise.
3. Method `contains:` takes a value `v` and returns true if that value is in the set, and false otherwise.
4. Method `size:` returns the number of elements in the set.
5. Method `remove:` takes a value `v` and removes `v` from the set. If `v` is not in the set, `remove:` throws an exception that describes the problem in an easy to understand way. For example, "Could not find a gnome with that sparkly rainbow" is not understandable, while "Element not found in set" is.
6. Method `forEach:` takes a block of code `b` as an argument and invokes `b` once with each value in the set. (This is conceptually similar to sequencing through the elements of a container using an iterator and a while loop in Java.) Assuming that `testSet` is a set of numbers, the following code will compute the sum of the elements in the set and store the result in the variable `x`.

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

Creating an instance of your set with `MySet new` should produce an empty set. You may have to override the `new` method in order to initialize your set in an appropriate manner.

You may define additional methods for internal use in your implementation, as well as create additional helper classes if you need them.

How you implement the set is up to you, with the following restrictions. You should use some simpler data structure to store the elements of the set, such as an array or a tree. You may *not* write methods that just call methods in a Smalltalk library collection class to do the work. (In other words, you need to manage a collection of data and implement the operations to add, remove, and search for items in that collection.)

Saving your work. 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 produce a text file in the Squeak directory named `Category-Name.st` (with the selected category name, of course) containing the code from that category.

Turn-in instructions

- Use the online turnin form to submit the text file produced by the `FileOut` command.