SE 2A04 Fall 2002

Lab Exercise 6

Instructor: William M. Farmer

Assigned:	22 November 2002
Demonstration due:	29 November 2002
Lab report due:	06 December 2002

The purpose of this lab exercise is to implement using recursion a definitional extension of the `Trees` module.

Step 1

Write a copy of the Oberon-2 module named `Trees` presented in the lectures.

Step 2

Write an Oberon-2 module named `TestTreesExt` that "black box" tests any module named `TreesExt` implementing the MIS given below.

Step 3

Write an Oberon-2 module named `TreesExt` that implements the MIS given below. Your implementation is required to implement each procedure of the interface using recursion.

Step 4

Construct an Oberon-2 program named `TestMine` that uses your `TestTreesExt` module to test your `TreesExt` module.

Step 5

During the lab session on November 29, demonstrate the `TestMine` program.

Step 6

Write a lab report that includes the following:

(1) A copy of the Lab Exercise 6 Marking Scheme (which is available on the course Web site) stapled to the front of your report.

(2) A copy of the Trees modules.

(3) A copy of your TestTreesExt module and a brief explanation of its design.

(4) A copy of your TreesExt module and a brief explanation of its design.

(5) The results of the test of your TreesExt module.

(6) A discussion of the test results and what you learned doing the lab exercise.

(7) A discussion of any problems you found with the MIS.

(8) A copy of the part of your log book relevant to this lab exercise.

The lab report must be given to the instructor no later than December 6.

Input/Output MIS for a Definition Extension of Trees

- Imported modules: Trees.
- Interface:

```
INTERFACE TreesExt;
  PROCEDURE Width(t: Tree): INTEGER;
  PROCEDURE Zeroize(t: Tree): Tree;
  PROCEDURE Mirror(t: Tree): Tree;
  PROCEDURE Compose(t1,t2: Tree): Tree;
END TreesExt.
```

- Exceptions: none required.
- Informal axioms:
  - For all t : Tree, Width(t) is the number of leaves in t.
  - For all t : Tree, Zeroize(t) is a copy of t in which the integers of the leaves in t have been changed to 0.
  - For all t : Tree, Mirror(t) is the "mirror" of t.
  - For all t_1, t_2 : Tree, Compose(t_1, t_2) is the tree obtained by replacing each leaf in t_1 with t_2.