See the handout for their effects on the structure.

Consider the following set of operations:

I. removeFront();
I. removeFront();
I. insertFront(new Integer(3));
I. insertFront(new Integer(7));
L = L;
L = new LRStruct();

LRStruct: An Enhanced AList
Program #1: Hangman

(secret word: polymorphism)
Player loses!

po mop i m

Program #1: Hangman (cont.)
of its actual character value:

- When it is visible, it converts to a string as the string consisting
  the hidden state or the visible state.
- When it is hidden, it converts to a string as " ".

In the Hangman game, a character in the target word can be either in

Program #1: Hangman (cont.)
1. Define class WordChar to represent the characters in a Hangman Word.

2. Define abstract class ACharState.

3. Define classes Hidden and Visible as concrete subclasses of ACharState.

4. Define a field in WordChar to reference an ACharState, its current state.

- All method calls in WordChar are delegated to its state.

Program #1: Hangman (cont.)
Secret Word: cat

Program #1: Hangman (cont.)
This design makes use of the composite pattern, the state pattern.

The UML diagram on the handout illustrates the above design.

Program #1: Hangman (cont.)