

## Homework #4 Lexi and Bridge

**Issued:** Tuesday, March 26

**Due:** Thursday, April 4

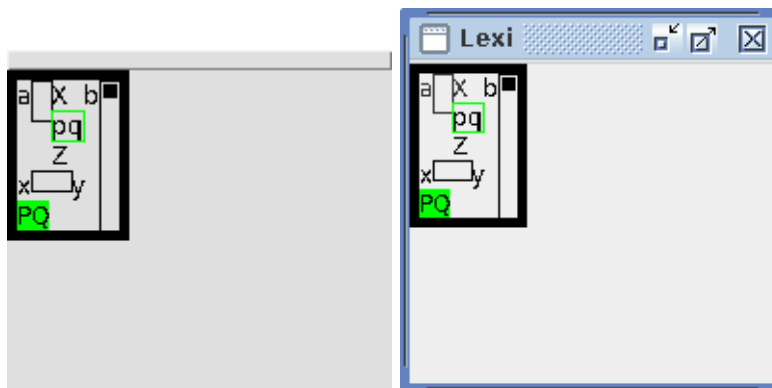
### Purpose

This assignment allows you to learn about the Bridge(151) design pattern. You should also use AbstractFactory(87), FactoryMethod(107), and Singleton(127).

### Assignment

Design (in UML) and implement (in Java) the Multiple Window Systems part of the Lexi editor, as described in Section 2.6 of our textbook. Support two window systems: AWT and Swing.

Test your solution with a simple graphical demonstration. For example, these are the results of my tests:



The left one is AWT. The right one is Swing.

## Notes and Suggestions

- Use the design suggestions from our textbook.
- Use the window interface and implementations in:

pub/hw4

You may want to rearrange them a little, for your factories.

- Regarding Singleton(127): There should be only one WindowFactory object, either a SwingWindowFactory object or an AwtWindowFactory object. From it, a client can create any number of SwingWindow or AwtWindow objects (as appropriate). For example, each ApplicationWindow object should contain a reference to its own SwingWindow or AwtWindow object.
- As usual, casting is prohibited!
- As usual, indicate with a comment at the top of each source file, the relevant patterns and participants.
- Use the value of an environment variable to select the window system. For example:

```
String s=System.getenv("LexiWindow");
if (s!=null && s.equals("Awt"))
    ...
else
    ...
```