

- Suppose that you have a `cSalesData` object with a `drawGraph()` method. Draw a UML including a few lines of code to show how you could use a Strategy Pattern to select at runtime between drawing a bar graph or a pie graph of the data.

- When you use Visual Studio you can open various sorts of files, edit them, and then save them. Text files are shown as text and bitmap files are shown as images. We might suppose that there is a `FileOpen(CString filename)` method. Some of this code will act the same no matter what kind of file you open, but part of the code will act differently depending on the kind of file. Describe how this might be accomplished by using the Template Method. Draw a UML diagram and write out some very rough code, simply using made-up names for the functions you use.

- Experiment a bit with the effect of Ctrl+Z (for Undo) and Ctrl+Y (for Redo) in your wordprocessor. What do you think might be a `WordProcessorCommand` in this context? What are the methods that `WordProcessorCommand` must implement? What kind of data structure might you need for holding the `WordProcessorCommand` objects? Draw a simple UML class diagram.

- How are design patterns different from frameworks?

- Old code expecting an iterator of type `Enumeration` is incompatible with newer aggregates such as `LinkedList`. To remedy this, write an adapter class `IterationAdapter` that will convert iterators of type `Iterator` to work with code expecting an iterator of type `Enumeration`. Here is some sample code using the adapter that should help clarify what is expected:

```
import java.util.*;
public class AdapterExample {
    public static void main(String[] args) {
        LinkedList container = new LinkedList();
        container.add(new Integer(1));
        container.add(new Integer(2));
        Iterator iterator = container.iterator();
        IterationAdapter iterationAdapter = new IterationAdapter(iterator);
        client(iterationAdapter);
    }
    // This client only understands the older
    // interface Enumeration
    public static void client(Enumeration e) {
        while(e.hasMoreElements()) {
            System.out.println(e.nextElement());
        }
    }
}
// Implement this class:
class
    IterationAdapter . . .
```

