Homework Assignment #2

Note

This assignment is due 9:10AM Thursday, December 16, 2010. Please write or type your answers on A4 (or similar size) paper. Put your completed homework on the instructor's desk before the class starts. For late submissions, please drop them in Yih-Kuen Tsay's mail box on the first floor of Management College Building II. Late submissions will be penalized by 20% for each working day overdue. You may discuss the problems with others, but copying answers is strictly forbidden.

Problems

1. Consider the implementation of an XML parser. The input for the parser is a well-formed XML document and the output is a Java object tree that represents the XML tree structure. (60 points)

   (a) Usually there is one and only one XML Parser factory within the system, and this factory will be used to create concrete parsers. Suppose we want the factory to be able to create 2 kinds of parser: XML DOM Parser or XML SAX Parser. Please write up a Java pseudocode of this parser factory with the property that there will be only one instance of this factory within the system.

   (b) Each node of the tree should provide at least two functions: 1) print the name of the current node and 2) return the number of descendants of the current node. Please describe how to apply the Composite pattern to model the whole-part relationship of the XML tree structure with the UML class diagram.

   (c) Suppose we want to search for the nodes in the parse tree. The Iterator pattern is a good choice to traverse the tree and let us perform operations on each of the traversed nodes. We may use the following code snippet to count the nodes that have names prefixed with "NTUIM":

```
int count = 0;

Node root = parser.parse ("sample.xml");    // parse the document and node will be the
root
Iterator iter = root.iterator ();
while (iter.hasNext())
{
      Node node = (Node) iter.Next();
      // suppose each node has a method, getName(), which returns the name of the
node
        if (node.getName().startsWith ("NTUIM") )
        {
                count ++;    // a match
        }
}
System.out.println ("There are " + count + " nodes prefixed with NTUIM");
```

Please apply the Iterator pattern to the parser and describe how the Iterator works for the composite pattern. Either breadth-first or depth-first traversal is acceptable.

2. For the admission application system that your team is designing and implementing, use OCL to specify the following requirements. (Improvise if the requirements do not apply to your system.) (40 points)
   (a) An applicant may apply for at most five departments/programs of the same university.
   (b) A “notice of incomplete application’’ is sent to an applicant (by calling some operation) only if the applicant’s application is still incomplete on the due date.