

Ex3 Preview, Swing tutorial Ex1 review

Amit Shabtay

Next Exercise- ex3- xmli

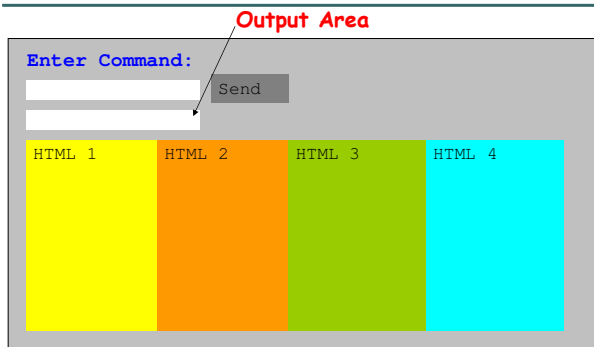
- In Ex3 you will build XML interpreter.
- This interpreter will receive an xml file. Simple commands will be accepted by the interpreter and will form HTML output.
- For the ex you will use your xmlp code.
- You will create a simple GUI for the xmli.

March 3rd, 2004

Object Oriented Design Course

2

Ex3- GUI



March 3rd, 2004

Object Oriented Design Course

3

Ex3- an XML input file

- ```
<Book id = "123" status = "available">
 <Name>Refactoring</Name>
 <Author>
 <FirstName>Martin</FirstName>
 <LastName>Fowler</LastName>
 <!-- Here's an example of a list within a list -->
 <OtherBooks>
 <BookName>UML Distilled</BookName>
 <BookName>Analysis Patterns</BookName>
 </OtherBooks>
 </Author>
</Book>
```

March 3rd, 2004

Object Oriented Design Course

4

## Ex3- Commands

- **edit [output-name]**
  - Selects the current view to be modified. The default is view 1.
  - Example: `edit 2`  
output: `Now editing view 2`
- **select [tag-name]**
  - Selects all tags [tag-name] and all their parents.  
Example: `select BookName`  
output: `Selected 5 tags`  
(BookName\*2, OtherBooks, Author, Book)
  - Acts only on the view we are currently editing!

March 3rd, 2004

Object Oriented Design Course

5

## Ex3- Commands

- **upcase [tag-name]**
  - Will change all tags with [tag-name] to upper case.  
Example: `upcase BookName`  
output: `Modified views: view2`
  - Acts on ALL the view we are editing!  
(views that were selected and have this tag)
- **locase [tag-name]**
  - The same, but to lower case.

March 3rd, 2004

Object Oriented Design Course

6

## Ex3- Commands

---

- **Undo**
  - Undoes the last command.
  - Can be called several times.
  - Example: `undo`  
output: `Modified views: view2`  
(This is in the case that the last command was upcase or locase)
- Exiting is by using the X of the window.

## Swing Tutorial

## AWT- Abstract Window Toolkit

---

- In 1995, Java technology introduced to the World Wide Web developers a network-centric, object-oriented language that provided client side processing for otherwise static pages.
- Java platform's graphical user interface library, the Abstract Window Toolkit (AWT), is used in the creation of dynamic Web pages and is a multiplatform tool.

## JFC- Java Foundation Classes

---

- Java Foundation Classes extends the original AWT by adding a comprehensive set of graphical user interface class libraries that is completely portable and delivered as part of the Java platform
- JFC is composed of the AWT, Swing and Java2D.

## SWT- Standard Widget Toolkit

---

- AWT was buggy beyond belief this was just poor code that needed fixing by Sun's coders. At IBM we hated Swing from day one. Big, buggy, and looks crap. (Simon Ritchie)
- SWT is a widget toolkit for Java designed to provide efficient, portable access to the user-interface facilities of the operating systems on which it is implemented.

## AWT vs. Swing

---

- [AWT vs. Swing](#) gives an overview on AWT and Swing and checks the pros and cons of each.
- In general, Swing has a better design with massive use of design patterns, look and feel options, event handling.

## Swing vs. SWT

- Great debate.
- Pro Swing: ([logemann](#))
  - \* Coding is easier to learn
  - \* Better API
  - \* Swing is a standard
- SWT uses **native code** for each platform, so it is faster and has better Look & Feel. (Eclipse is an Example for SWT instance)

## Ex1 review

## Ex1 Goals

- Our goal in this ex was to:
  - Practice in building a scalable design using "common language"- design patterns. (Structural and Traversal)
  - Practice with UML, Ant
  - Create a habit of testing the code.

## Ex1 Review- Design By Contract

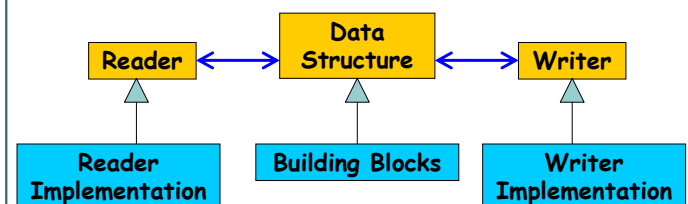
- API and comments are most important.
- Design documentation is not only for the grader but also serves as a contract with the package user.

## Ex1 Review- Trivialities

- These are not acceptable:
  - `String s = "Try";`  
`if (s == "Try")`  
`System.out.println("Wow");`
  - `Tag x= new Tag;`  
`visitor.visit((Tag)x);`

## Ex1 Review- Good Separation

- Your exercises design in general was good:



## Ex1 Review- DataBase or "Mediator"

- What about:



- VS.:



March 3rd, 2004

Object Oriented Design Course

19

## Ex1 Review-Nearly Empty Class

- ```
class Attribute {
    String key;
    String value;
}
```

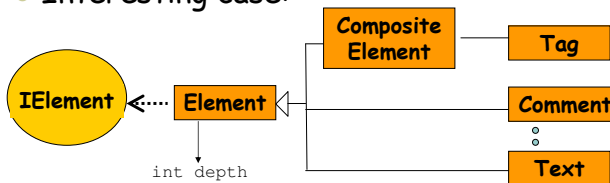
March 3rd, 2004

Object Oriented Design Course

20

Ex1 Review- Data Structure

- Most used Element, CompositeElement and deriving Tag, Comment, ...
- Interesting case:

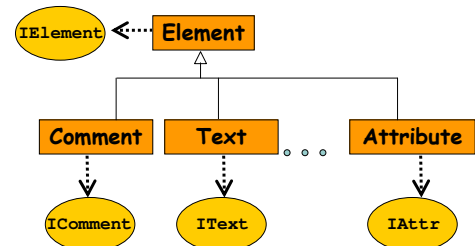


March 3rd, 2004

Object Oriented Design Course

21

Ex1 Review- Interfaces use



March 3rd, 2004

Object Oriented Design Course

22

Ex1 Review-Interfaces methods

- ```
abstract class Element{
 public Iterator getSons(){
 return null;
 }

 public void addSon(Element e){
 return;
 }
}
```

This can be avoided using a visitor, though the composite design pattern specifies this design.

//This visitor method is wrong, because it should be an //empty implementation  
 abstract public void visit(Element e);

March 3rd, 2004

Object Oriented Design Course

23

## Ex1 Review-Traversal

- Three examples:
  - While (iterator.hasNext()){
 Element x= iterator.next();
 x.accept(visitor); //For each element, use i.next()
 }
  - X.accept(visitor) // Traverse using only the visitor
  - Traverse elements using visitor, but children of an element using an iterator

March 3rd, 2004

Object Oriented Design Course

24

## Ex1 Review-Filtering

---

- Now it is your turn:

Possible ways for filtering  
(comments and attributes)