

UML

Class Diagram Example

1

Class Diagram Example

Here is a brief description of writing a text document:

- Suppose that you're writing a document in some of famous text processing tools, like Microsoft Word for example.
 - You can start typing a new document, or open an existing one.
 - You type a text by using your keyboard.
 - Every document consists of several pages, and every page consists of header, document's body or/and footer.
 - In header and footer you may add date, time, page number, file location etc.

2

Class Diagram Example

- Document's body has sentences. Sentences are made up of words and punctual signs.
- Words consists of letters, numbers and/or special characters.
- Also in the text you may insert pictures and tables.
- Table consists of rows and columns.
- Every cell from table may hold up text or pictures.
- After finishing the document, user can choose to save or to print the document.

3

Candidate Classes for our Model

- This is simplified explanation of creating a text document.
- If we extract the list of nouns form previous text, a following list can be achieved:

4

Candidate Classes for our Model

- Document
- Text
- Header
- Footer
- Document's body
- Date
- Time
- Page number
- Location of file
- Page
- Sentence
- Word
- Punctual sign
- Letter
- Number
- Special character
- Picture
- Table
- Row
- Column
- Cell
- User

5

Document Class

- Let's start with the document. As you can see this example deals with documents, thus a document will be the central class in our class diagram.
- Document has a several pages, therefore a *numberOfPages* will be one of the attributes.
- For the operations we have: *open()*, *save()*, *print()* and *new()*.
- Every document consists of pages. The Page will be also a candidate for the class.

6

Document Class

Document
numberOfPages
open() save() print() new()

7

Page Class

- The **Page** class will hold *pageNumber* as an attribute, and operations allowed here can be: *newPage()*, *hideHeader()* and *hideFooter()*. Operations for the header and footer tells us that the Header and The Footer can be also candidate classes.

Page
pageNumber
newPage() hideHeader() hideFooter() insertPicture() insertTable()

8

Header & Footer Classes

- The **Header** class and the **Footer** class has common attributes:
 - *date*, *time*, *pageNumber* and *fileLocation*. These attributes are optional for every header or footer and user may configure them.
- This will guide us that a common class can be introduced. This will be a good time to make an **inheritance**.

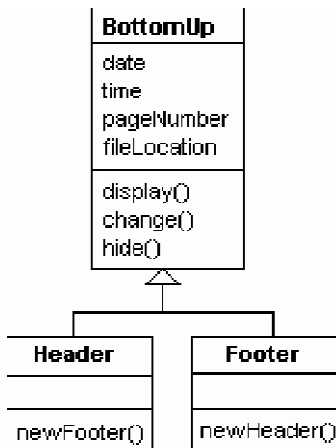
9

BottomUp Class

- Parent class will be **BottomUp** (this name is chosen because headers and footer appear in upper and bottom parts of every page) and will hold common attributes for header and footer, and these operations:
 - *display()*
 - *hide()* and
 - *change()*
- Header and Footer classes (children of BottomUp) will have only operations:
 - *newHeader()* and
 - *newFooter()*

10

BottomUp Class & It's Children



11

Document's Body

- Before examining the document's body or text, let's take a look at the making of text:
 - Document's text is made up of sentences.
 - Sentences are made up of words.
 - Words are made up of characters.
- If words are array of characters and sentence is array of words, then a sentence is also an array of characters. Therefore a document's body can be an array of characters. For this purpose to make a document's text we'll use the **Character** class with its children.

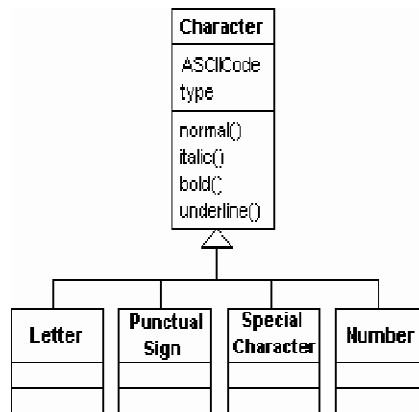
12

Character Class & It's Children

- The **Character** class will have:
- Attributes: *ASCIICode* and *type* (type tells the type of the character - normal, italic, bold or underline), and
- Operations: *normal()*, *bold()*, *italic()* and *underline()*
- The Character class children will be:
- **Letter**, **PunctualSign**, **SpecialCharacter** and **Number**.
Also in the document's body we can find **tables** and **pictures**. These are new classes in our class diagram.

13

Character Class & It's Children



14

Table & Picture Classes

- The **Table** class has:
- Attributes: *numbRows* and *numbColumns*
- Operations: *insertRow()*, *insertColumn()* and *newTable()*, *insertPicture* as operations.

Table	Picture
numbRows	
numbColumns	
insertRow()	
insertColumn()	
newTable()	
insertPicture()	

15

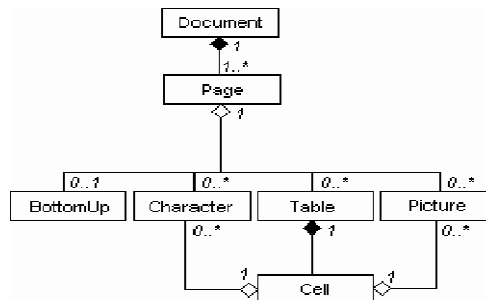
Cell Class

- Every table consists of one or more cells.
- Every **cell** can have:
 - Text or
 - Pictures

16

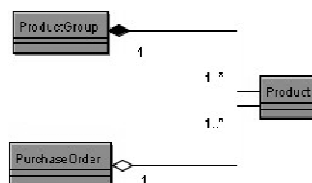
Putting it all together

- Having all this classes in front of us we can draw out the associations between them and gain the completed class diagram for this problem.



17

Aggregation & Composition



- The **composition** association is represented by the solid diamond. It is said that *ProductGroup* is **composed** of *Products*. This means that if a *ProductGroup* is destroyed, the *Products* within the group are destroyed as well.
- The **aggregation** association is represented by the hollow diamond. *PurchaseOrder* is an **aggregate** of *Products*. If a *PurchaseOrder* is destroyed, the *Products* still exist.

18

Are We Done?

- This model can grow and grow ...
- Making all the necessary steps of analysis and design you may reach the point, where a new text processing tool is modeled.

19

Next Time

- **Sequence Diagram** with Example
- We still have the following topics:
 - **Software Testing**
 - **Cost Estimation**
- **Midterm II (First week of December) on UML**

20