**Hacettepe University**
**Department of Computer Engineering**
**Bil138 Programming Laboratory**

**Subject :** Polymorphism in object-oriented programming
**Programming Language :** Java
**Submission Date :** 27/4/2011
**Deadline :** 9/5/2011
**Advisors :** R.A. Levent Seçkin, Dr. Sevil Şen
**Experiment number :** 3

## AIM

In this experiment you are aimed to get familiar with Polymorphism in Object Oriented Programming. Two classes, which you will use to develop and test your program, will be given to you. One of them is an abstract class, which is prepared for you to implement in your experiment by overriding its abstract methods. Other one is a test class, which is suitable to program specialties and it is in format of Junit testing framework. You will use Junit testing framework to test your program. You need to use polymorphic arrays to develop your application.

***Polymorphism*** *is a programming language feature that allows values of different data types to be handled using a uniform interface. The concept of parametric polymorphism applies to both data types and functions. A function that can evaluate to or be applied to values of different types is known as a polymorphic function. A data type that can appear to be of a generalized type (e.g., a list with elements of arbitrary type) is designated polymorphic data type like the generalized type from which such specializations are made.[wikipedia]*

## Experiment

For this experiment you are expected to develop an **Information Page Serving System**. In this page server system you have different type of elements that you will use to construct an information page. These elements can be listed as articles, images and links. All elements have an array of keys and id to define that item. All ids –no matter which type they are- are unique in whole application scope. You will use these elements to create information pages.

Articles have a text body that will be read from given text file. The text file's name will be given as parameter of add article function. Images have the file path, width and height attributes. Links have name and url attributes. Surely, all of these attributes will be defined as parameters too.

There are two different ways to create information pages: Search Pages, Article Pages. To create search pages, related function will give item type and algorithm type to search items by keys. To create article page, wanted topic id will be given by the function.

While creating the information pages, all of the items will be produced as tags, whose formats can be automatically prepared by abstract class methods.

You need to override some methods to implement program competence. Here are detailed explanations of abstract methods:

```
      public abstract boolean addArticle(int id, String[] keys, String
filePath);
```

Add article is a simple method to add articles to server sources. To add an article, method needs to get file that is defined by path and read that text file to prepare article text body. You can use your IO classes and packages from your previous projects. Just as all other items, articles have id and keys which define that article.

```
      public abstract boolean addImage(int id, String[] keys, String
imgPath, int width, int height);
```

addImages is another method to be implemented in your source code. Unlike adding an article to system, there is no need for any reading operation. Just keeping the path and size of image is enough.

```
      public abstract boolean addLink(int id, String[] keys, String url,
String text);
```

Like images, links are attribute based items –no need to read a file- . Keeping url and definition of that link is enough.

**Note:** All adding operations have to give information about their success or failure by their returning value.

```
      public abstract String addArticleReference(int articleId, int
referenceId);
```

Articles can have references which could be any type of item. Second argument of function defines an item and that item needs to be added to the article which is declared by the first argument of function. Returning value should give information about execution result.
Example return codes:

```
<articleId> is not found
<articleId> is not article
<referenceId> is not found
<articleId> referenced by <referenceId>
```

**Note:** Page creator methods are different. These three methods return a value which contains information about how many different items does the generated page contain. To generate a page, you need to create tags which are specific for that items type. Tag creator methods are already written in abstract class and ready to use. By giving correct arguments you can easily get the prepared tag. You have to write that tag to output file as <u>one line</u>.

```
      public abstract int searchPage(String resultFile, String[] keys,
ItemType itemType, SearchType algType);
```

To implement search operation, you need to search for given keys in given type of elements' type using the given algorithm. AND – OR are two different algorithm types which is written in the abstract class as an enum. If the algorithm type is AND, you need to write all of argument keys that are included in all of matching items. You can write an item if any of the keys are included for OR algorithm. Another enum is ‘types enum’. ALL, ARTICLE, IMAGE and LINK are four different types of elements to look for their keys. Another important point while creating pages is adding the time information which shows when the page is created.

```
    public abstract int infoPage(String resultFile, int id);
```

This method triggers creating of an information page. You need to write tag of that article which is defined by id parameter. After that you should write reference tags. Referencing time is important while creating article pages. You need to generate reference item tags in the order they are given via input file. While creating an article page, you should use reference array and write them by their order. You need to use reference start and end tags which are given by abstract class, at the start and end of reference writing operation.

```
    public abstract int extendedInfoPage(String resultFile, int id);
```

This is a bonus method. Its only difference is that, if an article is referenced by another article, you need to write referenced article’s references for the extended information page. This referencing operation may continue recursively for many times but to prevent infinite loops you not need to write elements which isn’t already written in that page.

**Skeleton of Project**

| | |
|---|---|
| +test | package |
| InfoServicePagesTestCases.java | already written test class |
| InfoServiceTemplate.java | already written abstract class |
| + infoServicePages | package |
| InfoServiceManager.java | inherits abstract class, you need to write this class |
| ... | all of your other classes and packages must be written in this scope |

**Important Notes**

- You have to design your application in complete object oriented design approach. You are expected to define a java class for everything that is suitable for a class.
- Class diagram for full program must be given in your report. You have to prepare a basic class diagram.
- Another important point is designing your code in encapsulation paradigm. Encapsulation is an obligation in OOP approach.
- Tag creator functions have been written in a program paradigm which forces you to encapsulate these methods too.
- You can use *your InputDelegate and OutputDelegate* classes from your first experiment for IO operations.
- Generated output files can be viewed as web pages but evaluation will be done on their text format. Be careful about new lines and other subjects.

**SUBMISSION**

- Your submission will be in the format below

```
< StudentID>
|-- report
        |-- Report.pdf
|-- source
        |-- infoServicePages
                |-- InfoServiceManager.java
                |-- other  *.java files and packages.
```

- You must use “Online Experiment Submission System”.
- http://submit.cs.hacettepe.edu.tr. Other type of submissions, especially by e-mail, WILL NOT BE ACCEPTED.
- Submission deadline is 9/5/2011, 16.59 pm.
- Respect the office hours of your advisor.

    Wednesday 09:00-12:00, Friday 13:00-15:00 Office: 122

REFERENCES

- http://www.google.com
- http://en.wikipedia.org/wiki/Polymorphism_(computer_science)
- http://www.junit.org/
- http://en.wikipedia.org/wiki/Encapsulation_(object-oriented_programming)
- http://web.cs.hacettepe.edu.tr/~ersiner/prj/minio/
- http://en.wikipedia.org/wiki/UML
- http://en.wikipedia.org/wiki/Class_diagram
- http://en.wikipedia.org/wiki/HTML
- http://www.w3schools.com/html/