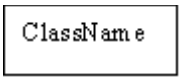
Hacettepe University
Department of Computer Engineering
BBM104 Programming Laboratory

Experiment Number	**:** 2
Subject	**:** Object Oriented Programming-Inheritance
Programming Language	**:** Java
Deadline	**:** March 20, 2014
Advisors	**:** Dr.Oğuz Aslantürk, R.A. Ali Seydi Keçeli

Introduction

Class Diagrams

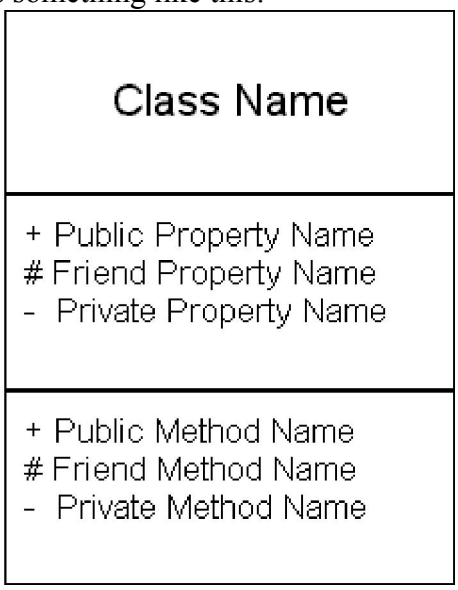
Class diagrams show the Public and Private methods and properties of our classes. The number of class diagrams required to model a system will depend on its size and complexity. In a class diagram, a class is represented in its most simple form as a box with the class name within it, as shown here:



As it stands, however, this is not very useful. We need some additional information about the class. We display this information by dividing the class diagram into three sections:

- The top section contains the name of the class
- The middle section contains the properties of the class
- The lower section contains the methods of the class

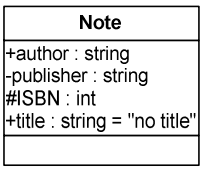
A typical class diagram looks something like this:



We place a symbol next to each of these properties and methods to indicate its level of visibility. The plus signs + indicate Public properties and methods, while the negative signs - indicate Private properties and methods. We use the hash sign # to indicate Friend properties and methods.

Examples :

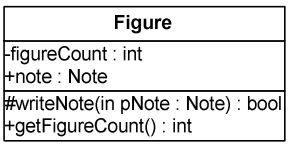
Class Diagram



Java Implementation

```
public class Note
{
        public String author;
        private String publisher;
        protected int ISBN;
        public static String title=”no title”;
}
```

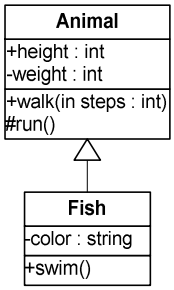
Class Diagram



Java Implementation

```
public class Figure
{
        private int figureCount;
        public Note note;
        protected bool writeNote(Note pNote)
        {
                …
        }
        public int getFigureCount()
        {
                …
        }
}
```

Class Diagram



Java Implementation

```
public class Animal
{
        public int height;
        private int weight;
        public void walk(int steps)
        {
                …
        }
        protected void run()
        {
                …
        }
}

public class Fish extends Animal
{
        private String color;
        public void swim()
        {
                …
        }
}
```

Experiment

In this experiment you will implement a library system that handles library items. You are expected that you find an Object Oriented Solution to this problem.

Suppose that two kinds of people using library items. These are students and academic stuffs. Students can borrow only one item at a time, and academic stuffs can borrow at most 3 items at the same time. Also people that use library items have common properties. For example everyone has a name, surname and unique number for each person.

At the library we have items and people are allowed to borrow them for a while. We have 3 types of items at the library. Books, Magazines and Compact Discs (CDs). All items have a unique number called serial number, shelf number that the item put on that shelf and shelf index, the index of the item at the shelf. Items at the library have additional different properties changing to item type. For example books have name, publisher name, and author name. CDs have title property and magazines have name, publisher properties. But people are allowed to borrow only books and CDs not magazines.

Implementation

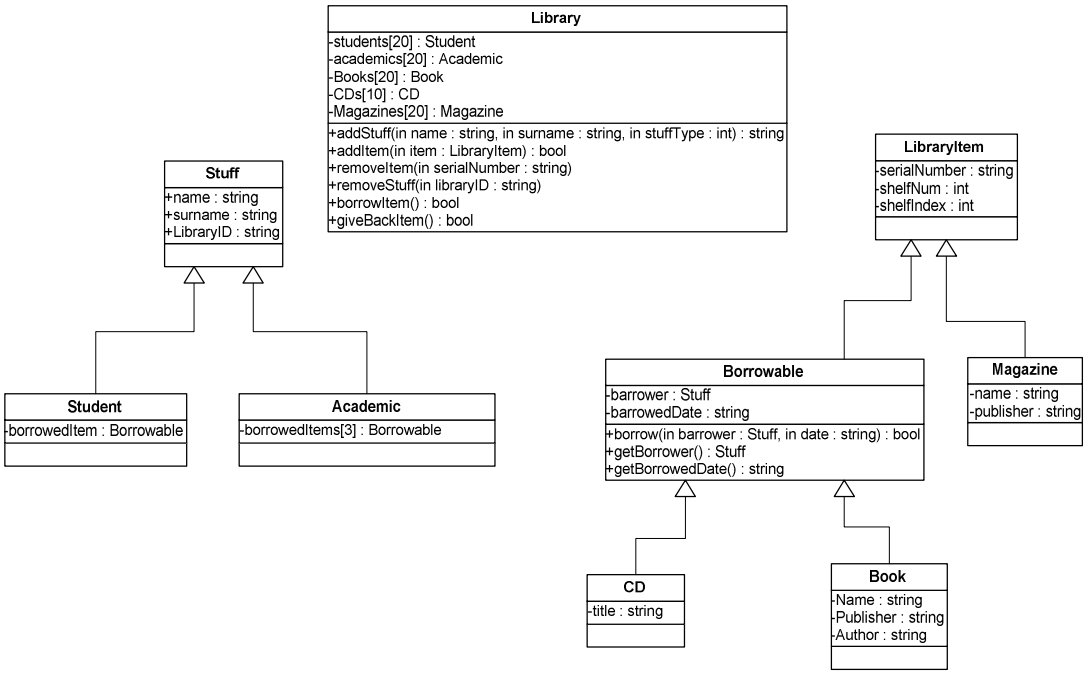
You are given a class diagram below. Implement your solution through this class diagram. You can add methods and properties for your requirement if you want. Extend this class diagram and draw it in "Main Data Structures" section at your reports. This class diagram is hint for you, also you must add more methods and properties to these classes or you can also add extra classes as well.

Your program ("libmain.class") reads the commands from the keyboard and executes them. You are given sample input as files. You must redirect your keyboard input to these files when running your programs at the dos prompt as below. Output the results to the screen.

Example:

```
java libmain.class < input.txt
```

Also you do not need to write source for reading from keyboard. You can download sample input, output files from the course ftp site.



Input Commands:

- AddItem <Serialnumber> <Shelf number> <Shelf Index> <Item Type> <…>

This command adds an item to the library. Get the attributes of the item and create the relevant object for the item in your program. For related item type read other attributes as well. You can see other attributes for the item types below.

<Item Types>:

BOOK: <"name"> <"publisher"> <"author">
CD: <"title">
MAGAZINE: <"name"> <"publisher">

Examples:

AddItem sd143 12 2 BOOK "Data Structures In C" "Addison-Wesley" "Horowitz"
AddItem sr345 10 5 CD "File Structures"
AddItem rt194 2 19 MAGAZINE "Computer World Num:567" "TIMES"

- AddStuff <"Name"> <"Surname"> <LibraryID> <Stuff Type>

This command add a library user to the system. According to stuff type attribute create different object for the user.("Student" or "Academic")

Examples:
AddStuff "Selman" "Duatepe" sde346 ACADEMIC
AddStuff "Cemal" "Koplay" fjh386 STUDENT

- RemoveItem <Serial Number>

Removes an item from the library whose serial number is given. You can not remove an item that has already been borrowed but do not give an error message.

- RemoveStuff <Library ID>

Removes a user whose library ID is given from the library. You can not remove a stuff that has already borrowed an item but do not give an error message.

- Borrow <Library ID> <Serial Number> <"date">

Borrows one item (Serial Number) to a person whose library ID is given

- GiveBack <Library ID> <Serial Number>

Gives back one item. Makes it free for the borrowing again.

- ListItems

  Lists all items (books, CDs and magazines) in the Library
  Output format:
  <serial number> <shelf number> <shelf index> <item type> <…>

  Examples:
  rt456 4 12 BOOK "File Structures" "Addison-Wesley" "Michael Folk"
  er403 12 6 BOOK "Discrete Mathematics" "Addison-Wesley" "Horowitz"
  gh419 6 1 BOOK "Data Structures in C" "Wrox" "Eric Riccardi"
  dj256 3 14 CD "Algorithms"
  hj684 13 1 CD "Computer Graphics"
  df254 12 5 MAGAZINE "COMPUTER WORLD" "TIMES"

- ListStuffs

  Lists all library users.

Examples:
"Selman" "Duatepe" sde346 ACADEMIC
"Ferda" "Ergüneş" djf863 ACADEMIC
"Cemal" "Koplay" fjh386 STUDENT

- ListBorrowedItems
  Lists all items that have been borrowed
  Output format :
  <serial number> <library ID> <"borrowed date">
  Examples:
  se143 sde346 "10.3.2002"
  jf863 dkn295 "12.3.2002"
  jh386 aem103 "30.1.2003"

- End

Indicates that your program finish to read commads.

Notes and Restrictions

- Define all properties as "private" and use "get" and "set" methods while accessing these properties.

Example:

```
private String myName;
public String getMyName()
{
        …
        return myName;
}
public void stMyName(String name)
{
        …
        myName=name;
}
```

- do not use default constructor, use constructor that sets all initial values to the properties.

Example:

```
public class Student
{
        private int ID;
        private String name;
        private String surname;
        public Student(int pID, String pName, String pSurname)
        {
                ID=pID;
                name=pName;
                surname=pSurname;
        }
}
```

- Use "javadoc" rules for documenting your source codes.
- In your reports please keep your problem and solution sections longer. (At least one page for each.)
- Do not define any packages
- Submision format is defined below:
  - Student_number.zip
    - --report
      - report.pdf
    - --src

    - libmain.java
    - *.java

- All reports and project files will not be accepted in hard copy, e-mail or diskette. You can give your project files via submission system
- All experiments will be examined using JDK 1.7 environment, and your main class name must be "Libmain.java". So I will compile your source code with command "javac libmain.java". Do not use any other name for the main program class.
- Don't pass the deadline. No experiment will be accepted after **April 4, 2014**.