Hacettepe University
Department of Computer Science & Engineering
BiL138 Programming/Software Laboratory
Experiment 3

Subject : Introduction to Java
Submission Date : 03.04.2012
Design Report Deadline : 10.04.2012 - 17.00
Final Deadline : 17.04.2012 - 17.00
Programming Language : Java
Development Environment : Eclipse IDE for Java Developers
Advisors : Dr. Ebru SEZER, R.A. Cumhur Yiğit ÖZCAN, R.A. Gültekin IŞIK

Introduction
In this experiment, you are expected to implement application logic and file IO parts of a company's employee management system using Java programming language. Main focus point of this experiment is getting you familiar with developing code synchronously with someone else. For this purpose, you will be provided with some interfaces. These interfaces will let you communicate with GUI part of the software which is going to be developed by another developer etc.

Problem
The software that you will implement is going to be used by a game company called 'Dilizzart' to manage wages of their employees. This company currently holds 8 different types of employees. Information in the parenthesis indicates department names.

1. Project Manager (Managers)
In charge of one or more game projects that are run by the company. Takes regular monthly salary.

2. Lead Developer (Developers)
In charge of development process of a game project run by the company. Takes regular monthly salary.

3. Developer (Developers)
Works at one of the projects going on as software programmer. Takes regular monthly salary.

4. Senior Graphics Designer (Graphics)
Designs main characters and objects of the games. Doesn't have fixed work hours. Takes a regular monthly salary and 50$ for each day worked that month.

5. Junior Graphics Designer (Graphics)
Designs relatively unimportant models, effects etc. for the games. Doesn't have fixed work hours. Takes a regular monthly salary and 30$ for each day worked that month.

6. Sales Supervisor (Sales Staff)
Makes meetings with producers and sells finished games of the company. Takes regular monthly salary plus a %10 of total sales amount for that month.

7. Sales Assistant (Sales Staff)
Assists sales supervisor in the meetings held. Takes regular monthly salary plus a %5 of total sales amount for that month.

8. Intern (Interns)
Interns only get a regular monthly salary.

Input File
Every time the program starts, it should read from an input file to initialize its employee list. Path of the input file will be given as first argument to the main function. Format of the input file will be like :
<employee_id>;<position>;<name>;<surname>;<base_salary>

```
A01;Project Manager;Levent;SECKIN;3500;
A02;Project Manager;Oner;BARUT;3500;
B01;Lead Developer;Cumhur Yigit;OZCAN;3200;
B02;Lead Developer;Murat;YILDIZ;3200;
B03;Developer;Melih;CEYHAN;2750;
B04;Developer;Okan;CAKIRHAN;2750;
B05;Developer;Ugur Eray;TAHTA;2500;
B06;Developer;Sinan Onur;ALTINUC;2500;
B07;Developer;Senol;YUKSEL;2300;
C01;Senior Graphics Designer;Orhun;DALABASMAZ;750;
C02;Senior Graphics Designer;Naim;TURKDOGAN;750;
C03;Junior Graphics Designer;Barbaros;BAL;750;
C04;Junior Graphics Designer;Burak;BALDIRLIOGLU;750;
D01;Sales Supervisor;Firat;YILDIRIM;1500;
D02;Sales Assistant;Berkay;TURKMEN;1000;
D03;Sales Assistant;Ezgi;KATIPOGLU;1000;
E01;Intern;Hüseyin;CAVDAR;750;
E02;Intern;Ali;YILDIRIM;750;
```

Output Files
You will create two different types of output files in this experiment. These files will contain a complete summary of employee list and their salary information. Please check ftp server for output file samples.

1. Type A
This file will contain employee information divided into parts for each department. Employees in each department will be listed ordered by their employee ID. Department names and the 20-dash-separators should be placed carefully.

2. Type B
Much like Type A but in this file, employees in each department will be ordered by their surnames. Also after each department is listed, total salary value for that department should be printed. Finally at the end of the report, a final total of all salaries should also be printed.

Design Considerations
As explained before, for this experiment you will implement the application logic and file IO parts of the software. The GUI part will be developed by another developer. After the meeting held with the person who will develop the GUI part, you have decided that your application logic

should implement 2 interfaces named IEmployeeManager, which will be implemented by a class named EmployeeManager in the application logic part, and IEmployee which will be implemented by a class named Employee in the application logic part. Definitions of these interfaces are as follows:

```
public interface IEmployeeManager {
	public void initialize(String filePath);
	public void addEmployee(String name, String surname, int baseSalary, int type);
	public void removeEmployee(String employeeID);
	public void editEmployee(String employeeID, String name, String surname, int
baseSalary);
	public IEmployee[] getEmployees();
	public IEmployee[] getEmployees(int type);
	public IEmployee findEmployee(String employeeID);

	public void operateEndOfMonth();

	public void printOutputFileA(String filePath);
	public void printOutputFileB(String filePath);

	public void changeSalesSupervisorShare(int share);
	public void changeSalesAssistantShare(int share);

	public void changeSeniorGraphicsDailyPremium(int dailyPremium);
	public void changeJuniorGraphicsDailyPremium(int dailyPremium);

	public void addMonthlySale(int amount);
	public void addWorkdayToDesigner(String employeeID, int amount);
}
```

Initialize
Initialize method of EmployeeManager class should make necessary operations to read input file and create objects that are needed.

AddEmployee
AddEmployee should add a new entry for an employee of the given type. The types are as follows :
1 - Project Manager
2 - Lead Developer
3 - Developer
4 - Senior Graphics Designer
5 - Junior Graphics Designer
6 - Sales Supervisor
7 - Sales Assistant
8 - Intern

RemoveEmployee
RemoveEmployee method should remove the employee whose id is given from its registery.

EditEmployee
EditEmployee method should edit the employee's information, whose id is given, with new parameters.

GetEmployees
This method should return an array of all employees in the system. If the overloaded method, that takes an integer as a parameter, is run then it should return only the employees whose type match with the given parameter.

FindEmployee
This method should return the object that represents the employee whose id is given.

OperateEndOfMonth
When this method is called, all workday information and sales amount that causes premium wages to be paid to the sales staff should be reset to zero.

PrintOutputFileA
When this method is called, an output file type A should be created by the system at the given path.

PrintOutputFileB
Same as PirntOutputFileA but this time an output file of type B should be created.

ChangeSalesSupervisorShare
Share amount for Sales Supervisors is 10 percent by default. This method will be used to change this amount.

ChangeSalesAssistantShare
Same as ChangeSalesSupervisorShare but this time the Sales Assistant's share, which is 5 percent by default, will be effected.

ChangeSeniorGraphicsDailyPremium
Daily premium amount for Senior Graphics Designer is 50$ by default. This method will be used to change this amount.

ChangeJuniorGraphicsDailyPremium
Same as ChangeSeniorGraphicsDailyPremium but this time the Junior Graphics Designer's daily premium, which is 30$ by default, will be affected.

AddMonthlySale
This method will be used to add monthly sale amount for the company. This value affects all sales staff automatically. This value will be reset to zero when OperateEndOfMonth method is called.

AddWorkdayToDesigner
When this method is called, specified amount of work days will be added to the given employee. Only the given employee will be affected by this operation. This value will be reset to zero when OperateEndOfMonth method is called.

```
public interface IEmployee {
        public String getName();
        public String getSurname();
        public int getBaseSalary();

        public void setName(String name);
        public void setSurname(String surname);
        public void setBaseSalary(int salary);

        public int calculateMonthlySalary();
        public String getEmployeeInfo();
}
```

CalculateMonthlySalary
This method should calculate monthly salary of the employee in an appropriate way and return it.

GetEmployeeInfo
This method should return a string whose format will be :
<employee_id> <name> <surname> <monthly_salary>

You should implement each function of each interface very carefully so that the system will run correctly when two parts are joined together.

You are expected to develop a fully object-oriented design for as the solution of this experiment. Use interfaces and inheritance when necessary. As you can see, by making use of interfaces, GUI part is abstracted from the application logic part. In your design, you should preserve same perfective for the application logic part and the file IO part.

You will submit your work in two parts. First part will contain only a design report (no code). In your design report, you should give a brief scheme of your design as a class diagram, which is one of the diagrams of UML. Your diagram should clearly indicate implementations, inheritance relations, associations (“has a”) and also each class representation should contain attributes and method definitions that are necessary. So don’t put getter-setter methods etc. to your class diagram. You may use a free software or eclipse plug-in for this purpose. In addition to the class diagram, give a very short plain explanation for each class and interface.

Design report doesn’t have to represent your final design. Changes to be made on the design are allowed -and encouraged- after design report submission date. But you are expected to explain each of those changes and their reasons on your final report.

Design Notes
* You don't have to check for inaccurate inputs. However if you do, explain your error messages in report.
* Class that contains your main method should be named "Main.java".

Final Report
You should write these topics on the report:
1. Cover Page,
2. Software Usage,
3. Software Design Notes: **Problem, Solution.** In the solution part, give an updated version of the class diagram you have created for the design report. Explain each change you have made and their reasons.

Notes and Restrictions
1) Do not submit any file via e-mail. You should upload your files via “Online Experiment Submission System” which is at http://submit.cs.hacettepe.edu.tr

2) Design Report Submission Format:
<student_number>.zip
|--- report
|--- report.pdf

3) Final Submission Format:
<student_number>.zip
|--- src
|--- Main.java
|--- *.java
|--- report
|--- report.pdf

3) You should submit only project files and source files; not compiled class files.
4) Your work should obey general software engineering principles and conventions. (Design, Comments, Indentation)
http://java.sun.com/docs/codeconv/html/CodeConvTOC.doc.html
http://java.sun.com/j2se/javadoc/writingdoccomments/

5) Save all your work until the assignment is graded.
6) This is not a team project. You are expected to provide an **INDIVIDUAL** work.
7) You should follow announcements on “Dersler.Bil138” newsgroup which is located in http://news.cs.hacettepe.edu.tr
8) No submission will be accepted after deadlines.