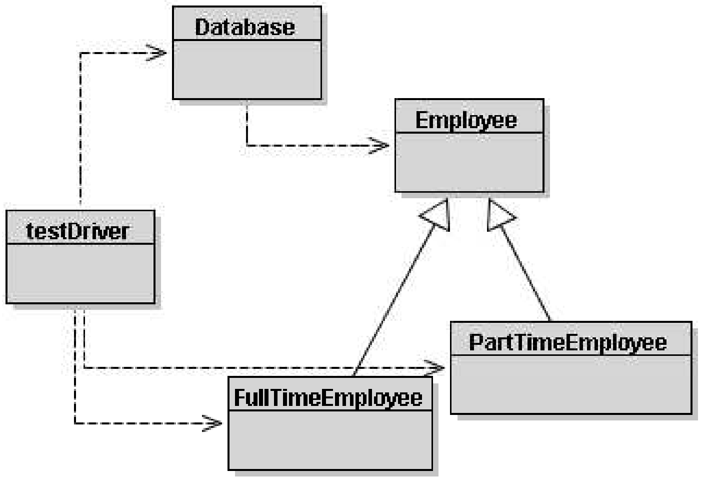
Project 4: Payroll Calculations

Introduction

This program is supposed to calculate the yearly salaries for two different types of employees: part-time and full-time. The full-time employees' monthly salary is fixed, based on their employee level. The part-time employee' salary is based on the number of hours they work. You are supposed to use inheritance in this project to represent the relationships between an Employee class and the two other types of classes: FullTimeEmployee and PartTimeEmployee.

Program Details

You should to implement five different classes: Database, testDriver, Employee, FullTimeEmployee and PartTimeEmployee. The classes and their relationships are indicated in the class diagram shown below.



The details of each of the five classes are indicated below.

testDriver

This class is used to test the other classes. This class should read the employee details from an input file called ***employeedata.txt*** and create a database of employees. The input file contains an arbitrary number of lines. Each line follows the format indicated below

```
Last First ID Dept Code Level PayRate Ded Months Hrs
```

Each field is explained below
Last: The last name of the employee
First: The first name of the employee
ID: The employee ID (a nine digit number)
Dept: The department code (a four character string)
Code: A single letter code indicating whether the employee is part-time ('P') or full-time ('F')
Level: This is applicable only for full-time employees and indicates their salary levels. The acceptable levels are 0, 1, 2, 3 and 4. The monthly salaries are based on this code is indicated in the table below. The value of this field will be 5 for part-time employees.

Table 1: Codes and salaries for full-time employees

Employee Level	Monthly Salary
0	3000
1	4000
2	5000
3	5500
4	6000

PayRate: This is applicable only for the part-time employees and indicates the pay rate per hour for each part-time employee. The value of this field will be 0 for full-time employees.
Ded: This indicates the total deductions for each employee.
Months: This is applicable only for the full-time employees and indicates the number of months they have worked for. The value of this field will be 0 for part-time employees
Hrs: This is applicable for part-time employees and indicates the number of hours they worked. The value of this field will be 0 for full-time employees.

The testDriver class should read these lines from the input file, create appropriate FullTimeEmployee or PartTimeEmployee objects and add them to a database object. After adding them, it should invoke the list method of the database to print out the salary details about each employee.

Database

This class should build a database of Employee entries using an ArrayList. This class should have a list method that lists the details of each method including their calculated salaries

Employee

This class should contain the basic information about each employee. This should include the last name, first name, employee id and department code. These should be declared as private.

FullTimeEmployee

This class extends the Employee class and should add information that is pertinent to a full-time employee only. This class should have a calculateSalary method that calculates the salary based on Table 1. This class should also have a print method that prints out the details about each employee including their salary.

PartTimeEmployee

This class extends the Employee class and should add information that is pertinent to a part-time employee only. This class should have a calculateSalary method that calculates the salary based on the pay rate and number of hours worked. This class should also have a print method that prints out the details about each employee including their salary.

Hints

1. Refer to the Java API at http://java.sun.com/j2se/1.4.2/docs/api/ whenever you need to get more information about particular classes.
2. Refer to Chapters 8 and 9 of your book to understand better the concepts of inheritance.

Programming Standards

You'll be expected to observe good programming and documentation standards. All the discussions in class about formatting, structure, and commenting your code will be enforced. The given code satisfies the documentation standards below. You will be expected to follow these standards and others in all of your later programs. Some, but not necessarily all, specifics include:

- You must include header comments specifying your name, the compiler and operating system used and the date your source code and documentation were completed.
- The header comment must also include a brief description of the purpose of the program (sort of a user guide) — this should be in your own words, not copied from this specification.
- You must include a comment explaining the purpose of every variable you use in your program.
- You must use meaningful, suggestive (of function or purpose!) variable names.
- Precede every major block of your code with a comment explaining its purpose. You don't have to describe how it works unless you do something so sneaky it deserves special recognition.

Testing

You should test your program within the BlueJ environment to make sure that you implemented all the above requirements correctly. Later, you will be provided with test files to test your program with. You should be certain that your program produces the output given above when you use the given input file and the sample test files provided on the course web site. However, verifying that your program produces correct results on a few test cases does not constitute a satisfactory testing regimen. You should make up and try additional input files as well; of course, you'll have to determine what the correct output would be.

Your program for this project is a stand-alone program. No other program on the curator will invoke its methods. **You should invoke the main method of the class in the BlueJ environment to run the program.**

Evaluation

Your program will be submitted to the Curator system for evaluation of correctness (details are given below). You will have five submissions to allow you an opportunity to correct mistakes that your own testing has not found. **You will not be given more submissions for any reason!** Your submitted program will be assigned a grade based on the correctness of your program and evaluation of your program by a TA. The grade you see from the Curator is your correctness grade. The grade for the project for future assignments is computed by deducting any points for style and design from the correctness grade. Therefore, the correctness grade is the *maximum* that your program will receive. **The TA will always grade the *first* submission that has the highest points.** No exceptions will be made, so be sure that you have done everything you need to before submitting. Note: Check your grade a few minutes after submitting. **If you get "N/A" as a grade, do not submit again until you get a numeric grade.**

Submitting your Program

You will submit this assignment to the Curator System (read the *Student Guide*), and it will be graded automatically. Instructions for submitting, and a description of how the grading is done, are contained in the *Student Guide*. **Exact details of what you need to submit will be posted to the class listserv and also on the class site.** You will be allowed up to ***five*** submissions for this assignment. Use them wisely. Test your program thoroughly before submitting it. Make sure that your program produces correct results for every sample input file posted on the course website. If you do not get a perfect score, analyze the problem carefully and test your fix with the input file shown as part of the Curator output, before submitting again. The highest score you achieve will be counted. The *Student Guide* can be found at:
`http://www.cs.vt.edu/curator/`

Program Compilation

Your program must compile and run under JDK 1.4.

Pledge

Every program submission for this class must include an Honor Code pledge. Specifically, you **must always** include the following pledge statement in the header comment for your program:

```
/**********************************************************************
 *
 * On my honor:
 *
 * - I have not discussed the Java language code in my program with
 * anyone other than my instructor or the teaching assistants
 * assigned to this course.
 *
 * - I have not used Java language code obtained from another student,
 * or any other unauthorized source, either modified or unmodified.
 *
 * - If any Java language code or documentation used in my program
 * was obtained from another source, such as a text book or course
 * notes, that has been clearly noted with a proper citation in
 * the comments of my program.
 *
 * - I have not designed this program in such a way as to defeat or
 * interfere with the normal operation of the Curator System.
 *
 * Your Name
 * Your PID
**********************************************************************/
```

Failure to include this pledge in a submission is a violation of the Honor Code.