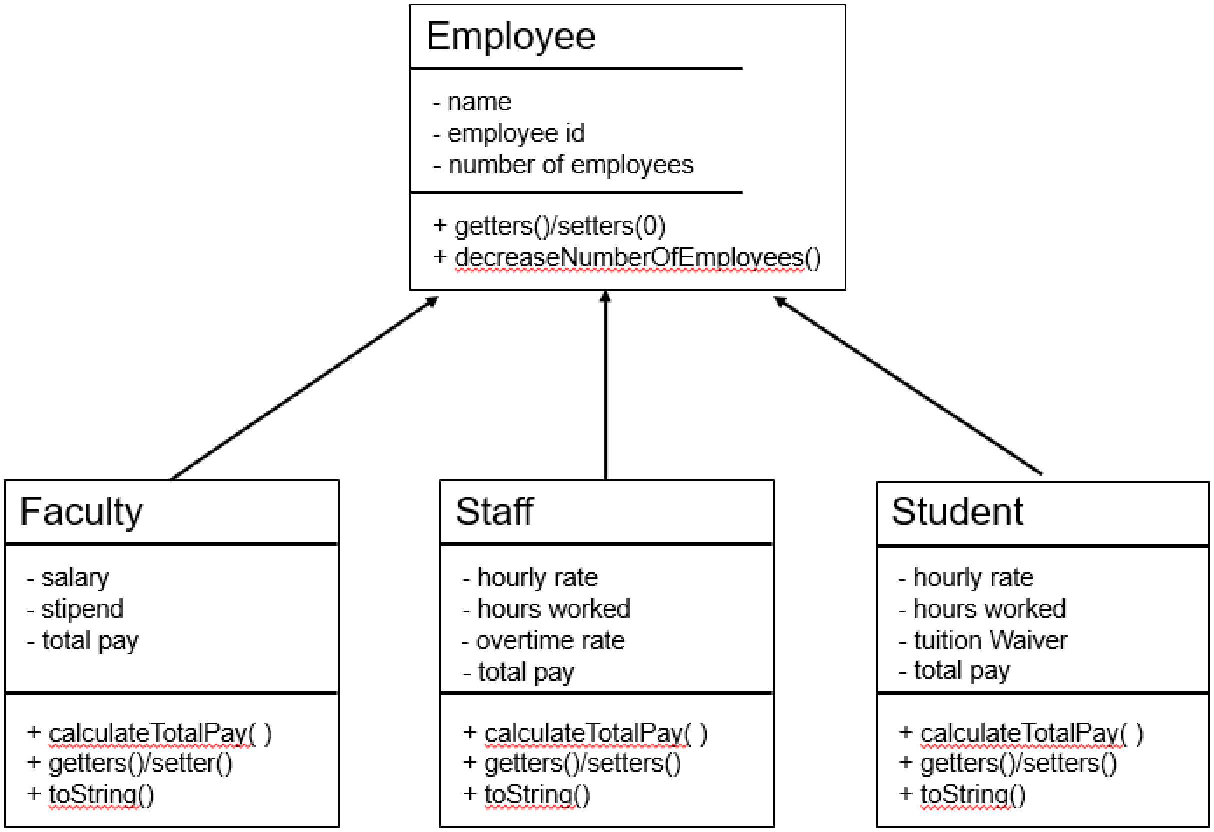
CSE1322L Assignment 3

Background:

Topics

- Inheritance
- Super/Base
- ArraysList (Java) / List (C#)
- Static variables/methods
- ToString Override

For this assignment, you will code “a portion” of a basic payroll management system, which is depicted in the following UML (unified modeling language) diagram:



This assignment is designed to emphasize the topics covered in this week’s classes. In the coming weeks, as we cover more advanced topics (i.e. interfaces, abstraction, casting, and polymorphism), our assignments will include the new topics.

Your task:

For this assignment, you will implement the Employee and Faculty classes, utilizing the fundamentals of inheritance and ArrayLists (Java) / Lists (C#).

Define the following two classes:

1) Employee Class
    a) Must have the following attributes:
        - Name, string, private, employee's full name
        - Employee ID, int, private, unique employee id
        - Number of employees, int, private, static, initialized with zero (0), ongoing number of objects created (incremented each time an object is created)
    b) Must have a default constructor which:
        - assigns a default value of "new employee" to the name attribute
        - assigns the value of ++numberOfEmployees to the employee id attribute (i.e. the creation of each object increments the numberOfEmployees attribute, which is assigned to the employee id attributes)
    c) Must have an overloaded constructor which:
        - takes in a single parameter of type string and assigns it to the name attribute
        - assigns the value of ++numberOfEmployees to the employee id attribute (i.e. the creation of each object increments the numberOfEmployees attribute, which is assigned to the employee id attributes)
    d) Must have getter/setter methods for the name and employee id attributes
    e) Must have a static method called getNumberOfEmployees, which returns the value of the static attribute numberOfEmployees
    f) Must have a static method called decreaseNumberOfEmployees, which returns no value, takes in no parameters, and decrements the static attribute numberOfEmployees by one (1) (i.e. numberOfEmployees --)

2) Faculty Class
    a) The Faculty class inherits the Employee class
    b) Must have the following attributes:
        - Weekly salary: double, private, a faculty member's weekly salary amount

- Weekly stipend: double, private, a faculty member’s weekly stipend amount
- Weekly pay: double, private, a faculty member’s total weekly pay amount

c) Must have an overloaded constructor which:
   - takes in a parameter of type string and calls the default constructor in Employee class to assign the parameter value to the name attribute in the Employee class
   - takes in a parameter of type double and assigns it to the weekly salary attribute
   - takes in a parameter of type double and assigns it to the weekly stipend attribute

d) Must have getter/setter methods for the weekly salary, weekly stipend, and weekly pay attributes

e) Must have a public method called calculateWeeklyPay, which takes in no parameters, returns no values, and calculate the weekly pay amount (i.e. weeklySalary + weeklyStipend) and assigns it to the weekly pay attribute.

f) Must override the tostring method to return a string which includes the label and value of all attributes (i.e. name, weekly salary, weekly stipend, weekly pay)

Driver Program:

In the driver class:

- Create an ArrayList (Java) / List (C#) of Faculty objects.

- Create a method addFaculty, which returns no values and takes in one (1) parameter, an arrayList (Java) / List (C#) of Faculty objects:
  - prompt the user to enter and read in a name
  - prompt the user to enter and read in a weekly salary
  - prompt the user to enter and read in a weekly stipend
  - use the name, weekly salary, and weekly stipend to create a Faculty object and store the object into the arrayList (Java) / List (C#)

- Create a method removeFaculty, which returns no values and takes in one (1) parameter, an arrayList (Java) / List (C#) of Faculty objects:
  - prompt the user to enter and read in a name
  - search the arrayList (Java) / List (C#) for the object containing the user provided name

- if the object is found, remove the object from the arrayList (Java) / List (C#), update the numberOfEmployees attribute, and end the search

- Create a method calculateFacultyPay, which returns no values and takes in one (1) parameter, an arrayList (Java) / List (C#) of Faculty objects:
  - Traverse the entire arrayList (Java) / List (C#)
  - For each object in the arrayList (Java) / List (C#), invoke the calculateWeeklyPay method to compute the weekly pay for the faculty member.

- Create a method printFaculty, which returns no values and takes in one (1) parameter, an arrayList (Java) / List (C#) of Faculty objects:
  - Traverse the entire arrayList (Java) / List (C#)
  - For each object in the arrayList (Java) / List (C#), **print the object**

- Using a loop, prompt the user with the following menu and read in the user's response:

  1 – Add Faculty
  2 – Delete Faculty
  3 – Calculate Faculty Pay
  4 – Print Faculty
  5 – Exit

  Enter Choice:

  - If the user enters 1, invoke the method addFaculty
  - If the user enters 2, invoke the method deleteFaculty
  - If the user enters 3, invoke the method calculateFacultyPay
  - If the user enters 4, invoke the method printFaculty
  - If the user enters 5, terminate the program
  - If the user enters any character other than a 1, 2, 3, 4, or 5, the following error message should display: Error: Please enter valid input, and the user should be allowed to reenter a valid choice.
  - Hint: In Java, you may want to make Scanner a static object in the main class (i.e. *static Scanner input = new Scanner(System.in)*). This allows you to read in input from the console from inside you user-defined methods.

Sample Output:

```
1 – Add Faculty
2 – Delete Faculty
3 – Calculate Faculty Pay
4 – Print Faculty
5 – Exit

Enter Choice: 9
Error: Please Enter Valid Input

1 – Add Faculty
2 – Delete Faculty
3 – Calculate Faculty Pay
4 – Print Faculty
5 – Exit

Enter Choice: 1

Enter Employee Name:
Fred Flintstone
Enter Weekly Salary:
2000.00
Enter Weekly Stipend:
500.00

1 – Add Faculty
2 – Delete Faculty
3 – Calculate Faculty Pay
4 – Print Faculty
5 – Exit

Enter Choice: 3

1 – Add Faculty
2 – Delete Faculty
3 – Calculate Faculty Pay
4 – Print Faculty
5 – Exit
```

```
Enter Choice: 4

Name: Fred Flintstone, Salary: 2000.00, Stipend: 500.00, Weekly Pay: 2500.00
Name: Barney Rubble, Salary: 5000.00, Stipend: 600.00, Weekly Pay: 5600.00

1 – Add Faculty
2 – Delete Faculty
3 – Calculate Faculty Pay
4 – Print Faculty
5 – Exit

Enter Choice: 2

Enter Employee Name: Barney Rubble

1 – Add Faculty
2 – Delete Faculty
3 – Calculate Faculty Pay
4 – Print Faculty
5 – Exit

Enter Choice: 4

Name: Fred Flintstone, Salary: 2000.00, Stipend: 500.00, Weekly Pay: 2500.00

1 – Add Faculty
2 – Delete Faculty
3 – Calculate Faculty Pay
4 – Print Faculty
5 – Exit

Enter Choice:
```

Submitting your answer:

Please follow the posted submission guidelines here:
https://ccse.kennesaw.edu/fye/submissionguidelines.php

Ensure you submit before the deadline listed on the lab schedule for CSE1322L here: https://ccse.kennesaw.edu/fye/courseschedules.php

Rubric:

- Employee Class (20 points total)
  - Includes all attributes, with correct data types and modifiers (i.e. public, private, static) (4 points)
  - Includes default constructor, which initializes attributes (4 points)
  - Includes overloaded constructor, which takes in parameter(s) and initializes attributes (4 points)
  - Includes getter/setter methods for each attribute (4 points)
  - Includes decreaseNumberOfEmployees methods (4 points)

- Faculty Class (30 points total)
  - Inherits Employee class (5 points)
  - Includes all attributes, with correct data types and modifiers (i.e. public, private, static) (4 points)
  - Includes overloaded constructor, which takes in parameters and initializes attributes (4 points)
  - Overloaded constructor invokes parent (base) constructor (5 points)
  - Includes getter/setter methods (4 points)
  - Includes CalculateWeeklyPay method (4 points)
  - Overrides toString method (4 points)

- Driver program (50 points total)
  - Creates ArrayList (Java) / List (C#) of Faculty objects (3 points)
  - Presents menu and reads user input (total 10 points)
    - Prompts user with correct menu and reads in user choice (2 point)
    - Provides error message for invalid user input (2 points)
    - Allows user to reenter choice in case of invalid user input (4 points)
    - Invokes correct method based on user input (2 points)
  - Includes printFaculty method (total 8 points)
    - includes correct parameter(s) (2 points)
    - traverses entire ArrayList (Java) / List (C#) (3 points)
    - prints object via toString method (3 points)
  - Includes calculateFacultyPay method (total 7 points)
    - includes correct parameter(s) (2 points)
    - traverses entire ArrayList (Java) / List (C#) (3 points)
    - invokes correct method to compute faculty pay (2 points)

- Includes removeFaculty method (total 12 points)
  - includes correct parameter(s) (2 points)
  - prompts user for input and reads in input (2 points)
  - searches ArrayList (Java) / List (C#) to find object (3 points)
  - removes object from ArrayList (Java) / List (C#) (3 points)
  - update number of employees attribute (2 points)
- Includes addFaculty method (total 10 points)
  - includes correct parameter(s) (2 points)
  - prompts user for input and reads in input (2 points)
  - creates new Faculty object (3 points)
  - adds object to ArrayList (Java) / List (C#) (3 points)