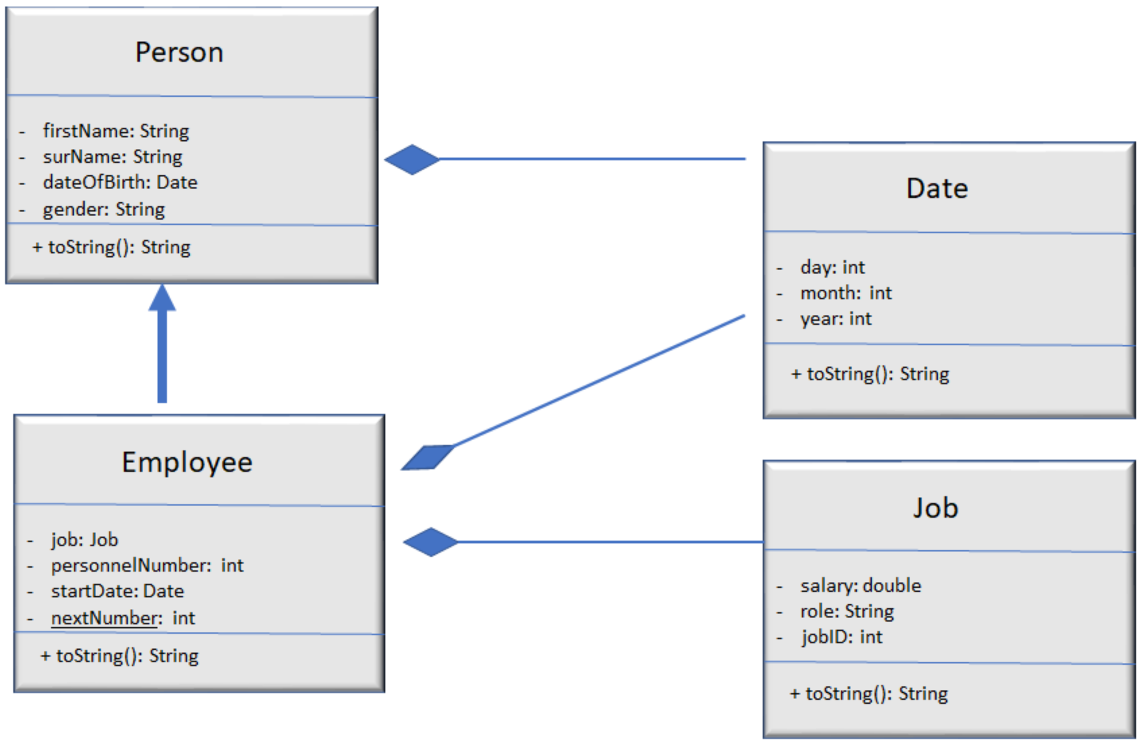
The purpose of this lab is to some file processing and take a look at Composition in Java.

To Debug your java code in Eclipse, follow the notes put up in Webcourses.



Part 1 – set up Job and Date class

Set up a **Date** class and **Job** class as shown, noting the following:

- Encapsulate attributes, using getters and setter methods
- **Date** class constructor: should prevent invalid days as follows: days must be from 1 to 31, and months 1 – 12.
- **Job** class constructor: set up all attributes with incoming values.
- Write a toString() method to print the attribute values

Test your code by instantiating sample Date and Job objects and "system.out.println"ing the objects from a "main" method in another "Control" class (or whatever you decide to call it).

Part 2 – set up Classes that have (i.e. " are composed of") these classes

Set up a P**erson** and **Employee** classes as shown, noting:

- Employee inherits from the Person class (“extends”);
- personnelNumber, which is a unique number, increments by 1 for each new employee object, using static attribute nextNumber as a counter;
- Set up a constructor for each of the classes, setting all attributes of the class;
- Include toString() method in each class;
- Encapsulate the attributes.

Test your code by instantiating sample Employee and Person objects and “system.out.println”ing the objects from a “main” method in another “Control” class (or whatever you decide to call it).

Part 3 – Working with files – reading

Edit your **Job class constructor**, so that it validates the “role” value being sent in to the constructor as follows: the role is valid if it exists in the text file, “roles.txt”. Your constructor will need to read in the values from the file, and compare to the “role” values used when creating a “job” object.

To do this, create a separate **FileProcessor** class that has various methods to connect to the file, read the file etc – and then use this File class from your Job class constructor.

Part 4 – Working with files - writing

Add another method to your **FileProcessor** class so that it can be used by the **Employee** class to write text to a file. The toString() method of the Employee class should write the employee names to a **file** too – “names.txt” - as well as returning a String.

(Don’t forget to close the PrintWriter object each time, or your file writing won’t work).