Let us assume the pays of employees on a weekly basis. The employees are of four types:

> Salaried employees are paid a fixed weekly salary regardless of the number of hours worked,

> hourly employees are paid by the hour and receive overtime pay (i.e., 1.5 times their hourly salary rate) for all hours worked in excess of 40 hours,

> commission employees are paid a percentage of their sales and

> base-salaried commission employees receive a base salary plus a percentage of their sales.

For the current pay period, the company has decided to reward Salaried-commission employees by adding 10% to their base salaries. The company wants you to write an application that performs its payroll calculations polymorphically.

We use abstract class Employee to represent the general concept of an employee. The classes that extend Employee are SalariedEmployee, CommissionEmployee and Hourly-Employee. Class BasePlusCommissionEmployee—which extends CommissionEmployee—represents the last employee type.

Abstract superclass Employee declares the "interface" to the hierarchy—that is, the set of methods that a program can invoke on all Employee objects. We use the term "interface" here in a general sense to refer to the various ways programs can communicate with objects of any Employee subclass. Be careful not to confuse the general notion of an "interface" with the formal notion of a Java interface.

Each employee, regardless of the way his or her earnings are calculated, has a first name, a last name and a social insurance number, so private instance variables firstName, lastName and social-InsuranceNumber appear in abstract superclass Employee.

We do not list superclass Employee's get methods because they're not overridden in any of the subclasses—each of these methods is inherited and used "as is" by each subclass.

| | Earnings | toString |
|---|---|---|
| Employee | abstract | firstName lastName<br>Social insurance number: sin |
| Salaried-Employee | weeklySalary | salaried employee: *firstName lastName*<br>social insurance number: *SIN*<br>weekly salary: *weeklySalary* |
| Hourly-Employee | if (hours <= 40)<br><br>wage * hours<br>else if (hours > 40)<br>{<br>40 * wage +<br>( hours - 40 ) *<br>wage * 1.5<br>} | hourly employee: *firstName lastName*<br>social insurance number: *SIN*<br>hourly wage: *wage*; hours worked: *hours* |
| Commission-Employee | commissionRate * grossSales | commission employee: *firstName lastName*<br>social insurance number: *SIN*<br>gross sales: *grossSales*;<br>commission rate: *commissionRate* |
| Base- plus-Commision employee: | (commissionRate *grossSales)+baseSalary | base salaried commission<br><br>*firstName lastName*<br>social insurance number: *SIN*<br>gross sales: *grossSales*;<br>commission rate: *commissionRate*;<br>base salary: *baseSalary* |