**Payroll System Using Polymorphism**

**due 12.14.2024**

* Use an abstract method and polymorphism to perform payroll calculations based on the type of inheritance hierarchy headed by an employee.
* Enhanced employee inheritance hierarchy requirements:

A company pays its 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 to write a Java application that performs its payroll calculations polymorphically.

Polymorphic interface for the Employee hierarchy classes

|  |  |  |
| --- | --- | --- |
| Classes | Earnings | EmployeeInfo / toString Method |
| Abstract Superclass Employee |  | firstName lastName  social security number:SSN |
| Salaried-Employee | weeklySalary | salaried employee:firstName lastName  social security number: SSN  weekly salary:weeklysalary |
| Hourly-Employee | If (hours<=40)  wages \*hours  else if (hours > 40) {  40 \* wages+ (hours – 40) \* wage \*1.5  } | hourly employee: firstName /lastName  social security number: SSN  hourly wage: wage;  hours worked: hours |
| Commission-Employee | commisionRate \* grossSales | commission employee: firsrName lastName  Social security number:SSN  gross sales: grossSales  commission rate: commissionRate |
| BasePlus-Commission-Employee | (commisionRate \* grossSales) + baseSalary | base salaried commission employee: firstName lastName  social security number:SSN  gross sales: grossSales;  commission rate: commissionRate;  base salary: baseSalary |