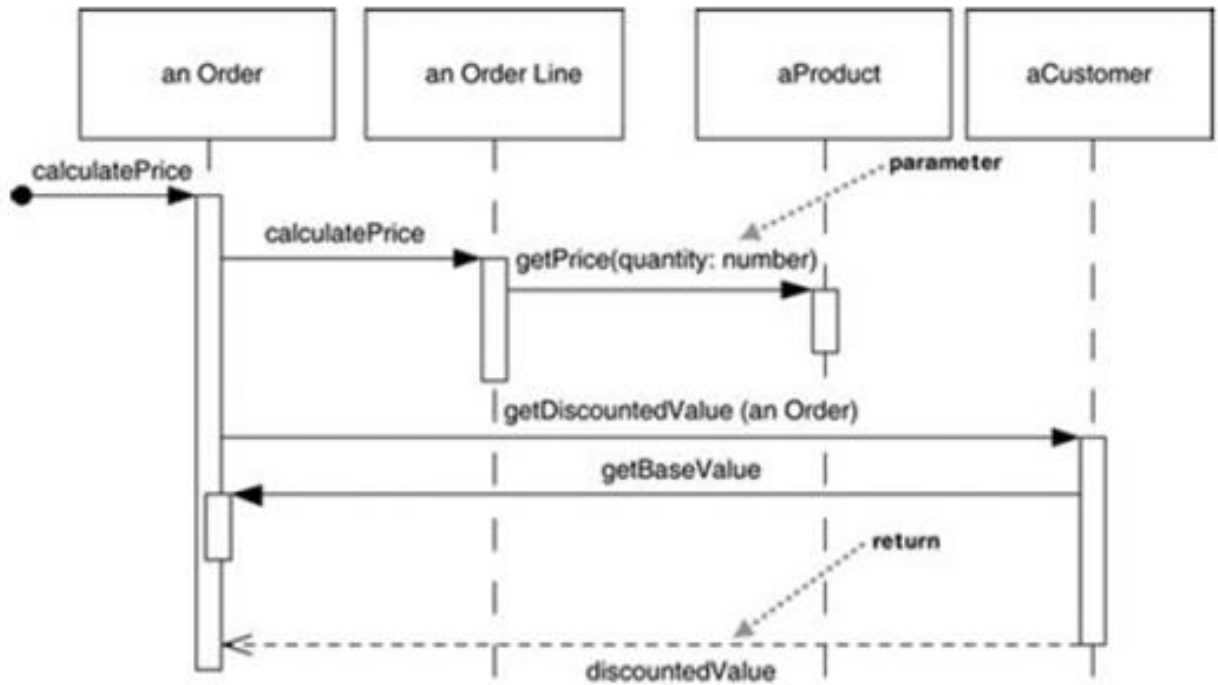


Lab 4 Parts A, B

A. Re-do the sequence diagram on Slide 16 of Lesson 4 (reproduced below).



In your diagram, be sure to include the following:

- Message numbering
- An actor who initiates action
- Proper UML syntax for the objects displayed at the top.
- Iteration markers where looping occurs

- [illegible]

Lab 4 C

POLYMORPHISM

Payroll Calculation

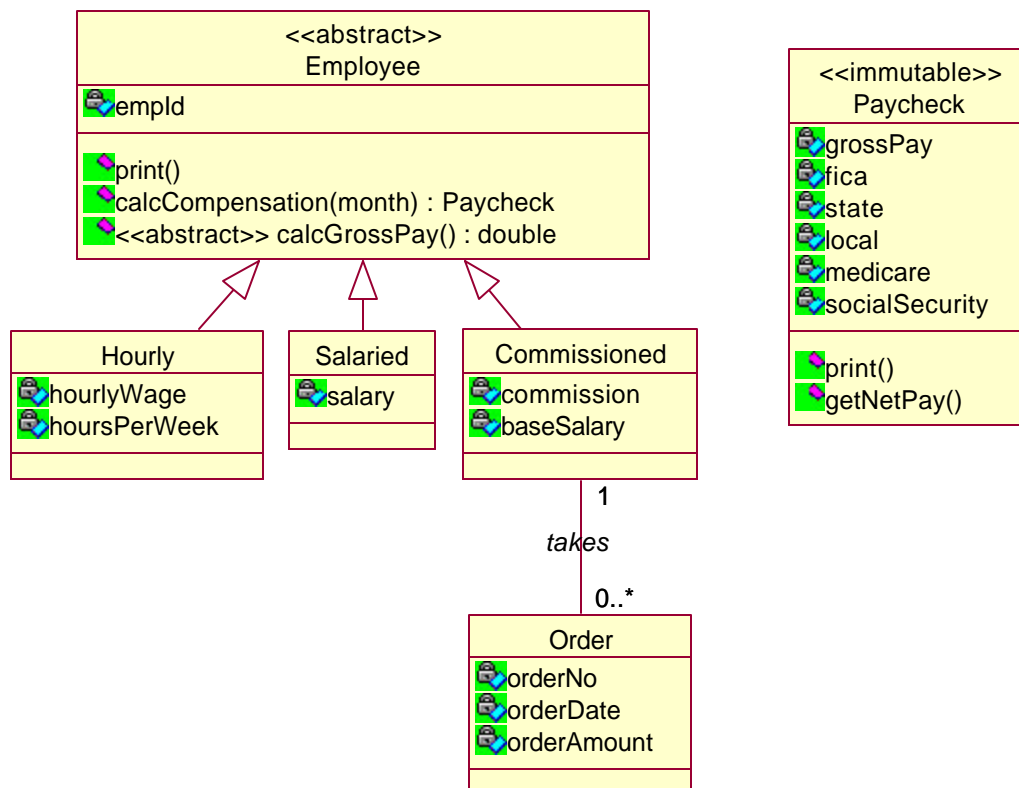
Objectives:

- Practice implementing and using polymorphic methods.
- Understand how the Template Method design pattern works.

Human Resource Application:

The HR department has identified three types of Employees based on how they are paid: Hourly, Salaried, and Commissioned. The paycheck for each type is calculated differently:

- Hourly employees are paid monthly and their paycheck amount is calculated based on their hourly wage and the hours per week they work. For simplicity just assume four weeks for each month.
- Salaried employees are paid monthly and their paycheck amount is a fixed amount every month.
- Commissioned employees are also paid monthly. They receive a small base salary, plus a percentage (commission) on the total value of all orders they sold during the previous month.



Lab 4

Page 2

POLYMORPHISM
Payroll Calculation

Tasks:

1. Add a concrete `calcCompensation()` method to `Employee`.
 - a. This method takes the month and year as arguments for which to calculate the compensation.
 - b. For hourly and salaried employees the amount is the same every month.
 - c. For Commissioned employees the amount depends on the orders they sold that month.
2. The `Employee.calcCompensation()` method delegates to the respective derived class to calculate the gross pay amount by invoking the abstract `Employee.calcGrossPay()` method.
3. The `Employee.calcCompensation()` method then calculates the FICA, state & local taxes, medicare and social security contributions based on the gross pay. Assume the following fixed tax percentages:
 - FICA is 23%
 - State tax is 5%
 - Local tax is 1%
 - Medicare is 3%
 - Social Security is 7.5%

Important details about the UML class diagram:

- `Paycheck` class are immutable, i.e. all data needs to be passed to the constructor and no setter methods should be provided.
- `Employee` is an abstract class!
- `Employee.calcCompensation()` returns a `Paycheck` object!
- In order to calculate the paycheck for a Commissioned employee you need to access all the `Order` objects that each Commissioned employee is responsible for and add up the order amount of all orders during a given month.

Lab 4, Part D

- D. Create a sequence diagram for the problem described in Lab 4, Part C. Create a distributed control solution. As you distribute control, make sure that the object that handles a step of processing really should be responsible for that behavior, based on the purpose of the class that was determined in the class diagram.