Employee (interface)
- amountDue() : Integer
- amountDue(int) : Integer

AbstractEmployee
- NEGATIVE_OVERTIME_MSG : String
- rate : Integer
- hours : Integer
- AbstractEmployee(Integer, Integer)
- getRate() : Integer
- getHours() : Integer
- equals(Object) : boolean
- hashCode() : int
- amountDue() : Integer
- amountDue(int) : Integer

IManagement (interface)
- DISMISS_MSG : String
- dismiss() : void

IWorker (interface)
- COMPLAIN_MSG : String
- complain() : void

Management
- subordinates : Integer
- subordinateCost : Integer
- Management(Integer, Integer, Integer, Integer)
- Management(Integer, Integer, Integer)
- getSubordinateCost() : Integer
- getSubordinates() : Integer
- amountDue() : Integer
- dismiss() : void

Worker
- Worker(Integer, Integer)
- complain() : void

IllegalOvertimeException
- IllegalOvertimeException(String)

Supervisor
- COMPENSATION_PER_EACH_WAREHOUSEWORKER : int
- Supervisor(Integer, Integer, Integer)

Dispatcher
- COMPENSATION_PER_DRIVER : Integer
- Dispatcher(Integer, Integer, Integer)

Manager
- COMPENSATION_PER_EACH_CASHIER : Integer
- Manager(Integer, Integer, Integer)

Cashier
- Cashier(Integer, Integer)

WarehouseWorker
- ALLOWED_OVERTIME : int
- OVERTIME_NOT_ALLOWED_MSG : String
- WarehouseWorker(Integer, Integer)
- amountDue(int) : Integer

Driver
- OVERTIME_NOT_ALLOWED_MSG : String
- Driver(Integer, Integer)
- amountDue(int) : Integer

Level 0: (Employee)
The highest level of parent interface/class is the interface of Employee

Level1: (IManagement, AbstractEmployee, IWorker)
They are subclass/interface of Employee:
IManagement: follow LSP, as it doesn't change any behavior defined in Employee.

AbstractEmployee: doesn't follow LSP. The method amountDue(int) needs normal precondition, but it throws exceptions when argument is negative, so it has weaker postcondition.

IWorker: follow LSP, as it doesn't change any behavior defined in Employee.

Level 2: (Management, Worker)
The class Management implements IManagement.
Management doesn't change the behavior of methods defined in IManagement. So it follows LSP when it inherits from IManagement.

The class Worker implements IWorker.
Worker doesn't change the behavior of methods defined in IWorker. So it follows LSP, when it inherits from IWorker.

Both Management and Worker classes also extends AbstractEmployee.

Woker doesn't change the behavior of methods defined in AbstractEmployee. So it follows LSP when it inherits from AbstractEmployee. The precondition for AmountDue() requires two fields

hours and rate, which are same to that from AbstractEmployee. And the output is also the same to the method defined in AbstractEmployee.

However, Management change the behavior of methods AmountDure() from AbstractEmployee, so it doesn't follow LSP, when inherit from AbstractEmployee.
The precondition for AmountDue() in Management requires four fields hours, rate, subordinates and subordinateCost. The condition is stronger than that of AbstractEmployee. And the output is different to the method defined in AbstractEmployee.

Level 3: Supervisor, Dispatcher and Manager; Cashier, Driver and WarehouseWorker.
Management has three subclasses: Supervisor, Dispatcher and Manager.
All three subclass doesn't follow LSP. As the precondition for method in the subclasses are stronger than that in the Management class. All three class require specific subordinateCost value.

Worker has three subclasses: Cashier, Driver and WarehouseWorker.
Here only Cashier follows LSP.
Driver and WarehouseWorker don't follow LSP, because each of them has stronger precondition requirement.
Driver are not allowed to work overtime.
WarehouseWorker has the work overtime limitation.