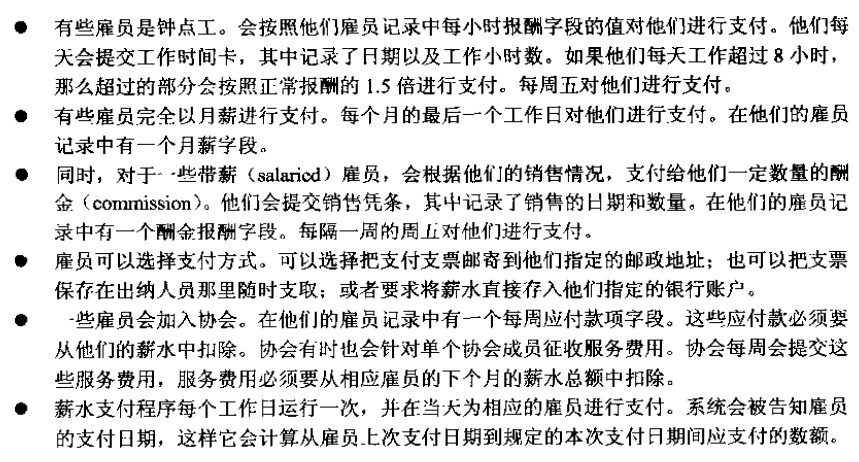
**用户素材**：



**用例分析（细化用户素材，大致列出每个用例的输入输出）**：

1. 新增雇员

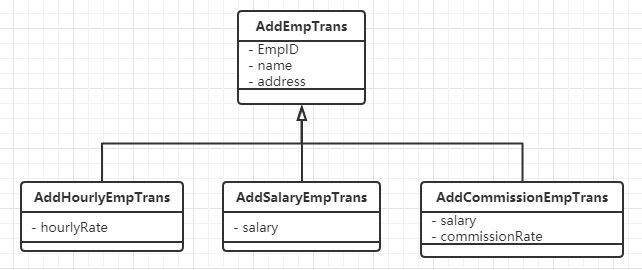
增加钟点工：AddEmp <EmpID> <name> <address> <hourly-rate>

增加月薪工：AddEmp <EmpID> <name> <address> <monthly-salary>

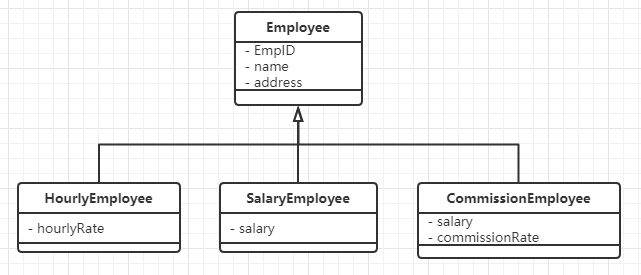
增加带新雇员：AddEmp <EmpID> <name> <address> <monthSalary> <commission-rate>

**提取抽象，隔离稳定和变化，进行初步设计**

这三个操作有三个相同的字段，可以提取出来作为基类



三种雇员：



1. 删除雇员

DelEmp <EmpID>

1. 修改雇员

修改名字：ChangeEmp <EmpID> Name <name>

修改地址：ChangeEmp <EmpID> Address <address>

修改时薪：ChangeEmp <EmpID> Hourly-Rate <hourly\_rate>

修改月薪：ChangeEmp <EmpID> Salary <salary>

修改酬金：ChangeEmp <EmpID> Commission <salary> <commission\_rate>

邮寄支票：ChangeEmp <EmpID> Mail <mail\_address>

持有支票：ChangeEmp <EmpID> Hold

薪水打到银行账户：ChangeEmp <EmpID> Direct <bank> <account>

加入协会：ChangeEmp <EmpID> Member <memberID> Dues <rate>

退出协会：ChangeEmp <EmpID> NoMember

**提取抽象，隔离稳定和变化，进行概要设计**

1. 月薪雇员可以修改为钟点工，钟点工也可以修改为月薪雇员 -- 雇员对象的类型可以修改，是变化的。如果采用之前的雇员类型设计，雇员对象类型的修改就很麻烦，会出现僵化性等臭味。

**隔离稳定和变化**

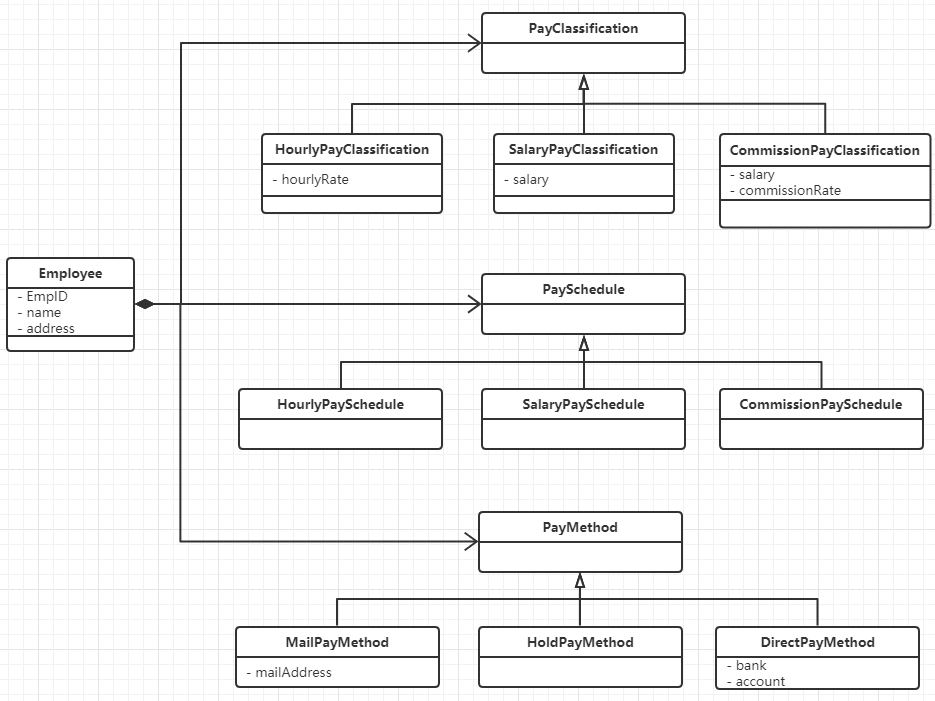
稳定的部分是雇员的基本属性：姓名和地址；

变化的部分是雇员薪水的计算方式和发放时间，钟点工、月薪雇员和带薪雇员的本质区别就是薪水的计算方式和发放时间。

我们将稳定的雇员属性抽象为Employee类，从变化的部分（薪水的计算方式和发放时间）抽象出稳定的接口PayClassification，将不稳定的部分隐藏在PayClassification接口之后。稳定的Employee类依赖于稳定的PayClassification接口。

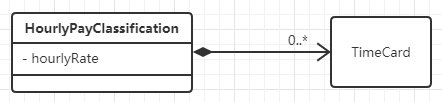
进一步分析，薪水发放时间也可能会独立于薪水的计算方式发生变化，它们之间是相互独立的。所以薪水发放时间应该抽象为一个稳定的接口PaySchedule，不应该藕合进薪水的计算方式。

1. 支付方式也是变化的，雇员可以选择任意一种支付方式。我们从支付方式中提取出接口PayMethod，将支付方式的变化隐藏在该接口之后。Employee依赖接口PayMethod。



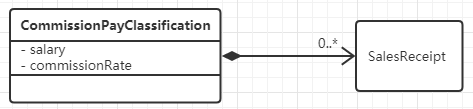
1. 登记工作时间卡

只对钟点工有效：TimeCard <EmpID> <date> <hours>



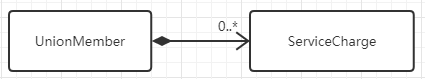
1. 登记销售凭条

只对带薪雇员有效：SalesReceipt <EmpID> <date> <amount>



1. 登记协会服务费

ServiceCharge <memberID> <amount>



1. 支付薪水

Payday <date>

* 钟点工：每周五支付工资，根据工作时间卡计算小时数。超过8小时的按1.5倍算；
* 月薪工：每个月的最后一天支付
* 带薪工：每个月的最后一天支付工资，每隔周五支付酬金，酬金记录在销售凭条中。
* 服务费用：需要在薪水中扣除当期的协会会费和服务费用。

注意：

1. 不要支付当前支付期外的工资和酬金

2. 根据支付方式支付