

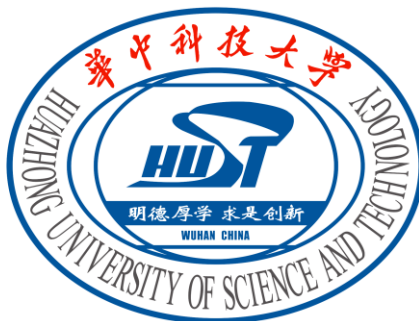
# 基于Java的面向对象程序设计

陈维亚

*weiya\_chen@hust.edu.cn*

华中科技大学软件学院

## 第16讲：类的关系综合练习



1. 报纸与新闻
2. 分组的策略
3. 薪水支付

## □ 问题描述

java.util.ArrayList

```
ArrayList<T> al = new ArrayList<T>;  
T x = al.get(i);  
al.set(i, x);
```

```
T[] array = new T[5];  
T x = array[i];  
array[i] = x;
```

一份报纸Journal中包含了一系列事件Event，每个事件包含该事件发生的日期和内容（一段文字描述）。报纸类应该具有如下功能：

aJournal.add();

aJournal.toString();

aJournal.toString(String s);

## □ 问题描述

- A. Journal类继承ArrayList类，请写出Journal类和Event类。
- B. 完成一个包含main函数的TestJournal类，使其可处理如下输入：
  - 1) +text：在Journal中增加一个事件（当前系统时间）；
  - 2) ?：显示Journal中的所有事件；
  - 3) ? string：列出所有包含了string的所有事件；
  - 4) \*：退出程序。
- C. 重写Journal类，ArrayList不再是其父类，而是Journal的一个成员变量。不要改动TestJournal类。
- D. 请对两种Journal类的实现方式做出评论。

## 2. 分组的策略



一个整数分类器IntegerGroup可将1-n之间的整数按照奇偶性分类，请对其进行修改，使该分类器可按照不同的分类策略对整数进行分类：

- 1) 对3求余分成3组;
- 2) 个位是否为5分2组;

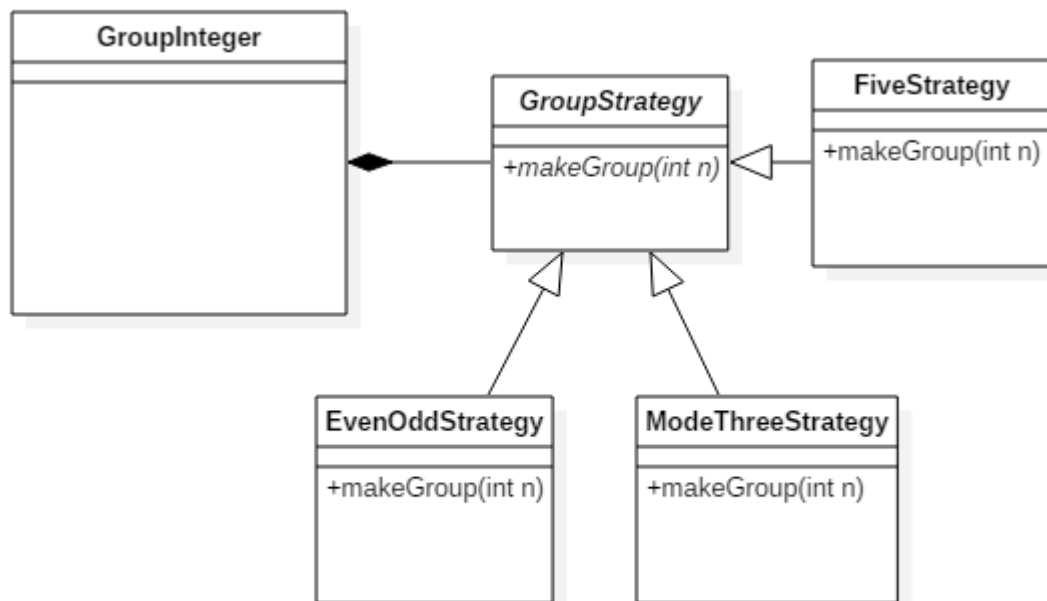
```
public class IntegerGroup {  
    public void divideInteger(int n){  
        // 按奇偶性进行分类  
    }  
}
```

## 2. 分组的策略

一个整数分类器IntegerGroup可将1-n之间的整数按照奇偶性分类，请对其进行修改，使该分类器可按照不同的分类策略对整数进行分类：

- 1) 对3求余分成3组
- 2) 个位是否为5分2组

方案一

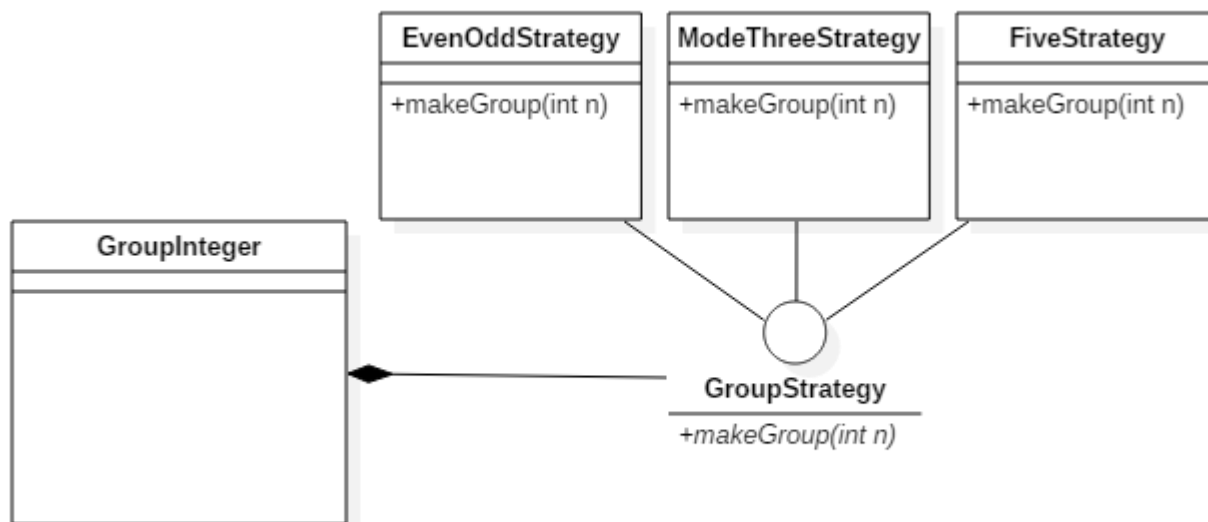


## 2. 分组的策略

一个整数分类器IntegerGroup可将1-n之间的整数按照奇偶性分类，请对其进行修改，使该分类器可按照不同的分类策略对整数进行分类：

- 1) 对3求余分成3组
- 2) 个位是否为5分2组

方案二



### 3. 薪水支付

#### ❑ 华山派的薪水支付案例（IBM 刘欣）

令狐冲在华山之巅向风清扬请教面向对象的绝学。

风前辈说：“我给你出个考题，你先来尝试着用你学的面向对象设计一下。让我想想啊，嗯，你的师傅岳不群虽然人品不端，但是经营华山派还算马马虎虎，除了你们几个师兄弟外，还雇了不少佣人帮着干活。咱们就拿这个作为例子来学一下，听好了！”





你师傅那里有个表格，记录着各种佣人的情况，佣人主要分为这么几类：

1. 正式工，每天都来华山干活，每个月的最后一天给他们付工钱，在他们的佣人记录中有个月薪字段。
2. 有些佣人是钟点工，他们每天提交工作时间卡，其中记录了日期以及工作时辰数，如果每天工作超过6个时辰，按1.5倍进行支付。每周五发工钱。每个时辰的报酬也是你师傅定的。
3. 还有一些带薪的佣人，和正式工类似，会帮助销售咱们华山的土特产，你师傅根据他们的销售情况，支付给他们一定数量的佣金，他们会提交销售凭条，其中记录了销售的日期和数量，对这种佣人，每隔一周的周五发一次工钱。

这些佣人可以选择支付方式，可以把银票邮寄到他们指定的邮政地址，也可以保存在你师傅那里随时支取，或者要求直接存入他们指定的票号里去。

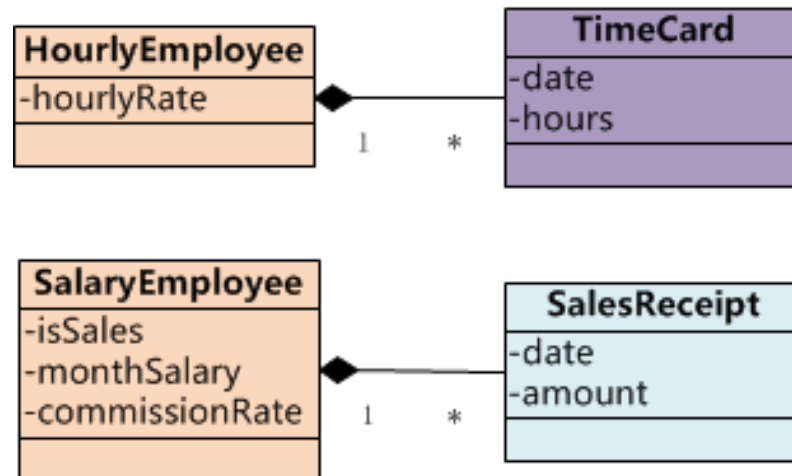
### 3. 薪水支付



你师傅那里有个表格，记录着各种**佣人**的情况，佣人主要分为这么几类：

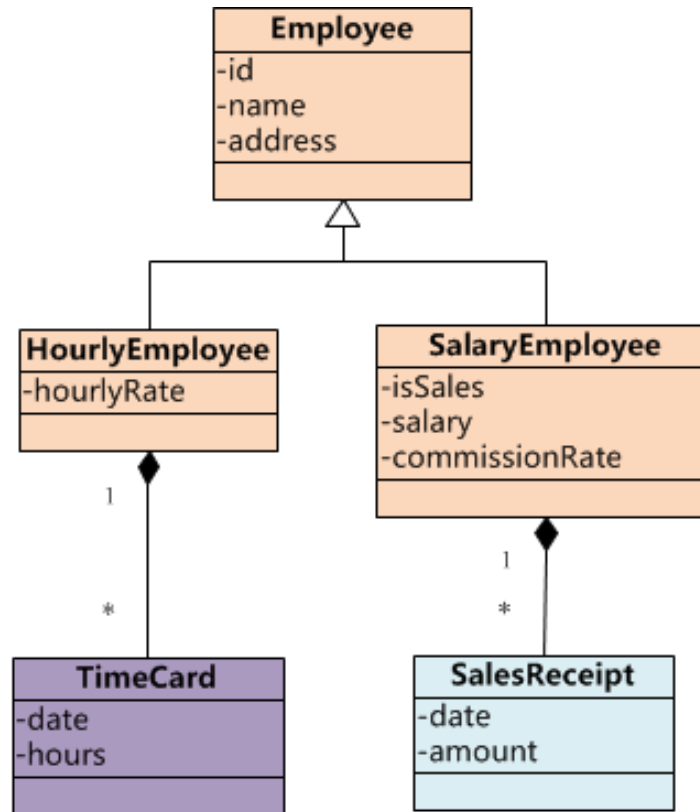
1. **正式工**，每天都来华山干活，每个月的最后一天给他们付工钱，在他们的佣人记录中有个**月薪**字段。
  2. 有些佣人是**钟点工**，他们每天提交**工作时间卡**，其中记录了**日期**以及工作**时辰数**，如果每天工作超过6个时辰，按1.5倍进行支付。每周五发工钱。每个时辰的报酬也是你师傅定的。
  3. 还有一些**带薪的佣人**，和正式工类似，会帮助销售咱们华山的土特产，你师傅根据他们的销售情况，支付给他们一定数量的佣金，他们会提交**销售凭条**，其中记录了销售的**日期**和**数量**，对这种佣人，每隔一周的周五发一次工钱。
- 这些佣人可以选择**支付方式**，可以把银票邮寄到他们指定的邮政地址，也可以保存在你师傅那里随时支取，或者要求直接存入他们指定的票号里去。

### 3. 薪水支付

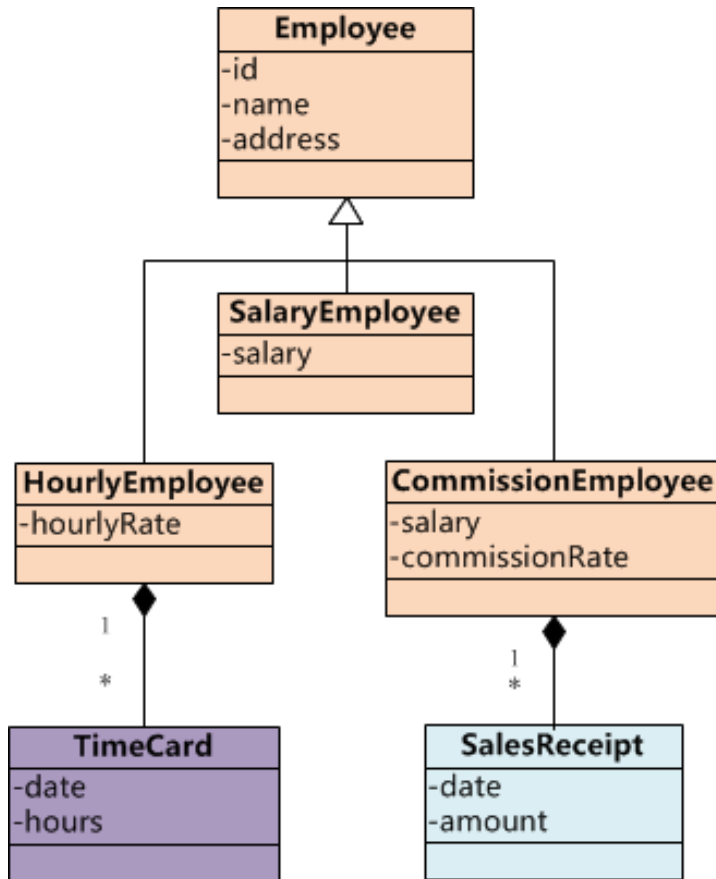


有没有可改进的地方呢？

### 3. 薪水支付



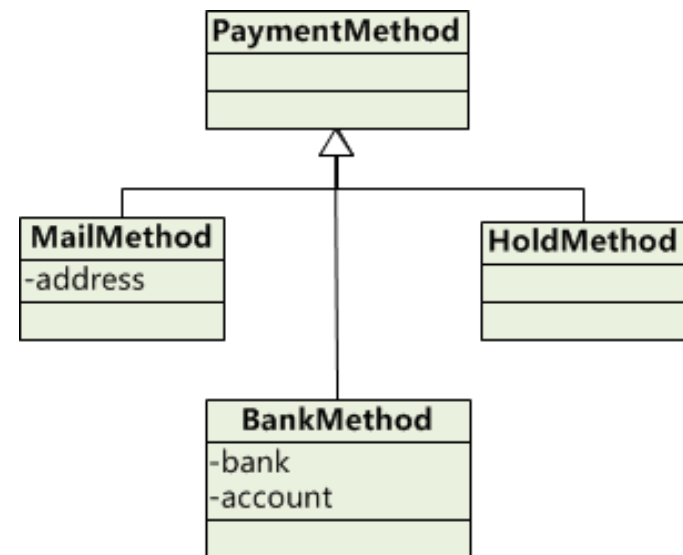
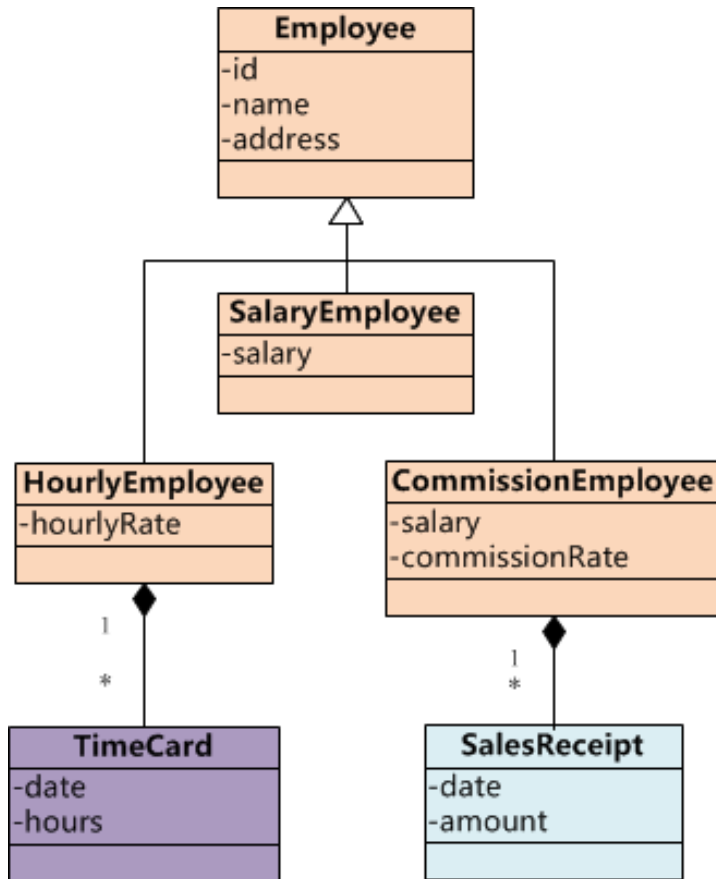
### 3. 薪水支付



风清扬说：“现在考虑下支付的方式吧。  
一种是邮寄，一种是直接取，一种是存入票号。”

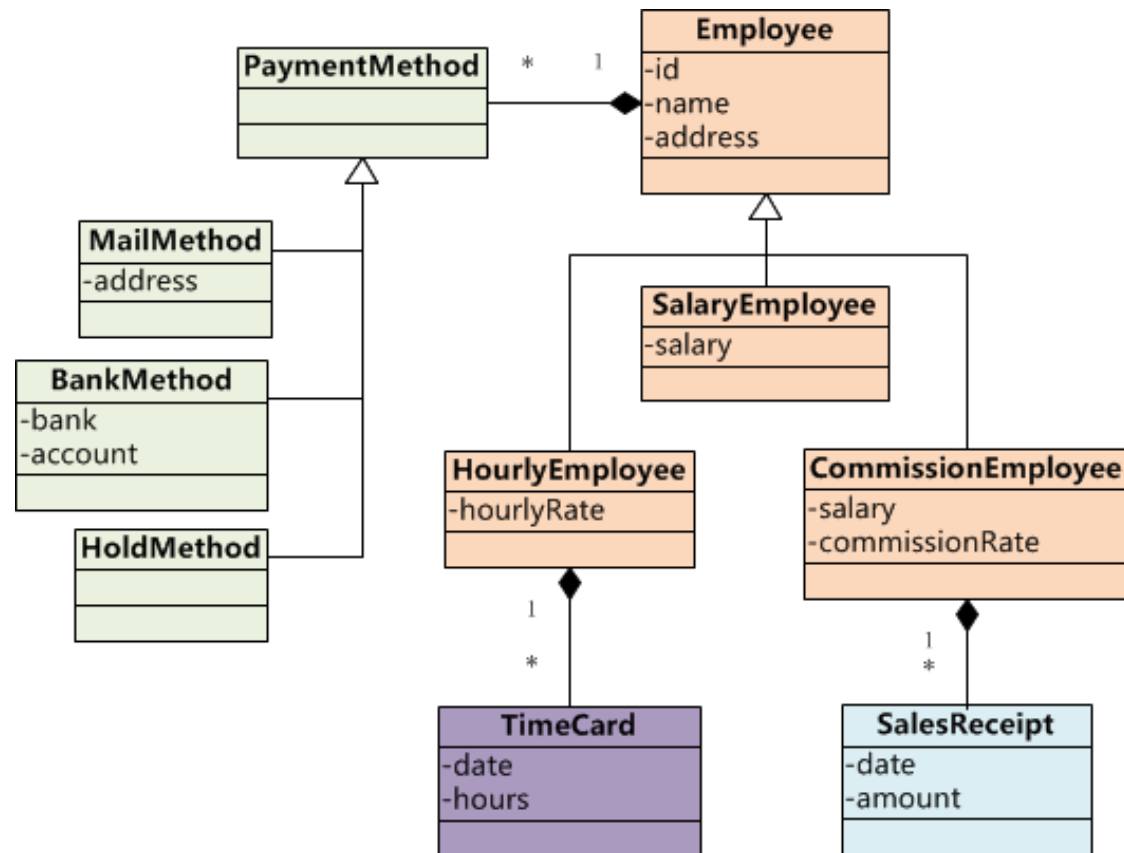
### 3. 薪水支付

风清扬说：“现在考虑下支付的方式吧。  
一种是邮寄，一种是直接取，一种是存入票号。”



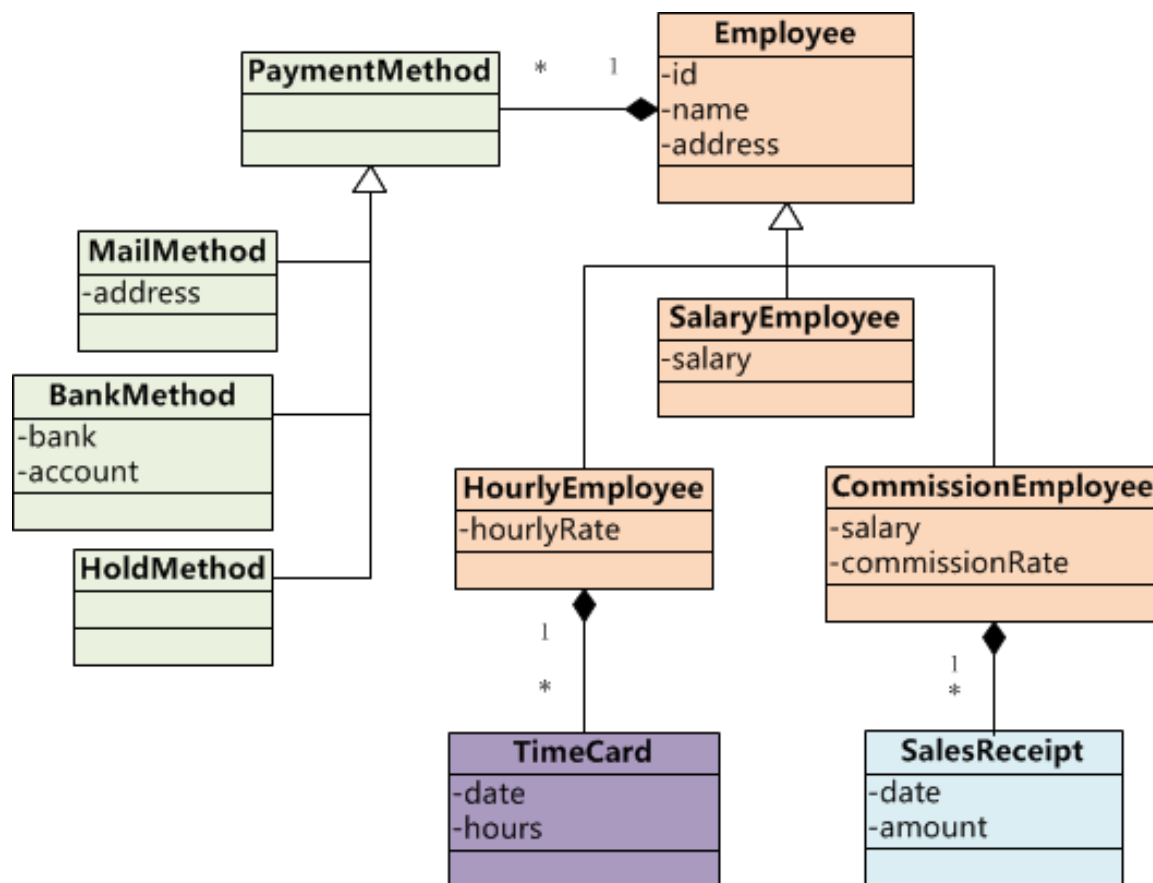
### 3. 薪水支付

风清扬说：“现在考虑下支付的方式吧。  
一种是邮寄，一种是直接取，一种是存入票号。”



### 3. 薪水支付

现在重点的部分来了，你师傅有一天突然决定，**佣人的类型可以来回改变**，一个钟点工可以改成正式工，正式工可以改成卖土特产的佣人，卖土特产的也能改成其他两类。

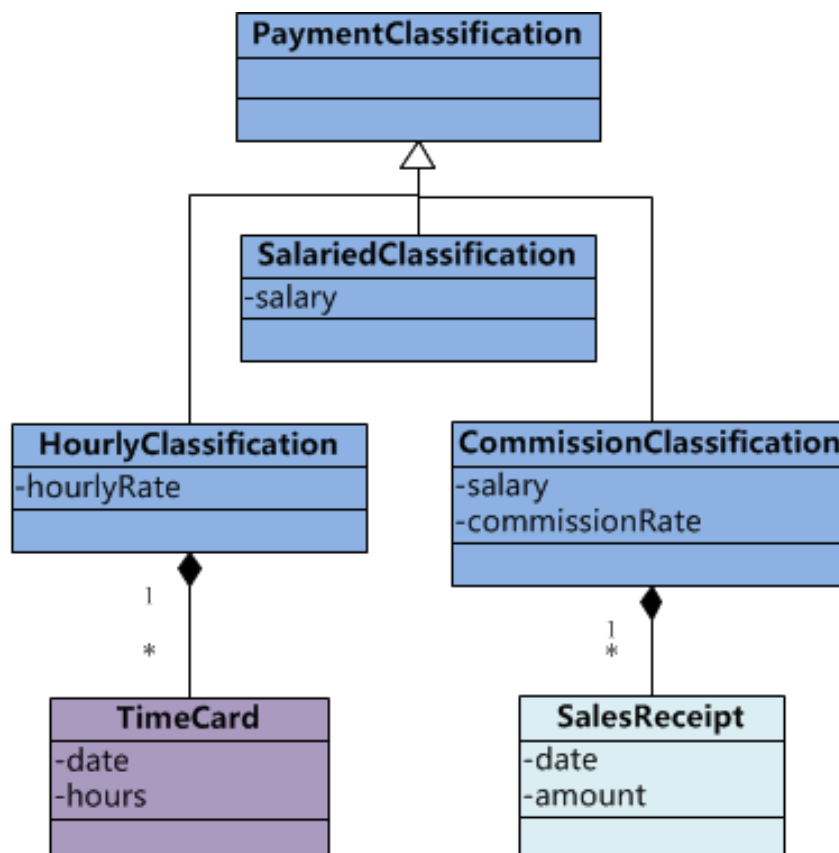




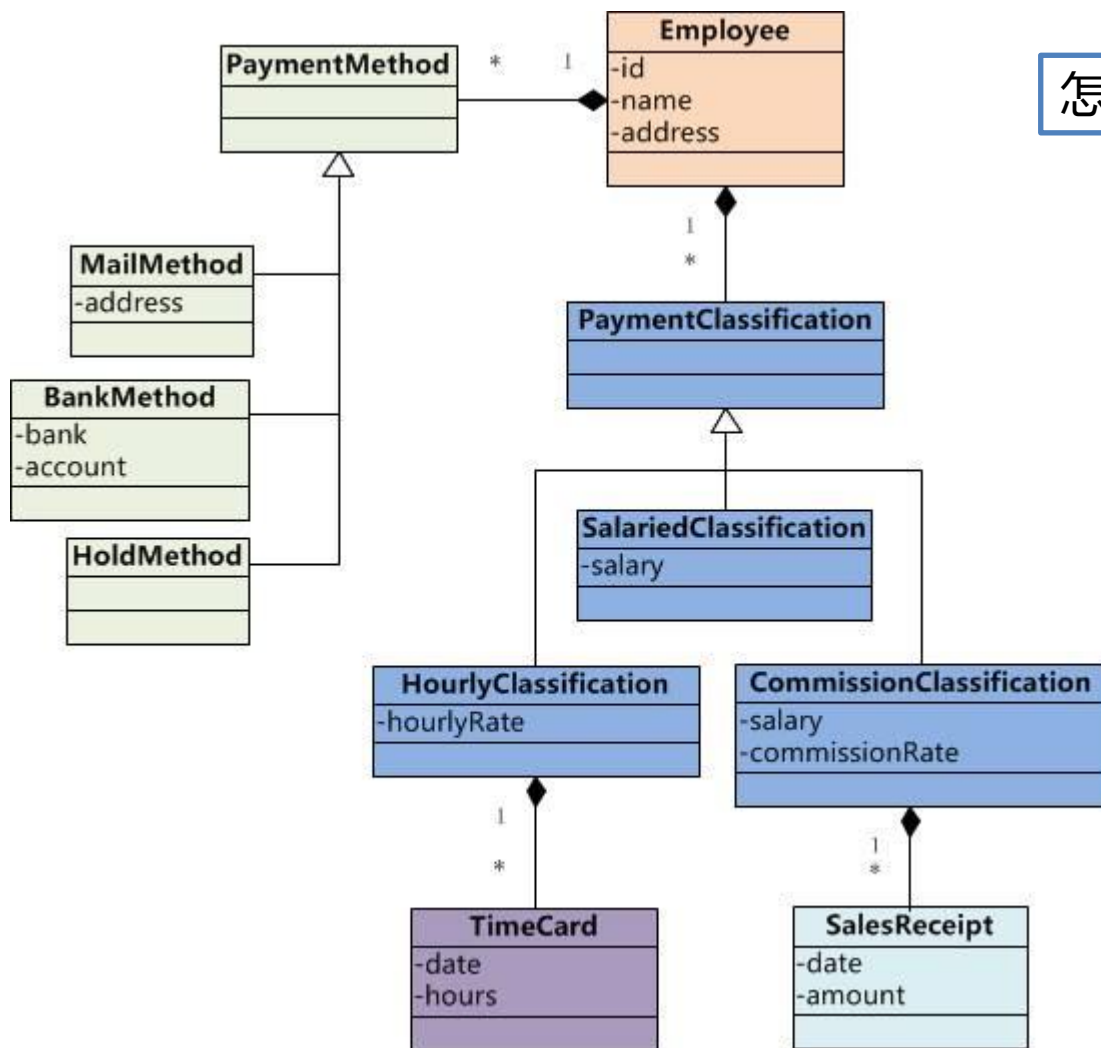
### 3. 薪水支付



仔细想想，做一个**抽象**的时候到了，一些佣人按时辰获得工钱，一些佣人按月获得工钱，还有一些按销售获得酬金，这其实暗含着：**所有佣人都被支付，只是支付的策略不同**

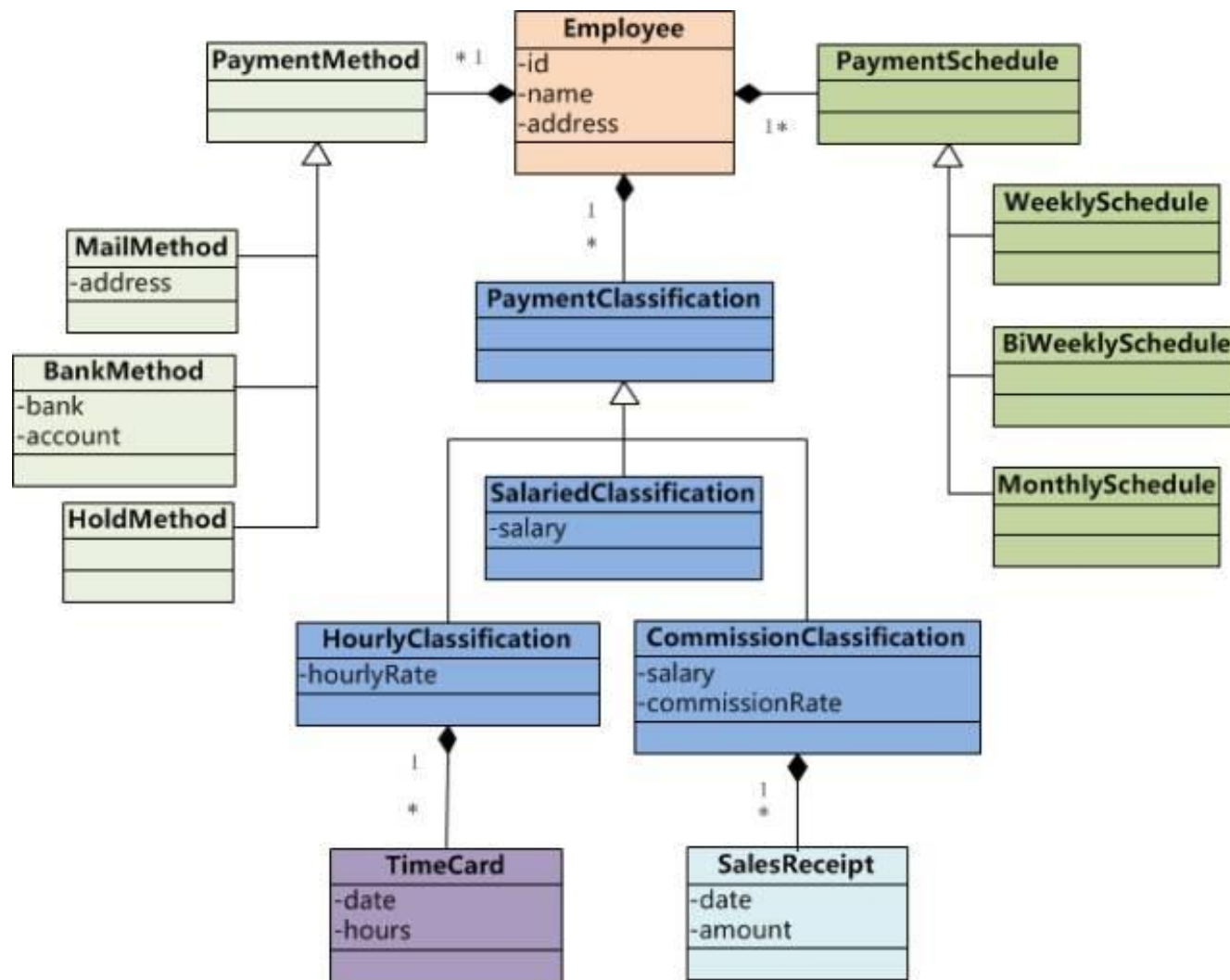


### 3. 薪水支付



怎么处理发工钱的日期呢？

### 3. 薪水支付



你学会了封装，继承，多态，也学了设计模式，但是这些都是具体的招式，只有学会了抽象，才是应付千变万化的终极招式！

## Java标准库介绍