

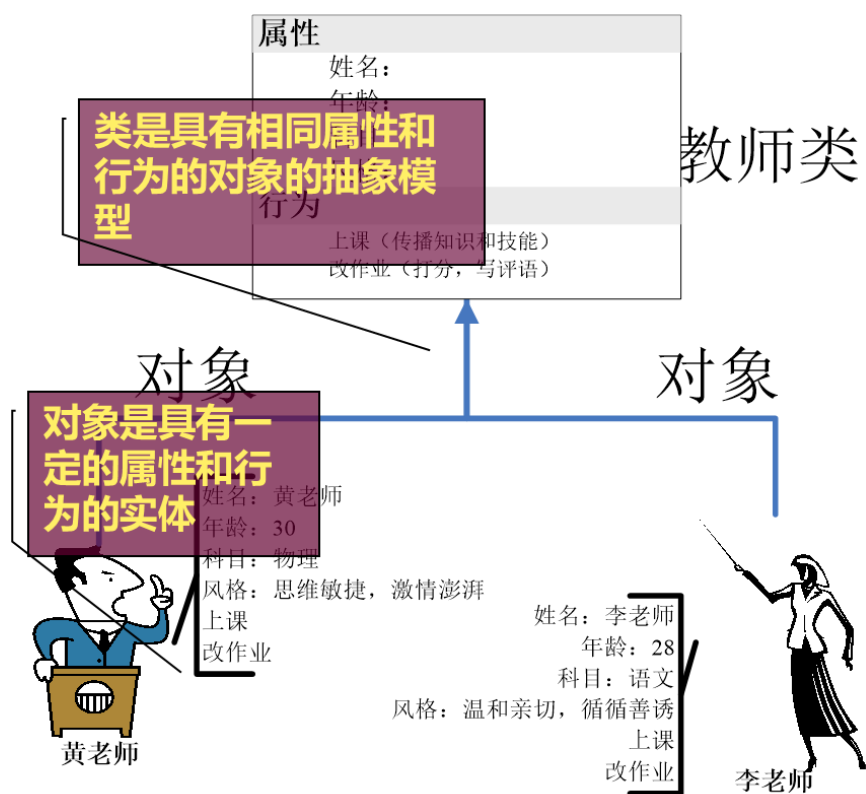
# 学习任务一 技能节报名小程序

## 子任务 1.1 回顾设计类

### 1、关于 Java 基础的回顾（微课学习内容）

#### 1.1 设计类

我们都知道，Java 是面向对象程序设计语言，他需要分析应用领域的类和对象，构建类图，所以类是 Java 程序设计的核心元素。那么如何理解类呢？如下图所示，我们人类描述世界，也喜欢用归纳的方法，例如如果让你归纳老师的特征，一般人都会回忆某某老师，他们都有什么特点，他们一般都做什么，所以我们会把老师都具有的特点归纳为老师的属性，例如：他们有姓名、有教的科目，有年龄等属性；我们把所有都有的行为归纳为老师类的方法，例如老师都会教书，会改作业。这样我们就定义了一个老师类。

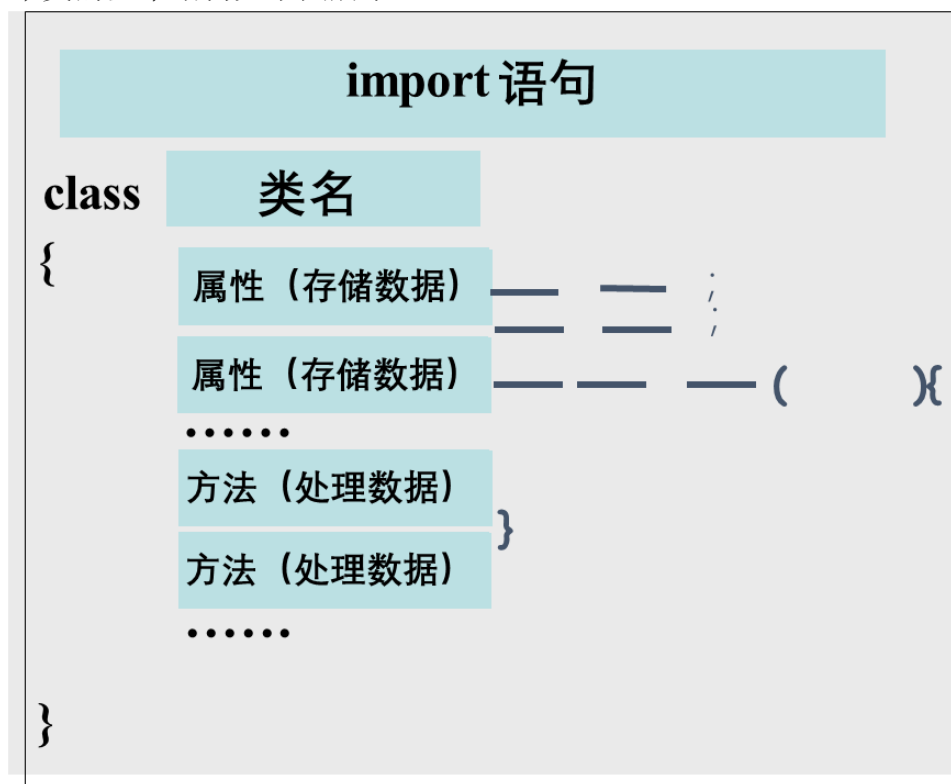


老师类是一个抽象的“模型”，而黄老师，李老师是老师类的具体对象，当然显示世界中同一类型对象共有的属性和行为很多，哪些属于我们要归纳的范围，这就取决于我们要解决的问题，这就是我们说的需求分析，从需求出发，我们才能按需求提炼类和对象。

## 1.1 定义类

如果知道了我们需要哪些类，知道了这些类应该具备哪些属性和方法，我们就可以在程序世界里实现他们。

一个类的基本结构如下图所示：



## 1.2 定义属性

属性一般是变量，变量定义的语法如下：

语法格式：

数据类型 变量名；

或者是

数据类型 变量名=初始值；

例如：

```
int age=10;
```

```
char c='x';  
float rate=0.3f;  
double price=23.3;  
boolean flag=true;
```

### 1.3 定义方法

类的行为使用方法实现的，结构如下图所示：



类中的方法有几种特殊的形式

#### 1) 构造方法和方法的重载

```
Class Boy{  
    int age;  
    public Boy(){age=0;}  
    public Boy(int age){this.age=age;}  
}
```

构造方法和类名相同，没有返回类型，他是创建新对象的时候会执行的方法，一般都有重载的现象，也就是多个方法，同名但是参数不同。就如同上图中的 Boy 类有两个构造器，一个不带参数，一个带一个参数，用于满足不同的

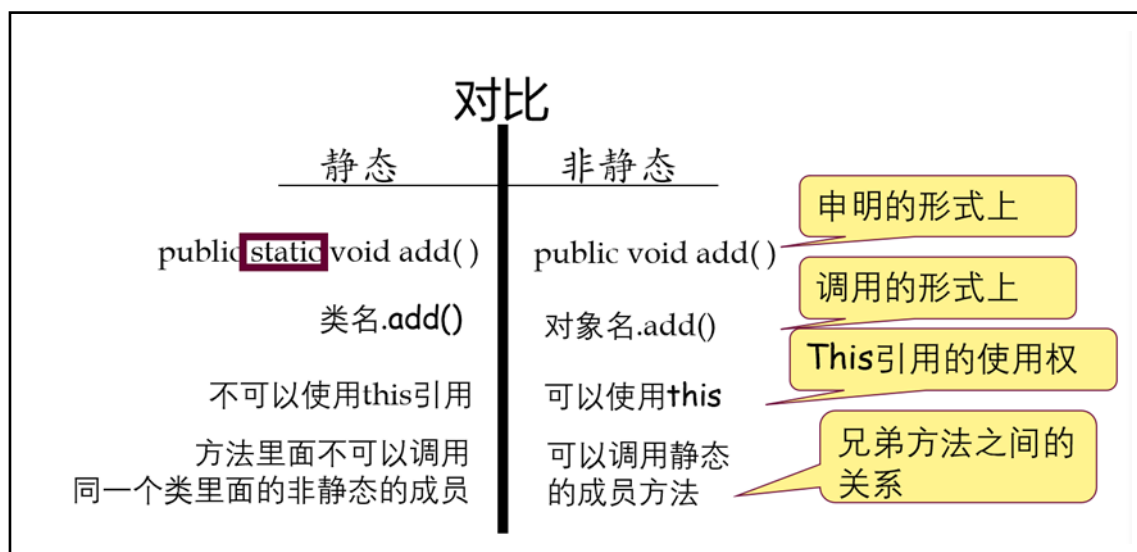
对象构建需求，例如创建普通的男孩对象，年龄默认是 0，但是如果要创建哪吒，就要给一个参数，默认年龄就不用 0 了。

## 2) Get/set 方法

```
private int age;  
public void setAge(int age){this.age=age;}  
public int getAge(){return this.age;}
```

基于类的封装性，我们在设计属性的时候，一般把类的属性设置为私有的，对外就像是一个黑匣子，无法访问，那么外界如果要访问数据，怎么办，我们为类设计对应属性的 get 和 set 方法，满足属性的访问需求。

## 3) 静态方法



## 4) 方法的重写

```

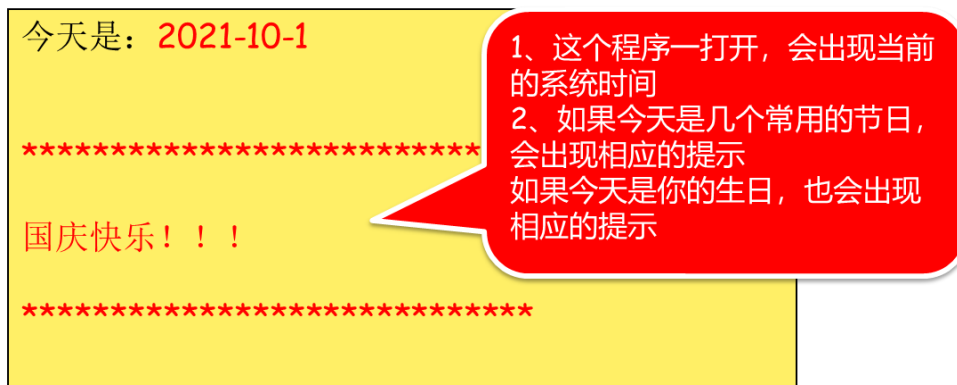
abstract class Employee {
    int basic = 2000;
    abstract void salary();//抽象方法
}
class Manager extends Employee{
    void salary() {
        System.out.println("薪资等于 "+basic*5);
    }
}
class Worker extends Employee {
    void salary() {
        System.out.println("薪资等于 "+basic*2);
    }
}

```

在类的继承中，经常会发生方法的重写，这也是实现多态的重要手段。

## 2、技能训练

现在我们利用一个小的任务，将先行课程学过的内容进行检测。具体任务需求如下图所示：



- 1、要求：两个人一个小组，在[www.processon.com](http://www.processon.com)绘制类图
- 2、提示：这个小任务，需要几个类，每个类里有什么属性和方法