

# 软件建模分析与设计

总复习（三）

## UML大题包含

1. 用例图： 会绘制用例图
2. 类图： 代码->类图 类图->代码
3. 构件图、布署图： 简单
4. 次序图和协作图： 协作转次序 或 次序转协作
5. 状态图和活动图： 状态图 或 活动图

**注意：** 100%不会考这个PPT上原题

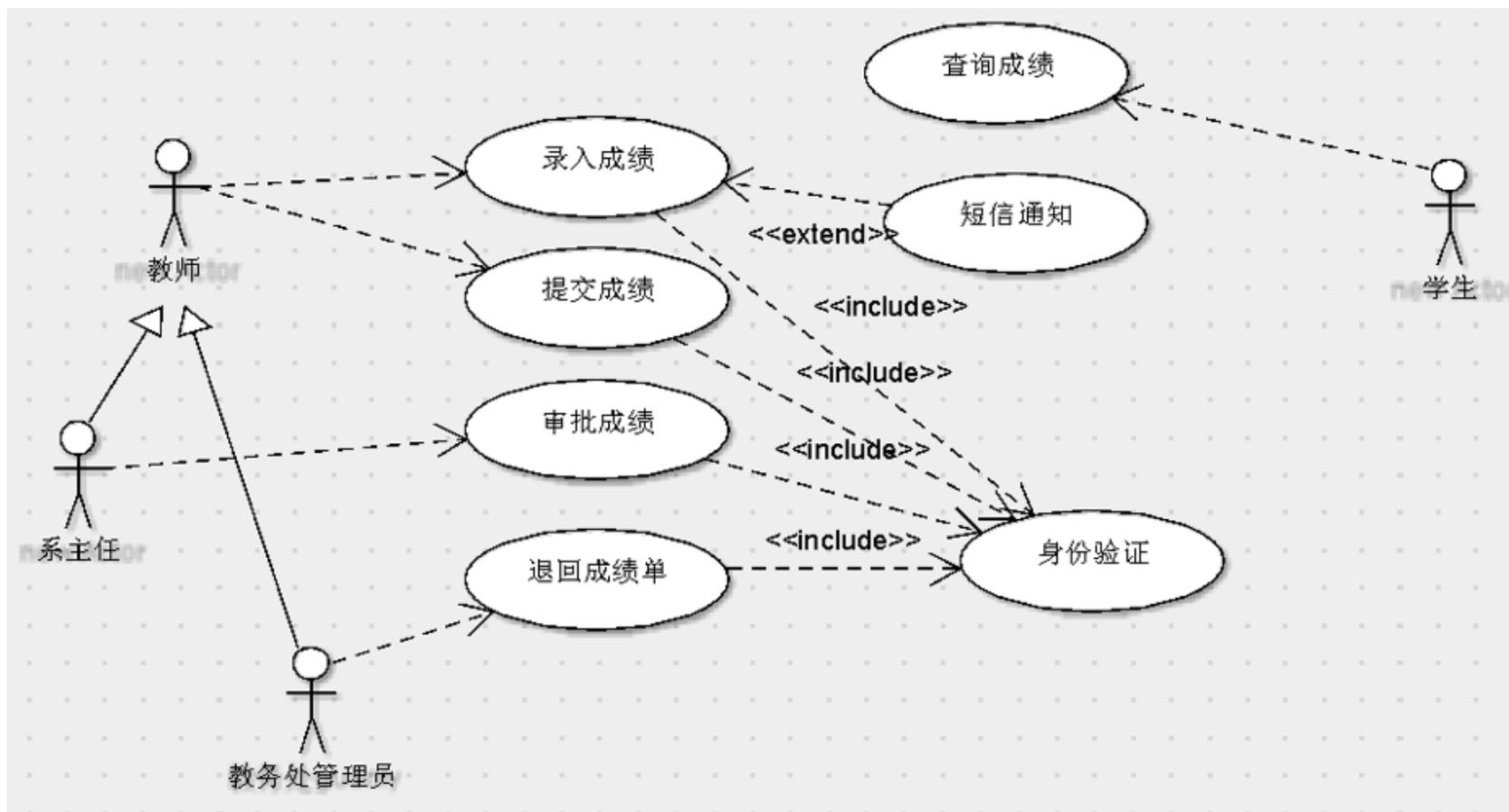
为预防剽窃， 题目内容较练习题多1-2个对象

- 某高校需要制作一个学生成绩管理系统，教师分为：普通教师、系主任、教务处管理员。全部教师均能够录入成绩和提交成绩，系主任能够同意成绩单，教务处管理员能够退回成绩单。全部这些教师功效均需要身份验证。提交成绩单之后如不及格那么可能会进行短信通知。学生能够查阅自己成绩。

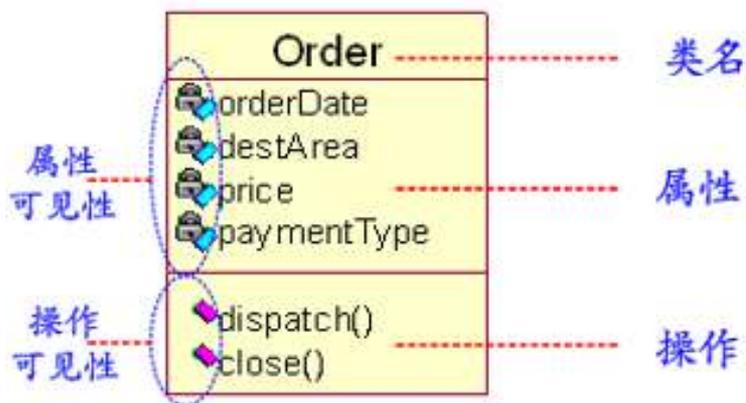
### 问，系统中角色和用例有那些？

角色：普通教师、系主任、教务处管理员、学生

用例：录入成绩、提交成绩、同意成绩单、退回成绩单、身份验证、查阅、短信通知



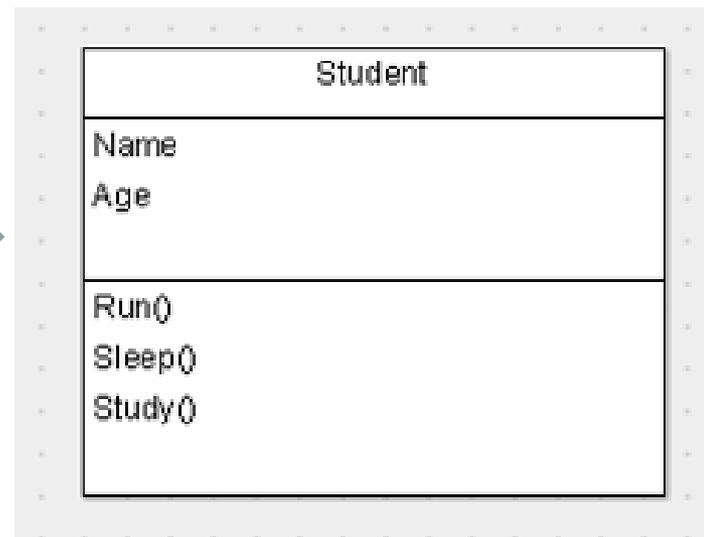
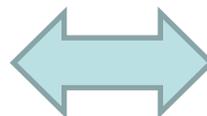
- 类图包含，类名，属性名和方法名
- 对于访问级别，为简单起见仅需要了解public



例子：将代码转换为类图

Class Student

```
{  
    public String Name;  
    public int Age;  
    public void Run(){.....}  
    public void Sleep() {.....}  
    public void Study() {.....}  
}
```



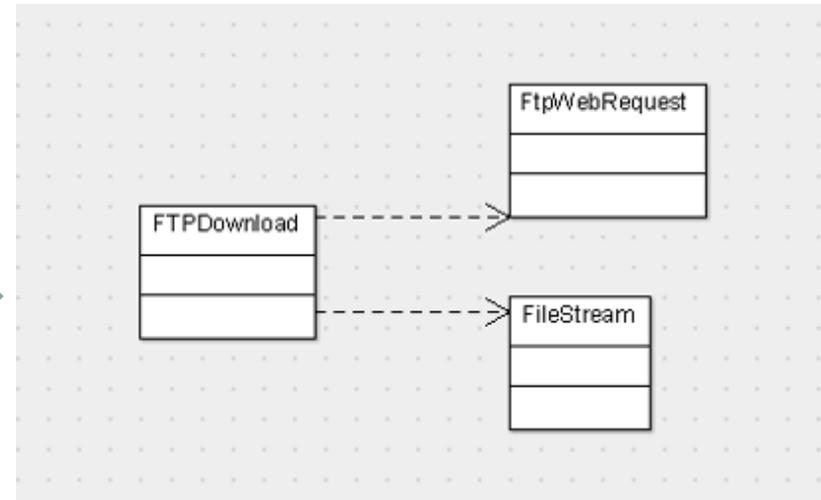
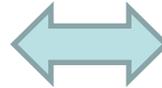
依赖关系： 一个类引用另外一些类

- 依赖关系UML符号表示是带箭头虚线



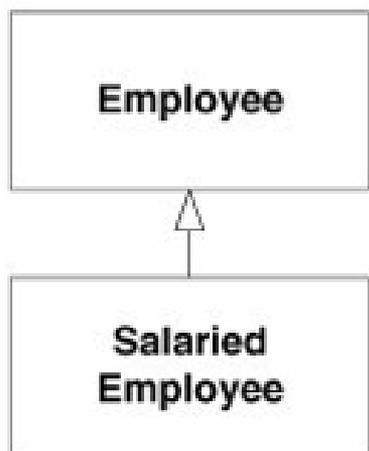
Class FTPDownload

```
{  
    private FtpWebRequest fwr;  
    private FileStream stream;  
}
```



类继承关系：

采取一个空心三角箭头指向父类

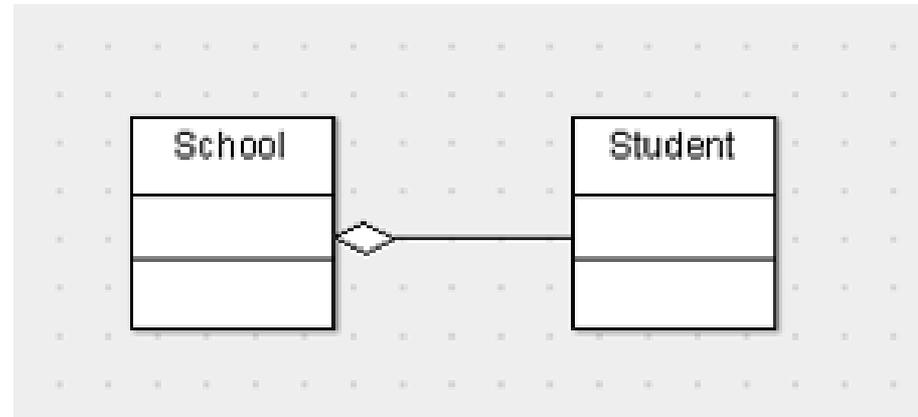


```
public class Employee
{
    ...
}

public class SalariedEmployee : Employee
{
    ...
}
```

关联关系-组合或者聚合关系

采取菱形箭头（空心或者实心，考试时不给予区分）

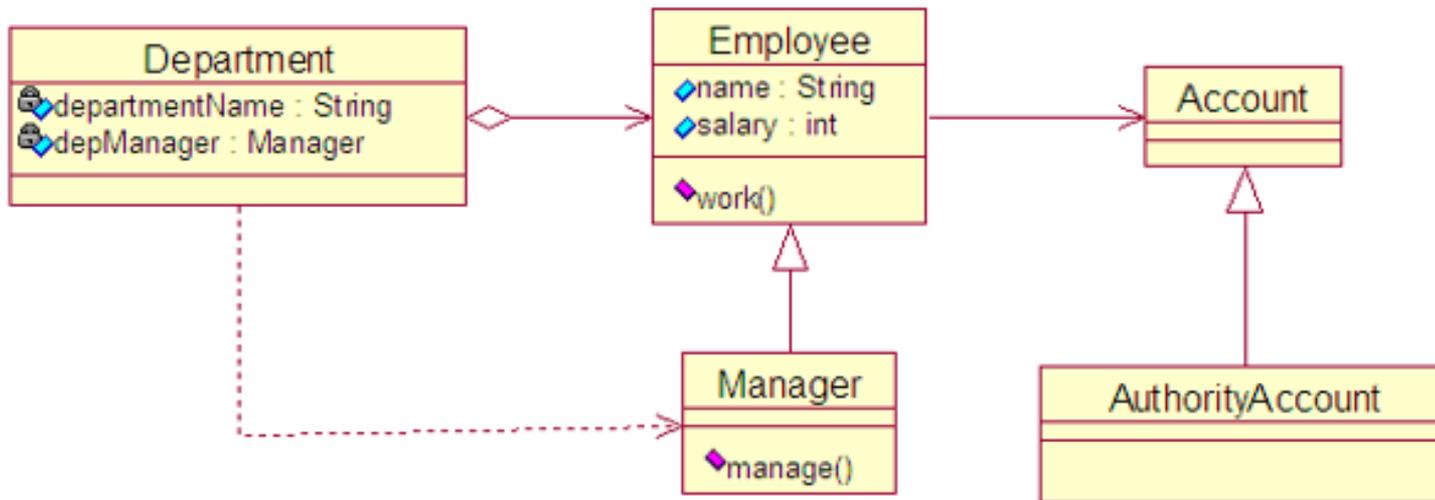


```
Class Student  
{  
}
```

```
Class School  
{  
    public Array<Student>  
    stdlist;  
}
```

2024/12/29

类图练习1: 阅读以下类图, 并写出对应程序代码框架:



- class Account
- {
- }
  
- class AuthorityAccount extends Account
- {
- }
  
- class Employee
- {
- Account account;
- public String name;
- public int salary;
- public void work(){}  
2024/12/29
- }

类图练习2: 阅读以下代码, 绘制类图

Class Vehicle

```
{  
    public int wheelNumber;  
    public void Run();  
}
```

Class Motorcycle extends vehicle

```
{  
    private int runSpeed();  
}
```

Class Car extends Vehicle

```
{  
    private Radar radar;  
    private ArrayList<CarItem> carItems;  
}
```

Class Radar

```
{  
    protect int radarNumber;  
}
```

```
Class CarItem
```

```
{  
}
```

```
Class 车门 extends CarItem
```

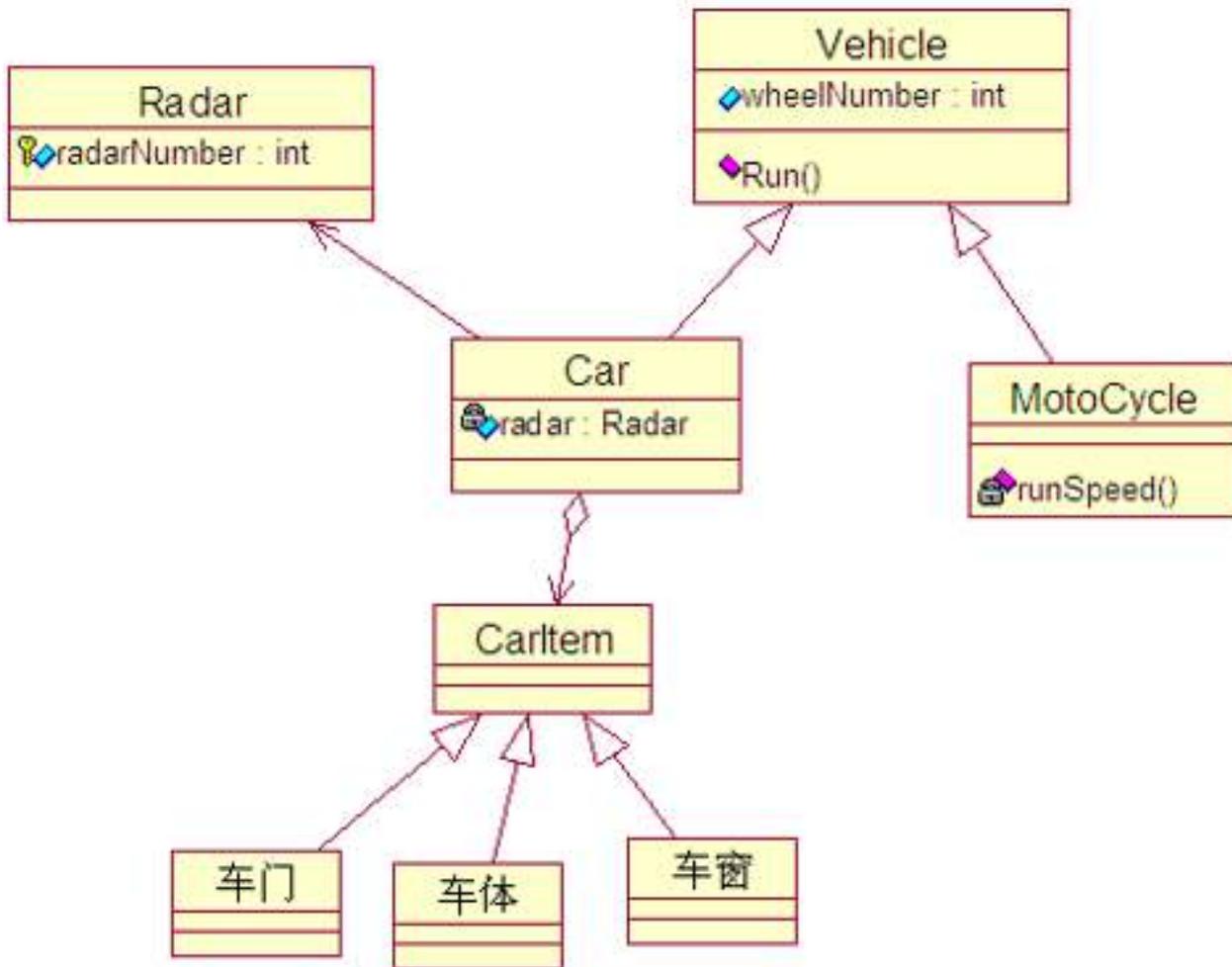
```
{  
}
```

```
Class 车体 extends CarItem
```

```
{  
}
```

```
Class 车窗 extends CarItem
```

```
{  
}
```



- 有一些构件图标表示形式和标准构件图形表示形式相同，它们包含ActiveX、Applet、Application、DLL、EXE以及自定义结构型构件，它们表示形式是在构件上添加相关结构型，是一个结构型为Applet构件。



以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/087033055034006154>