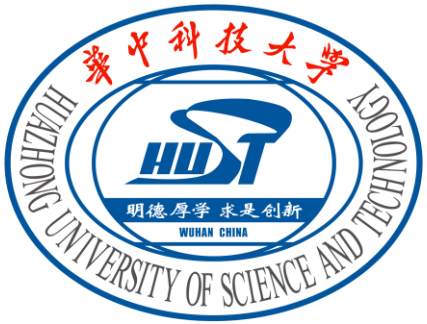
基于Java的面向对象程序设计

陈维亚

weiya_chen@hust.edu.cn

华中科技大学软件学院

第6讲：抽象类和方法



目录

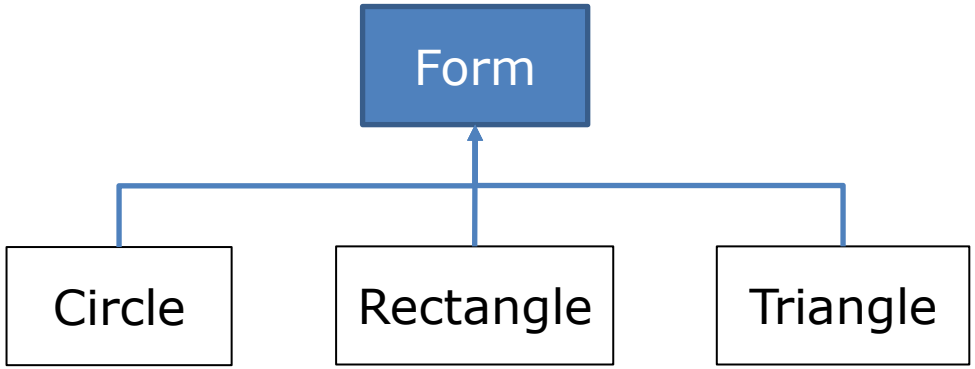
1. 概念

❑ 抽象类

▪ 在面向对象的概念中，所有的对象都是通过类来描绘的，但是反过来，并不是所有的类都是用来描绘对象的，如果一个类中没有包含足够的信息来描绘一个具体的对象，这样的类就是抽象类。

具体类：可以被实例化的类

抽象类：不能被实例化的类



▪ 抽象类通常只作为继承层次中的超类使用

▪ 定义抽象类的**基本目的**是提供合适的超类，使其他类可以继承它，以实现共享。

1. 概念

❑ 抽象方法

- 如果你想设计这样一个类，该类包含一个特别的成员方法，该方法的具体实现由它的子类确定，那么你可以在父类中声明该方法为抽象方法。
- 抽象方法只包含一个方法名，而没有方法体。
- 抽象方法没有定义，方法名后面直接跟一个分号，而不是花括号。

```
public abstract class Form{
   …
   public abstract void draw();
}
```

1. 抽象类

❑ 归纳

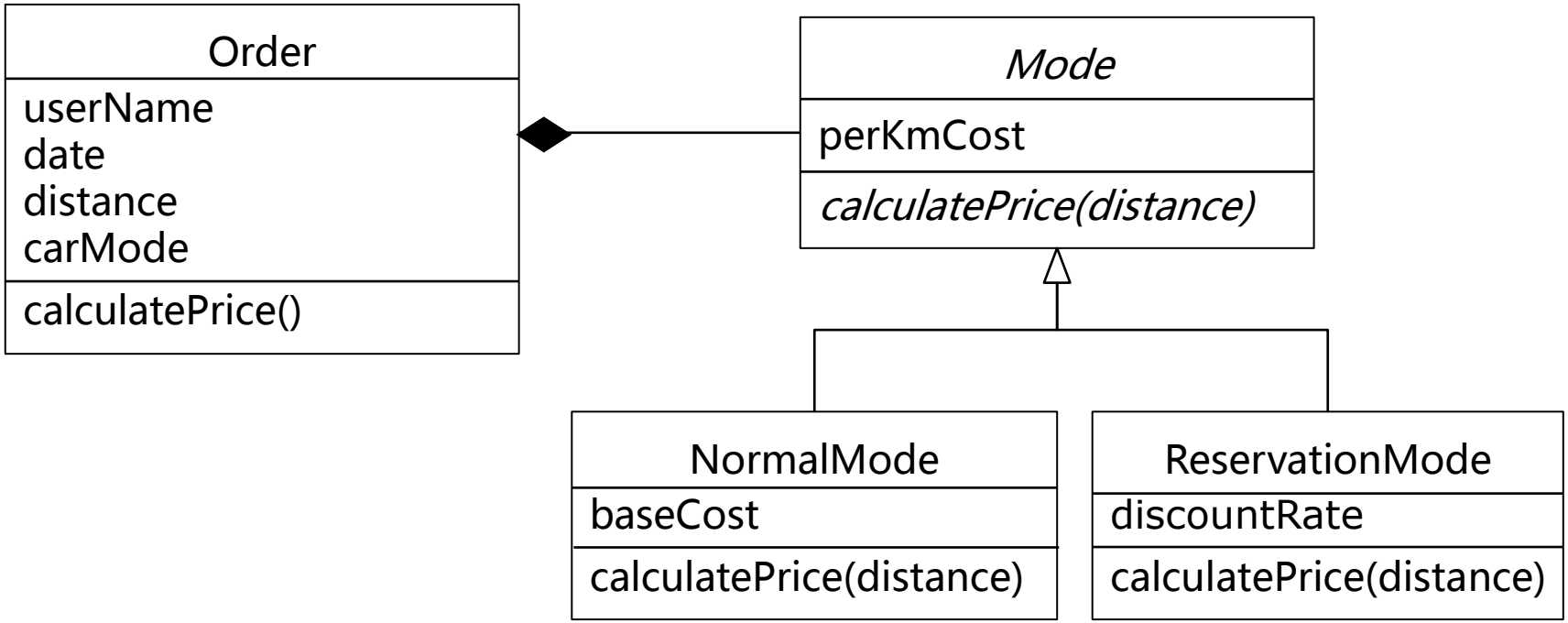
```
public abstract class Form{
  private int posX;
  private int posY;

  public void getInfo(){
    System.out.println("This is a 2D form.");
  }

  public abstract void draw();
}
```

1）抽象类中不一定包含抽象方法，但是有抽象方法的类必定是抽象类。

2）抽象类的子类必须给出抽象类中的抽象方法的具体实现，除非该子类也是抽象类。

2. 应用举例

【练习】迪迪打车共支持2种打车模式-**快车**和**顺风车**，分别有不同的计费规则，请帮助他们实现不同的订单计费规则。



3. 章节小结

抽象类（abstract class）和抽象方法 （abstract method）

- 定义
- 应用举例

下节预告

静态方法与变量