



类的封装

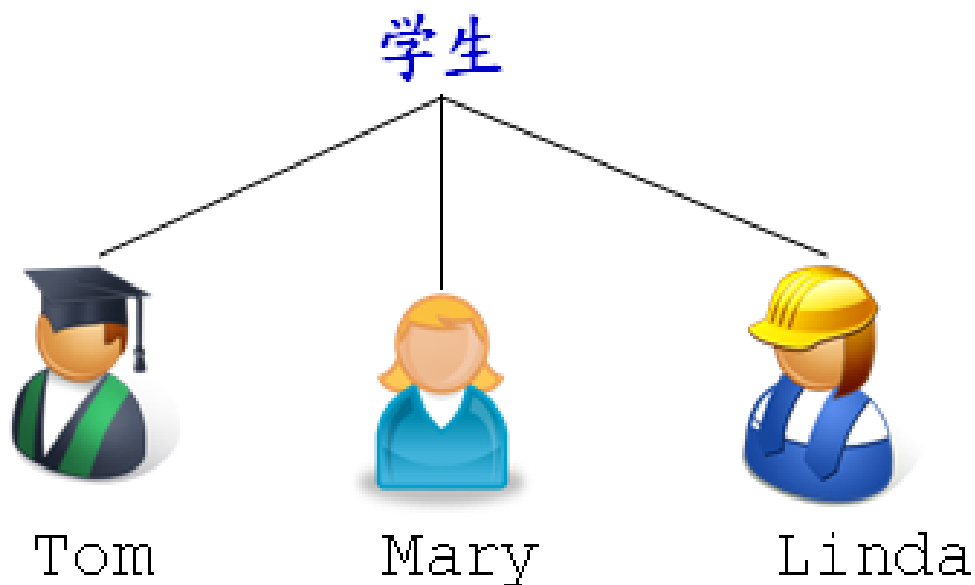
李玮玮

讲授思路

- 封装的概念
- 封装的好处
- 类与封装
- 访问修饰符

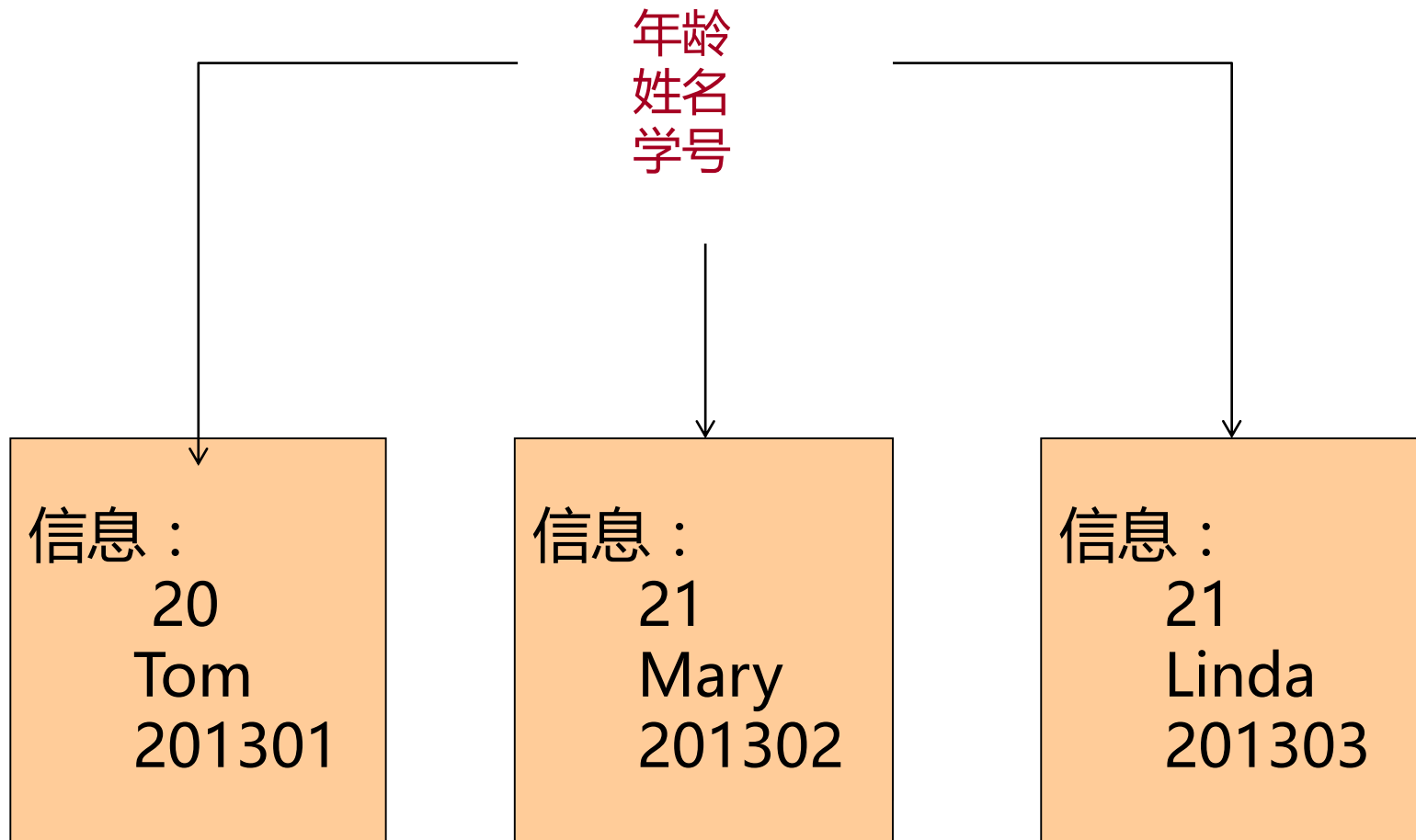
封装的引入

- 举例：学生成绩管理系统中，对于学生类，如何在计算机中表示学生的信息？



封装的引入

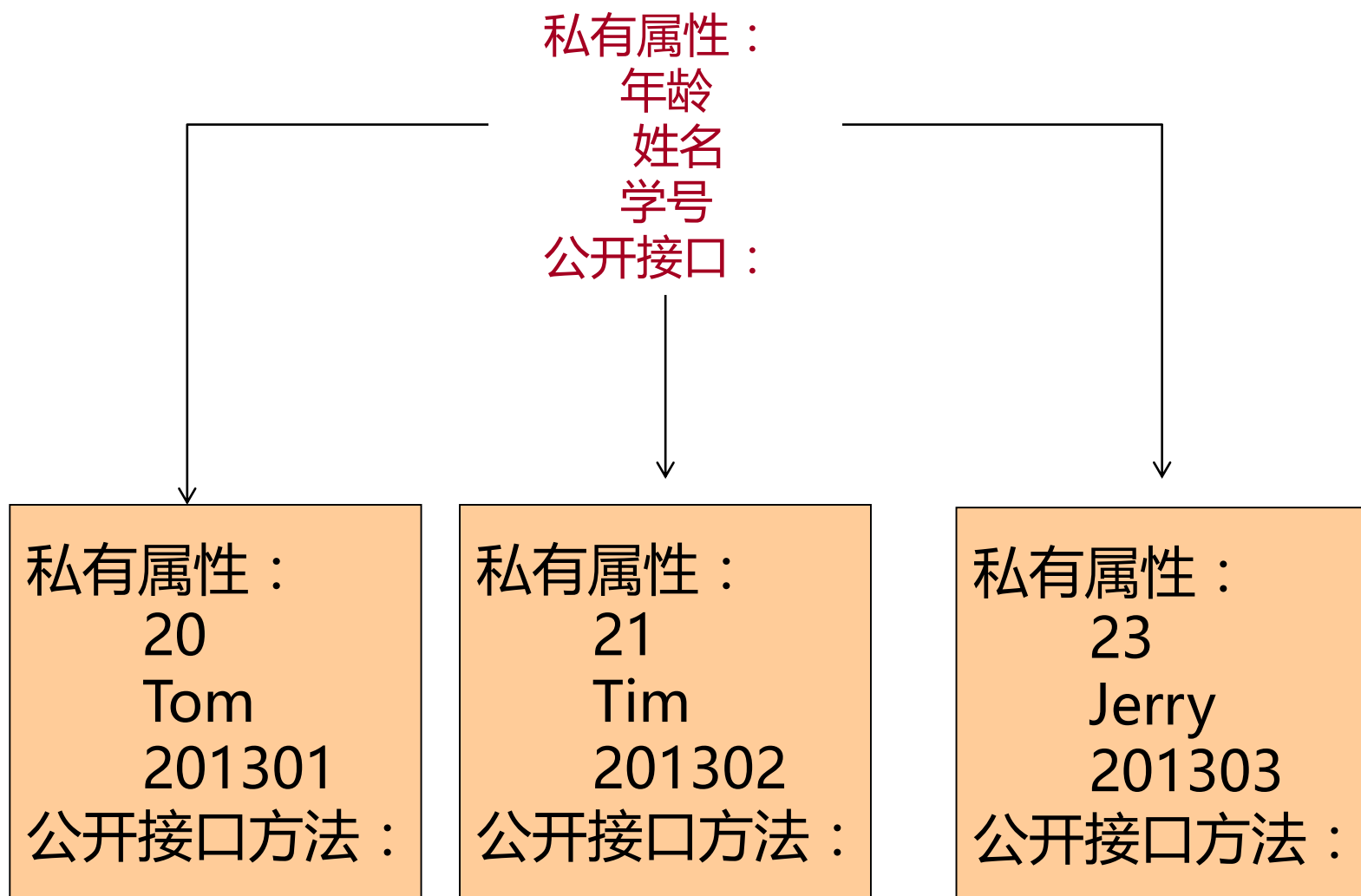
- 信息：



封装 ≈ “包装” + “隐藏”

- 封装 (Encapsulation) 摘自 : [zh.wikipedia.org/wiki/封装_\(物件导向程式设计\)](http://zh.wikipedia.org/wiki/封装_(物件导向程式设计))
 - 一种将抽象性函数接口的实现细节部分包装、隐藏起来的方法。
 - 也是一种防止外界调用端，去访问对象内部实现细节的手段，这个手段是由编程语言本身来提供的。
- 封装，摘自：
http://baike.baidu.com/view/1520586.htm#1_2
 - 一是把对象的全部属性和行为结合在一起，形成一个不可分割的独立单位。对象的属性值（除了公有的属性值）只能由这个对象的行为来读取和修改；
 - 二是尽可能隐蔽对象的内部细节，对外形成一道屏障，与外部的联系只能通过外部接口实现。

封装的引入



封装的好处

- 封装机制将对象的使用者与设计者分开，使用者不必知道对象行为实现的细节
 - 或者说“接口”与“实现”分开
- Simplicity and clarity
- Low complexity
- Better understanding

现实世界中的封装

- 封装机制将对象的使用者与设计者分开，使用者不必知道对象行为实现的细节
 - 或者说“接口”与“实现”分开
- 比如：
 - “人”隐藏了什么？
 - 内脏
 - 你如何与他交互？(接口)
 - 手、脚、五官

类与封装

- 通过类来实现封装，通过访问修饰符来实现信息隐藏
- 类声明的语法：
 - class 类名{
 - [private/protected/public] 成员的声明和定义;
 - }
 - 其中，private、public及protected被称为访问修饰符。

访问修饰符与封装

- 4种访问权限修饰符(3个关键字)
 - public
 - private
 - Protected
 - 缺省
- 不写访问权限(默认权限，或称包权限)

访问修饰符与封装

- public公共成员
 - 成员方法或变量声明为public，称为公共成员
 - 可以被所有的类访问的成员（前提所属类本身是可见的）
- private私有成员
 - 不能被其所在类以外的任何类访问
 - 声明为private的成员，称为私有成员
- protected保护成员和默认成员
 - 声明为protected的成员，称为保护成员
 - 可以被同一包内的类访问和被子类继承
- 没有任何修饰符的成员，称为默认成员
 - 只能被同一包内的类访问

访问修饰符与封装

```
public class Student{  
    public String name;  
    public int id;  
    private int age;  
    public void print(){  
        System.out.println("姓名="+name  
            +"学号="+id + "年龄=" +age);  
    }  
}
```

封装



隐藏



类成员的访问

- 对象的private成员，被隐藏，不可被对象使用者访问;
- 对象的public成员，被公开，可被对象使用者访问;
- 关于protected成员，将在“继承”一讲中阐述。

```
public class Student{  
    public static void main(){  
        Student tom= new Student();//xiaoming对象  
        tom.name= "Tom" ;    //name是public成员  
        //tome.age = 22;    // 编译出错，age是private成员  
    }  
}
```

类成员的定义

- 类成员
 - 属性，如：name
 - 方法，如：void print()
- 类的成员
 - 抽象而来：忽略不必要的，取我们所需要的成员
 - 同类对象拥有相同的成员
- 成员的封装
 - 都可以选择private、protected、public或默认

总结

- 封装的概念
- 封装的好处
- 类与封装
- 访问修饰符

课后阅读

- private、protected、public修饰符的访问权限。



Thank You