## Day07

### 7-1 面向对象概念

* 面向对象：虚拟世界”模拟现实”生活，必须保证模拟一致
* 面向过程：
* 类：对同一类事物的**抽象的**描述，也就是：**不是具体的**
  + 如：电脑类、水杯类、键盘类、人类………
* 对象：万物皆对象，也就是说：**具体的实例**
  + 我的电脑、张三的水杯………
* 类与对象之间关系
  + 抽象与具体的关系
* **描述对象的特征，称为属性**，如：颜色、价格、尺寸…….
* **对象所做的事情，称为方法或行为**

张三对象

属性：名字、性别、年龄、身高、体重、住址……

方法：学习、说话、吃饭……..

李四对象

属性：姓名、年龄、住址、性别………

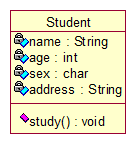
方法：睡觉、学习、发红包、游泳……

**将多个对象找到相同的属性和方法组合在一起，形成类**

学生类

属性Attributes：姓名、年龄、性别、住址

方法Operations：学习study



### 7-2 编写类的操作步骤

第一步：编写类

语法格式：

[修饰符] class 类名{

}

第二步：编写属性也叫做成员变量

语法格式：

[**修饰符**] 数据类型 属性名称 [**=值**];

* 属性编写在类中
* []中的内容是可有可无的
* 属性名称也是标识符，建议采用驼峰命名法，见名知意
* 当属性或成员变量没有赋值时，则是有默认值的，数据类型不同则默认值不同：
  + 整数类型默认值为0
  + 浮点类型默认值为0.0
  + 字符类型默认值为\u0000
  + 布尔类型默认值为false
  + 所有引用类型的默认值为null

第三步：编写方法

语法格式：

[修饰符] 返回类型 方法名称([参数列表]){

方法体

}

* 方法必须编写在类中
* 在面向对象中，方法的修饰符暂时编写为public

### 7-3 编写测试类

也就是包含main方法的类

### 7-4 创建对象，也叫做实例化

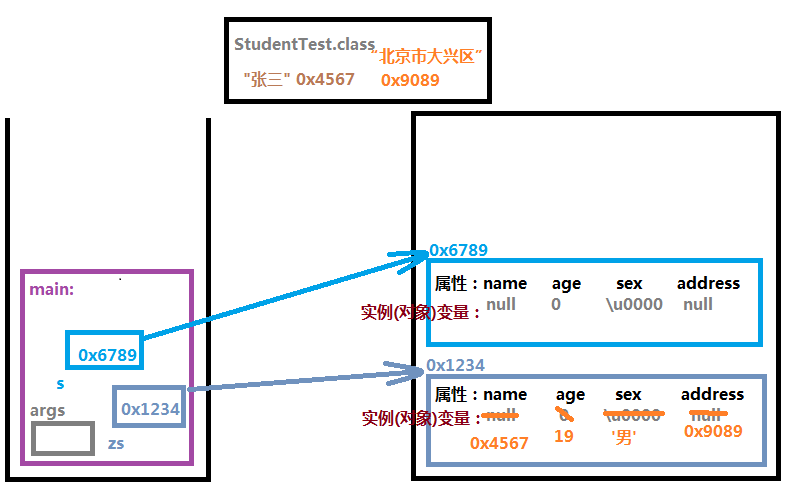
* 语法格式：

**类名 引用名称 = new 类名();**

**注意：一个可以创建N个对象**

### 7-5 访问对象中的属性和方法

* 访问对象中的属性：
  + 引用名称.属性名称
* 访问对象中的方法
  + 语法格式：引用名称.方法名称();



### 7-6 成员变量和局部变量区别

* 声明位置不同
  + 成员变量：在类中声明
  + 局部变量：在方法中声明，其中形参也属于局部变量
* 初始值不同
  + 成员变量：在没有赋值时，是有默认值的，数据类型不同，则默认值不同
  + 局部变量：没有默认值，也就是说必须先声明，然后再赋值，最后才可以使用
* 在同一个类中，成员变量的名字可以与局部变量的名字同名，但是局部变量名字优先，也就是就近优先，如果非要访问成员变量则必须添加this.
  + this代表当前这个对象，也就说当前谁调用这个方法则这个对象就是谁

### 7-7 对对象中的属性赋值

* 第一种方法：使用引用名称.属性名称 = 值;
  + 如：d.strain = "拉布拉多犬";
* 第二种方法：使用构造方法完成

### 7-8 构造方法以及重载

* 作用：完成对对象中的属性赋值
* 构造方法
  + 构造方法是一个特殊的方法，构造方法名字必须与类名一致，构造方法必须没有返回类型，也就是不编写返回类型
  + 语法格式：

**public 方法名称([参数列表]){**

**}**

* 构造方法如何执行的？
  + 当创建对象时**自动执行**相匹配的构造方法
* 构造方法分类：
  + 隐式构造方法
    - 当在一个类中，没有手动编写构造方法，则系统会提供一个默认的无参的构造方法
  + 显式构造方法
    - 当在一个类中，手动编写构造方法，则系统不会提供默认的无参的构造方法
    - 建议：当手动编写构造方法时，先编写无参构造方法，然后再编写需要的构造方法
* 构造方法重载：
  + 在同一个类中，构造方法的名字必须相同，参数列表不同(个数不同、类型不同、顺序不同)