构造方法：（构造器） 🡪 支持方法的重载

构造方法必须和类名相同

构造方法没有返回值，但也不能写void

构造方法的作用，一般用于初始化成员变量 🡪 创建对象使用

在一个类中，若果没有写任何的构造方法，编辑器会默认的追加一个无参的构造方法

若果有其它的构造方法，则编辑器不再追加无参构造方法，

This：

构造器的调用：可以通过this关键字，调用当前类中已经存在的对应的构造方法

一般即使我们不使用无参的构造方法时，也要将无参构造方法写出，防止编译报错

内存管理：

1. 对象生存环境，和空间 编译好的java程序运行在jvm虚拟机中。
2. Jvm分为“堆”，“栈”，“方法区”三个区域。
3. 方法区分为静态区和非静态区
4. 基本数据类型存储在栈中。引用数据类型的地址存储在栈中，然后在堆中开辟一个空间留着存储数据，然后栈中存储的地址，指向堆中的对象
5. 成员变量属于对象。所以存储在堆内存中

生命周期：

1.

变量：

根据有效域：局部变量，成员变量。

1. 局部变量：
   1. 定义在方法区；
   2. 没有默认值，使用之前必须初始化；
   3. 方法调用时，存储在栈帧中，方法调用完，栈帧清除相应的数据。
2. 成员变量：
   1. 成员变量定义在类中，方法外；
   2. 有默认初始值，可以不用初始化，
   3. 实例化对象后，存储在堆中，对象使用完后，被gc（）（垃圾回收器—线程）回收，默认调用（写出来理论上会快一点回收）。
   4. 根据类型：基本类型，引用类型变量。

面向对象的三大特性：封装，继承，多态。

一个文件中只能有一个public修饰的class，且public修饰的class必须是和文件名相同

1. 封装：方法的封装，类的封装（组件）。
   1. JavaBean概念：
      1. JavaBean是一种Java语言写成的可重用组件，为了写成JavaBean，类必须是具体的和公开的。
      2. JavaBean其实就是一个按照特定规范定义的类（一种封装方式），里面需要有无参的构造方法，成员变量私有，方法公开，类具体，公开。
   2. 封装的好处：
      1. 对外部提供可供调用的稳定的功能
      2. 降低代码的出错率，提高代码的复用性和可维护性
      3. 当封装的方法内部出现细节改变时，只要保证对外的功能不变，其他模块不会因此受到影响
2. 继承：让类与类之间产生关系，子父类关系（强耦合） --- 继承者是子类，被继承是父类（父类大于子类）
   1. 关键字 ： extends --- 让类与类之间产生子父类的关系
   2. 将公共属性或者方法抽出来放到一起形成一个新的事物 --- 父类
   3. 注意：子类可以使用继承自父类的成员变量和方法 ---- 非私有
   4. 优点：提高代码的复用性，可维护性
      1. 让类与类之间产生关系 ---- 是多态的前提
3. 开发遵循原则：
   1. 高内聚，低耦合
      1. 高内聚：一个类自己完成某件事的能力
      2. 低耦合：类与类之间的关系
   2. 弊端：类与类之间产生强耦合，不利于开发中的灵活使用
4. 继承的特点：
   1. 只能继承父类中非私有的成员变量和方法
   2. Java中继承，只支持单继承，不支持多继承
   3. 支持多层继承（继承体系） --- 继承具有传递性
5. 如果想看整个继承体系中共有属性和方法，用最顶层的类创建对象
6. 如果想看整个继承体系中所有属性和方法（非私有），用最底层的类创建对象
7. 注意事项：子类只能继承父类中非私有的成员变量和方法
   1. 子类不能继承父类的构造方法，但是可以通过super关键字调用（访问）
   2. 在子类的构造方法中 默认调用父类中无参的构造方法，有且只能子类构造方法的第一句， 🡪 super(); 调用父类中无参构造方法
   3. 也可以调用对应参数的构造方法，初始化对应成员变量super(name , sex);
   4. 谨慎使用继承 – 不要因为部分功能而强制使用继承
8. This和super的区别
   1. This：指代当前对象的引用
   2. Super：指代父类的引用
      1. 区别：
         1. 调用成员变量：
            1. This.成员变量：调用本类中的成员变量，也可以调用父类中的成员变量（非私有）
            2. Super.成员变量：只能调用父类中成员变量（非私有）
         2. 调用构造方法:
            1. This(…):调用本类中构造方法
            2. Super(…):调用父类中的构造方法
            3. 注意：在同一个构造方法中，super和this不能同时调用—防止出现重复调用父类中构造方法
         3. 调用成员方法
            1. This.方法名():调用本类或者父类中的成员方法
            2. super.方法名():只能调用父类中的成员方法
   3. 子类中每个构造方法第一句都是默认调用父类中的无参构造方法 – super();
      1. 为什么？ 因为子类会继承父类中的数据，可能还会使用父类中数据，所以子类在初始化之前，必须先完成父类数据的初始化
      2. Object是所有父类的顶级父类