1. 对象
2. 对象的进化
   1. 基本数据类型： 适用于数据少无数据管理时代
   2. 数组： 数据多了，将同类数据分类放一起 弱管理
   3. 结构体：数据多了，类型复杂，将不同类型数据放一起 强管理
   4. 对象：数据多了，类型复杂，讲不通类型数据放一起 强管理
3. 对象和对象
   1. 对象是具体的，类是模板，通过类可以new出对象
4. 类
   1. 格式：

public class（类名）{ 规范：首字母大写， pubic 修饰的类需要跟我们的文件名相同

//属性

(一般定义成)private 数据类型 属性名 //建议增加相应的getter setter方法

//构造器(构造方法)

//方法

}

1. 方法
   1. 构造方法
      1. 方法名必须类名保持一致
      2. 无返回类型
      3. 通过new调用
      4. 如果不显式定义构造方法，系统默认添加一个无参构造方法
      5. 如果显式定义了构造方法，系统不会默认添加构造方法
      6. 构造方法的第一句总是super,即调用父类的构造方法
   2. 普通方法
   3. 方法的重载(普通方法和构造方法)
      1. 实现重载的条件：两同(同一个类，相同方法名)，三不同(参数列表不同、类型、个数、顺序)
      2. 返回值不同不能构成重载
      3. 形参名称不同不能构成重载
   4. 方法的重写
      1. 重写的必要条件：

必须有继承

=：方法名相同

>=: 子类权限修饰符大于等于父类的

<=: 子类的返回值类型小于等于父类的，子类异常对象不能超过父类的

1. 重要的关键字
   1. this关键字
      1. 构造方法中，this代表正要初始化的对象
      2. 构造方法中，可以使用this调用其他构造方法
      3. 普通方法中，this代表调用本方法的对象
   2. super关键字
      1. 调用直接父类的构造器
   3. static
      1. 修饰变量 –> 静态变量
      2. 修饰方法 –> 静态方法
      3. 修改{…} –> 静态代码块
   4. Final
      1. 修饰变量 –> 常量
      2. 修饰方法 –> 不能被重写
      3. 修饰类 –> 不能被继承
   5. import
      1. 引入外部的类
      2. import static 导入类的静态属性
   6. package
      1. 写在类的第一句
      2. 命名规则：一般域名倒置
2. 面向对象三大特性
3. 封装
   1. 如何实现：通过private, default, protected, public实现属性、方法、类的封装
   2. 作用：高内聚、低耦合
   3. 作用域

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 修饰符 | 同一个类 | 同一个包 | 不同包的子类 | 不同包的非子类 |
| private | √ |  |  |  |
| default | √ | √ |  |  |
| protected | √ | √ | √ |  |
| public | √ | √ | √ | √ |

1. 继承 extends
   1. 好处：代码复用，实现对现实世界更加准确的建模
   2. Object类是所有类的基类
   3. Object的方法：hashCode(), equals(), notify()， notifyAll(), wait()
2. 多态
   1. 三个必要条件： 继承，方法重写，父类引用指向子类对象
   2. 实现方式：静态绑定， 动态绑定(反射)