10 包-修饰符-内部类

- 包
  - 概念
    - 包就是文件夹，用于区分相同的类名
  - 声明格式
    - package 包名1.包名2…; // package:是个关键字
  - 带包的类的编译和运行
    - javac －d . 文件名.java
  - 导包
    - import 报名1.包名2…;
    - 如果我在一个类中需要用到不同包下的两个类，并且这两类名同名
      - 方式1：通过关键字import导入一个包，另一个用类全名
      - 方式2：两个都用类全名
  - package ,import ,class 的出现顺序
    - package-->imoprt-->class
    - 注意
      - package 最后面有;
      - class 类名后面是大括号
      - import,package是关键字，不是类名
- 修饰符
  - 权限修饰符
    - private，默认的，protected，public
      - private : 本类
      - 默认的 : 本类 // 同一包下(子类和无关类)
      - protected : 本类 //同一包下(子类和无关类) // 不同包下(子类)
      - public : 本类 //同一包下(子类和无关类) // 不同包下(子类) // 不同包下(无关类)
  - 状态修饰符
    - static，final
  - 抽象修饰符
    - abstract
  - 类
    - 权限修饰符 : 默认修饰符,public //内部类可以private,protected
    - 状态修饰符 : final // 内部类可以static
    - 抽象修饰符 : abstract
    - 最常用 : public
  - 成员变量
    - 权限修饰符 : private,默认,protected,public
    - 状态修饰符 : static,final
    - 最常用 : private
  - 构造方法
    - 权限修饰符 : private,默认,protected,public
    - 最常用 : public
  - 成员方法
    - 权限修饰符 : private,默认,protected,public
    - 转态修饰符 : static,final
    - 抽象修饰符 : abstract
    - 最常用 : public
  - 组合规则
    - 成员变量 : public static final
    - 成员方法 :
      - public static
      - public abstract
      - public final
- 内部类
  - 1、什么时候用
    - B类如果要直接使用A类中的 所有 成员的时候就把B类定义到A类里面
  - 2、内部类和继承的区别
    - 内部类可以直接访问外部类的私有成员
    - 但是子类要想访问外部类的私有成员的话只能通过get/set方法间接访问
  - 3、访问特点
    - 1、内部类可以直接访问外部类 所有 成员
    - 2、外部类想要访问内部类成员，必须内部类创建对象
  - 4、内部类分类
    - 1、成员内部类
      - A、其他类如何使用该内部类?
        - 外部类名.内部类名 变量名 = new 外部类构造方法.new 内部类构造方法;
        - private:一般就是为了安全
        - static:一般就是为了方便调用
      - B、注意问题
        - 如果一个内部类含有静态方法，那么该内部类必须被静态修饰
        - 如果一个类被定义为静态时，该类中可以没有静态方法
    - 2、局部内部类
      - 概念
        - 定义在方法中的类
      - 结论
        - 局部内部类访问局部变量，该局部变量必须被final修饰,因为被final 修饰就变成常量，在方法区中，他的生命周期延长了
        - ⛔⛔ 被final修饰的变量的生命周期存中小时呢？没有引用的时候以，为什么 被final修饰变量是常量，所以在方法区的常量池中，常量什么时候从内在外部类中定义一个方法，这个方法里面定义一个静态变量，测试可不可
  - 5、匿名内部类
    - 概念：
      - 没有名字的局部内部类
    - 前提：
      - 必须存在一个类，抽象类或者接口
    - 格式：
      - new 类名或接口名(){ //重写类或者接口中的方法 //也可自定义方法
    - 本质：
      - 其实就是继承了一个类或者实现了一个接口的匿名的子类对象
  - 6 .使用
    - 无论是匿名内部类还是匿名对象，一般使用方式有以下两点：
    - 1、只调用该类中的一个方法，这种情况一般该抽象类或者该接口只有一个抽象方法时使用
    - 2、作为实际参数传递(一般该匿名对象或匿名内部类中的方法不超过3个)
  - 7 . 命名规则
    - 成员内部类命名规则: 外部类$内部类.class
    - 局部内部类命名规则: 外部类$1第一级内部类$2第二级内部类$3第三级内部类.class，$后跟数字，数字与内部类对应的层级有关系
    - 匿名内部类命名规则: 外部类$1.class,1代表外部类下的第几个内部类
    - $跟 类名 表示是成员内部类,
    - $后面跟 数字 是匿名内部类，且数字代表个数,
    - $后面跟 数字$类名 是局部内部类，数字表示层级
  - 匿名内部类与匿名对象异同
    - 不同点
      - 匿名内部类是一个继承了类或者实现了接口的子类对象
      - 而匿名对象就是这个类的本类对象
    - 相同点
      - 都作为实际参数传递时使用，或者该对象的方法只调用 1次的时候使用,注意一般情况匿名对象或者匿名内部类对象里面最多有3个方法