

# 对象构造与初始化

2020年2月26日 星期三 下午9:55

对象都有构造方法, default 为空

## 调用本类或父类的构造方法

- 调用本类或父类的构造方法
  - this调用本类的其他构造方法。
  - super调用直接父类的构造方法
  - this或super要放在第一条语句,且只能有一条
- 如果没有this及super, 则编译器自动加上super(), 即调用直接父类不带参数的构造方法
- 因为必须令所有父类的构造方法都得到调用, 否则整个对象的构建就可能不正确。

## 一个问题

- Class A
- {
- A(int a){}
- }
- Class B extends A
- {
- B(String s){} //编译不能够通过。
- }

无参数, 而A中没有无参的构造方法

- 编译器会自动调用B(String s){ super(); } 出错。
- 解决方法:

- 在B的构造方法中, 加入super(3);
- 在A中加入一个不带参数的构造方法, A(){}
- 去掉A中全部的构造方法, 则编译器会自动加入一个不带参数的构造方法, 称为默认的构造方法。

## 创建对象时初始化

双括号对相应字段赋值

- p = new Person(){ age=18; name="李明"; };
- 这样可以针对没有相应构造函数, 但又要赋值
- 注意双括号

## 实例初始化与静态初始化

- 实例初始化 ( Instance Initializers )
- 在类中直接写
  - { 语句.... }
  - 实例初始化, 先于构造方法{}中的语句执行
- 静态初始化 ( Static Initializers )
  - static { 语句.... }
  - 静态初始化, 在第一次使用这个类时要执行,
  - 但其执行的具体时机是不确定的
    - 但是可以肯定的是: 总是先于实例的初始化