Exercise

```
public class ExtendExercise {
    public static void main(String[] args) {
        B b = new B();
    }
}
class A {
    A(){
        System.out.println("a");
    }
    A(String name){
        System.out.println("a name");
    }
}
class B extends A{
    B(){
        this( name: "abc");
        System.out.println("b");
    }
    B(String name){   super();
        System.out.println("b name");
    }
}
```

输出：1> a
2> b name
3> b

① 默认调用父类无参构造器
② this 与 super 二者不能同时在类.

P298 - P301 Java_super

基本介绍

super 代表父类的引用，用于访问父类的属性、方法和构造器.

注意细节

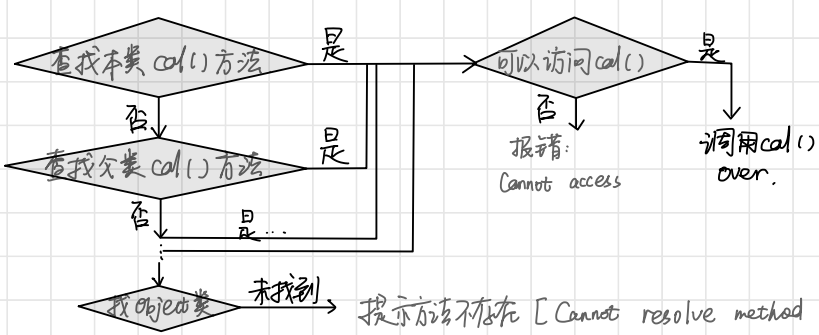
1) 访问父类的属性，但不能访问 private 修饰的属性；⇒ 同理方法.

2) 访问父类构造器：super(参数列表)；
只能放在构造器第一句，且只能出现一句.

3) 调用父类构造器的好处：
分工明确，父类属性由父类初始化，子类的属性由子类初始化

4) 当子类中有和父类中的成员(属性和方法)重名时，为了访问父类的成员，必须通过 super. 如果没有重名，使用 super、this、直接访问效果一样.

在子类对象调用某方法时，流程为：(以 cal() 为例)



提示方法不存在 [Cannot resolve method 'cal' in 'A']
方法 A类中.

5> super的访问不限于直接父类，如果爷爷类和本类中有同名的成员，也可以使用super去访问爷爷类的成员；如果多个基类(上级类)中都有同名的成员，使用super访问遵循就近原则

P302 - P305 Java_方法重写

定义

方法覆盖(重写)就是子类有一个方法，和父类的某个方法的名称、返回类型、参数一样，则是子类的这个方法覆盖了父类的方法.

注意事项和使用细节

1> 子类的方法的形参列表，方法名称，要和父类方法的形参列表，方法名称完全一样

2> 子类方法的返回类型和父类方法返回类型一样，或者是父类返回类型的子类

例：父类返回类型是Object，子类返回类型为String.

3> 子类方法不能缩小父类方法的访问权限 public > protected > 默认 > private

方法重写和方法重载比较

名称	发生范围	方法名	参数列表	返回类型	修饰符
重载(overload)	本类	必须一样	类型、个数或顺序至少有一个不同	无要求	无要求
重写(override)	父子类	必须一样	相同	重写方法返回类型与父类方法返回类型一致或是其子类	子类方法不能缩小父类方法的访问范围.

P306 - P Java_多态

基本介绍

方法或对象有多种形态，是面向对象的第三大特征，多态是建立在封装和继承基础之上的

多态的具体体现

1> 方法的多态：重写和重载就体现多态

2> 对象的多态：

① 一个对象的编译类型和运行类型可以不一致

② 编译类型在定义对象时，就确定了，不能改变

③ 运行类型是可以变化的

④ 编译类型看定义时=的左边，运行类型看=号右边.

注意事项和细节讨论

1) 多态的前提是：两个对象(类)存在继承关系

2) 多态的向上转型：

① 本质：父类的引用指向了子类的对象

② 语法：父类类型 引用名 = new 子类类型();

③ 特点：编译类型看左边，运行类型看右边

可以调用父类中的所有成员(需遵守访问权限)；

不能调用子类中特有成员；

最终运行效果看子类的具体实现！

3) 多态的向下转型：

① 语法：子类类型 引用名 = (子类类型) 父类引用；

② 只能强转父类的引用，不能强转父类的对象

③ 要求父类的引用必须指向的是当前目标类型的对象

④ 可以调用子类类型中的所有成员