1 数组和字符串都是引用类型。

2 重写：父子中出现同名变量和方法以及构造函数的问题。

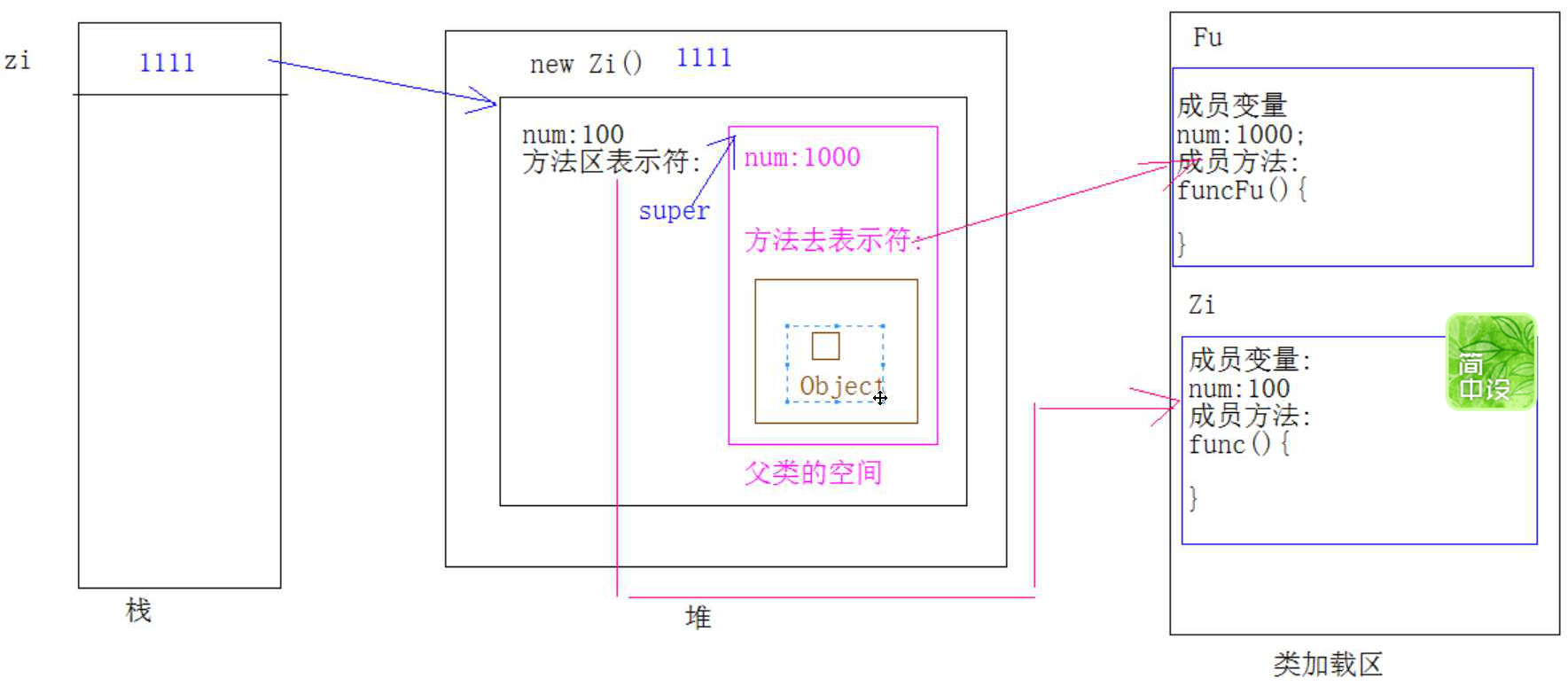
3 父类，超类，基类指的都是父类。

4 父子出现同名变量（super只能访问父类，不能访问祖先类。实际开发中不会用到super）：



5 先加载父类（或者祖父类），总之就是最祖先的类。任何类加载时首先加载object类，因为。。

6 继承的内存图：



7 父子类出现同名方法称为重写（区分重写和重载）。重写的意义：



由于父类的方法是子类同名方法的一部分，所以：

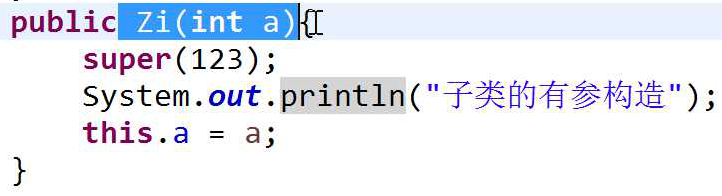


8 重写方法的返回值必须和父类中的方法返回值相同。

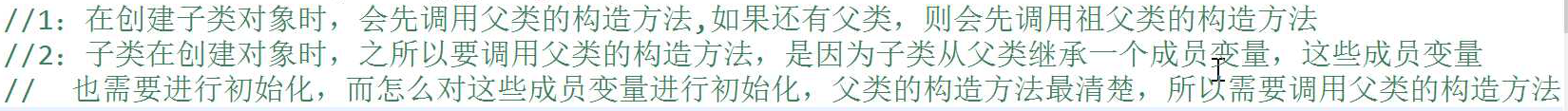
9 重写方法的权限问题：子类方法的权限必须不小于父类该方法的权限。

10 创建子类对象时，会先创建父类（祖父类）的构造方法。

11 不管子类是不是有参，都要调用父类的无参构造函数，因为子类的构造方法的第一行隐藏了super()。也可以显式写出super()。要想调用父类的有参构造，可以在super()中传入参数。并且即使super()有参数，也要放在第一行。



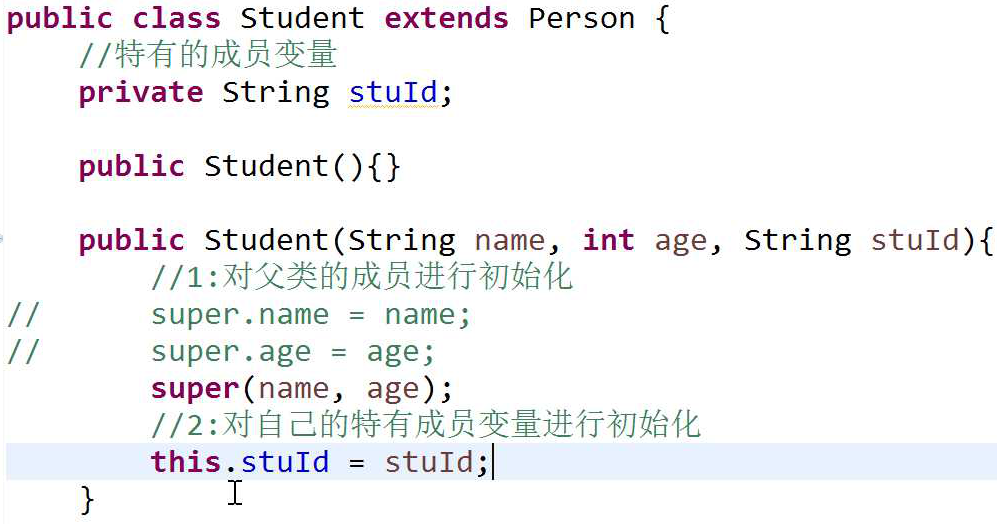
12 子类继承父类的构造函数的原因是：**(private的变量子类不可见，但是父类定义，子类不用再定义，只是用构造函数初始化即可。依据是内存图。)**



13 类中必须有无参构造函数，因为子类会默认有super()，写子类时会出错。

14 一旦写了有参构造函数，系统不会再自动提供无参构造函数。

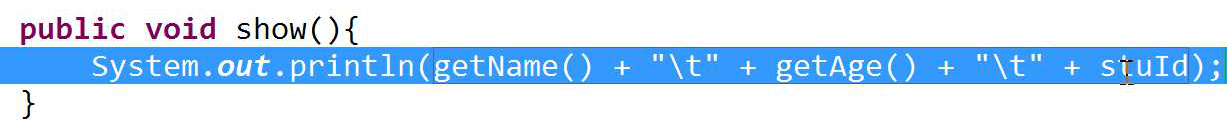
15 子类构造函数必须包括父类的参数（父类的私有成员变量）：



要想自动出现先全部参数，用shift alt+s+o再选择另外一个选项。

16 子类的set()和get()方法不包括父类的，因为可以直接使用父类成员变量的set,get方法。

17 父类私有成员变量的访问方式：



18 有抽象方法的类肯定是抽象类，但是反之不成立。

19 抽象类的特点：不能直接建立对象。必须有子类再创建对象。

20 当一个类继承一个抽象类但是没有重写所有的抽象方法，他仍然是抽象类。

21 继承抽象类中的方法可以在子类名字上用ctrl+1再选择抽象方法。

22 一般来说顶层的类都是抽象类，因为他说不清子类的具体实现。

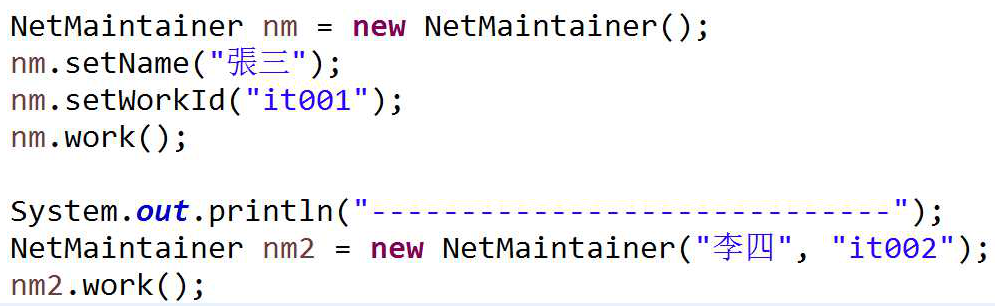
23 在创建类时就可以选择创建成抽象类。

24 ctrl+shift+f切换繁体字和简体字。

25 抽象类的构造方法是为子类准备的。

26 抽象类可以没有抽象方法（少见）。这种抽象类存在的意义是不让我们创建该类的子类。

27 用set get方法好一些，因为如果用有参构造的话就要写构造函数（有参构造函数）。并且所有的中间类都要有全参构造函数。用set get的话一直用的是最祖先的那个抽象类。



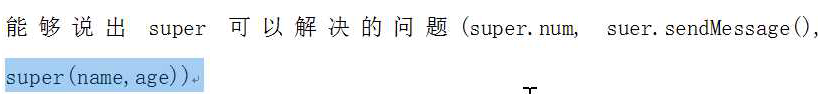
28 奖金：bonus

29 子类还是抽象类的时候用不用把未改变的抽象方法抄一遍？不用

30 区分重载和重写，前者在一个类里面，后者在父子类里面。

31 super只能访问父亲不能访问爷爷。

32 super的作用：



33 注意加数据类型（集合中的元素类型）：

**public** **void** method(ArrayList<Student1> arr){

**for** (**int** i = 0; i < arr.size(); i++) {

}

}

34 本类中的方法可以直接调用，省略了this。如果子类中没有构造函数，就只能用set get方法，如果加上构造方法就可以在创建对象的同时初始化。

35 r,c,o分别是set get 方法，无参构造函数，有参构造函数。

36 **被继承的父类不能只写带参数的构造函数不写空参数的构造函数，否则子类无法让系统添加空参构造函数。其实父类不写构造函数只用系统自动添加的空参构造函数即可。**

37