

JAVA SE Lesson 1

1. 类是一种**抽象**的概念，对象是类的一种具体表示形式，是**具体**的概念。先有类，然后由类来生成对象（Object）。对象又叫做实例（Instance）。
2. 类由两大部分构成：属性以及方法。属性一般用名词来表示，方法一般用动词来表示。
3. 如果一个 java 源文件中定义了多个类，那么这些类中**最多只能有一个类是 public 的**，换句话说，定义的多个类可以都不是 public 的。
4. 在 Java 中进行方法的参数传递时，无论传递的是原生数据类型还是引用类型，参数传递方式统一是传值（pass by value）。Java 中没有传引用（pass by reference）的概念。
5. 方法重载（Overload）。表示两个或多个方法名字相同，但方法参数不同。方法参数不同有两层含义：1）参数个数不同。2）参数类型不同。 **注意：方法的返回值对重载没有任何影响。**
6. 构造方法重载：只需看参数即可。 **如果想在一个构造方法中调用另外一个构造方法，那么可以使用 this()的方式调用，this()括号中的参数表示目标构造方法的参数。this()必须要作为构造方法的第一条语句，换句话说，this()之前不能有任何可执行的代码。**
7. 继承（Inheritance）：Java 是单继承的，意味着一个类只能从

另一个类继承（被继承的类叫做父类【基类，base class】，继承的类叫做子类），Java 中的继承使用 `extends` 关键字。

8. 当生成子类对象时，Java 默认首先调用父类的不带参数的构造方法，然后执行该构造方法，生成父类的对象。接下来，再去调用子类的构造方法，生成子类的对象。【要想生成子类的对象，首先需要生成父类的对象，没有父类对象就没有子类对象。比如说：没有父亲，就没有孩子】。

9. `super` 关键字：`super` 表示对父类对象的引用。

10. 如果子类使用 `super()` 显式调用父类的某个构造方法，那么在执行的时候就会寻找与 `super()` 所对应的构造方法而不会再去寻找父类的不带参数的构造方法。与 `this` 一样，`super` 也必须作为构造方法的第一条执行语句，前面不能有其他可执行语句。

11. 关于继承的 3 点：

- a) 父类有的，子类也有
- b) 父类没有的，子类可以增加
- c) 父类有的，子类可以改变

12. 关于继承的注意事项

- a) 构造方法不能被继承
- b) 方法和属性可以被继承
- c) 子类的构造方法隐式地调用父类的不带参数的构造方法
- d) 当父类没有不带参数的构造方法时，子类需要使用 `super` 来显

式地调用父类的构造方法，**super** 指的是对父类的引用

e) super 关键字必须是构造方法中的第一行语句。

13. 方法重写（Override）：又叫做覆写，子类与父类的方法返回类型一样、方法名称一样，参数一样，这样我们说子类与父类的方法构成了重写关系。

14. 方法重写与方法重载之间的关系：重载发生在同一个类内部的两个或多个方法。重写发生在父类与子类之间。

15. 当两个方法形成重写关系时，可以在子类方法中通过 **super.run()** 形式调用父类的 **run()** 方法，其中 **super.run()** 不必放在第一行语句，因此此时父类对象已经构造完毕，先调用父类的 **run()** 方法还是先调用子类的 **run()** 方法是根据程序的逻辑决定的。

16. 在定义一个类的时候，如果没有显式指定该类的父类，那么该类就会继承于 **java.lang.Object** 类（JDK 提供的一个类，Object 类是 Java 中所有类的直接或间接父类）。

17. 多态（Polymorphism）：我们说子类就是父类（玫瑰是花，男人是人），因此**多态的意思就是：父类型的引用可以指向子类的对象。**