**面向对象**

1. 面向对象：将数据和数据的操作方法放在一起，作为相互依存的整体，即对象。对同类对象抽象其共性，即类。类中的大多数数据，只能被本类的方法进行处理。类通过一个简单的接口与外界发生关系，对象与对象之间通过消息进行通信。程序流程由用户在使用中决定。
2. 面向过程：一种以事件为中心的开发方式，自顶向下顺序执行，逐步求精。其程序结构是按功能划分为若干个基本模块，形成一个树形结构。
3. 面向对象的原则：高内聚、低耦合。
4. 面向对象的三个原则：封装、继承和多态。
5. 类成员初始化顺序
6. 基类静态变量和全局变量
7. 基类静态初始化块
8. 派生类静态变量和全局变量
9. 派生类静态初始化块
10. 基类成员变量
11. 基类初始化块
12. 基类构造函数
13. 派生类成员变量
14. 派生类初始化块
15. 派生类构造函数
16. 面向对象经典例子：

public class Person {

private static String TAG = Person.class.getSimpleName();

public String name;

public String age;

public Person(String name, String age) {

Log.i(TAG, "Person");

this.name = name;

this.age = age;

init();

}

protected void init() {

Log.i(TAG, "name:" + name + ", age:" + age);

}

}

public class Student extends Person {

private static String TAG = Student.class.getSimpleName();

public String name;

public String age;

public Student(String name, String age) {

super(name, age);

Log.i(TAG, "Student");

this.name = name;

this.age = age;

init();

}

protected void init() {

Log.i(TAG, "name:" + name + ", age:" + age);

}

}

new Student("张三", “16”);

08-21 00:14:09.579 21185-21185/com.example.zxg.myprogram I/Person: Person

08-21 00:14:09.579 21185-21185/com.example.zxg.myprogram I/Student: name:null, age:null

08-21 00:14:09.579 21185-21185/com.example.zxg.myprogram I/Student: Student

08-21 00:14:10.392 21185-21185/com.example.zxg.myprogram I/Student: name:张三, age:16

注：子类调用父类构造函数时，父类构造函数调用了init()函数，由于父类的init()函数被子类覆盖了，因此父类构造函数调用的init()函数实际是子类的init()函数；父类构造函数调用init()函数时，实际调用的是子类的init()函数，此时子类的变量还没初始化，所以打印的name和age都是null。