**核心：封装 继承 多态**

**从C语言引入的知识扩展：**

**Java环境：JDK:**java开发工具包**；eclipse：**是一个集成开发环境，便于代码编辑。

**main方法：**在Java的工程文件里只写一个main方法，main方法所在的类是测试类，不要加逻辑。

**输入输出方法：**输入采用system.in的方法输入，在文件中加入Java.util包文件，写入import java.util.Scanner 从而进行输入，首先定义一个Scanner对象，

例如：Scanner a=new Scanner(system.in)

对象建立之后，进行内容的输入。

输出：使用system.out 进行输出。

注：输入输出语句最好在main函数（测试类）里面写，不要在其它类里面写输入输出。

**标识符：**在Java语言中，对变量、常量、函数、语句块也有名字，我们统称为标识符。标识符：是用来给对象 方法 变量 接口和自定义数据类型命名的。标识符由数字字母下划线和美元符号组成，只能以字母或下滑线或美元符号开头。区分大小写，未规定具体长度，不可使用Java关键字和保留字。

**数据类型：**Java数据类型分两种：①基本类型：整数类型 byte,short,int,long

浮点类型 double,float

字符类型 char

布尔类型 Boolean

②引用类型：类、接口、数组、枚举

**基本数据类型与引用数据类型区别：**

**基本数据类型**在被创建时，在栈上给其划分一块内存，将数值直接存储在栈上。

**引用数据类型**在被创建时，首先要在栈上给其引用（句柄）分配一块内存，而对象的具体信息都存储在堆内存上，然后由栈上面的引用指向堆中对象的地址。

**数据类型转换：**隐式类型转换当对数据进行加宽转换时可以进行隐式类型转换，类似：int类型转换为float类型。显示类型转换：当数据进行收缩转换时必须进行显示类型转换（强制转换）例如：float类型转换为int类型。

**==，equals：内置数据类型用双等，引用数据类型用equals；**

**=(内置数据类型)，引用数据类型(拷贝，不使用“=”进行赋值)**

**变量定义：赋值，初始化**

**运算符表达式：**

**控制流程：顺序，分支，循环**

**数组：**长度不变，类型相同，有先后顺序；数组和循环最好一起使用，减少代码书写量。

**基本类型的数组**

**引用类型的数组**

**数组初始化(引用类型的数组)：**

**数组引用的初始化**

**数组元素的初始化**

**数组操作要用for循环**

**Arrays类的使用：**java.util.Arrays类可以对数组进行一些列的操作，copyOf实现数组的复制；fill实现数组元素的初始化；sort实现数组的排序；binarySearch实现排序后的数组元素查找。

**方法的返回值，方法名，参数要很清晰（基本类型传参，引用类型传参）**

**方法重载：**

**封装(类和对象)**

**类，对象基本概念**

**封装：类要包含哪些属性，哪些方法，访问权限**

**方法：普通方法（set**(赋值)**，get**（取值）**，toString，equals），构造方法**（赋值和初始化）

**继承**

**为什么要继承，什么情况下能继承**

**基本语法extends，构造方法（super）**

**方法重写**（核心：基类和派生类有同名的函数方法，）

**抽象类：抽象方法**（可以包含抽象方法，在派生类继承时要实现抽象方法）

**接口类：只能包含抽象方法**（不能有属性，不能有空方法）

实现接口类：就是实现接口类的所有方法

**包**

**多态(动态绑定)**

**向上转型：**基类引用指向派生类的对象

**多态：**

**包装器类**

基本数据类型与引用数据类型的转换；

**内部类**

在某个类内部定义的类

**枚举**

一种数据类型，

**Try catch finally 语句块的执行流程:**

程序首先执行可能发生异常的try语句块。如果try语句没有出现运行时异常则执行完后跳至finally语句块执行；如果try语句出现异常，则中断执行并根据发生的异常类型跳至相应的catch语句块执行处理。catch语句块可以有多个，分别捕获不同类型的异常。catch语句块执行完后程序会继续执行finally语句块。finally语句是可选的，如果有的话，则不管是否发生异常，finally语句都会被执行。