一. OOP中的基本概念

Java的编程语言是面向对象的，采用这种语言进行编程称为面向对象编程(Object-Oriented Programming, OOP)， 它允许设计者将面向对象设计实现为一个可运行的系统。Java的编程单位是类，对象最后要通过类进行实例化(即“创建”)。

面向对象编程有三个特性：(先简单的了解一下这三个特点)

. 封装 Encapsulation：以前我们讲过java是以类为基础的，所有的属性和方法都是封装在类中的，不像C++在类外还可以定义函数。

. 继承 Inheritance：不容置疑，从父亲那里继承什么家产、金钱或者产业什么的，运用到我们java中又是怎么回事呢？它又继承哪些东东呢？

. 多态 Polymorphism：表面看是多种状态的意思。(这里先不过多讨论，下面会详细讨论)

接下来我们就要对这三个特性进行详细的分析，那么在java中我们一切是以类为基础，当然这三个特性跟类是分不开的，那我们认识下java中的类。

二. 抽象数据类型(类就是模板)

在C++中，我们可以用struct 来表示一个类，不了解也不要紧。

在java中，我们用class 这个关键字来表示一个类，类是一个抽象的数据类型，那怎么抽象法呢？

(思考一下我们之前讲过的java中的类是怎么来的)

三. 类和对象

面向对象的开发方法把软件系统看成各种对象的集合，对象就是最小的子系统，一组相关的对象能够组合成更复杂的子系统。面向对象的开发方法将软件系统看成各种对象的集合，接近人的自然思维方式。

猪八戒.吃(西瓜) 吃(猪八戒，西瓜)

对象是对问题领域中事件的抽象。对象具有以下特性：

1) 万物皆为对象。问题领域中的实体和概念都可以抽象为对象。例如学生，成绩单、教师、课和教室。

2) 每个对象都是惟一的。正如世界上不存在一模一样的数子(或者人)。

3) 对象具有属性和行为。

例如小张，性别女，年龄22，身高1.6m, 体重40kg, 能够学习，唱歌。

小张的属性包括姓名、性别、年龄、身高和体重，行为包括学习、唱歌。

例如一部手机，牌子是诺基亚、价格是2000元，银白色，能够拍照、打电话和收发短信等。

这部手机的属性包括品牌类型type、价格price和颜色color，行为包括拍照takePhoto()，打电话call()，收发短信receiveMessage()和发短信sendMessage().

成员变量由英语单词field意译而来的，早期有些书籍将成员变量称为属性，但实际上在JAVA世界里属性（由property翻译而来）指的是一组setter方法和getter方法。比如说某个类有age属性，意味着该类包含setAage()和getAge()两个方法。另外，也有些资料、书籍将field翻译为字段、域。

4) 对象具有状态。状态是指某个瞬间对象的各个属性的取值。对象的某些行为会改变对象自身的状态，即属性的取值。

例如小张本来体重为40kg，经为减肥后，体重减到45kg.

肥胖状态: 40kg

|

| 减肥行为

|

肥胖状态: 45kg

5) 每个对象都是某个类的实例。小张和小王都属于学生类、中国和美国都属于国家类、中文和英文都属于语言类。

类是具有相同属性和行为的对象的集合。

同一个类的所有实例都有相同属性，但属性取值不一事实上相同，但是它们的状态不一定相同。例如小张和小王都属于学生类，都有姓名、性别、年龄、身高和体重这些属性，但是他们的属性取值不同。

同一个类的所有实例都有相同行为，意味着它们具有一些相同的功能。

类是一组具有相同属性和行为对象的模板。面向对象编程的主要任务就是定义对象模型中的各个类。

1) 类是一种类型：是引用类型；

2) 类是元数据： 描述数据的数据，因为数据在面向对象领域里以对象的形式存在，所以类是对象共有属性和方法的抽象描述。

Java程序是各种对象相互交互作用、而不是类。举例：

1) 早上到公司上班，在电梯中碰到总经理我们会说，张总早或王总早，会不会说人早呀！

2) 我们要看电视，是买台电视机，而不是买制作电视机的模具；

在java中，类的申明和实现在同一时间，而且必须在一起，前面我们已经做过很多的例子。

在C++中，类的申明和实现可以被分开，不知道也没有关系，这只是让我们了解下两者的区别。

四. 类中方法的定义

接下来我们来看一下类中定义的方法的格式。

修饰符 返回类型 方法名（参数列表）异常抛出类型{...}

1) 必须有返回值，如果方法没有返回值，必须用void申明返回类型。

2) 构造器没有返回类型，构造器加上了返回类型就变成了一个普通方法的声明了。

3) 方法的修饰符可以同时有多个(多个修饰符之间没有先后顺序)

注意:return\_type:如果方法定义了返回类型,那么在方法体里面就必须出现return语句,而且返回的数据类型要和声明一致,如果没有返回类型的话(void),那么方法体里面可以写return,也可以不写return。当代码执行了return语句,那么代码就不往下执行了,直接退出这个方法并返回

方法不能独立定义，方法只能在类体里定义。

从逻辑意义上来看，方法要么属于该类本身，要么属于该类的一个对象。

永远不能独立执行方法，执行方法必须使用类或对象作为调用者。

只要一个方法的方法体实现中再次调用了方法本身，就是递归方法。

递归一定要向已知方向递归。

方法中定义的参数我们通常叫做形参，调用有参数的方法时，我们通常会

传递一些实参给方法，那么在java中方法的参数是如何的传递呢？

（先思考形参和实参的区别）

//temp是形参

public void add(int temp){

.....

}

main{

//1是实参

add(1);

// temp = 1;

int a = 10;

//a是实参

add(a);

// temp = a;

}

五. 参数传递

参数传递分为两种：

1) 值传递:对于基本数据类型，参数通过值传递。(把实参的值复制一份再传给形参)

2) 引用传递:对于引用类型，参数通过引用(对象的引用)传递。(把实参引用中地址值复制一份再传给形参)

3) 只有引用传递的内容能被改变，而按值传递不会变化。

参考例子ch05.ParameterPassTest.java，并结合示意图理解参数传递。

每当用Java命令启动一个Java虚拟机进程时，Java虚拟机就会创建一个主线程，该线程从程序入口main()方法开始执行。主线程在Java栈内有一个方法调用栈，每执行一个方法，就会向方法调用栈中压入一个包含该方法的局部变量及参数的栈桢。

主线程首先把main()方法的栈帧压入方法调用栈，在这个栈帧中包含2个局部变量 i和s，当主线程开始执行increment()方法时,会把该方法的栈帧也压入方法调用栈。在这个栈帧中包含1个i参数，它的初始值由main()方法的i局部变量传递。

其它与上类同。

六. this关键字

在方法调用、参数传递过程中，极有可能出现参数名称与实例变量名同时的情况。在一个方法内，可以定义和成员变量同名的局部变量或参数，此时成员变量被屏蔽。