JAVA手记

2019.3.1

JAVA语言程序设计 辛运帏 饶一梅 编著 手记

**第一章**

一个完整的程序应该包含大致以下内容：

1. package语句：个数不限但必须在文件开始的地方；
2. import语句：个数不变，但必须放在所有类定义之前；
3. 类定义：分为公有类和非公有类。每个文件公有类最多只有一个，非公有类的个数没有限制。
4. 接口定义：每个文件包含的接口定义个数没有限制；

Package和import语句必须放在所有类定义之前，且package必须放到最前；

如：

Package Transportation;

Import java.awt.Graphics;

Import java.applet.Applet;

输出String args[]和String[] args都可执行 建议使用String[] args 这样可以避免歧义和误读

String[] args也可写成String …args 前者为数组后者为可变参数

凡是应用程序都需要一个main()方法，**main方法前有三个修饰符一个都不能少**，但次序可有变化，如：static public void .

**如果方法不是静态的需要先创建类的实例，然后才能调用实例的方法**

而void则表明main()方法不返回任何值。

如果方法没有返回值则声明为void，不可忽略，若方法有返回值，则返回值类型替代void;

JAVA API

/\*一个类中的信息：

Filed Summary

Constructor Summary

Method Summary

Filed Detail

Constructor Detail

Method Detail

Filed Summary :列出类中成员变量的信息，包括名字、类型及含义。

Constructor Summary：中列出类构造方法的信息，包括参数列表并解释所创建的实例

Method Summary:可以查找到要使用的方法名，在Method Detail中将详细介绍该方法的使用方法

\*/

**第二章：**

面向对象程序设计技术

2019.03.03

面向对象的程序设计过程包含三个步骤：

1. 面向对象的分析
2. 面向对象的设计
3. 面向对象的程序设计

什么是类和对象：

例如楼房为一个实体，它有楼层、外观、面积等各项属性，同时也可以对楼房进行装修、翻盖等工程，盖楼房之前需要图纸。

**图纸就相当于类，也就是一个模板，而盖好的楼房就是具体的对象。**

**除了可以表示为这种具体的概念外，也可以表示为抽象的概念或者过程**。例如：可以定义有关三角形或者矩形的类，定义有关的化学方程式。当表示一个过程时，他的属性表示了过程的进展，也就是不同的状态。**当满足一定的条件的时候，状态就可以发生改变，即这个对象的属性值有变化。**

**对象包含属性和方法，属性值可以表示对象的状态，方法则用来表示对对象的操作**

类的定义相当于设定了一个模板，模板定义时各个成员变量还没有具体的值。当给模板的各个变量赋值后，代表创建了一个实例。使用模板可以闯将不同的实例，每个实例可以带有自己独特的值。每个模板创建的实例个数是不限定的。实例也称为对象，值描述了对象的属性。

面向对象的程序设计方法的重要特征是**封装、继承和多态。**

1. **封装**

将数据及对数据的操作捆绑在一起成为类，这就是封装技术。

封装技术使数据和操作成为一体，对敏感数据的操作只能通过在类中提供的方法来实施。

封装有两个基本含义：

一方面是指对象的全部属性数据和对数据的全部操作结合在一起，形成一个统一个体，也就是对象。另一方面是指尽可能的隐藏对象的内部细节，只保留有限的对外接口，**对数据的操作都通过这些接口实现。这些接口也就是能访问这些数据的成员方法。**

1. **继承**

一个类包含另一个类。外层类称为父类，内层类称为子类、例如机动车包含货车，机动车为父类，而货车则是机动车的子类。

父类可包含的子类个数不定，子类又可以有他的子类，如轿车可以分为自动挡和手动挡，这样的包含关系可以进行嵌套。

继承关系具有传递性，子类可以继承父类的特性，父类又可以继承自己父类的特性，原则上子类可以继承所有祖先类的特性。

在JAVA程序中只有一种基本的结构，即类。具有传递性，特殊的是JAVA中类的关系表现为层次关系，可以使用数据结构中的树来表示。

继承有两方面的含义，一是子类继承父类的共性，二是子类可以有自己独有的特性。

1. **多态**

不同类中完全可以定义同名的方法，因为在系统内部，由类名加方法名组合起来的全名是不同的。由于类名的不同，所以不同类可以存在同名的方法，这就是多态的一种表现形式。

多态不仅仅表现在不同类之间，同一个类中也可以定义多个同名的方法，称为方法的重载。

定义在方法时的参数列表称为形式参数，调用时给定的具体参数称之为实际参数。

在JAVA中可以定义相同方法名但参数列表完全不同的多个方法。

调用时，系统根据方法名称以及实参列表找到对应的方法，这是多态的另一种方法。

例如：要打印一个数据，要求不同的数据类型要选择不同的打印方式，可以选择为每一种数据类型写一种方法，每一种方法接受相应的实参。如：PrintfInt()接受整型参数并打出来，PrintFloat()接受浮点型参数并打印出来。。。但过于繁琐。在OOP（Object Oriented Programmin 面向对象程序设计）中使用多态，这些方法可以取相同的名字，只是参数列表不同，哪怕方法名相同，系统也能分清调用那个方法。

接口与类的差别在于，接口定义的方法可以没有方法体。

JAVA规定继承了空方法的类中必须重写父类中的空方法，只有这样才能创建实例。

子类是父类的子集，所以子类对象属于父类对象集合，子类中任何一个成员也是父类中的一个成员。

**第三章**

//单行注释

/\*一行或多行注释\*/

/\*\*文档注释\*/ 可以用来生成JAVA文档

常用关键字：

abstract boolean break byte case

catch char class const continue

default do double else extends

false final finally float for

goto if implements import instanceof

int interface long native new

null package private protected public

return short static strictfp super

switch synchronized this throw throws

transient true try void volatile

while

标识符不能以数字开头 标识符区分大小写，但长度并无限制。

JAVA中基本的数据类型

|  |  |  |
| --- | --- | --- |
| 基本数据类型 | 数值类型 | 整型byte(8),short(16),int(32),long(64) |
| 实型float(32),double(64) |
| 字符类型 | char(16) 注Unicode码 |
| 逻辑类型 | Boolean(8)(true/false) |
| 复合数据类型 | 类 | class |
| 字符串 | String StringBuffer |
| 数组 | 类型[] |
| 接口类型 | interface |