电脑软件必备知识之软件分类

1. 计算机编程类：

计算机硬件是具体的、形象的，而电脑软件则是逻辑的、抽象的，对于机械硬件是则只有开或关（相当于真或假、0或1），所以计算机能读懂的语言只能是0或1的二进制，即机器语言；而用符号代替二进制编码0和1，则叫汇编语言；机器语言和汇编语言都是贴近计算机的语言，统称低级语言。

而作为高智商的人呢，因为有10根手指头，所以在人类的麻瓜世界普遍采用的是十进制，即人类的语言（中文、英语、日语等等）；那么这就需要有一门既能让计算机读懂，又能让人类读懂的语言，所以编程语言（即高级语言）应运而生。

编程语言按编程思想大体分为面向过程编程POP（Process Oriented Programming）和面向对象编程OOP（Object Oriented Programming），而现在的程序员中最流行的就是Java面向对象语言。无论是哪类语言，它们都有各自的区别和联系，不要认为面向过程中没有面向对象，面向对象中没有面向过程，这种思想是万万不可取的，它们只是从两种不同的角度来看待世界。面向过程思想就是分析出解决问题需要哪些步骤，然后用模块化的函数把这些步骤一步一步地实现，使用的时候一个一个依次调用就行了。而面向对象思想则是把构成问题的事务系统按属性分解成各个对象，建立对象的目的不是为了完成一个步骤，而是为了描述某个事物在解决问题的步骤中的特定行为。

举个简单的例子，例如下五子棋，

（1）用面向过程的思想设计解决思路：

首先也是最重要的是分析解决问题的步骤，

1. 开始游戏
2. 黑子先下
3. 绘制画面
4. 判断输赢
5. 白子再下
6. 绘制画面
7. 判断输赢
8. 返回步骤2
9. 输出最后结果

把上面的步骤建立用户自定义函数，用不同的函数来实现，问题就能得到解决。

1. 用面向对象的思想设计解决思路：

整个五子棋可以分为，

1. 黑白双方，这两方的行为是一模一样的
2. 棋盘系统，负责绘制画面
3. 规则系统，负责判定输赢、犯规等游戏规则

第一类对象（玩家对象）负责接受用户输入，并告知第二类对象（棋盘对象）棋子布局的变化，棋盘对象接收到了棋子的变化就要负责在棋盘上面显示出这种变化，同时利用第三类对象（规则系统）来对棋局进行判定。

可以明显地看出，面向对象是以功能来划分问题，而不是步骤。同样是绘制棋局，这样的行为在面向过程的设计中众多的步骤中，很可能会出现不同的版本，因为通常程序员会考虑到实际情况进行各种各样的简化。而面向对象的设计中，绘图只可能在棋盘对象中出现，从而保证了绘图的统一。

功能上的统一保证了面向对象涉及的可扩展性。例如：我要加入悔棋的功能，如果要改动面向过程的设计，那么从输入到判断到显示这一连串的步骤都要改动，甚至步骤之间的循序都要进行大规模调整。

如果是面向对象的话，只用改动棋盘对象即可，棋盘系统保存了黑白双方的棋谱，只需要简单回溯就行了，而显示和规则判断则不用顾忌，同时整个对象功能的调用顺序都没有变化，改动的只是局部。

面向过程语言：C、Fortran 、Algol 、Cobol、Lisp、Perl

面向对象语言：C#、C++、basic、Delphi、Java、JavaScript、Python、Ruby、Flex

无论是以哪种思想为主的编程语言，最重要的是程序员要有一个清晰的头脑，逻辑性思维对于软件开发人员是至关重要的，有的软件初学者认为会的语言越多越好，其实并不是这样的，学再多的语言也只是工具，关键是你得会有现有的工具来解决实际问题才行。

时代在不断发展，软件技术也在不断提高，在编程中又出现了第三种编程思想：面向切面编程AOP（Aspect Oriented Programming），它也是现在一个比较热门的话题，举个例子，我们现在提供一个查询学生信息的服务，但是我们希望记录有谁进行了这个查询。这样的话，假如我们要实现的服务有多个呢？那就要在每个实现的类都添加这些记录过程。这样做的话就会有点繁琐，而且每个实现类都与记录服务日志的行为紧耦合，违反了面向对象的规则。那么怎样才能把记录服务的行为与业务处理过程中分离出来呢？看起来好像就是查询学生的服务自己在进行，但却是背后日志记录对这些行为进行记录，并且查询学生的服务不知道存在这些记录过程，这就是我们要讨论AOP的目的所在。AOP的编程，好像就是把我们在某个方面的功能提出来与一批对象进行隔离，这样与一批对象之间降低了耦合性，可以就某个功能进行编程。

也许随着互联网科技的进步，在未来还会有更加简单的思想来协助我们解决问题，但是无论是用什么思想进行编程，敲代码才是最好的变成提高方式。