

名词解释

计算机基础

1. XML:

- 可扩展标记语言，eXtensible Markup Language，简单文本格式
- 用于表达结构化信息，通常用于描述和配置型文档，传输交易数据的格式等；
- 被设计用来传输和存储数据；
- 本身没有动作含义，也即不可以执行，也不代表任何动作。

2. XML树结构:

- XML文档会呈现树状结构，类似操作系统中不同层级的目录，以及目录内的文件
- 从根部开始，然后扩展到树叶
- 根 (root)
 - 枝干
 - 树枝
 - 树叶

```
<root>
  <child>
    <subchild>.....</subchild>
  </child>
</root>
```

3. XML元素:

- 如下例子即为一个XML元素，所有XML元素都必须有关闭标签，开始和结束标签是同一字符，但结束是以 / 开头

```
<开始标签>元素内容</结束标签>
```

- XML标签对大小写敏感，即<a>和<A>完全不同
- XML标签必须正确嵌套，即开始和结束必须一一对应，如同信封一样，有左口和右口才能密封，而且是小信封在大信封内，以此类推，如果顺序错乱，例如，小信封的左口对应的是大信封的右口，XML文件将解析错误
- XML必须有根元素，即文档中必须有一个元素是所有其它元素的父辈元素，父与子即代表层级关系

Android

1. 资源：即应用程序App所需要使用的字符串，图片等，如各个不变的显示文字内容，例如应用程序名称，程序内所使用的图片
2. 位图：即Bitmap，在计算机中一种常见图片格式，常规来说即每个像素点都是由三种颜色（RGB）混合而成
3. 布局参数：即如何摆放我们的View控件所需要使用的参数，例如，大小，对齐方式等
4. 输入控件：即应用程序内接受用户输入的元素，用户可以通过触摸，动作，输入法键盘来告诉应用程序用户的意图，常见的有按钮-Button，输入框-EditText等
5. 软键盘类型：当使用Android系统提供软键盘，也即虚拟键盘，一般在屏幕的正下方，为了帮助用户便捷输入提供的不同类型的辅助键盘布局

Java

1. 虚拟机：执行Java程序的模拟计算程序，要注意区分是可执行程序不是我们写的源代码，源代码需要编译后才能被虚拟机执行，也即javac People.java 做的事情，把我们写的Java源代码变成class文件，如同Windows系统的exe文件可以被Windows操作系统执行一样，class文件可以被虚拟机（JVM）执行，即java People做的事情
2. 进程：就是单个exe文件，单个class（大多数情况会是很多class组合出一个进程）文件开始执行的一个大的资源单元，例如部队中的一个班，或者更高的单位，在这个单位内CPU，内存，还有磁盘等其他资源协同完成工作
3. 线程：就好像部队中一个单兵或者一个小的作战单位，由于CPU太快，其

他资源跟不上速度，只好用一个更小单位来充分利用CPU的计算资源，就好像指挥官的思维通常都比较快，指挥官可以轮流指挥不同的作战单位，但同一时间只有一个在执行，而且一个作战单位绝大多数情况都只属于一个指挥官一样，线程只能属于一个进程

4. 类-Class：为了更好的组织，书写和维护代码，程序员更倾向于模拟真实世界的情况来组织代码，所以近几十年来，大部分程序都在使用面向对象的方法开发，这里的面向对象就是面向真实对象（英文原义是物体）的含义，通过模拟真实物体，把一类物体的相关属性，状态和动作写在同一个模版内，这个模版也就是类
5. 对象-Object：模版只是一个抽象的描述，并不代表真实的物体，例如人是一个类，但是真实的人，张三，李四就是具体的某一个人，这个张三李四就是人这个类的实例对象
6. 方法-Methods：代表类的一些动作，但更通常是指一类问题的Java语句的组合，例如人有写字的方法，这个方法会包括拿笔，握笔，写笔画等等，通过这一类动作完成写字，我们把这一个组合称之为方法，也可以叫做函数，和数学的函数概念基本类似，有X值，通过函数或者公式，或者若干个步骤，最终求解出所需要的结果Y
7. 数据类型：对于计算机的绝大部分资源来说，如CPU和内存，都只认0或者1，但是对于程序或者源代码来说，用不同的类型，可以帮助解决不同精度等问题，例如对于年龄来说，绝大部分情况都只是整数，例如20，30，40等，而对于货币来说，会有两位小数，而对于科学计算来说需要更高的精度才能得到正确的结果，另外，正确使用类型可以帮助我们节省内存，优化程序执行效率
8. String：是Java以及其他编程语言的重要数据类型，可以帮助我们表达文字类信息
9. Java的内存分类：通常用栈和堆，栈内存空间小但速度快，堆内存空间大但速度慢，初期大家可以简单理解为基础数据类型，也就是我们说的那八种数据类型（byte到char）都是在栈内存上工作，而基础类型的组合，比如人这个类的实例对象，是很多基础数据类型的组合，他就需要在堆上分配，因为类和其它复合类型的大小不好预测，所以都会放在一个大的内存区域内，而字符串本质是char类型的数组（数值的一组集合），所以字符串是放在堆上的
10. Java的访问修饰：为了控制不同类之间的访问，需要定义类的属性和方法的访问级别，保护类的行为和属性不被随意访问和修改，如同我们在家里

可以自由自在做任何事，但在单位，有些动作和外表却不能和家里一样，
比如，在家里可以穿睡衣，到单位穿睡衣多半是会惹人笑话

11. Java运算符：即我们常规理解的四则运算，**加减乘除**的代码表达方式