1. Java语言的主要特点：  
   　　1. 跨平台性  
   　　所谓的跨平台性，是指软件可以不受计算机硬件和操作系统的约束而在任意计算机环境下正常运行。这是软件发展的趋势和编程人员追求的目标。之所以这样说，是因为计算机硬件的种类繁多，操作系统也各不相同，不同的用户和公司有自己不同的计算机环境偏好，而软件为了能在这些不同的环境里正常运行，就需要独立于这些平台。  
   　　而在Java语言中， Java自带的虚拟机很好地实现了跨平台性。 Java源程序代码经过编译后生成二进制的字节码是与平台无关的，但是可被Java虚拟机识别的一种机器码指令。 Java虚拟机提供了一个字节码到底层硬件平台及操作系统的屏障，使得Java语言具备跨平台性。  
     
   　　2. 面向对象  
   　　面向对象是指以对象为基本粒度，其下包含属性和方法。对象的说明用属性表达，而通过使用方法来操作这个对象。面向对象技术使得应用程序的开发变得简单易用，节省代码。Java是一种面向对象的语言，也继承了面向对象的诸多好处，如代码扩展、代码复用等。  
     
   　　3. 安全性  
   　　安全性可以分为四个层面，即语言级安全性、编译时安全性、运行时安全性、可执行代码安全性。语言级安全性指Java的数据结构是完整的对象，这些封装过的数据类型具有安全性。编译时要进行Java语言和语义的检查，保证每个变量对应一个相应的值，编译后生成Java类。运行时Java类需要类加载器载入，并经由字节码校验器校验之后才可以运行。 Java类在网络上使用时，对它的权限进行了设置，保证了被访问用户的安全性。  
     
   　　4. 多线程  
   　　多线程在操作系统中已得到了最成功的应用。多线程是指允许一个应用程序同时存在两个或两个以上的线程，用于支持事务并发和多任务处理。 Java除了内置的多线程技术之外，还定义了一些类、方法等来建立和管理用户定义的多线程。  
     
   　　5. 简单易用  
   　　Java源代码的书写不拘泥于特定的环境，可以用记事本、文本编辑器等编辑软件来实现，然后将源文件进行编译，编译通过后可直接运行，通过调试则可得到想要的结果。
2. 面向过程和面向对象区别

面向过程就是分析出解决问题所需要的步骤，然后用函数把这些步骤一步一步实现，使用的时候一个一个依次调用就可以了。   
  
面向对象是把构成问题事务分解成各个对象，建立对象的目的不是为了完成一个步骤，而是为了描叙某个事物在整个解决问题的步骤中的行为。  
  
例如五子棋，面向过程的设计思路就是首先分析问题的步骤：1、开始游戏，2、黑子先走，3、绘制画面，4、判断输赢，5、轮到白子，6、绘制画面，7、判断输赢，8、返回步骤2，9、输出最后结果。把上面每个步骤用分别的函数来实现，问题就解决了。  
  
而面向对象的设计则是从另外的思路来解决问题。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，负责判定诸如犯规、输赢等。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i变化就要负责在屏幕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分散在了总多步骤中，很可能出现不同的绘制版本，因为通常设计人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设计的可扩展性。比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，只用改动棋盘对象就行了，棋盘系统保存了黑白双方的棋谱，简单回溯就可以了，而显示和规则判断则不用顾及，同时整个对对象功能的调用顺序都没有变化，改动只是局部的。  
  
再比如我要把这个五子棋游戏改为围棋游戏，如果你是面向过程设计，那么五子棋的规则就分布在了你的程序的每一个角落，要改动还不如重写。但是如果你当初就是面向对象的设计，那么你只用改动规则对象就可以了，五子棋和围棋的区别不就是规则吗？（当然棋盘大小好像也不一样，但是你会觉得这是一个难题吗？直接在棋盘对象中进行一番小改动就可以了。）而下棋的大致步骤从面向对象的角度来看没有任何变化。  
  
当然，要达到改动只是局部的需要设计的人有足够的经验，使用对象不能保证你的程序就是面向对象，初学者或者很蹩脚的程序员很可能以面向对象之虚而行面向过程之实，这样设计出来的所谓面向对象的程序很难有良好的可移植性和可扩展性。