浅析java的环境 变量☐

配置java环境变量：
    JAVA_HOME：指向JDK的目录
    CLASSPATH：指向类路径，即查找类的路径，第一个为当前路径
    PATH：指向可执行文件的路径

    LINUX：修改" .bash_profile "文件，添加下面的内容
        JAVA_HOME=/opt/jdk1.5.0_06
        CLASSPATH=.:$JAVA_HOME/lib/tools.jar:$JAVA_HOME/lib/dt.jar
        PATH=$PATH:$JAVA_HOME/bin:.
        export JAVA_HOME CLASSPATH PATH   (将环境变量导出成为全局变量)

    windows系统：
        我的电脑右键-->属性-->高级-->环境变量

Java的编译执行：
    区别于脚本语言，也区别于C++。比如g++生成a.out后，机器可以直接读懂
    并且执行，但是要依赖本地操作系统，不能跨平台。

    而java源码要先编译成.java文件，再成.class的字节码文件
        Java源程序阶段(.java文件)--->
        Java编译器(javac)--->
        Java字节码文件(.class，与平台无关)--->
        Java解释器(java)，运行Java字节码

Java的内存管理：
    内存的释放是由gc完成的，但是
    程序员可以通过设置对象引用为空或者调用方法暗示gc回收，但不确定
    具体的回收时间，不能被强制执行
    java.lang.System.gc();
    内存泄露是指该内存空间使用完毕之后未回收，如果该堆内存的最小生命周期与程序的
    生命周期相同，例如：
        单例中引用的对象(生命周期与类的生命周期相同)

java的字节码：
    字节码的好处，增加了代码安全性
    字节码的运行时的优势：
        一次编译到处运行，跨平台不跨语言
        保护源代码，类加载器的保护

□□□□□□□□□□□□□□□□
　　　　□□□□□□□□□□□
　　　　□□□□□□□□□□□□□
　　　　□□□□□□□□□□□□□□□□□□□□□

□□□□
　　package mypack;　　//□□□□□□□

　　public class HelloWorld{
　　　　public static void main(String[] args){
　　　　　　System.out.println("Hello World");
　　　　}
　　}
　　□□

　　1□□□□□□□public□□□□□□□□□□.java□□□□□□□□□□□□□□□□□public□□□□□□□□□□□□□□□
□

　　　　2□□□.java□□□□□□□□□class□□□□□□□public□□□□□□
　　　　3□java□□□□□□□□□□□□□□□□□□□□□.class□□□

　　4□□□java□□□□□□□□□□□□main()□□□
□□□□public□□□□□□
　　　　□□□javac -d . HelloWorld.java
　　　　　　□□□□□□□□□□□□□□□□□□□□□□□□-d□□□□□□□□□□□□□□□□□□□□□□□□□□" .
"□□□□□□□□□□□□□□
　　　　　　□□□□□□□□□□□□□□□javac
HelloWorld.java□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□.class□□□□□□□□□□□
　　　　□□□java mypack.HelloWorld
　　　　　　□□□□□□□□□Java□□□□□□□□□□
　　　　　　□□□□□□□□□□□□□□□.□□□□□□□□

□(package)□□□□□□□□□□□□□
　　□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
　　□□□□□□□□□□□□□java□□□□□□□□□
　　□□□□□□□□□□□□□□package p;(□□□□□□□□□□□□□□□□□□□□)
　　□□□□□□□□□□□□□□□□□□□□□□□.□□□□□□□package p1.p2.p3.p4;
　　□"javac HelloWorld.java -d □□□□"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
　　□□□□□□□□□" java □□.□□ "
　　□□□□□□□□□□import □□.□□;
　　□□□□□□□□□□import □□.*;

□□□
　　// □□□□□　□□□□□□□□□□□□□□□□□
　　/* */　□□□□□　□/*　*/□□□□□□□□□□□□□□

/** */ 　□□□□□ java□□□□□/** */□□□□□□□□□□□□□□□□

　　□□□javadoc□java□□□□□□□□□□□□□□□html□□(□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□)

□□□□

　　□□□□□
　　　　(1) □□□□□□□□□□□□$□□□□□□□□□□□□
　　　　(2) □□□□□□
　　　　(3) □□□□java□□□□□□□□□□□

　　□□□□□□□□□□jdk1.2□□strictfp(□□□□□□)□□□□ jdk1.4□□assert(□□)□□□□
jdk1.5□□enum(□□) □□□
　　　　□□□□□□□□□□□□□□□□□
　　　　true□false□null□□□□□□□□□□□□□□□□□□□□□□□

　　□□□
　　　　(1) □□□□□□□□□□□
　　　　(2) □□□□□□□□□
　　　　(3) □□□□□□□□□□□□□□□□ //TarenaStudent
　　　　(4) □□□□□□□□□□□□□□□□□□□□□□□□□□ //tarenaStudent
　　　　(5) □□□□□□□□□□□□□□□□□□" _ "□□

□□□□□□□□8□
　　1) □□
　　　　byte　　　1B 8□　　　　-128□127
　　　　short　　2B 16□　　　-2^15□(2^15)-1
　　　　int　　　　4B　32□　　　-2^31□(2^31)-1
　　　　long　　　8B　　64□　　　-2^63□(2^63)-1
　　2) □□□□
　　　　float　　　4B　　32□
　　　　double　　8B　64□
　　3) □□□□
　　　　char　　　2B 16□
　　4) □□□　　1B
　　　　boolean　　false/true

　　□□
　　　1□Java□□□□□□□□□□□□□□
　　　　1)□□□□□□□□□□□□□□□□□□□□□□□
　　　　　　byte->short->int->long->float->double
　　　　2)□□□□□□□□□□□□□□□□□□□□□□□□□
　　　　　　□□int a = (int)4.562□
　　　　　　□□□□□□□□□□□□□□□
　　　2□boolean□□□□true□false□

3、char在Java中用" \u□□□□□□□□□□□□
(□□□□□□□□\u□□□□□□□□□4□□□□□□□□□□)"□□□□□□□□□□□□unicode□□□□□□□□□□□□□□□□□□□
4、□□□□□□□□double和float□□□□□□□□□□□□" f "或" F "；
5、long□□□□□□□□□□□□□□" l " 或□" L "

□□□□□□□□□
    □□□□□□□□□
    □□□□ □□□ = new □□□□□(□□);   //new□□□□□□□□□□□□□□□□□□
    □□□□□□□□□□□□□□□□

□□□□□□□□□□
    □□□□□□
        □□□□□□□□□□□□□(□□□□□□□□□□□)□□□□□□□□□□□□□□□□□□□□□□□□□□□□
        □□□□□□□□□□□□□□□□□

□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□
    (1)□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
    (2)□□□□□□□□□□□□□□□□□□□□□□□□
    (3)□□□□□□□□□2□□□□□□□□□□□

    □□□□□□□□□□□□□□□□ (□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□)

□□□□□□□□□
    byte a = 1;
    byte b = 2;
    a = a+b;   //□□□□□□□□□□□□□□□int
    a += b;   //□□□□□□□□□□□□□□□
    □□□□□□□□□□□□
        a□b□□□□□□□
        a□b□□double□□□□□□double
        a□b□□float□□□□□□float
        a□b□□long□□□□□□long
        □□□□□□□□□□int
    □□□□□□□□□□,□□□□□□□□□□□. byte c=(byte)a+b;

□□□□□□□□□□□□
    >>   □□□□□□□□□□□
        □□□□□□□□□□□□32□□□□□□□
        □□□□2□□□□□□□□□□□□□
    >>>   □□□□□□□□□0

□□□□□□□

&/|□□□□□□□□□□□

&& □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

|| □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

" + "□□□□

□□□□□□□□□□□□□□□□

□□□□□□□□□□□□□□□□□□□□

□□□□□□□□

□Core C++

switch□□□□□□□□□□byte□ short□int□char□□□□□□enum□□

switch(exp) exp □□□□□□□□□□□enum□□□□

□□□

□□□□□

□□□□□□□□□□□

□□[] array;

□□ array[];

□□

JAVA□□□□□□□[] array;

□□□□□□□□□

□□□□□□□□□□□□□□

□□□□□□□□□

□□□□□

□□□□□□□□□□int[] i = new int[2];

□□□□□□□□□□Student[] s = new Student[100];

□□□□□□□□□□□□□

□□	□□□
byte	0
short	0
int	0
long	0l
float	0.0f
double	0.0d
char	\u0000
boolean	false
reference types	null

□□