

JAVA编码转换的详细过程计算机等级考试 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/645/2021_2022_JAVA_E7_BC_96_E7_A0_81_c97_645411.htm 我们常见的JAVA程序包括以下类别：

- *直接在console上运行的类(包括可视化界面的类)
- *JSP代码类（注：JSP是Servlets类的变型）
- *Servlets类
- *EJB类
- *其它不可以直接运行的支持类

这些类文件中，都有可能含有中文字符串，并且我们常用前三类JAVA程序和用户直接交互，用于输出和输入字符，如：我们在JSP和Servlet中得到客户端送来的字符，这些字符也包括中文字符。无论这些JAVA类的作用如何，这些JAVA程序的生命周期都是这样的：

- *编程人员在一定的操作系统上选择一个合适的编辑软件来实现源代码并以.java扩展名保存在操作系统中，例如我们在中文win2k中用记事本编辑一个java源程序；
- *编程人员用JDK中的javac.exe来编译这些源代码，形成.class类(JSP文件是由容器调用JDK来编译的)；
- *直接运行这些类或将这些类布署到WEB容器中去运行，并输出结果。

那么，在这些过程中，JDK和JVM是如何将这些文件如何编码和解码并运行的呢？这里，我们以中文win2k操作系统为例说明JAVA类是如何来编码和被解码的。第一步，我们在中文win2k中用编辑软件如记事本编写一个Java源程序文件(包括以上五类JAVA程序)，程序文件在保存时默认采用了操作系统默认支持GBK编码格式(操作系统默认支持的格式为 file.encoding格式)形成了一个.java文件，也即，java程序在被编译前，我们的JAVA源程序文件是采用操作系统默认支持的 file.encoding编码格式保存的，java源程序中含有中文信息字符和英文程序代码；要查看

系统的file.encoding参数，可以用以下代码：

```
public class ShowSystemDefaultEncoding { public static void main(String[] args) { String encoding = System.getProperty("file.encoding"); System.out.println(encoding); } }
```

第二步，我们用JDK的javac.exe文件编译我们的Java源程序，由于JDK是国际版的，在编译的时候，如果我们没有用-encoding参数指定我们的JAVA源程序的编码格式，则javac.exe首先获得我们操作系统默认采用的编码格式，也即在编译java程序时，若我们不指定源程序文件的编码格式，JDK首先获得操作系统的file.encoding参数（它保存的就是操作系统默认的编码格式，如WIN2k，它的值为GBK），然后JDK就把我们的java源程序从file.encoding编码格式转化为JAVA内部默认的UNICODE格式放入内存中。然后，javac把转换后的unicode格式的文件进行编译成.class类文件，此时.class文件是UNICODE编码的，它暂放在内存中，紧接着，JDK将此以UNICODE编码的编译后的class文件保存到我们的操作系统中形成我们见到的.class文件。对我们来说，我们最终获得的.class文件是内容以UNICODE编码格式保存的类文件，它内部包含我们源程序中的中文字符串，只不过此时它已经由file.encoding格式转化为UNICODE格式了。这一步中，对于JSP源程序文件是不同的，对于JSP，这个过程是这样的：即WEB容器调用JSP编译器，JSP编译器先查看JSP文件中是否设置有文件编码格式，如果JSP文件中没有设置JSP文件的编码格式，则JSP编译器调用JDK先把JSP文件用JVM默认的字符编码格式（也即WEB容器所在的操作系统的默认的file.encoding）转化为临时的Servlet类，然后再把它编译成UNICODE格式的.class类，并保存在临时文件夹中。如：在中文win2k上

，WEB容器就把JSP文件从 GBK编码格式转化为UNICODE格式，然后编译成临时保存的Servlet类，以响应用户的请求。

100Test 下载频道开通，各类考试题目直接下载。详细请访问
www.100test.com