

将控制台重定向到GUI组件上 PDF转换可能丢失图片或格式  
，建议阅读原文

[https://www.100test.com/kao\\_ti2020/273/2021\\_2022\\_\\_E5\\_B0\\_86\\_E6\\_8E\\_A7\\_E5\\_88\\_B6\\_E5\\_c104\\_273293.htm](https://www.100test.com/kao_ti2020/273/2021_2022__E5_B0_86_E6_8E_A7_E5_88_B6_E5_c104_273293.htm) 最近为一个项目的服务器的做一个界面的时候，为了能把原来在后台打印出来的相关信息重定向到GUI界面的时候，费了一些心思。都是以前在实现的时候大多数信息的打印和测试时使用的都是System.out和System.err之类，图个方便，没有使用log功能。当然这是个不好的习惯。刚开始的时候用Swing做了一个面板，采用JTextPane组件作为打印信息的容器。可能是太久没有用Swing了，在测试的时候这个东西的水平滑动块老是随着信息的显示自动地滑向最后的地方，而且打印信息看起来很凌乱。后来就换用SWT，把打印的信息容器换用Text组件，把样式定义为SWT.MULT | SWT.V\_SCROLL | SWT.WRAP，让它可以自动换行。查一下JDK的API文档，System类提供了可以重定向的方法setOut(PrintStream out)、setErr(PrintStream err)、setIn(InputStream in)。在此我们只需求使用setOut和setErr就够了，这二个方法都要传入一个PrintStream类型的参数，只要在调用打印信息的前面调用这二个方法重设输出流和错误流就可以达到我们的目的。那么我们继承PrintStream类，并把要显示信息的组件作为参数传入到这个自定义的打印流类中。

```
public class MyPrintStream extends PrintStream { private Text text.  
public MyPrintStream(OutputStream out, Text text) { super(out).  
this.text = text. } /** *//** * 在这里重截,所有的打印方法都要调  
用的方法 */ public void write(byte[] buf, int off, int len) { final  
String message = new String(buf, off, len). /**/* SWT非界面线程
```

访问组件的方式 \*/ Display.getDefault().syncExec(new Thread(){  
public void run(){ /\*\*/\*\* 在这里把信息添加到组件中 \*/  
text.append(message). } }). } } 把组件作为参数传入到这个打印  
流中，并重写父类的write(byte[] buf, int off, int len)方法，把写  
出的信息添加到组件上，注意到重载了带OutputStream的构造  
方法。这里有二点必须注意，确保组件在调用打印信息添加  
前是已经被正确创建的，另外必须注册不同GUI组件对线程  
的访问形式，像上面的SWT就对界面的访问有严格的规定。  
完成了PrintStream后，下面是如何使用，在我们启动界面后，  
使用下面的方式： MyPrintStream mps = new  
MyPrintStream(System.out, text). System.setOut(mps).  
System.setErr(mps). 这样就可以把原来程序里的所有System.out  
和System.err信息转移到你的GUI界面上。 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](http://www.100test.com)