

java 作业3

姓名：李文骏

学号：2020213969

班级：软件工程20-3班

1.如何将一个字节输入流转换成采用UTF-8编码的字符输入流，写出相应的代码并解释。

答：若字节输入流命名为inputStream，则将其转化为UTF-8编码的代码为

```
1 InputStreamReader inputStreamReader = new
  InputStreamReader(inputStream, charset.forName("UTF-8"));
```

2. 如何采用URL类实现“<http://www.hfut.edu.cn/logo.png>”的拷贝，写出代码并解释代码。

代码如下所示：

```
1 import java.io*;
2 import java.net*;
3
4 public class Demo {
5     public static void main(String[] args) {
6         try {
7             //创建一个URL对象
8             URL url = new URL("http://www.hfut.edu.cn/logo.png");
9             //通过URL对象的openStream方法获取字节输入流
10            InputStream inputStream = url.openStream();
11            FileOutputStream fos = new FileOutputStream("copy.png");
12
13            //采用一次读一个字节数组的方法
14            byte[] bytes = new byte[1024];
15            int len;
16            while ((len = inputStream.read(bytes)) != 0) {
17                fos.write(bytes, 0, len);
18                fos.flush();
19            }
20            //关闭资源
21            inputStream.close();
22            fos.close();
23        } catch (IOException e) {
24            e.printStackTrace();
25        }
26    }
27 }
```

对于该程序，首先我们将我们需要用到的 java.io 包和 java.net 包导入，然后是创建URL对象，通过URL对象的openStream方法获取字节输入流，同时再创建文件字节输出流，采用一次读一个字节数组的方法，最后关闭资源

3.详述Java中Socket网络通信的过程和步骤。

答：Java中的网络通信是通过Socket实现的，Socket分为ServerSocket和Socket两大类，ServerSocket用于服务端，服务器端可以通过accept()方法监听客户端的请求，监听到客户端请求之后返回Socket，Socket用于具体完成数据传输，客户端直接使用Socket发起请求并传输数据，服务器端也是通过socket接受和发送数据，最后再关闭服务器端和客户端的相关资源

Socket通信的步骤如下：

对于服务器端：

1. 在服务器端中以客户端对应端口为参数开启一个ServerSocket，然后通过ServerSocket的方法accept()监听客户端请求，返回值为一个socket对象。
2. 通过socket的getOutputStream和getInputStream获取Socket的输入输出流
3. 获取输入输出流之后，通过输入流读取客户端发送的请求信息
4. 通过输出流向客户端发送相关信息
5. 关闭相关资源

对于客户端

1. 以IP地址和端口号为参数创建Socket对象与服务器端进行通信
2. 通过socket的getOutputStream和getInputStream获取Socket的输入输出流
3. 获取输入输出流之后，通过输入流读取客户端发送的请求信息
4. 通过输出流向客户端发送相关信息
5. 关闭相关资源

4.FlowLayout和BorderLayout有什么区别，分别是什么容器的默认布局管理器？

FlowLayout为流式布局管理，为Panel和Applet的默认布局管理器

BorderLayout为边框布局管理，是JFrame, JWindow, JDialog, JInternalFrame以及JApplet内容面板的默认布局管理器。

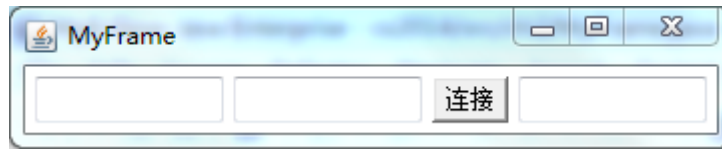
区别：

当布局管理器为**FlowLayout**时，是按照**行从左到右**向容器当中添加组件，有时候，我们向容器中添加的组件比较多时，也就是说当在一行中不能适应这么多的组件，它会另起一行，开始新的一行组件排列。是一种**顺序性布局**

当布局管理器为**BorderLayout**时，管理的容器被划分为**上、下、左、右、中五个区域**，分别对应North(北区), South (南区), West(西区), East (东区) 和Center(中区)，我们添加到哪个区，就在哪个区放入组件。如果在加入组件时没有声明添加的区域，那么默认的是中区

5.使用Java中的AWT编程实现如下功能：

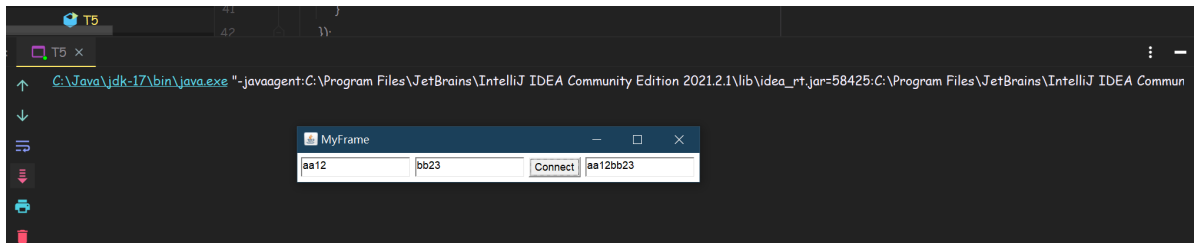
- (1) 使用Frame, TextField, Button等控件绘制如下图界面。
- (2) 点击按钮，将第三个文本框的字符串的值设置为第一个文本框的字符串+第二个文本框的字符串。
- (3) 注意：Frame的默认布局管理器的问题。



代码如下:

```
1  import java.awt.*;
2  import java.awt.event.WindowAdapter;
3  import java.awt.event.WindowEvent;
4
5  public class T5 {
6      Frame frame;
7      Button button;
8      TextField fieldA;
9      TextField fieldB;
10     TextField fieldC;
11     Panel panel;
12
13     public void init(){
14         fieldA = new TextField(10);
15         fieldB = new TextField(10);
16         button = new Button("Connect");
17         button.addActionListener(e -> {
18             fieldC.setText(fieldA.getText()+fieldB.getText());
19         });
20         fieldC = new TextField(10);
21
22         panel = new Panel();
23         panel.add(fieldA);
24         panel.add(fieldB);
25         panel.add(button);
26         panel.add(fieldC);
27
28         frame = new Frame("MyFrame");
29         frame.add(panel);
30         frame.setVisible(true);
31         frame.pack();
32
33         //frame不像JFrame那样方便 若要点击窗口关闭 则需要添加事件监听器
34         frame.addWindowListener(new WindowAdapter() {
35             @Override
36             public void windowClosing(WindowEvent e) {
37                 System.exit(0);
38             }
39         });
40     }
41
42     public static void main(String[] args) {
43         new T5().init();
44     }
45 }
```

运行结果如下:



程序设计思路与代码解释：

首先由于Frame的容器默认布局管理为BorderLayout，不能做到像我们案例中的窗口布局那样三个文本框和一个按钮是按照顺序从左到右排列的。故我们采用Panel，将按钮、文本框等组件放进Panel中，由于Panel的默认布局为FlowLayout，故可以达到程序效果

其次程序的核心在于对Connect按钮的事件监听，通过Button对象的addActionListener()方法，采用匿名类的方法以及Lambda表达式重写actionPerformed方法来进行事件监听，从而实现字符串相组合的功能

最后frame.pack()方法是根据组件自适应Frame窗口大小，同时对于Frame，它不像JFrame那样方便，若要点击窗口关闭，则需要添加事件监听器，通过调用方法addWindowListener方法，同样也是采用匿名类进行参数传递，并重写与事件相关的方法，来实现点击窗口右上角的X按钮，窗口就关闭，并且程序终止