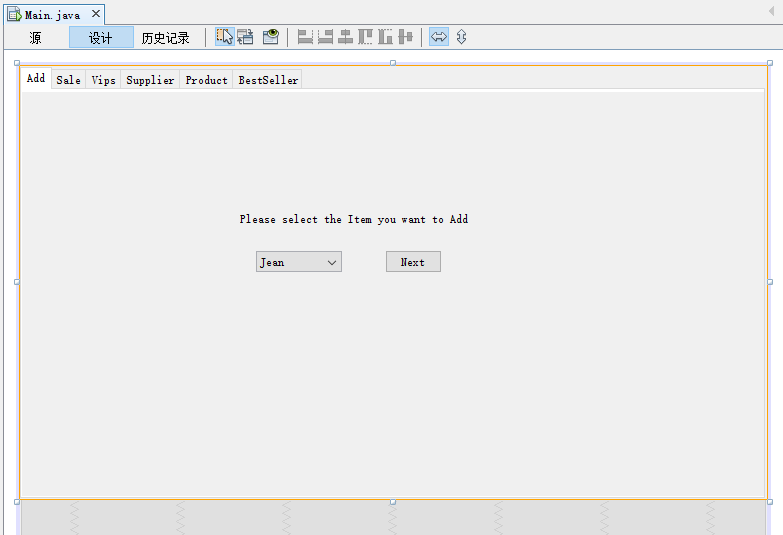
Techniques used:

1. main.java是整个GUI程序的框架部分，主要的界面都在这里设计编写好。之后再添加具体的响应事件。
2. 把需要添加的item信息,如jacket,jean,vip等内容，分别设计分类为不同的类，条理清晰，调用准确。
3. FileOperate.java集成了文件操作方法，有读取和写入。还根据不同的读写操作类型，如写入字符串，对象或ArarayList等，设计了不同的读写方法，如WriteFile(),SaveObject(),SaveObjectList()等。RandomAccessFile包括，sale.txt,vip.txt,supplier.txt,bestseller.txt,jacket.txt等
4. 每个界面切换操作时，都会自动触发一个更新事件，即重新获取界面需要的数据，实现了数据的实时性和准确性。
5. 利用NestBeans的Swing GUI可视化编程的工具，大大减少了编写界面的工作量。

Main.java

从图片就可以看出，这个是整个程序的框架部分，主要的界面都一次性写了（没办法，我也不想所有的代码都放在一个.java里面，不过用这种GUI可视化设计编程的方法，只能这么写），比较臃肿。



负责add,sale,supplier,product,bestseller的界面。

1.main.java里面具体的方法：

（这些方法主要都是可以由系统生成的方法名，然后自己往里面添实现的内容，

）

下面这个是重写的方法，每次切换界面时，都会自动响应这个事件，即执行这个方法。根据当前是哪个界面，来执行对应的操作。

@Override

public void stateChanged(ChangeEvent e) {

// TODO Auto-generated method stub

String name = jtp.getSelectedComponent().toString();

FileOperate filedemo=new FileOperate();

if(name.equals(vippane.toString()))

{

ArrayList<String> strList=new ArrayList<String>();

strList=filedemo.ReadObject("vip.txt");

//System.out.println(strList);

String[][] itemList = {

};

......

......

1. vip界面

从vip.txt里获取文件数据,放入ArrayList里，来显示在界面的表格里。

ArrayList<String> strList=new ArrayList<String>();

strList=filedemo.ReadObject("vip.txt");

//System.out.println(strList);

String[][] itemList = {

};

int len=strList.size();

if(len>0)

{

itemList=new String[len][5];

for(int i=0;i<strList.size();i++)

{

String []str=strList.get(i).split("%");

itemList[i]=str;

}

}

......

1. product界面

从jean.txt,jacket,tshirt.txt里获取文件数据,放入ArrayList里，来显示在界面的表格里

if(name.equals(productpane.toString()))

{

ArrayList<String> jeanList=new ArrayList<String>();

ArrayList<String> jacketList=new ArrayList<String>();

ArrayList<String> tshirtList=new ArrayList<String>();

jeanList=filedemo.ReadObject("jean.txt");

jacketList=filedemo.ReadObject("jacket.txt");

tshirtList=filedemo.ReadObject("tshirt.txt");

String[][] itemList = {

};

int len=jeanList.size()+jacketList.size()+tshirtList.size();

if(len>0)

{

itemList=new String[len][7];

int row=0;

for(int i=0;i<jeanList.size();i++)

{

String []str=jeanList.get(i).split("%");

itemList[i]=str;

}

row=jeanList.size();

for(int j=0;j<jacketList.size();j++)

{

String []str=jacketList.get(j).split("%");

......

1. bestseller界面

从bestseller.txt里获取文件数据,放入ArrayList里，来显示在界面的表格里。

if(name.equals(bestsellerpane.toString()))

{

ArrayList<String> strList=new ArrayList<String>();

strList=filedemo.ReadObject("bestseller.txt");

String[][] itemList = {

};

int len=strList.size();

if(len>0)

{

itemList=new String[len][3];

for(int i=0;i<strList.size();i++)

{

String []str=strList.get(i).split("%");

.......

1. supplier界面

从supplier.txt里获取文件数据,放入ArrayList里，来显示在界面的表格里。

if(name.equals(supplierpane.toString()))

{

ArrayList<String> strList=new ArrayList<String>();

strList=filedemo.ReadObject("supplier.txt");

String[][] itemList = {

};

int len=strList.size();

if(len>0)

{

itemList=new String[len][3];

for(int i=0;i<strList.size();i++)

{

String []str=strList.get(i).split("%");

......

(5)add界面：

点击next按钮，就会执行下面这个方法（即触发对应的事件）。这个方法是根据选择的item名称，来打开一个新的信息添加界面，例如，如果选中jean,就跳转到Add jean界面。

private void nextBtnActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

String item=(String)ItemList.getSelectedItem();

if(item.equals("Jacket"))

{

jacket = new JacketItem();

jacket.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

jacket.setVisible(true);

System.out.println(item);

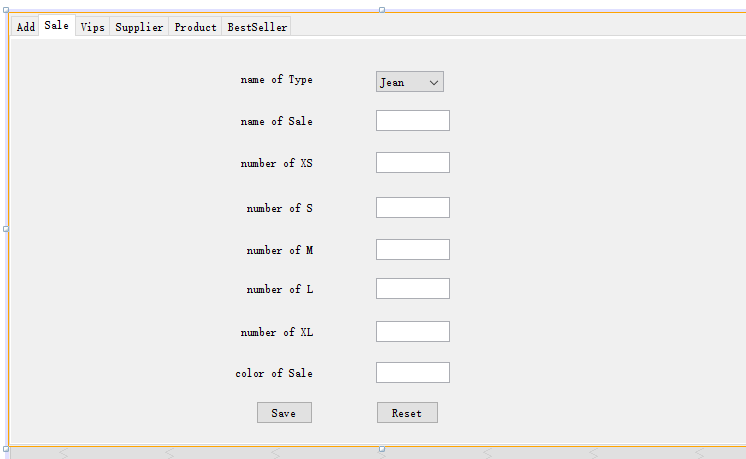
}

......

}

（6）sale界面

点击保存按钮，就会触发下面这个方法。获取sale界面每个输入框的内容，保存到sale.txt中。



private void saveBtnActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

FileOperate filedemo=new FileOperate();

String name=nameInput.getText().trim();

String xs=XSInput.getText().trim();

String s=SInput.getText().trim();

String m=MInput.getText().trim();

String l=LInput.getText().trim();

String xl=XLInput.getText().trim();

String color=colorInput.getText().trim();

int num;

String str="";

......

}

1. FileOperate.java里面
2. WriteFile():

根据txt文件路径，追加写入一个字符串类型的数据。

public static void WriteFile(String content,String path){

File file = new File(path);

file.setWritable(true);

try {

BufferedWriter fw = new BufferedWriter(new FileWriter(file,true));

fw.write(content);

fw.newLine();

fw.flush();

fw.close();

//file.setWritable(false);

System.out.println("Records written to "+path+" successfully !");

} catch (IOException e) {

System.out.println("Error writing to file");

}

}

1. ReadObject():

根据txt文件的路径，获取文件内的数据，返回字符串类型的ArrayList。

public static ArrayList<String> ReadObject(String path){

ArrayList<String> AllObjects = new ArrayList<String>();

File file = new File(path);

String line;

try {

BufferedReader fr = new BufferedReader(new FileReader(file));

while ((line = fr.readLine()) != null) {

String A = line;

AllObjects.add(A);

}

fr.close();

System.out.println("Success to reading "+path+" !");

} catch (IOException e1) {

System.out.println("Error reading file");

} catch (ArrayIndexOutOfBoundsException e2) {

System.out.println("Error. Wrong number of fields");

} catch (NumberFormatException e3) {

System.out.println("Error. Date is not a long or id number not an integer");

}

return AllObjects;

}

1. SaveObjectList():

将字符串类型的ArrayList写入指定路径的txt文件中，覆盖原有的文件内容。

public static void saveObjectList(ArrayList<String> strList,String path)

{

//clear the file

File file1 =new File(path);

file1.setWritable(true);

try {

if(!file1.exists()) {

file1.createNewFile();

}

FileWriter fileWriter =new FileWriter(file1);

fileWriter.write("");

fileWriter.flush();

fileWriter.close();

} catch (IOException e) {

e.printStackTrace();

}

//write new data to the file

File file = new File(path);

file.setWritable(true);

try {

BufferedWriter fw = new BufferedWriter(new FileWriter(file,true));

for(int j=0;j<strList.size();j++)

{

fw.write(strList.get(j));

fw.newLine();

}

fw.flush();

fw.close();

//file.setWritable(false);

//System.out.println("Records written to "+path+"successfully !");

} catch (IOException e) {

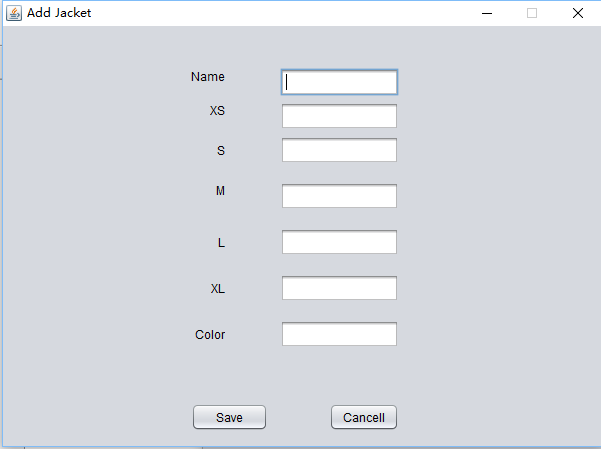
System.out.println("Error writing to file");

}

}

1. JacketItem.java,JeanItem.java,VipItem.java,BestsellerItem.java等

如图，是信息添加的界面。保存按钮，获取每个输入框的信息，将信息保存到对应的txt文件中，取消则退出该界面。另外还有信息提示的对话框。



private void saveBtnActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

FileOperate filedemo=new FileOperate();

String name=nameInput.getText().trim();

String xs=xsInput.getText().trim();

String s=sInput.getText().trim();

String m=mInput.getText().trim();

String l=lInput.getText().trim();

String xl=xlInput.getText().trim();

String color=colorInput.getText().trim();

int num;

String str="";

try {

str="XS";

num=Integer.valueOf(xs);

str="S";

num=Integer.valueOf(s);

str="M";

num=Integer.valueOf(m);

str="L";

num=Integer.valueOf(l);

str="XL";

num=Integer.valueOf(xl);

} catch (Exception e1) {

JOptionPane.showMessageDialog(null, str+" must be a Number","ERROR", JOptionPane.ERROR\_MESSAGE);

return;

}

String all=name+"%"+xs+"%"+s+"%"+m+"%"+l+"%"+xl+"%"+color;

filedemo.WriteFile(all, "jacket.txt");

JOptionPane.showMessageDialog(null, "Saved Successfully !","Congratulations", JOptionPane.INFORMATION\_MESSAGE);

}