



# 廣東工業大學

## JAVA实验 (1)

课程名称: Java 面向对象程序设计

学 院: 管 理 学 院

专业班级: 信息管理与信息系统(班

学 号: 3 1 0 8 0 0 6 4 0 6

姓 名: 张 钟 权

任课教师: 聂 小 东

2010年05月01日

# 广东工业大学

管理学院 信息管理与信息系统专业 08 信管 6 班 \_\_\_\_\_ 组

学号 3108006406 姓名张钟权 协作者 \_\_\_\_\_ 教师评定 \_\_\_\_\_

实验题目 窗体界面程序设计

## 一、实验目的与要求

- (1) 实验目的：训练对 Java 开发工具的安装和环境配置，并能在指定的窗口界面完成程序的编写和调试工作。
- (2) 实验要求：按照 Java 程序设计教程的要求完成第一章中第 1、2、3 节程序的编写和调试。

## 二、实验方案

### (1) 界面布局：

I、主界面程序类 TextFindReplace，实现了 MouseListener 和

WindowsListener 接口：

①Creat()方法：创建主界面，安排查找与替换信息的对话框；

②mouseClicked()方法：处理鼠标单击事件。

II、查找和替换字符串算法类 matchFun：

①strFind()方法：实现字符串查找功能，返回匹配的次数；

②strReplace()方法：实现字符串替换功能，返回替换的次数。

III、查找字符串的核心算法，用到了 String 类的两个方法：

①getCaretPosition()方法：得到当前光标位置，即开始比较的位置；

②CharAt(int index)方法：得到当前字符串 index 位置上的一个字符。

IV、替换字符串的核心算法，重复前面的查找过程，找到要替换字符

的位置，然后将要替换的主字符串转换为 `StringBuffer` 类型，再利用 `StringBuffer` 类的 `replace(int i, int j, String str)` 方法，从主串的第 `i` 位开始替换，共替换掉 `j` 个字符，用字符 `str` 替换。

## (2) 程序启动界面：

I、定义了一个 `JSplashWindow` 类，其父类为 `JWindow`，实现了 `Runnable` 接口：

① `JSplashWindow()` 方法

② `start()` 方法

③ `run()` 方法

④ `showFrame()` 方法

## (3) 树形分层结构：

A、最简单的树形结构：

I、定义了一个 `TreeDemoFrame` 类，其父类为 `JFrame`：

① `TreeDemoFrame()` 方法：该方法中调用了默认的树形结构。

B、处理节点事件：

I、`Apptree` 类，其父类为 `JFrame`，实现了接口 `TreeSelectionListener`：

① `Apptree()` 函数：构造函数，实现界面初始化；

② `valueChanged()` 方法：节点事件。

II、`TreeDemo` 类：

① `pack()` 方法：调整窗口大小以适合子组件的首选大小和布局。

C、文件目录树：

I、`FolderExplorer` 类，其父类为 `JFrame`：

① `FolderExplorer()` 方法：构造方法，实现对界面的初始化；

②main()方法：程序运行的入口；

③jbInit()方法：完成对界面进行初始化的任务；

④initTree()方法：初始化树状结构；

⑤getTreeNode()方法：获取最新被选择的节点，并返回；

⑥getFileNode()方法：返回节点，其类型为 FileNode。

II、MyExpansionListener 类，实现了接口 TreeExpansionListener:

①treeExpanded()方法

②treeCollapsed()方法

III、MySelectionListener 类，实现了接口 TreeSelectionListener:

①valueChanged()方法

IV、FileNode 类，获取系统中的文件目录节点:

①FileNode()方法：构造方法；

②getFile()方法

③toString()方法：若为文件则返回文件名，否则返回节点路径；

④expand()方法：判断各节点是否有子节点；

⑤hasSubDirs()方法

⑥compareTo()方法：判断文件目录是否相同，返回一 int 型的数；

⑦listFiles()方法：得到子文件列表。

D、结合树和输入输出流实现日历记事本:

I、Remember 类，其父类为 JFrame，实现了 TreeSelectionListener

和 ActionListener 两个接口:

①Remember()方法：构造方法，完成界面及树节点初始化；

②valueChanged()方法：处理树节点事件；

③actionPerformed()方法

④del()方法：删除文件。

## II、AppTree 类：

①pack()方法：调整窗口大小以适合子组件的首选大小和布局。

## 三、 实验结果和数据处理

### 1、 界面布局：

①strFind()方法：实现字符串查找，返回匹配次数：

```
public int strFind(String s1,String s2,int pos)
{ /*变量 i 和 j 分别表示主串和模式串中当前字符串位置， k 表示匹配次数*/
int i,j,k=0;
//pos 代表主串中开始比较的位置
i = pos;
j = 0;
while(i<s1.length()&&j<s2.length())
{
if(s1.charAt(i) == s2.charAt(j))
{
++i;
++j;
if(j==s2.length())
{ //j=s2.length() 表示字符串匹配成功， 匹配次数加 1
k = k+1;
//将指示主串和模式串中当前字符的变量 i 和 j 进行回退
i = i-j+1;
j = 0;
}
}
else {i = i-j+1; j = 0;}
}
return k;
}
```

②strReplace()方法：实现字符串替换，返回替换次数：

```
public int strReplace(String s1,String s2,String s3,int pos)
{ /*变量 i 和 j 分别表示主串和模式串中当前字符串位置， k 表示匹配次数*/
int i,j,k=0;
i = pos;
```

```

j = 0;
//将 s1 转化成 StringBuffer 型进行操作
repStr = new StringBuffer(s1);
while(i<repStr.length()&& j<s2.length())
{
if(repStr.charAt(i) == s2.charAt(j))
{
++i;
++j;
if(j==s2.length())
{ /*j=s2.length() 表示字符串匹配成功，匹配次数加 1，此外对主串进行字符串替换 */
k = k+1;
repStr.replace(i-j,i,s3);
//将 j 进行重新赋值开始新的比较
j = 0;
}
}
else
{i = i-j+1; j = 0;}
}
return k;
}
}

```



2、程序启动界面：

showFrame()方法：

```

static void showFrame(String title){
    JFrame frame = new JFrame(title);
    frame.setSize(400,300);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}

```

```
/* 测试屏幕分辨率 */
Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();
Dimension frameSize = frame.getSize();
if (frameSize.height > screenSize.height) {
    frameSize.height = screenSize.height;
}
if (frameSize.width > screenSize.width) {
    frameSize.width = screenSize.width;
}
/* 使窗口自动居中 */
frame.setLocation((screenSize.width - frameSize.width) / 2,
    (screenSize.height - frameSize.height) / 2);
    进入应用系统
frame.getContentPane().add(lb, BorderLayout.CENTER);

frame.setVisible(true);
}
```





3、树形分层结构：

A、最简单的树形结构：

TreeDemoFrame()方法：

```
public TreeDemoFrame()
{
    getContentPane().add(jPanel);
    //调用默认的树型结构
    jPanel.getViewPort().add(jtree, null);
    setSize(400,300);
    //演示怎样使用
    //显示
    show();
    //点击”关闭窗口“按钮时，则退出系统
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
```

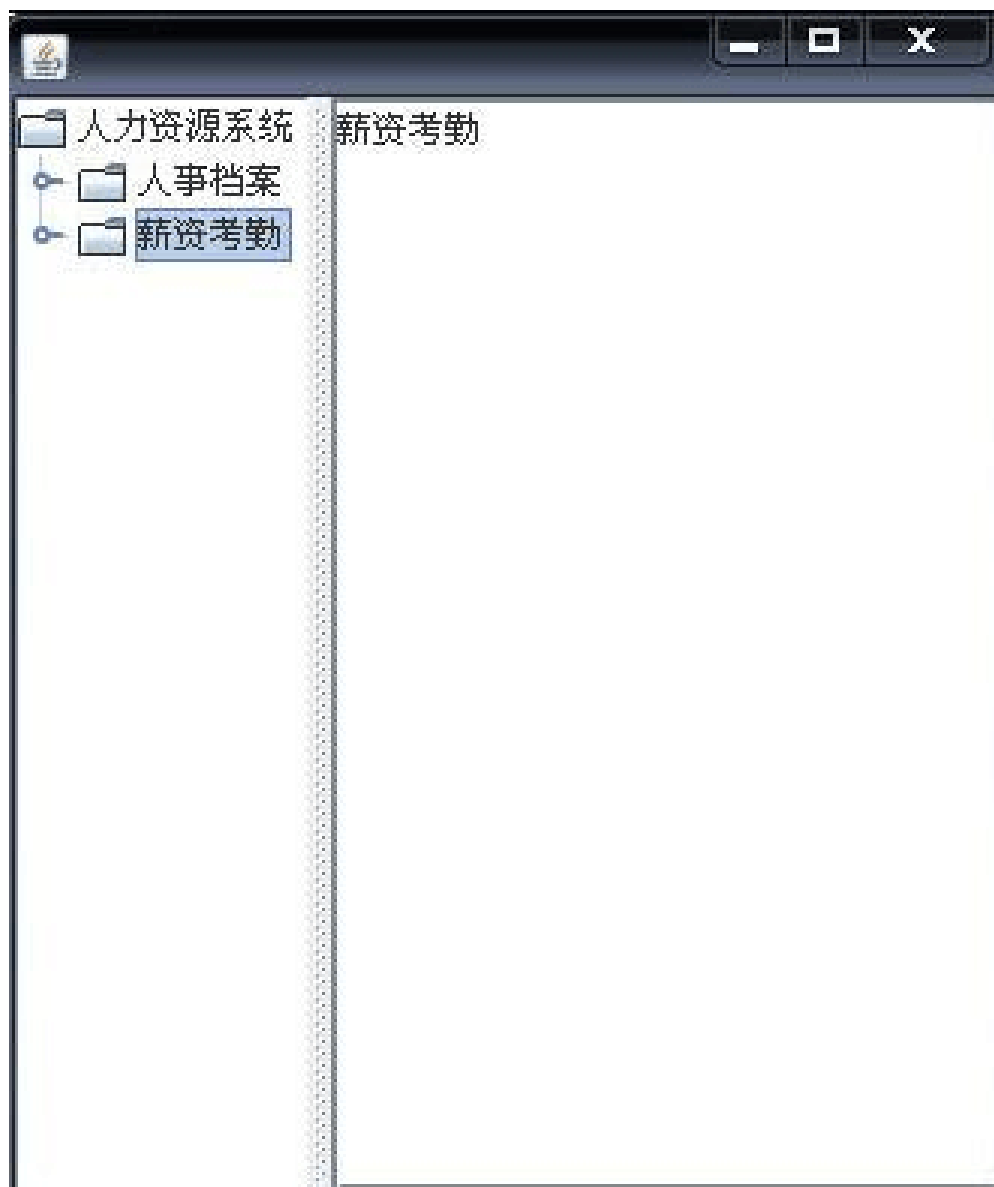




B、处理节点事件：

valueChanged()方法：

```
public void valueChanged(TreeSelectionEvent e)
{
    if(e.getSource()==tree)
        { //定义被选中的节点
        DefaultMutableTreeNode node=
        (DefaultMutableTreeNode)tree.getLastSelectedPathComponent();
        if(node.isLeaf()){
            //获取节点所定义的文本信息
            String str=node.toString();
                人员基本信息
                {
                    这里显示'人员基本信息
                    业绩考核
                    这里显示'业绩考核
                    考勤管理
                    这里显示'考勤管理
                    计件工资
                    这里显示'计件工资
                }
        }
        else {
            text.setText(node.getUserObject().toString());
        }
    }
}
```



C

```
JScrollPane jScrollPane1 = new JScrollPane();
JTree tree;
DefaultTreeModel treeModel;
JLabel lbStatus = new JLabel();

public FolderExplorer() {
    try {
        //    jbInit()  界面初始化方法
        jbInit();
    } catch (Exception e) { e.printStackTrace(); }
}

public static void main(String[] args) {
    FolderExplorer FolderExplorer1 = new FolderExplorer();
}

//jbInit()    方法是对界面进行初始化
private void jbInit() throws Exception {
    initTree();

    this.setSize(new Dimension(400, 300));
    this.getContentPane().add(jScrollPane1, BorderLayout.CENTER);
    this.getContentPane().add(lbStatus, BorderLayout.SOUTH);
    jScrollPane1.getViewport().add(tree, null);
```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/126135215104010115>