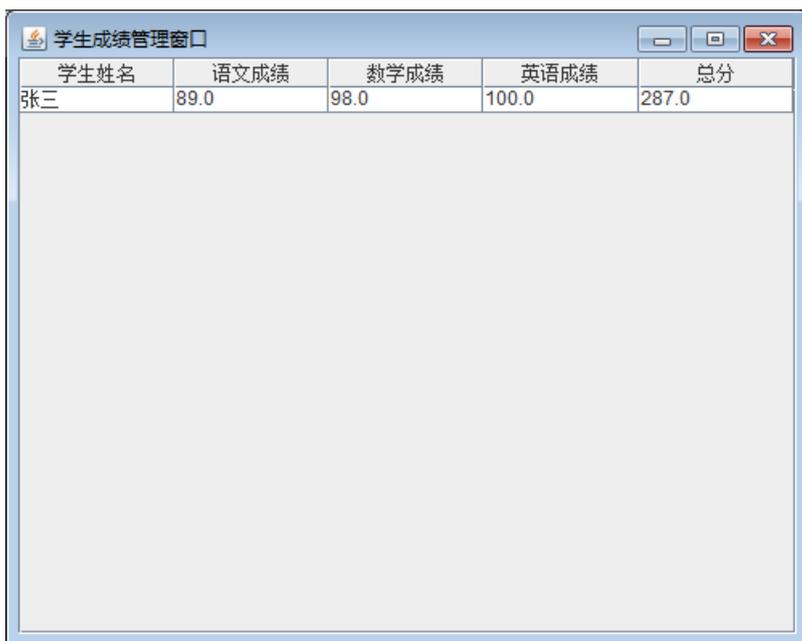




		<pre>         }         对于类 B，它包含一个类型 A 的引用就是持有对方引用         A a1 = new A();         B b = new B();         b.a = a1;         这样我们就能够通过 b.a 来操作对方 而不是直接使用对方引用         2.如何将 A 传递给 B 呢？我们一般用构造方法的方式。     </pre>														
新课导入		请做一个计算班级评价分的程序，实现如下功能： <ol style="list-style-type: none"> <li>依次输入学生的姓名，语文成绩、数学成绩和英语成绩，点击保存后，计算出该同学的平均分。</li> <li>将该同学的信息保存在表格中。</li> <li>显示共录入了多少学生信息，语数外的当前平均分分别为多少。</li> </ol>	提出问题	讨论												
		1.请同学们根据需求思考一下，你将如何设计这个程序，并在纸上画出你的设想。 <ol style="list-style-type: none"> <li>2.抽出一个小组，说出你的想法。</li> <li>3.演示我们最重要的程序运行，并要求同学们看后说一下应该用什么样的步骤来实现。</li> </ol>	分析	思考												
		本单元的学习安排如下： <table border="1" data-bbox="268 1216 1198 1599"> <thead> <tr> <th>单元</th> <th>功能要求</th> <th>课前准备</th> <th>课上学习练习</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>录入学生信息，并显示平均分</td> <td>持有对方的引用</td> <td>1.Jtable 2.应用“持有对方的引用”实现子母窗口</td> </tr> <tr> <td>2</td> <td>处理输入成绩，计算平均分时产生的一场</td> <td>异常和异常处理</td> <td>1.应用异常处理解决实际问题。 2.Collection 集合 3.ArrayList 链表</td> </tr> </tbody> </table>	单元	功能要求	课前准备	课上学习练习	1	录入学生信息，并显示平均分	持有对方的引用	1.Jtable 2.应用“持有对方的引用”实现子母窗口	2	处理输入成绩，计算平均分时产生的一场	异常和异常处理	1.应用异常处理解决实际问题。 2.Collection 集合 3.ArrayList 链表	分析	讨论 观看
单元	功能要求	课前准备	课上学习练习													
1	录入学生信息，并显示平均分	持有对方的引用	1.Jtable 2.应用“持有对方的引用”实现子母窗口													
2	处理输入成绩，计算平均分时产生的一场	异常和异常处理	1.应用异常处理解决实际问题。 2.Collection 集合 3.ArrayList 链表													
任务一	任务引入	1.请同学们根据老师的演示，说一下你的编程思路。 2.我们可以用几个类来解决这个问题？	布置任务	思考	2											
	任务部署	我们将建立 3 个类 <ol style="list-style-type: none"> <li>（1）学生成绩类，封装学生的姓名、语数外成绩以及平均分。</li> <li>（2）主窗口类，用来记录学生的成绩。</li> <li>（3）输入窗口，用来输入学生的成绩。</li> </ol>	讲解	思考 听讲	5											

任务实施 1	定义一个学生类， (1) 学生类应该具有如下属性： <table border="1" style="margin-left: 20px;"> <tr> <td>1.姓名</td> <td>String</td> </tr> <tr> <td>2.语文成绩</td> <td>double</td> </tr> <tr> <td>3.数学成绩</td> <td>double</td> </tr> <tr> <td>4.外语成绩</td> <td>double</td> </tr> <tr> <td>5.平均分</td> <td>double</td> </tr> </table> (2) 将各个属性进行封装	1.姓名	String	2.语文成绩	double	3.数学成绩	double	4.外语成绩	double	5.平均分	double	引导 辅导 修正	学习 编程	
	1.姓名	String												
2.语文成绩	double													
3.数学成绩	double													
4.外语成绩	double													
5.平均分	double													
修正：对于平均分，修改 set 方法怎么写？	引导	讨论												
任务引入	定义一个成绩录入窗口，由用户录入学生的成绩。 	引入	思考	2 0										
任务部署	1. 可以采用什么样的布局管理器来实现？ 2. 如何让窗口的大小不能改变？	讲解 总结	思考											
任务实施 1	1. 我们使用 GridLayout 布局管理器来实现本界面。 2. 由学生自行完成。 3. 代码修改：大多数同学都是这样做的： <pre> JLabel l1=new JLabel("学生姓名 :"); JLabel l2=new JLabel("语文成绩 :"); JLabel l3=new JLabel("数学成绩 :"); JLabel l4=new JLabel("英语成绩 :"); JTextField t1=new JTextField(); JTextField t2=new JTextField(); JTextField t3=new JTextField(); JTextField t4=new JTextField();                 </pre> 依次定义各组件	编程	思考 计算											

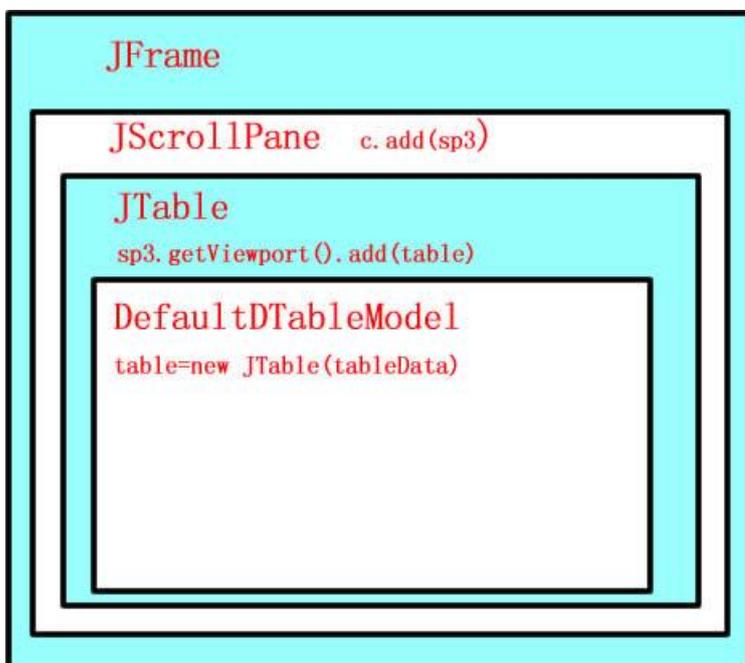
	<pre> c.add(l1); c.add(t1); c.add(l2); c.add(t2); c.add(l3); c.add(t3); c.add(l4); c.add(t4); c.add(b1); c.add(b2); </pre> <p>依次将组建添加到容器里。</p> <p>可以想象，在读取组建内容时，也是依次读出。</p> <p>代码重复度高。</p>			
<p>任务实施 2</p>	<p>3. 提示：我们可以用数组来解决问题</p> <p>(1) 定义 2 个数组，一个是 JTextField 数组，一个是表单信息数组</p> <pre> String[] l_names={"学生姓名：","语文成绩：","数学成绩：","英语成绩："}; JTextField[] ts=new JTextField[4]; </pre> <p>(2) 用一个循环，将标签和文本框依次添加到窗口上。</p> <pre> for(int i=0;i&lt;4;i++){     JLabel l=new JLabel(l_names[i]); 设置对其方式     l.setHorizontalAlignment(JLabel.CENTER);     c.add(l);     ts[i]=new JTextField();     c.add(ts[i]); } </pre> <p>(3) 在清除文本框信息时，操作变得十分简单：</p> <pre> public void clear(){     for(int i=0;i&lt;ts.length;i++){         ts[i].setText("");     } } </pre> <p>重新输入要调用完成录入后也要调用。</p> <p>4. 将输入的信息，封装到一个学生对象里面去。</p>	<p>演示讲解</p>	<p>学习修改</p>	
<p>任务引入 三</p>	<p>定义一个有表格的窗口，用来显示录入的学生成绩信息。</p> <p>提示：用 BorderLayout 布局管理器。</p>	<p>引入</p>	<p>思考</p>	<p>2 0</p>



JTable 的使用:

### 1. 表格的原理

如下图所示，我们可以想象成一个个的容器的包含关系，最底层是我们的窗口 JFrame 类对象，上面先添加一个 JScrollPane 滚动面板对象，滚动面板对象又包含一个 JTable，JTable 中才包含数据对象 DefaultTableModel。我们添加数据的时候事实上是添加在 DefaultTableModel 中的。



### 2. 常用方法

#### (1) 创建空表格

```
String[] headers = { "表头一", "表头二", "表头三" };
```

```
DefaultTableModel model = new DefaultTableModel(headers, 0);
```

任务部署

引导

思考  
讨论

```
table = new JTable(model);
```

(2) 设置列不可随容器组件大小变化自动调整宽度。

```
table.setAutoResizeMode(JTable.AUTO_RESIZE_OFF);
```

(3) 限制某列的宽度。

```
TableColumn firsetColumn = table.getColumnModel().getColumn(0);
firsetColumn.setPreferredWidth(30);
firsetColumn.setMaxWidth(30);
firsetColumn.setMinWidth(30);
```

(4) 取得表格列数

```
int cols = table.getColumnCount();
```

(5) 添加列

```
DefaultTableModel tableModel = (DefaultTableModel) table.getModel();
tableModel.addColumn("新列名");
```

(6) 删除列

```
table.removeColumn(table.getColumnModel().getColumn(columnIndex));
// columnIndex 是要删除的列序号
```

(7) 设置行高

```
table.setRowHeight(20);
```

(8) 置当前行数

```
DefaultTableModel tableModel = (DefaultTableModel) table.getModel();
int n=5;
tableModel.setRowCount(n);
```

(9) 取得表格行数

```
int rows = table.getRowCount();
```

(10) 添加表格行

```
DefaultTableModel tableModel = (DefaultTableModel) table.getModel();
tableModel.addRow(new Object[]{"sitinspring", "35", "Boss"});
```

(11) 删除表格行

```
DefaultTableModel tableModel = (DefaultTableModel) table.getModel();
model.removeRow(rowIndex); // rowIndex 是要删除的行序号
```

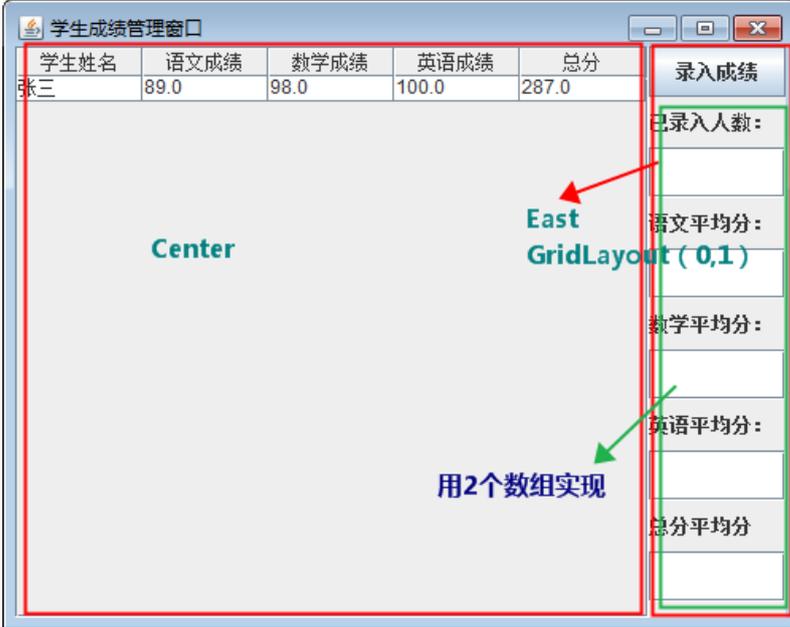
(12) 取单元格数据

```
DefaultTableModel tableModel = (DefaultTableModel) table.getModel();
String cellValue=(String) tableModel.getValueAt(row, column); // 取
单元格数据, row 是行号, column 是列号
```

(13) 获取用户所选的单行

	<pre> int selectedRowIndex = table.getSelectedRow(); // 取得用户所选单行 (14) 取得用户所选的多行 int selectRows=table.getSelectedRows().length;// 取得用户所选行的 行数 DefaultTableModel tableModel = (DefaultTableModel) table.getModel(); if(selectRows&gt;1)     int[] selRowIndex=table.getSelectedRows();// 用户所选行的序列     for(int i=0;i&lt;selRowIndex.length;i++){         // 用 tableModel.getValueAt(row, column)取单元格数据         String cellValue=(String) tableModel.getValueAt(i, 1);     } }  (15) 添加表格的事件处理 table.addMouseListener(new MouseListener() {     public void mousePressed(MouseEvent e) {         // 鼠标按下时的处理     }     public void mouseReleased(MouseEvent e) {         // 鼠标松开时的处理     }     public void mouseEntered(MouseEvent e) {         // 鼠标进入表格时的处理     }     public void mouseExited(MouseEvent e) {         // 鼠标退出表格时的处理     }     public void mouseClicked(MouseEvent e) {         // 鼠标点击时的处理     } });                 </pre>			
<b>任 务 实 施</b>	1. 定义表格 (1) 定义表格标题 (2) 定义在表格中显示的数据 (3) 定义表格	演示 讲解	学习 记录	

	<pre>String[] head={"学生姓名","语文成绩","数学成绩","英语成绩","总分"}; DefaultTableModel data=new DefaultTableModel(head,0); JTable table=new JTable(data);</pre> <p style="text-align: center;">创建表格标题      0行</p> <p>2. 将表格放在 JScrollPane 上。这样当数据量比较大的时候，会自动出现滚动条。</p> <pre>JScrollPane sp=new JScrollPane(table);</pre> <p>3. 将表格放在窗口上，事实上我们放的是 JScrollPane</p> <pre>c.add(sp,bl.CENTER);</pre> <p>4. 在表格中添加数据。</p> <p>添加数据的时候，每一行数据都是一个字符串类型的数组。为了更好的体现面向对象的编程思想，更好的封装数据，并使各个科目的成绩与总分之间存在逻辑关系，我们须要进行如下转换：把对象转换为字符串数组。</p> <p>思考：这个转换应该发生在那个类中更好？</p> <p>(1) 初始化对象</p> <pre>Student stu=new Student(); stu.setName("张三"); stu.setChinese(89); stu.setMath(98); stu.setEnglish(100); stu.setSum();</pre> <p style="text-align: center;">1.封装对象</p> <p>(2) 将对象转换为数组</p> <pre>} public String[] print(){     String[] str=new String[5];     str[0]=this.name;     str[1]=String.valueOf(this.chinese);     str[2]=String.valueOf(this.math);     str[3]=String.valueOf(this.english);     str[4]=String.valueOf(this.sum);     return str; }</pre> <p style="text-align: center;">2.将对象转换为数组</p> <p>(3) 将数组添加到行里面去。</p> <pre>public void addStu(Student stu){     String[] row=stu.print();     data.addRow(row); }</pre> <p style="text-align: center;">3.将对象添加行 独立一个方法</p>			
<p style="text-align: center;">任务部署</p>	<p>现在我们已经实现如下配件：</p> <p>1. 定义了信息录入窗口，该窗口将学生的信息录入并封装到了一个对象 stu 中。</p> <p>2. 实现了显示窗口，并能够显示一个学生的信息。</p> <p>下面我们要做的是，应用子母窗口，将录入的成绩显示在表格里。</p>			1 0

任务实施 1	<p>首先做出界面。</p> <p>提示：可以将按钮单独放在左侧的 pane 中。</p> <p>参照 InputFrame，用数组来简化代码实现。</p> 	辅导	编程
展示与点评	<p><b>体会：</b></p> <ol style="list-style-type: none"> <li>1. 为什么将添加行的方法定义在 Mainframe 中而不是 InputFrame 中？</li> <li>2. 在编写程序的时候，我们不能只停留在功能实现的层面上，要考虑好功能的划分。</li> </ol>	展示 点评 纠正	听讲 记录 思考
梳理脉络	<ol style="list-style-type: none"> <li>1. 定义右侧组件为成员变量</li> </ol> <pre> JPanel p=new JPanel(); JButton b=new JButton("录入成绩"); String[] l_names={"已录入人数:", "语文平均分:", "数学平均分:", "英语平均分:", "总分平均分"}; JTextField[] tfs=new JTextField[5];                 </pre> <ol style="list-style-type: none"> <li>2. 布局并形成右侧的 JPanel</li> </ol> <pre> GridLayout gl=new GridLayout(0,1); gl.setHgap(10); p.setLayout(gl); p.add(b); for(int i=0;i&lt;5;i++){     JLabel l=new JLabel(l_names[i]);     p.add(l);     JTextField tfs[i]=new JTextField();     tfs[i].setEditable(false);     p.add(tfs[i]); } c.add(p,bl.EAST);                 </pre>		
任	<ol style="list-style-type: none"> <li>1. 信息录入窗口，具有主窗口的对象，由此来操作主窗口的属性和方法。</li> </ol>		

<b>务部署</b>	2. 主窗口中，有按钮，能够打开信息录入窗口，并把自身当做参数传递过去。			
<b>任务实施</b>	1. 将 MainFrame 声明为 InputFrame 的成员变量，并通过构造方法赋值。 <pre> MainFrame2 mf; public InputFrame4(MainFrame2 mf){     this.mf=mf; }                 </pre> 2. MainFrame 将自身当做参数传递给 InputFrame <pre> public void actionPerformed(ActionEvent e) {     InputFrame4 input=new InputFrame4(this); }                 </pre> 3. 分别修改两个类的 actionPerformed 方法。			
<b>教学小结</b>	1. JTable 的使用步骤 2. 持有对方的引用实现子母窗口。	总结	听讲记录	5
<b>布置作业</b>	同步课外项目：模拟购票系统，实现买票功能。 本节复习：课堂录像、项目说明文档 作业要求：程序上传至云盘，放在 *组-姓名 文件夹下			
<b>课后学习资源</b>	下节预告 项目 2.4.2 成绩管理（二） 课前储备： 微课《异常和异常处理》			