

## 一、项目导入 (20) 分钟

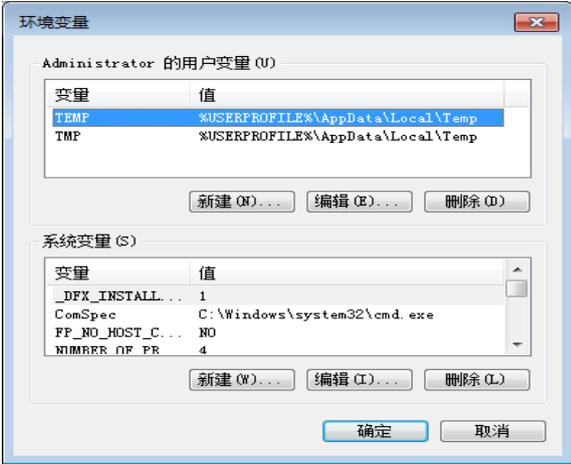
### 主题导入

教学环节	教学内容	教师活动	学生活动
提问	1. 下列关于类的说法中, 错误的是 ( ) A. Java 中创建类的关键字是 class B. 类中可以有属性与方法, 属性用于描述对象的特征, 方法用于描述对象的行为 C. Java 中对象的创建, 首先需要定义出一个类 D. 一个类只能创建一个对象 答案: D	提问 分析	讨论 回答
	2. 下列修饰符中, 成员内部类被 ( ) 修饰后, 可以被外界访问。 A. default B. protected C. public D. private 答案: C	提问 分析	讨论 回答
	4. 类的定义必须包含在以下哪种符号之间? A. 方括号[] B. 花括号{} C. 双引号"" D. 圆括号() 答案: B	提问 分析	讨论 回答
	5. 下列关于类与对象的说法中, 错误的是 ( )。 A. 类是对某一类事物的抽象描述, 而对象则是该类事物的个体。 B. 对象是类的实例化 C. 类用于描述多个对象的共同特征, 它是对象的模板 D. 类与对象之间没有关系 答案: D	提问 分析	讨论 回答
	6. 下列关于类和对象的描述, 错误的是 ( )	提问	讨论

	<p>A. 对象是类的实例</p> <p>B. 一个类只能有一个实例化对象</p> <p>C. 类是对象的抽象</p> <p>D. 类用于描述一组对象的共同特征和行为</p> <p>答案： B</p>	分析	回答
	<p>7.下列关于类与对象的说法中，正确的是（ ）</p> <p>A. 类可以看做是一个模型，可以用来创建对象</p> <p>B. 没有类，也可以创建对象</p> <p>C. 类是对某一类事物的抽象描述，而对象用于表示现实中该类事物的个体</p> <p>D. 以上说法都不对</p> <p>答案： C</p>	提问 分析	讨论 回答
	<p>8.在 Java 语言中，下列关于类的继承的描述，正确的是（B）。</p> <p>A.一个类可以继承多个父类</p> <p>B.一个类可以具有多个子类</p> <p>C.子类可以使用父类的所有方法</p> <p>D.子类一定比父类有更多的成员方法</p>	提问 分析	讨论 回答
	<p>9.Java 中，如果类 C 是类 B 的子类，类 B 是类 A 的子类，那么下面描述正确的是（A）</p> <p>A.C 不仅继承了 B 中的成员，同时也继承了 A 中的成员</p> <p>B.C 只继承了 B 中的成员</p> <p>C.C 只继承了 A 中的成员</p> <p>D.C 不能继承 A 或 B 中的成员</p>	提问 分析	讨论 回答
	<p>10.关于继承的说法正确的是：（ ）</p> <p>A.子类将继承父类所有的属性和方法。</p> <p>B.子类将继承父类的非私有属性和方法。</p> <p>C.子类只继承父类 public 方法和属性</p> <p>D.子类只继承父类的方法，而不继承属性</p> <p>答案： B</p>	提问 分析	讨论 回答

**新课导入**

教学环节	教学内容	教师	学生
------	------	----	----

		活动	活动								
<p>某制药公司希望配备设备的实验室外安装一个门禁系统，只有得到授权并取得保护密码的人员才被允许进入实验室。以下便提供了一些有效密码及所代表的雇员小组：</p> <table border="1" data-bbox="165 349 1054 618"> <thead> <tr> <th>保护密码</th> <th>雇员小组</th> </tr> </thead> <tbody> <tr> <td>1645</td> <td>技术人员</td> </tr> <tr> <td>8345</td> <td>管理人员</td> </tr> <tr> <td>9998, 1006, 1007, 1008</td> <td>科学家</td> </tr> </tbody> </table> <p>当输入保护密码后，应使该保护密码对任何可能站在门禁系统前的人员都是不可见的。而所输入的每一个保护密码，其请求要么是被允许，要么是被拒绝。所有试图进入实验室的请求都将显示在位于小键盘下的一个屏幕中，如果请求被允许，则相应的日期、时间及其所属的小组名称（科学家、管理员等）均会显示在屏幕中；如果请求被拒绝，则相应的时间、日期以及一条错误信息也会显示在屏幕中。</p>		保护密码	雇员小组	1645	技术人员	8345	管理人员	9998, 1006, 1007, 1008	科学家	提出问题 分析	讨论 思考
保护密码	雇员小组										
1645	技术人员										
8345	管理人员										
9998, 1006, 1007, 1008	科学家										
<p>我们在前期的学习中，都是用 JOptionPane 来输入，用 JOptionPane 或 System.out.print 来输出。在实际程序开发中，我们一般喜欢用图形界面来输入输出。这节课我们开始学习图形用户界面。</p> <p>什么是图形界面？</p> 		分析	讨论 观看								

## 二、课前测试与知识梳理 (20 分钟)

### 什么是图形用户界面？

教学环节	教学内容	教师活动	学生活动
	<p>所谓图形用户界面（Graphics User Interface，简称 GUI），是指使用图形的方式，以菜单、按钮、标识、图文框等标准界面元素组成的用户操作屏幕。大家最常见、使用最多的是 Windows 系统下的用户操作界面。</p> <p>在应用系统得开发中，根据需要可能要设计各种各样的用户界面。因此，我们所关心的是开发环境提供了哪些用于构成用户界面的组件元素，这些组件元素的功能及</p>	讲解	听讲

作用是什么，组件元素之间有无关系，如何利用这些组件元素构建用户操作界面。

在 Java 中，将构成图形用户界面的各种组件元素大致可分为以下三类：容器、组件和用户自定义成分。

### 1. 容器 (Container)

容器是用户屏幕上的一个特殊的窗口，它用来组织或摆放其它界面元素。一个容器上可以摆放若干个界面元素，这些界面元素本身可能也是一个容器，这些容器上也可摆放其它的界面元素，依次类推就可构成一个复杂的图形界面系统。例如，框架 (Frame) 容器是 Java 中的标准窗口，在它上边可以摆放窗格 (Panel) 容器和其他组件，在窗格容器上又可以摆放窗格容器和其他组件等等。

容器的引入有利于把复杂的图形用户界面分解为功能相对独立的子部分。在设计用户界面时，如果使用的界面元素较多，就可以按操作需要分类，将它们分别放在不同的容器中，然后以某种规则（嵌套、行列、顺序等）摆放在用户屏幕上。

如上所述，容器是构建用户界面的关键组件，它的主要特点如下：

1) 容器是一个窗口（矩形区域），作为一个组件对象被摆放在屏幕上，有其位置和大小，在它上边摆放的元素也被限制在这个窗口之内。

2) 容器作为一个对象可以现身或隐身，当容器现身时，它所包含的所有元素也同时显示出来，当容器被隐身时，它所包含的所有元素也一起被隐藏。

3) 容器上的元素可以按一定的规则来排列（布局）。

4) 容器的嵌套性，一个容器可以被嵌套在其它的容器之中。

### 2. 组件

组件是图形用户界面上最小的界面元素，它被放置在容器上，它里面不能再包含其它的组件。组件的作用是显示或接收信息，完成与用户的一次交互。例如，接收用户的一个命令、显示给用户一段文本或一个图形等等。

## 三、任务实施

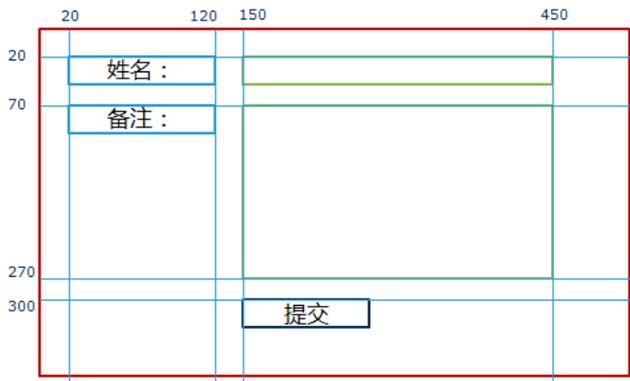
### 任务 1: (40 分钟)

教学环节	教学内容	教师活动	学生活动
任务引入	初始化一个带标题的窗口。	布置任务	思考
任务部署	<b>(二) 对 JFrame 类继承的理解:</b> 每一个 GUI 应用程序是由至少一个继承 Java API (java 中已经定义好的类库) 中的 JFrame 的类组成的，利用继承机制扩展 JFrame，应用程序可以将 JFrame 作为一个“模板”来使用。从 JFrame 中继承，能够带来一个关键性好处就是 Java API 中已经定义好了 JFrame 的含义，它已经具有了一个窗口的基本功能，即程序员不必再重新定义一个窗口的这些功能了。	讲解	思考 听讲

	<p>如何初始化一个窗口？</p> <ol style="list-style-type: none"> <li>1.导入必要的 jar 包：import javax.swing.*;</li> <li>2.继承类 JFrame</li> <li>3.写构造方法 <ul style="list-style-type: none"> <li>(1) 设置标题：this.setTitle("第一个窗口");</li> <li>(2) 设置大小：this.setSize(400, 300);</li> <li>(3) 设置为显示：this.setVisible(true);</li> </ul> </li> <li>4.初始化窗口</li> </ol>		
任务 实施 1	<pre>public class FirstFrame extends JFrame{      public FirstFrame(){          this.setTitle("第一个窗口");          this.setSize(400, 300);      }  }</pre> <p>问题：</p> <p>1.没有显示</p>	演示	学习 编程
	<pre>public class FirstFrame extends JFrame{      public FirstFrame(){          this.setTitle("第一个窗口");          this.setSize(400, 300);          this.setVisible(true);      }  }</pre> <p>问题：</p> <p>点击关闭按钮，不能关闭。</p>	演示	学习 编程
任务 实施 2	<pre>public class FirstFrame extends JFrame{      public FirstFrame(){</pre>	演示	学习 编程

	<pre> <b>this</b>.setTitle("第一个窗口");  <b>this</b>.setSize(400, 300);  <b>this</b>.setVisible(<b>true</b>);  <b>this</b>.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);      }  } </pre> <p>注意：习惯上，我们将<b>this.setVisible(true)</b>放在最后面。</p>		
任务部署	<p>补充：</p> <p>System.exit(0)是退出整个程序，如果有多个窗口，全部都销毁退出。</p> <p>frame.setDefaultCloseOperation()是设置用户在此窗体上发起"close"时默认执行的操作。必须指定以下选项之一：</p> <p>DO_NOTHING_ON_CLOSE (在 WindowConstants 中定义)：不执行任何操作；要求程序在已注册的 WindowListener 对象的 windowClosing 方法中处理该操作。</p> <p>HIDE_ON_CLOSE (在 WindowConstants 中定义)：调用任意已注册的 WindowListener 对象后自动隐藏该窗体。</p> <p>DISPOSE_ON_CLOSE (在 WindowConstants 中定义)：调用任意已注册 WindowListener 的对象后自动隐藏并释放该窗体。</p> <p>EXIT_ON_CLOSE (在 JFrame 中定义)：使用 System exit 方法退出应用程序。仅在应用程序中使用。</p> <p>默认情况下，该值被设置为 HIDE_ON_CLOSE</p> <p>也就是说没有设置的话,默认点关闭时只是隐藏窗体,在后台进程中还可以看到，如果有多个窗口，只是销毁调用dispose的窗口，其他窗口仍然存在，整个应用程序还是处于运行状态。</p>	讲解	思考 听讲
练习	做一个窗口，自己定义大小、标题和显示位置。	辅导	编程 讨论

**任务 2： (45 分钟)**

教学环节	教学内容	教师活动	学生活动
任务引入	现在，我们已经有了一个窗口，窗口上还要放一些组件，比如按钮、文本框、单选按钮、多选按钮等，在 swing 包中，有 JButton、JLabel、JTextField 等，那么如何将这些组件显示在窗口上呢？	引入	思考
任务部署	<p><b>JFrame: Swing</b> 有三个基本构造块：标签、按钮和文本字段；但是现在需要个地方放它们，希望用户知道如何处理它们。<b>JFrame</b> 类就是做这个的——它是一个容器，允许您把其他组件添加到它里面，把它们组织起来，并把它们呈现给用户。<b>JFrame</b> 在本机操作系统中是以窗口的形式注册的，可以最小化/最大化、改变大小、移动等。</p> <p><b>JLabel:</b> 用来显示不允许用户进行修改的文本，即标签。</p> <p><b>使用方法:</b></p> <ol style="list-style-type: none"> <li>1.声明并初始化: <code>JLabel l1=new JLabel("该批货物箱数:");</code></li> <li>2.定义显示: <code>l1.setBounds(16, 16, 130, 21);</code></li> <li>3.添加到容器中: <code>c.add(l1);</code></li> </ol> <p><b>JTextField:</b> 从键盘中获取输入或者将信息显示给用户。</p> <p><b>使用方法:</b></p> <ol style="list-style-type: none"> <li>1.声明并初始化: <code>JTextField tf1=new JTextField();</code></li> <li>2.定义显示: <code>tf1.setBounds(148, 16, 40, 21);</code></li> <li>3.定义其他属性(可选):  <code>tf1.setHorizontalAlignment(JTextField.RIGHT);</code>  <code>tf1.setText("0");</code></li> <li>4.添加到容器中: <code>c.add(tf1);</code></li> </ol> <p><b>JButton:</b> 允许用户命令应用程序完成一项操作。</p> <p><b>使用方法:</b></p> <ol style="list-style-type: none"> <li>1.声明并初始化: <code>JButton b=new JButton("计算");</code></li> <li>2.定义显示: <code>b.setBounds(204, 48, 126, 24);</code></li> <li>3.添加到容器中: <code>c.add(b);</code></li> </ol> <p>常用方法见 java 帮助文档。</p>	讲解总结	思考
任务实施	<p>画出结构图:</p> 	讲解引导	思考计算



```
public class Practice2_component extends JFrame{
```

```
    private JLabel l1=new JLabel();
```

```
    private JLabel l2=new JLabel();
```

```
    private JTextField t=new JTextField();
```

```
    private JTextArea ta=new JTextArea();
```

```
    private JButton b=new JButton();
```

```
    public Practice2_component(){
```

```
        Container c=this.getContentPane();
```

```
        c.setLayout(null);
```

```
        l1.setText("姓名: ");
```

```
        l2.setText("备注: ");
```

```
        b.setText("提交");
```

```
        l1.setBounds(20, 20, 100, 30);
```

任务  
实施  
2

演示  
学习  
思考

	<pre> l2.setBounds(20 , 70, 100, 30);  t.setBounds(150, 20, 300, 30);  ta.setBounds(150, 70, 300, 200);  b.setBounds(150, 300, 100, 30);  c.add(l1);  c.add(l2);  c.add(t);  c.add(ta);  c.add(b);  this.setTitle("第一个窗口");  this.setSize(500, 400);  this.setVisible(true);  this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); }  public static void main(String[] args){      Practice2_component p=new Practice2_component(); } } </pre>		
练习	模仿老师编写的程序，按照界面定义的步骤，做一个窗口	辅导	编程
思考	没有自动换行怎么办？ Ta.setLineWrap(true);	引导	讨论 解决

### 任务 3： (45 分钟)

教学环节	教学内容	教师活动	学生活动
------	------	------	------

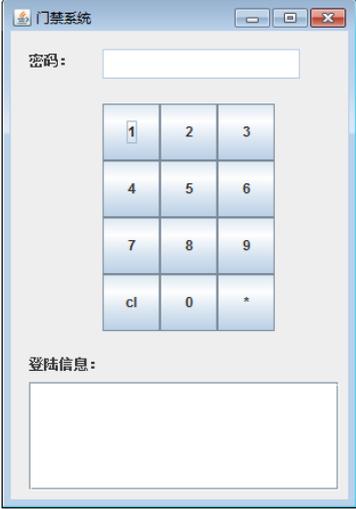
<p>任务引入</p>	<p>某制药公司希望配备设备的实验室外安装一个门禁系统，只有得到授权并取得保护密码的人员才被允许进入实验室。以下便提供了一些有效密码及所代表的雇员小组：</p> <table border="1" data-bbox="263 286 1145 448"> <thead> <tr> <th>保护密码</th> <th>雇员小组</th> </tr> </thead> <tbody> <tr> <td>1645</td> <td>技术人员</td> </tr> <tr> <td>8345</td> <td>管理人员</td> </tr> <tr> <td>9998, 1006, 1007, 1008</td> <td>科学家</td> </tr> </tbody> </table> <p>当输入保护密码后，应使该保护密码对任何可能站在门禁系统前的人员都是不可见的。而所输入的每一个保护密码，其请求要么是被允许，要么是被拒绝。所有试图进入实验室的请求都将显示在位于小键盘下的一个屏幕中，如果请求被允许，则相应的日期、时间及其所属的小组名称（科学家、管理员等）均会显示在屏幕中；如果请求被拒绝，则相应的时间、日期以及一条错误信息也会显示在屏幕中。</p>	保护密码	雇员小组	1645	技术人员	8345	管理人员	9998, 1006, 1007, 1008	科学家	<p>引入</p>	<p>思考</p>
保护密码	雇员小组										
1645	技术人员										
8345	管理人员										
9998, 1006, 1007, 1008	科学家										
<p>任务部署</p>	<div style="display: flex; justify-content: space-around;">   </div>	<p>引导</p>	<p>思考讨论</p>								
<p>任务实施</p>	<pre>public class SecurityPanel extends JFrame{      JButton b1=new JButton("1");      JButton b2=new JButton("2");      JButton b3=new JButton("3");      JButton b4=new JButton("4");      JButton b5=new JButton("5");      JButton b6=new JButton("6");      JButton b7=new JButton("7");      JButton b8=new JButton("8");      JButton b9=new JButton("9");      JButton b0=new JButton("0");      JButton bclear=new JButton("cl");      JButton benter=new JButton("*");</pre>	<p>辅导</p>	<p>编程</p>								

```
JLabel l=new JLabel("密码: ");
JLabel l2=new JLabel("登陆信息: ");
JPasswordField pf=new JPasswordField();
JTextArea ta=new JTextArea();
JScrollPane sp=new JScrollPane();

public SecurityPanel()
{
    this.createInterface();
}

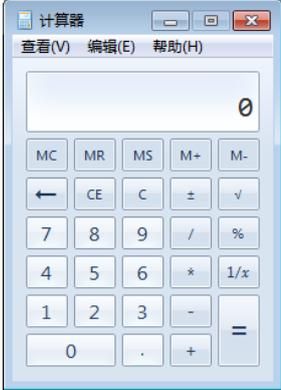
public void createInterface()
{
    Container c=this.getContentPane();
    c.setLayout(null);
    l.setBounds(16, 16, 90, 21);
    pf.setBounds(80, 16, 172, 26);
    pf.setEnabled(false);
    b1.setBounds(80, 64, 50, 50);
    b2.setBounds(130, 64, 50, 50);
    b3.setBounds(180, 64, 50, 50);
    b4.setBounds(80, 114, 50, 50);
    b5.setBounds(130, 114, 50, 50);
    b6.setBounds(180, 114, 50, 50);
}
```

```
b7.setBounds(80, 164, 50, 50);  
b8.setBounds(130, 164, 50, 50);  
b9.setBounds(180, 164, 50, 50);  
bclear.setBounds(80, 214, 50, 50);  
b0.setBounds(130, 214, 50, 50);  
benter.setBounds(180, 214, 50, 50);  
  
l2.setBounds(16, 285, 100, 16);  
sp.setBounds(16, 309, 270, 95);  
sp.setViewportView(ta);  
ta.setEditable(false);  
  
c.add(l);  
c.add(pf);  
c.add(b1);  
c.add(b2);  
c.add(b3);  
c.add(b4);  
c.add(b5);  
c.add(b6);  
c.add(b7);  
c.add(b8);  
c.add(b9);  
c.add(b0);  
c.add(bclear);
```

	<pre> c.add(benter);  c.add(l2);  c.add(sp);  <b>this</b>.setSize(310, 450);  <b>this</b>.setTitle("门禁系统");  <b>this</b>.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  <b>this</b>.setVisible(<b>true</b>);  }  <b>public static void</b> main(String[] args) {      SecurityPanel sp=<b>new</b> SecurityPanel();  }  } </pre>		
		<p>点评学生做的界面</p> <ol style="list-style-type: none"> <li>1.组件的对齐情况?</li> <li>2.按钮之间的距离?</li> <li>3.如何做到固定窗口大小?</li> <li>4.窗口的形状</li> <li>5.按钮的大小。</li> </ol>	<p>思考</p> <p>提问</p> <p>引导</p> <p>解决问题</p>
<p>补充</p>	<p><b>窗口居中的方式</b></p> <pre> <b>this</b>.setSize(310, 450);  <b>int</b> windowWidth = <b>this</b>.getWidth(); //获得窗口宽  <b>int</b> windowHeight = <b>this</b>.getHeight(); //获得窗口高 </pre>	<p>引导</p>	<p>讨论</p>

<pre>Toolkit kit = Toolkit.getDefaultToolkit(); //定义工具包  Dimension screenSize = kit.getScreenSize(); //获取屏幕的尺寸  int screenWidth = screenSize.width; //获取屏幕的宽  int screenHeight = screenSize.height; //获取屏幕的高  this.setLocation(screenWidth / 2 - windowWidth / 2, screenHeight / 2 - windowHeight / 2); //设置窗口居中显示</pre>		
<p><b>设置固定窗口大小</b></p> <pre>this.setResizable(false);</pre>	引导	讨论

#### 四、总结与课后安排

教学环节	教学内容	教师活动	学生活动
教学小结	1.初始化窗口的步骤 2.将组件添加到窗口上的步骤。	总结	听讲 记录
布置作业	 <p>根据 windows 中计算器，模拟一个计算器窗口。</p> <p>课后任务</p> <p>练一练：模拟 ATM 取款机窗口、计算器窗口，详见作业说明。</p> <p>本节复习：课堂录像、项目说明文档</p> <p>作业要求：程序上传至云盘，放在 *组-姓名 文件夹下。</p>		
课后学习资源	微课《Java 中的抽象类》、《Java 中的接口》		