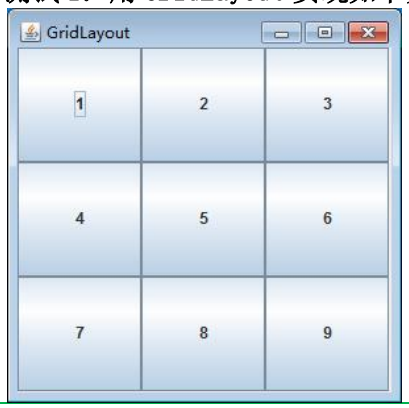
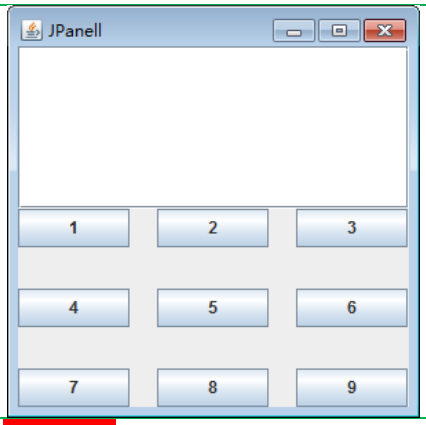
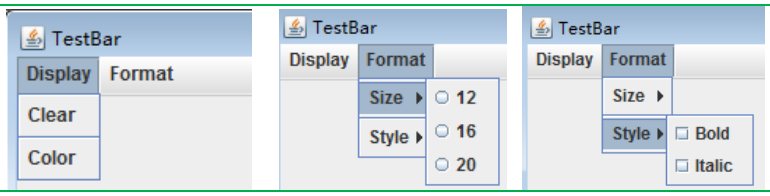
单元 7.2 键盘测试（二）

单元教学设计

<table>
<tr><td colspan="2">学习阶段</td><td colspan="2">二、实用程序开发</td><td colspan="2">学时</td></tr>
<tr><td colspan="2">项目 7</td><td colspan="2">键盘测试</td><td>学时</td><td>6</td></tr>
<tr><td colspan="2">单元 7.2</td><td colspan="2">键盘测试（一）</td><td>学时</td><td>2</td></tr>
<tr><td rowspan="3">教学目标</td><td>能力目标</td><td colspan="2">知识目标</td><td colspan="2">素质目标</td></tr>
<tr><td>能够熟练使用菜单设计程序功能，能够监听键盘事件。</td><td>1. 掌握菜单和菜单项的使用；
2. 掌握键盘监听器的实现；
3. 掌握颜色选择器的使用。</td><td>课上学习练习</td><td rowspan="2">引导学生关注工科领域里的学科人文，通过键盘的发展历史：
1. 让学生了解到小小键盘在布局、工艺精益求精的追求，这正是大国工匠精神的体现。
2. 通过布局的编号，字母排序的原因，引导学生对人机交互、人体工程学的思考。</td></tr>
<tr><td>1. 理解什么是持有对方引用
2. 学会定义内部类</td><td>课前知识储备</td></tr>
<tr><td>课前准备</td><td>视频</td><td colspan="3">马士兵讲座：《持有对方引用》、《内部类》</td></tr>
<tr><td></td><td>课前练习</td><td colspan="3">用内部类来改造我爱大明星程序和门禁系统程序。</td></tr>
<tr><td>项目引入</td><td colspan="4">我们在上一节课中，已经实现了游戏的界面，在本节课中，我们将把功能依次添加：
1. 按下键盘上的某个按键，则相应的按钮高亮显示，键盘按键释放后，按钮恢复原来的颜色。
2. 所按键盘的字母显示在 JtextArea 上。
3. 给程序添加菜单，实现选择字体、字号、文本颜色以及清除所显示的文本等各种功能。</td></tr>
<tr><td>项目进度安排</td><td colspan="4"><table><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><tr><td>1</td><td>点击按键，相应按钮高亮显示。</td><td>布局管理器。</td><td>1. java 的内部类和键盘监听器。
2. Jpanel 的使用。
3. 用 ascII 码显示字符。</td></tr><tr><td>2</td><td>添加菜单，选择字体、字号、文本颜色以及清除所显示的文本。</td><td>Java 中的内部类。</td><td>1. 菜单
2. 颜色选择器
3. 键盘监听器</td></tr></table></td></tr>
<tr><td>十分钟测试</td><td colspan="2">测试 1：用 GridLayout 实现如下界面。
 
</td><td colspan="2">窗口实现要求：
1. 用 GridLayout，将 1-9 共 9 个按钮，放到窗口上。
2. 用内部类实现事件监听器。
3. 在 9 个按钮上注册事件监听器，每点击一个按钮，则在控制台打印出该按钮上显示的文字。</td></tr>
</table>

<table>
<tr><td rowspan="5">能力训练任务</td><td>任务一</td><td colspan="2"> 

键盘监听程序要求：
1.用内部类定义键盘事件监听器，要求按下按键时，响应的按键高亮显示，释放按键后，恢复原样。

2.将 JtextArea 设置为不可编辑状态，按下按键时，将按键内容添加进去。</td></tr>
<tr><td></td><td colspan="2">课程思政
知识点：抽象类和接口含抽象方法不能创建对象，要利用子类或者实现类去生成对象；接口是一种规范，它只用来声明规则。面向接口编程具有相当大的灵活性。
思政元素：进行价值塑造与思想引领，以及人文精神培养。
融入方式：
1.提醒学生不要空想，一方面要树立远大理想，另一方面要为了实现理想去脚踏实地地努力奋斗，才能有所收获将来为国家做更大的贡献。
2.对于既定的标准与规则，每一位公民都要遵守。
3.标准及规范性与灵活性的辩证关系问题，可提升学生的人文精神素养。可引导学生在处理学习、生活等方面的事情时运用唯物辩证法思考问题：在大是大非面前讲规矩、讲纪律，在允许自由裁量范围内讲灵活性。</td></tr>
<tr><td>任务二</td><td colspan="2"> 

设计一个菜单，对窗口中的字体进行设置。
1.Display 下有两个菜单项 Clear 和 Color
2.Format 下有 2 个子菜单，Size 和 Style
3.Size 有 3 个菜单项，用来设置字体的大小。
4.Style 有 3 个菜单项，用来设置字体的类型。</td></tr>
<tr><td>任务三</td><td> 
</td><td>设置字体分别实现菜单项的功能。
1.设置字体的大小
2.设置字体的样式
3.设置字体的颜色。

深入了解两个类：
1.Font 类
2.Color 类</td></tr>
<tr><td>任务四</td><td colspan="2">将键盘监听器和菜单分别添加到 keyBoard 程序上。</td></tr>
<tr><td colspan="2">课后作业</td><td colspan="2">1.完善功能
2.课外同步项目</td></tr>
<tr><td colspan="2">课后学习资源</td><td colspan="2">在线教学平台</td></tr>
</table>

单元教学进度设计

Step1：项目导入（20）分钟

教学环节		教学内容	教师活动	学生活动
测试	测试1	1．下面关于内部类的说法，错误的是（）。 A、内部类不能有自己的成员方法和成员变量 B、内部类可用 abstract 修饰定义为抽象类，也可以用 private 或 protected 定义 C、内部类可作为其他类的成员，而且可访问它所在类的成员 D、除 static 内部类外，不能在类内声明 static 成员 答案：A	引导提问	抢答
	测试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	引导提问	抢答
	测试3	下列修饰符中，哪个修饰内部类后，会使内部类会随着外部类的加载而加载。 A. static B. protected C. public D. private 答案：A 解析：内部类被 static 修饰后，会随着外部类的加载而加载。所以，可以把一个静态内部类理解为一个外部类。	引导提问	抢答
	测试4	Outer 类中定义了一个成员内部类 Inner，需要在 main()方法中创建 Inner 类实例对象，以下四种方式哪一种是正确的？ A. Inner in = new Inner() B. Inner in = new Outer.Inner(); C. Outer.Inner in = new Outer.Inner(); D. Outer.Inner in = new Outer().new Inner(); 答案：D	引导提问	抢答
	测试5	2.问：什么是内部类呢？ 答：内部类（ Inner Class ）就是定义在另外一个类里面的类。与之对应，包含内部类的类被称为外部类。		
	测试6	3.问：那为什么要将一个类定义在另一个类里面呢？清清爽爽的独立的一个类多好啊！！ 答：内部类的主要作用如下： （1）内部类提供了更好的封装，可以把内部类隐藏在外部类之内，不允许同一个包中的其他类访问该类 （2） 内部类的方法可以直接访问外部类的所有数据，包括私有的数据 （3）内部类所实现的功能使用外部类同样可以实现，只是有时使用内部类更方便	引导提问	抢答
	测试	**4.阅读如下代码，说出会有什么样的输出**	引导提问	抢答

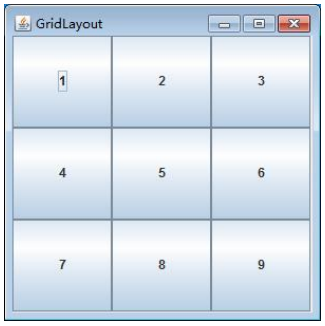
	7	 ``` public class Outer { private int a=22; public class Inner{ private int b=99; public Inner(){ System.out.println("outer的a="+a); System.out.println("inner 的b="+b); } } public void test(){ new Inner(); } public static void main(String[] args){ Outer o=new Outer(); o.test(); } } ``` outer的a=22 inner 的b=99 从上面的代码中我们可以看到，成员内部类的使用方法： 1、 Inner 类定义在 Outer 类的内部，相当于 Outer 类的一个成员变量的位置，Inner 类可以使用任意访问控制符，如 public 、protected 、private 等 2、 Inner 类中定义的 test() 方法可以直接访问 Outer 类中的数据，而不受访问控制符的影响，如直接访问 Outer 类中的私有属性 a 3、 定义了成员内部类后，必须使用外部类对象来创建内部类对象，而不能直接去 new 一个内部类对象，即：内部类 对象名 = 外部类对象.new 内部类( ); 4、 编译上面的程序后，会发现产生了两个 .class 文件 Outer$Inner.class 2014/6/2 13:02 CLASS 文件 Outer.class 2014/6/2 13:02 CLASS 文件 其中，第二个是外部类的 .class 文件，第一个是内部类的 .class 文件，即成员内部类的 .class 文件总是这样：外部类名$内部类名.class		引导讲解	**记录思考**
	测试8	**说出如下程序错误的原因：** ``` public class Outer2 { private int a=22; public class Inner{ private int b=99; public void print(){ System.out.println("outer的a="+a); System.out.println("inner 的b="+b); } } public void test(){ print(); } public static void main(String[] args){ Outer2 o=new Outer2(); o.test(); } } ```	外部类是不能直接使用内部类的成员和方法滴可先创建内部类的对象，然后通过内部类的对象来访问其成员变量和方法。	**引导提问**	**抢答**
	测试9	**请写出如下程序输出的结果：**	如果外部类和内部类具有相同的成员变量或方法，内部类默认访问自己的成员变量或方法，如果要访问外部类的成	**引导提问**	**抢答**

```
public class Outer3 {
    private int a=22;
    public class Inner{
        private int a=99;
        public  Inner(){
            System.out.println("a="+a);
            System.out.println("this.a="+this.a);
            System.out.println("Outer.this.a="+Outer3.this.a);
        }
    }
    public void test(){
        new Inner();
    }
    public static void main(String[] args){
        Outer3 o=new Outer3();
        o.test();
    }
}
```

员变量，可以使用 this 关键字。

十分钟测试

测试 1：用 GridLayout 实现如下界面。



窗口实现要求：

1.用 GridLayout，将 1-9 共 9 个按钮，放到窗口上。

2.用内部类实现事件监听器。

3.在 9 个按钮上注册事件监听器，每点击一个按钮，则在控制台打印出该按钮上显示的文字。

1.定义内部类

```
public class MyListerer implements ActionListener{
    public void actionPerformed(ActionEvent e) {
        System.out.println(e.getActionCommand());
    }
}
```

定义内部类MyListener，实现了监听器接口

输出按钮添加的标示。

2.生成内部类的对象

```
MyListerer listener=new MyListerer();
```

3.将内部类的对象注册到按钮上去

```
for(int i=0;i<b.length;i++){
    b[i]=new JButton(Integer.toString(i+1));
    c.add(b[i]);
    b[i].addActionListener(listener);
    b[i].setActionCommand("您点击的是第"+(i+1)+"个按钮！");
}
```

注册监听器

给每个按钮注册标示

新课导入

教学环节	教学内容	教师活动	学生活动
	我们在上一节课中，已经实现了游戏的界面，在本节课中，我们将把功能依次添加： 1.按下键盘上的某个按键，则相应的按钮高亮显示，键盘按键释放后，按钮恢复原来的颜色。 2.所按键盘的字母显示在 JtextArea 上。 3.给程序添加菜单，实现选择字体、字号、文本颜色以及清除所显示的文本等各种	提出问题	讨论 思考

功能。	分析	
学习过程 1.学习键盘监听器的使用 2.学习将菜单和菜单项添加到窗口上 3.学习设置字体和设置字体颜色 4.键盘测试程序实现	分析	讨论观看

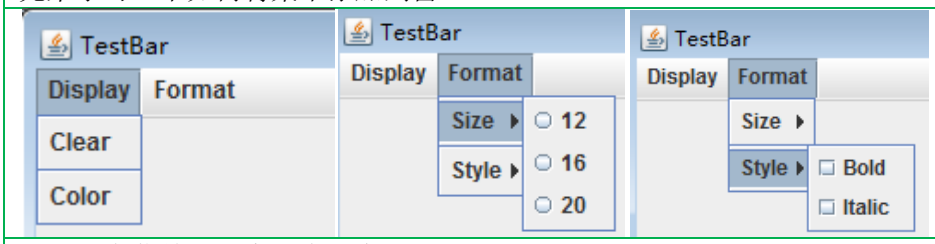
Step2：任务实施

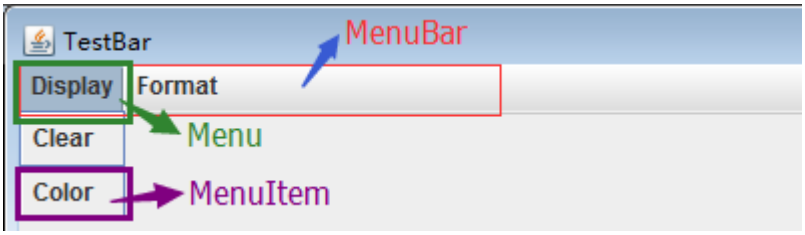
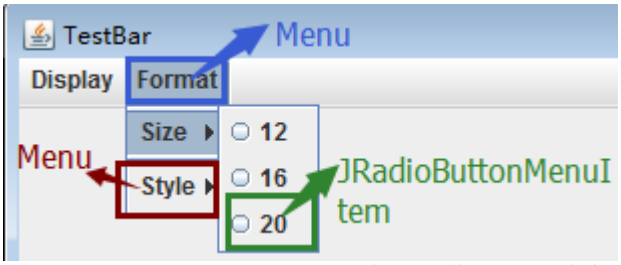
任务 1：（40 分钟）

教学环节	教学内容	教师活动	学生活动
任务引入	在构建自己的打字训练器程序前，首先让我们回顾一下此程序的功能： 当用户按下某个按键 　　将虚拟键盘上的某个相应的 JButton 显示为黄色 当用户释放此按键时 　　相应的 JButton 恢复原来颜色	布置任务	思考
任务部署	下面我们讲学习如何处理键盘事件。键盘事件是在按下或释放键盘上的某个按键的时候产生的一类事件，接口 KeyListener 针对键盘事件声明了 3 各处理程序：keyPressed，keyReleased、keyTyped。 keyPressed 事件处理程序是在按下某一按键时调用的， keyReleased 事件处理程序是在释放某一按键时调用的， keyTyped 事件处理程序是在按下然后释放某个字符集时调用的， 需要注意到是，当按下一个非字符按键时，如 shift 或 Backspace 键时，将调用 keyPressed 事件处理程序，而不会调用 keyTyped 事件处理程序。我们在本例中不会用到 keyTyped 事件处理程序，只需要实现相应的接口即可（即使用空语句体）。也就是实现接口中所有已经声明的方法。	讲解	思考听讲
	1.定义键盘监听器 `public class MyKeyListener implements KeyListener{` `  public void keyPressed(KeyEvent e) {` `  int n=e.getKeyCode();` 获得按键的asc码 `  b[n-96].setBackground(Color.yellow);` 被按下的按键显示为黄色 `  }` 按键0的asc码=96，一次类推 `  public void keyReleased(KeyEvent e) {` `  int n=e.getKeyCode();` `  b[n-96].setBackground(color);` 释放按键后，恢复原来的颜色。 `  }` `  public void keyTyped(KeyEvent e) {` `  }` `}` 2.初始化键盘监听器对象。 `MyKeyListener kl=new MyKeyListener();` 3.将键盘监听器添加到 JtextArea 上 `ta.setEditable(false);` `ta.addKeyListener(kl);` 注意： 1.如果只定义了监听器，但是监听器么有实例化，则监听器不会被触发。 2.如果不将监听器注册到一个组件上，则监听器不会被触发。		
课程	**知识点：**抽象类和接口含抽象方法不能创建对象，要利用子类或者实现类去生成对象；接口是一种规范，它只用来声明规则。面向接口编程具有相当大的灵活性。		

思政	**思政元素：**进行价值塑造与思想引领，以及人文精神培养。 **融入方式：** 1.提醒学生不要空想，一方面要树立远大理想，另一方面要为了实现理想去脚踏实地地努力奋斗，才能有所收获将来为国家做更大的贡献。 2.对于既定的标准与规则，每一位公民都要遵守。 3.标准及规范性与灵活性的辩证关系问题，可提升学生的人文精神素养。可引导学生在处理学习、生活等方面的事情时运用唯物辩证法思考问题：在大是大非面前讲规矩、讲纪律，在允许自由裁量范围内讲灵活性。		
任务实施1	1.学生根据案例实施一遍。	辅导	编程
	2.学生讲键盘监听器添加到keyBoard程序上。 3.一个程序里，可以定义多个内部类，并实例化多个内部类的对象。比如，可以在JtextArea上添加KeyListener，也可以在Jbutton上添加ActionListener，结合实际情况，说一下，这样做有什么好处？ 总结： 1. 内部类提供了更好的封装，可以把内部类隐藏在外部类之内，不允许同一个包中的其他类访问该类 2. 内部类的方法可以直接访问外部类的所有数据，包括私有的数据 3. 内部类所实现的功能使用外部类同样可以实现，只是有时使用内部类更方便	辅导	编程
	存在问题解析。		

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	已经实现了键盘监听，现在想用菜单实现字体设置的功能，怎么做？让我们先来学习一下如何将菜单添加到窗口上。	引入	思考
			
	设计一个菜单，对窗口中的字体进行设置。 1.Display 下有两个菜单项 Clear 和 Color 2.Format 下有 2 个子菜单，Size 和 Style 3.Size 有 3 个菜单项，用来设置字体的大小。 4.Style 有 3 个菜单项，用来设置字体的类型。		
任务部署	利用菜单可以将应用程序中的一些相关命令组合在一起。尽管不同应用程序中的菜单命令都不尽相同，但有一些菜单命令，如打开和保存命令，在大多应用程序中都是一样的。菜单是 GUI 设计中的一个重要部分，通过把多个命令组织在一起，使 GUI 看起来不太凌乱。但是菜单并非直接放在某些组件的内部，而是通常先将它们添加到一个菜单条中，菜单条一般位于一个应用程序界面的顶部。在这个例子中，我们将学习如何添加 JMenu，同 JMenu 控制 JTextArea 中的文本颜色、字号和字体，从而改进打字训练器应用程序。 步骤： 1.在应用程序中添加菜单条 JMenuBar	讲解总结	思考

	2.向菜单条中添加菜单 JMenu 3.向菜单中添加菜单项 JMenuItem 4.为 JMenuItem 添加事件响应		
任务实施	**（一）理解 Menu 的结构，MenuBar-Menu——MenyItem 的关系如下图所示：**  、 **（二）进一步理解菜单与子菜单的关系。菜单里可以包含子菜单，菜单最终包含菜单项。**  JradioButtonMenuItem，需要将一组单选按钮放在一个 ButtonGroup，这时每次只能有一个按钮被选中 **（三）具体步骤** 1.定义和初始化成员变量 `private JMenuBar mb=new JMenuBar();//定义菜单栏` `private JMenu displayM=new JMenu("Display");//定义Display菜单` `private JMenuItem clearMI=new JMenuItem("Clear");//定义菜单项` `private JMenuItem colorMI=new JMenuItem("Color");//定义菜单项` Display 菜单 `private JMenu formatM=new JMenu("Format");//定义Format菜单` `private JMenu styleM=new JMenu("Style");//定义styleM子菜单` `private JCheckBoxMenuItem[] styleMIs;//定义多选按钮类型的菜单项` fromat菜单 `private JMenu sizeM=new JMenu("Size");//定义Size子菜单` `private JRadioButtonMenuItem[] sizeMIs;//定义单选按钮类型的菜单项` `//需要将一组单选按钮放在一个ButtonGroup，这时每次只能有一个按钮被选中` `private ButtonGroup bg=new ButtonGroup();` size菜单 2.在应用程序中添加菜单条 JmenuBar `this.setJMenuBar(mb);` 3.向菜单条中添加菜单 Jmenu `mb.add(displayM);//将菜单添加到菜单栏上` `mb.add(formatM);//将菜单添加到菜单栏上` `displayM.add(clearMI);//将菜单项添加到菜单上` `displayM.addSeparator();` 分割线 `displayM.add(colorMI);//将菜单项添加到菜单上` `formatM.add(sizeM);//将子菜单添加到菜单上` `formatM.addSeparator();` `formatM.add(styleM);//将子菜单添加到菜单上` 4.向菜单中添加菜单项 JmenuItem		

```
for(int i=0;i<sizeNames.length;i++)
{
    //前面只是声明对象，在此生产并初始化对象
    sizeMIs[i]=new JRadioButtonMenuItem(sizeNames[i]);
    sizeM.add(sizeMIs[i]);//将菜单项添加到其菜单上
    bg.add(sizeMIs[i]);//将单选按钮添加到ButtonGroup上。
}

for(int i=0;i<styleNames.length;i++)
{
    styleMIs[i]=new JCheckBoxMenuItem(styleNames[i]);
    styleM.add(styleMIs[i]);
}
```

体会为什么用数组来方菜单项的内容。

5.为 JMenuItem 添加事件响应

任务实施	由学生仿照老师的讲解，在一个窗口上加上菜单。 1.如果构造方法非常长，能否将菜单部分的代码独立出来？ 2.思考：为什么字体大小和字体样式用数组标示？如果不用数组，行不行？		

任务 3：（45 分钟）

教学环节	教学内容	教师活动	学生活动
任务引入	 在窗口上添加 TextArea，里面显示一段文字，通过菜单，设置字体分别实现菜单项的功能。 1.设置字体的大小 2.设置字体的样式 3.设置字体的颜色。 深入了解两个类： 1.Font 类 2.Color 类	引入	思考
任务部署	1.将 TextArea 添加到窗口上，窗口使用 BorderLayout 布局管理器，ta 添加到 Center 部分。 （1）设置 ta 的换行方式 （2）设置 ta 的字体。	引导辅导	编程讨论
	2.定义事件监听器内部类，实现 clearMI，和 colorMI 的功能。 `public class Listener implements ActionListener{` `  public void actionPerformed(ActionEvent e) {` `  if(e.getSource()==clearMI){` `  ta.setText("");` `  }else if(e.getSource()==colorMI){` `  color=JColorChooser.showDialog(null,"请选择字体颜色", Color.green);` `  ta.setForeground(color);` `  }` `  }` `}` 因为这个类只实现了监听器接口，因此不能用this。 打开颜色选择器	编程讲解	听讲思考模仿
	3.定义事件监听器内部类，实现字体大小的设置		

	public class SizeListener implements ActionListener{ public void actionPerformed(ActionEvent e) { font=ta.getFont(); 保存原有字体 int size=Integer.parseInt(e.getActionCommand()); Font output=new Font(font.getName(),font.getStyle(),size); ta.setFont(output); } }		
	4.定义事件监听器内部类，实现字体样式的设置 public class StyleListener implements ActionListener{ public void actionPerformed(ActionEvent e) { font=ta.getFont(); 保存原有字体 int style=font.getStyle(); JCheckBoxMenuItem b=(JCheckBoxMenuItem)e.getSource(); 获得事件源，一定是一个JCheckBoxMenuItem类的对象 // "Bold","Italic" if(b.getText().equals("Bold")){ if(b.isSelected()){ style+=Font.BOLD; }else{ style-=Font.BOLD; } }else if(b.getText().equals("Italic")){ 如果加粗字体被选择，则设置为加粗，否则设置为正常非加粗样式。 if(b.isSelected()){ style+=Font.ITALIC; }else{ style-=Font.ITALIC; } } } }		
	5.修改 KeyBoard 代码，实现菜单的功能。		
	6.给菜单加上快捷键。 菜单. setMnemonic(KeyEvent.VK_F);可以设置快捷键，alt+F 键可以打开菜单。 菜单项. setMnemonic(KeyEvent.VK_F);可以设置快捷键，alt+F 键可以打开菜单项。 displayM.setMnemonic(KeyEvent.VK_D); formatM.setMnemonic(KeyEvent.VK_F); colorMI.setMnemonic(KeyEvent.VK_O);		
任务实施	1.学完成测试程序。 2.将功能整合到 KeyBoard 程序上去。 3.体会： 定义监听器的内部类，比在类上直接实现 ActionListener 接口要方便的多，代码也更加清晰。 进一步理解内部类的优点。	辅导	编程

Step3：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	1.什么是内部类？ 2.内部类和普通类有什么区别？ 3.使用内部类有什么优点？	总结	听讲记录
布置作业	1.整理并完成 KeyBoard 程序，进一步理解内部类的实现和应用。 2.预习：异常和异常处理，并完成如下任务：		

	（1）做一个除法运算，把除数设置为 0，看抛出什么样的异常。并处理。 （2）定义一个对象，不初始化，看抛出什么样的异常，并处理。 （3）定义一个数组 b，共有 10 个成员，访问数组成员 b[10]，看抛出什么样的异常，并处理。
课后学习资源	复习资料： 课上程序代码+项目文档
	预习资料： 1.《1-1 异常简介 (06:50)》（必学） 2.《1-2 使用 try..catch..finally 实现异常处理 (05:08)》（必学） 3.《1-3 通过案例学习 try...catch...finally (14:00)》（必学） 4.《1-5 Java 中的异常抛出以及自定义异常 (04:32)》（选学） 5.《1-7 Java 中的异常链 (06:09)》（选学）
课程思政	1.键盘布局在发展过程中遵循的人体工程学原理 2.键盘的工艺追求。