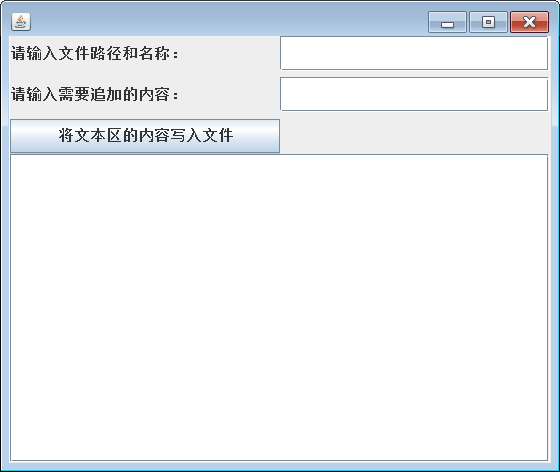
实验内容

1. 第一部分

编写一个Java源程序，其中包含：

编写一个包含主方法main的公共类（访问权限为public的类），该类继承自窗体类JFrame，并且，该类实现了接口ActionListener（实现接口ActionListener的方法actionPerformed）。

源程序实现的任务是实现如下图所示的图形用户界面（当窗口大小改变时，其中的组件的大小也会随之改变），并且完成以下要求：



（1）可以在“请输入文件路径和名称：”后面的文本框中输入文件名和路径。

（2）可以在文本区中写文本。

（3）可以在“请输入需要追加的内容：”后面的文本框中输入内容，按回车键之后，其中的内容会添加到文本区中（添加的内容另起一行）。

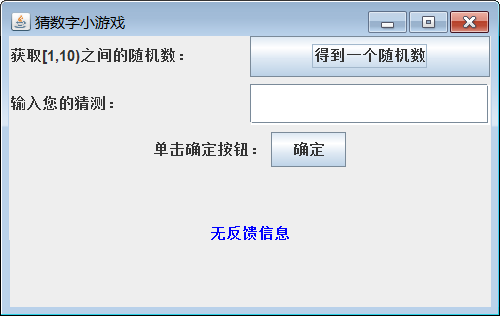
（4）单击“将文本区的内容写入文件”按钮，文本区中的内容就会被写到指定的文件中。

1. 第二部分

编写一个Java源程序，其中包含：

编写一个包含主方法main的公共类（访问权限为public的类），该类继承自窗体类JFrame，并且，该类实现了接口ActionListener（实现接口ActionListener的方法actionPerformed）。

源程序实现的任务是实现如下图所示的图形用户界面（当窗口大小改变时，第1行和第2行的组件的大小也会随之改变，其他组件的大小不发生改变），并且完成以下要求：



（1）窗体的标题是“猜数字小游戏”。

（2）单击“得到一个随机数”按钮，系统会产生一个1~10之间的随机整数（使用Math类的方法random( )）。

（3）在“输入您的猜测：”后面的文本框中输入你要猜测的整数，之后单击“确定”按钮。

（4）对你猜测的整数，图形用户界面的最下面会有提示信息，没有猜的时候，显示“无反馈信息”，如果猜大了，显示“猜大了”，如果猜小了，显示“猜小了”，如果猜对了，显示“猜对了”。（其中，蓝色字体是JLabel类的对象调用方法setForeground(Color.blue)实现的。）

实现方法

分为两部分，分别以主类的简介、图形界面介绍、事件响应说明和其中涉及的异常事件处理来阐述每一部分的实现思路。

1. 第一部分
2. Main 类：

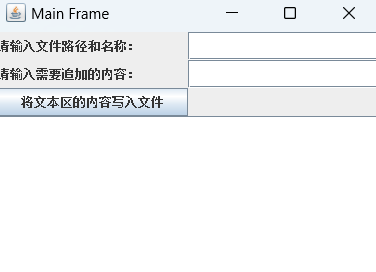
继承了JFrame 类，实现了 ActionListener 接口。

构造函数Main()中创建了窗口图形界面，在成员方法actionPerformed(ActionEvent e)中实现了按钮点击和文本输入的事件处理方法。

1. 图形用户界面设计：

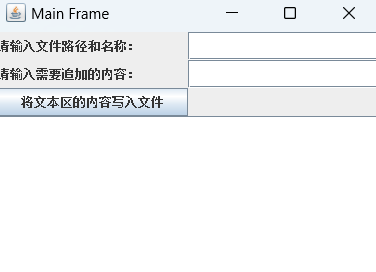
在Main（）构造函数中，setTitle("Main Frame")设置窗口的标题为 "Main Frame"，setLayout(new BorderLayout())使用 BorderLayout 进行整体布局，setSize(400,300)设置窗口大小为 400x300。

创建一个 inputPanel，使用网格布局管理器GridLayout(3, 2)，包含了文件路径和名称的标签和文本框，以及需要追加的内容标签、文本框和一个写入文件的按钮。将输入面板放置在窗口的北部（上部）。如图所示，组件按照三行两列的样式排列。



接下来创建一个 textArea 作为文本显示区域，通过滚动面板 JScrollPane 放置在窗口的中央，以支持显示大量文本。

将输入面板放在北部（上部），文本区放在中央,因为South没有组件，所以文本区自动侵占下方的South区域，如下图所示。



**North**

**Center & South**

1. 事件的响应

actionPerformed(ActionEvent e)来实现 ActionListener 接口来处理用户的输入事件。

监听了两个组件的事件appendStringField（回车键事件）和 writeToFileButton（按钮点击事件），实现了两种事件的处理，一种是从追加内容文本框中获取内容并追加到文本显示区域，另一种是将文本显示区域的内容写入到指定文件中，并处理可能出现的写入异常。

1. 追加内容文本框事件处理：

当事件源是 appendStringField（追加内容文本框）时，即用户按下了回车键时：从 appendStringField 获取用户输入的文本，将获取的文本追加到下面的文本显示区域中，并在结尾添加换行符。完成上述操作后清空 appendStringField，以便用户输入下一个内容。

1. 写入文件按钮事件处理：

当事件源是 writeToFileButton（写入文件按钮）时，表示用户点击了写入文件操作，此时做以下操作：从 filePathField 获取用户输入的文件路径，从 textArea 获取全部文本内容，将文本内容写入到指定路径的文件中。

如果写入成功，弹出提示框显示 "内容已写入文件！"；如果写入过程中出现异常（如文件路径错误等），捕获 IOException 异常，弹出带有错误信息的提示框。

4. 异常控制处理

使用 try-catch 块捕获了可能出现的异常，例如写入文件时可能发生的 IOException。

1. 第二部分
2. GuessNumberFrame类的实现

继承了 JFrame 类，实现了 ActionListener 接口。

构造函数GuessNumberFrame()中创建了窗口和 GUI 组件，成员方法actionPerformed(ActionEvent e)实现了按钮点击等事件的处理方法。

1. 窗口管理的应用

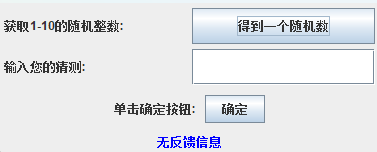
在构造函数GuessNumberFrame()中实现。

1. 图形用户界面设计：

整个JFrame采用GridBagConstraints布局，分别有：

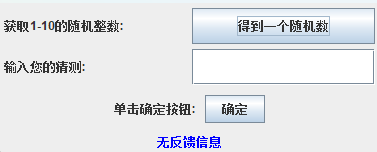
1. JPanel用来装载第一行和第二行：

采用GridLayout，在其中添加了四个组件：两个标签 (randomLabel 和 guessLabel)、一个按钮 (getIntBt) 和一个文本框 (guessTextField)。这些组件通过 GridLayout 进行布局，网格布局的行数为 2，列数为 2，并且各组件之间有 5 像素的水平和垂直间距。



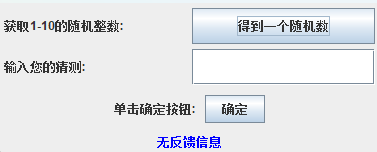
1. box来装载第三行：

先创建一个水平盒子，放置确定按钮和提示信息，再在 Box 内部添加 Glue 组件来实现水平居中。水平盒子box中放置了一个标签 (confirmLabel) 和一个按钮 (guessBt)，它们通过 Box.createHorizontalStrut(10) 进行了间隔设置。



1. 以及最后一行的feedbackLabel：

设置其颜色为蓝色。



1. GridBagConstraints的应用具体为如下：

用weightx和weighty设置组件在水平方向上拉伸和组件在垂直方向上拉伸，将JPanel的这两个参数设置为1，下面组件的这两个参数设置为0，便可以实现第一行和第二行在窗口大小变化时跟随变化大小，下面的组件大小不变的效果。

1. 窗口的显示：

setTitle来设置窗口标题为"猜数字小游戏"、setSize设置大小为(400, 200)。这些初始化和上面的布局都放在窗口类GuessNumberFrame的构造函数中，在主函数中调用构造函数生成新窗口就可以完成这个自定义窗口的创建。

1. 事件的响应

在actionPerformed成员方法中实现。

GuessNumberFrame类实现 ActionListener 接口来处理按钮点击等用户交互事件，通过 addActionListener() 方法将按钮与事件监听器关联，实现事件的捕获和处理。分别对不同事件的处理如下：

1. 获取随机数按钮的事件处理：

当事件源是 getIntBt（获取随机数按钮）时，生成一个 1 到 10 的随机数并将其赋值给 randomNumber 变量，同时将反馈标签 feedbackLabel 的文本内容设置为 "无反馈信息"，以重置反馈信息。

1. 猜测按钮的事件处理：

当事件源是 guessBt（猜测按钮）时，首先检查是否已经获取了随机数（randomNumber != 0）。

如果尚未获取随机数，则显示 "请先点击按钮获取随机数" 的提示信息；如果已经获取了随机数，则尝试从文本框 guessTextField 中获取用户猜测的数字，使用 Integer.parseInt() 方法将文本转换为整数。

如果转换成功，则根据猜测与随机数的大小关系，更新反馈信息，可能是 "猜大了"、"猜小了" 或 "猜对了"；如果转换失败（用户输入不是有效的整数），捕获 NumberFormatException 异常，并显示 "请输入有效整数！" 的提示信息。

4. 异常控制处理

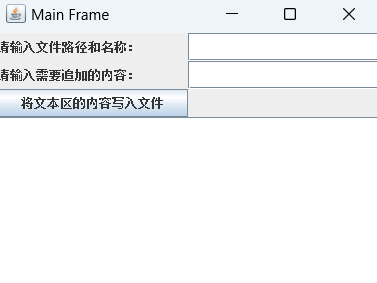
使用 try-catch 块捕获可能出现的异常，当用户不是有效的整数时，通过 NumberFormatException 来捕获非法输入异常，给出相应提示信息。

实验结果

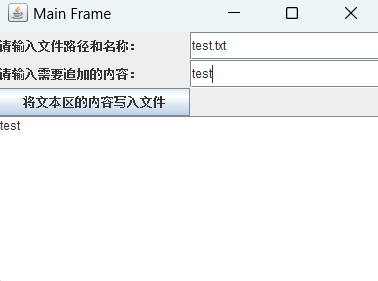
1. 第一部分

程序运行结果如下所示。

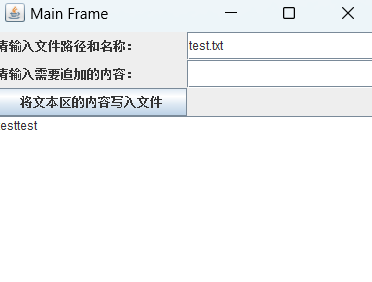
1. 运行后界面初始显示。



1. 向文本框中输入内容

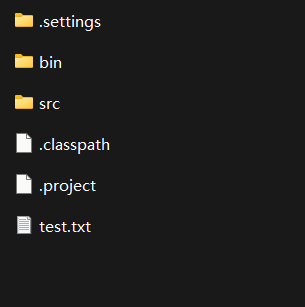


1. 回车键，追加内容

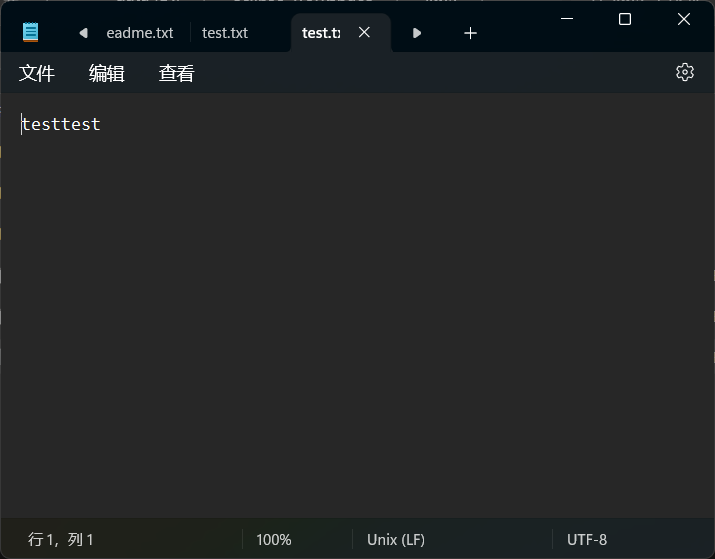


1. 点击按钮写入文件

在和src（源代码）同级的目录下出现新建的文件test.txt。



打开test.txt查看刚刚写入的内容。

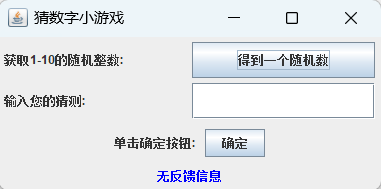


文件写入成功。

1. 第二部分

程序运行结果如下所示。

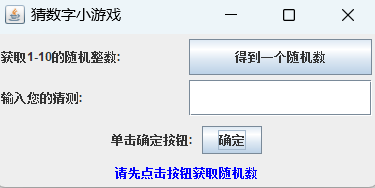
1. 运行后界面初始显示。



1. 当窗口大小改变时，第1行和第2行的组件的大小也会随之改变，其他组件的大小不发生改变。

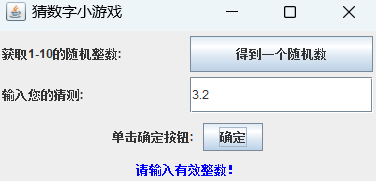


1. 异常输入：不生成随机数，直接点击确定，反馈区显示“请先点击按钮获取随机数”。

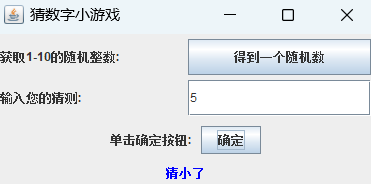


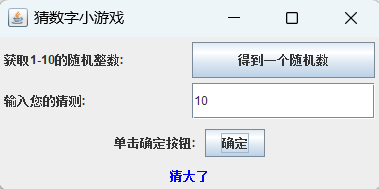
1. 点击得到一个随机数：

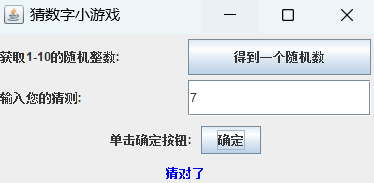
输入不是整数，提示“请输入有效整数”。



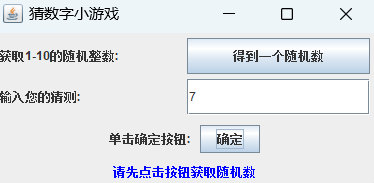
正常输入整数，反馈“猜小了”“猜大了”“猜对了”。



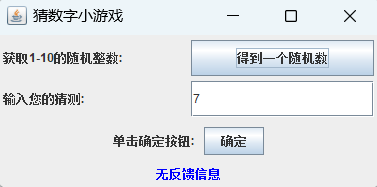




1. 猜对后重置随机数为0，等待下一轮游戏。若继续点击确定，提示“请先点击按钮获取随机数”。



1. 点击得到一个随机数后重置反馈信息为“无反馈信息”。



结论分析

对于java的图形界面，主要内容分为界面的布局和组件的使用，在界面布局上，选择合适的布局来实现理想的UI界面，比如要实现不同区域每个时候都占满整个窗口就可以采用BoarderLayout，或是要实现规整的分布或者是要实现规定行数和列数的时候就采用GridLayout网格布局。在此之外，可以采用中间窗口来完成复杂布局操作。对于组件的使用方面，主要是和事件响应函数相互配合使用，比如获得和改动文本框中的信息等。