

学习任务二 开发小游戏

子任务 2.1 实现游戏界面

1、任务引入：

这次的任务，我们要开发一个小游戏，大家在刚开始接触电脑的时候，为了练习打字，都玩过这个游戏，如下图，从天上掉下很多个字母，我们要在规定的时间内，敲中这些字母，如果敲中就得分，没有敲中就失去了加分的机会，最后根据分数可以了解我们的打字熟练程度。



这个界面很明显和我们接触到的第一个任务界面完全不同，上一个任务我们做的应用程序是基于文本的响应方式，所有和用户之间的互动要靠文字的输入和输出，显然是不方便的。而这个界面是我们熟悉的基于图形的用户界面，程序和用户之间的互动是靠图形组件的互动事件来实现的。

所以这个子任务，我们首先要学习如何构建一个图形化界面

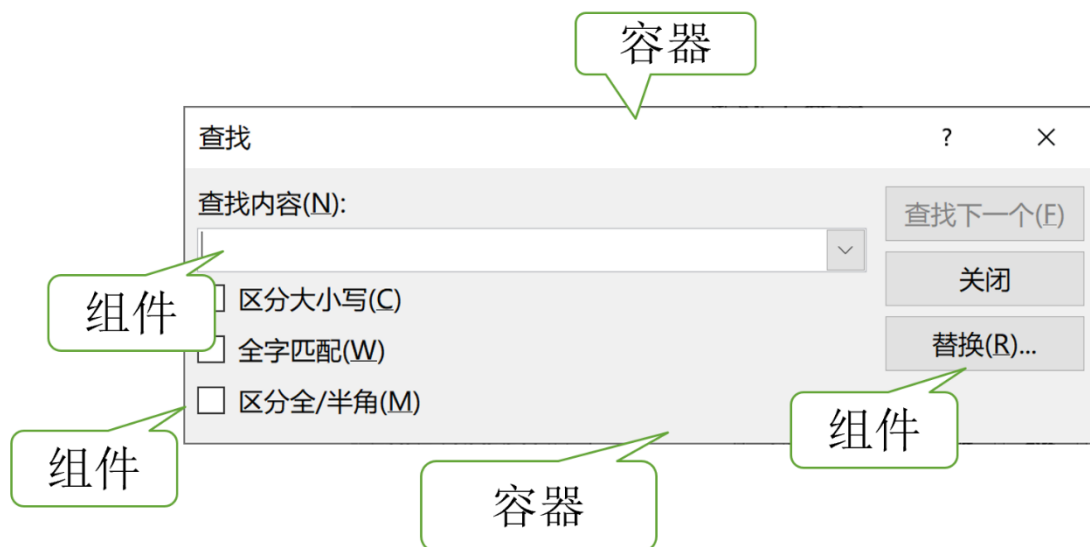
2、任务准备（微课学习内容）

1) 了解 Java 图像化界面实现的框架

Java 实现图形化界面的框架里有两个基本概念：容器和组件

- ✧ 容器---房子，柜子
- ✧ 组件---柜子，物件

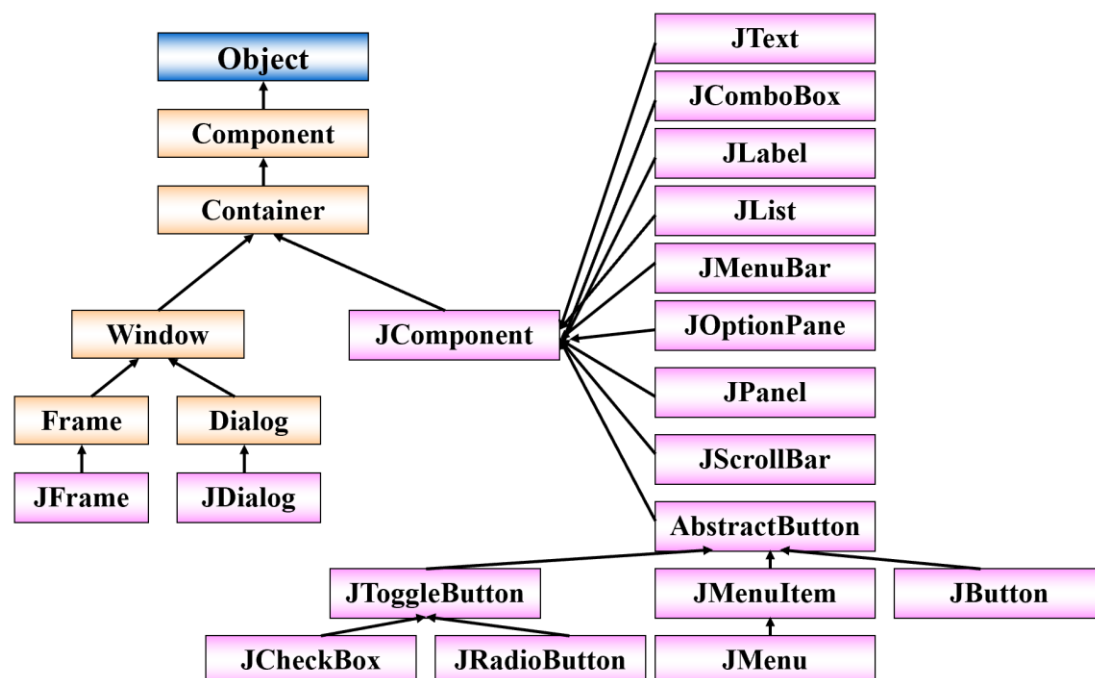
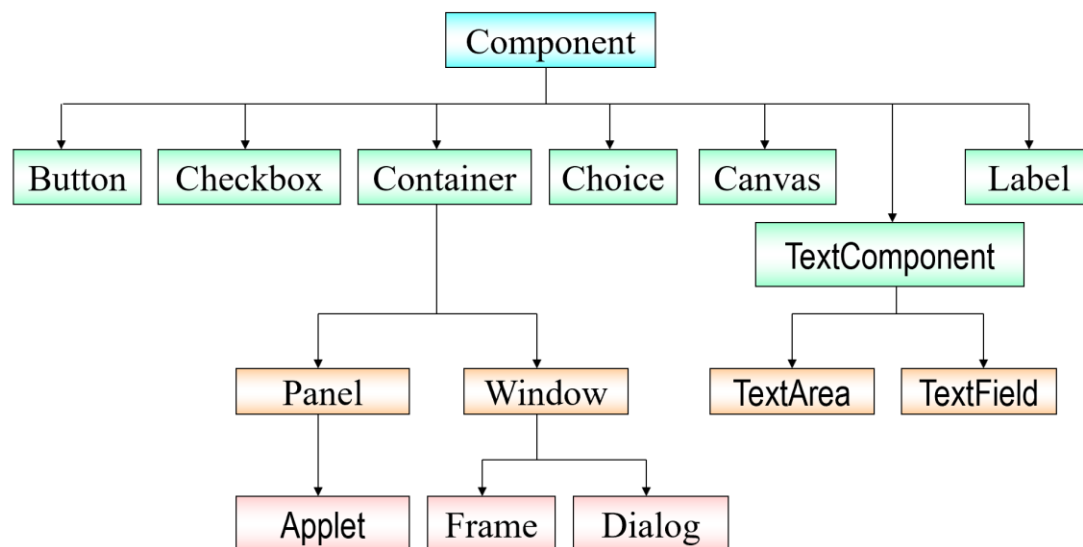
容器是我们看到的窗口和一些分区，组件是放入窗口和分区里的东西，例如按钮啊，文本框啊，我们做一个界面就是要确定，有几个容器，有哪些组件，然后把组件放到容器中去，如下图



组件有哪些呢？其实都是我们经常看到的，如下图：



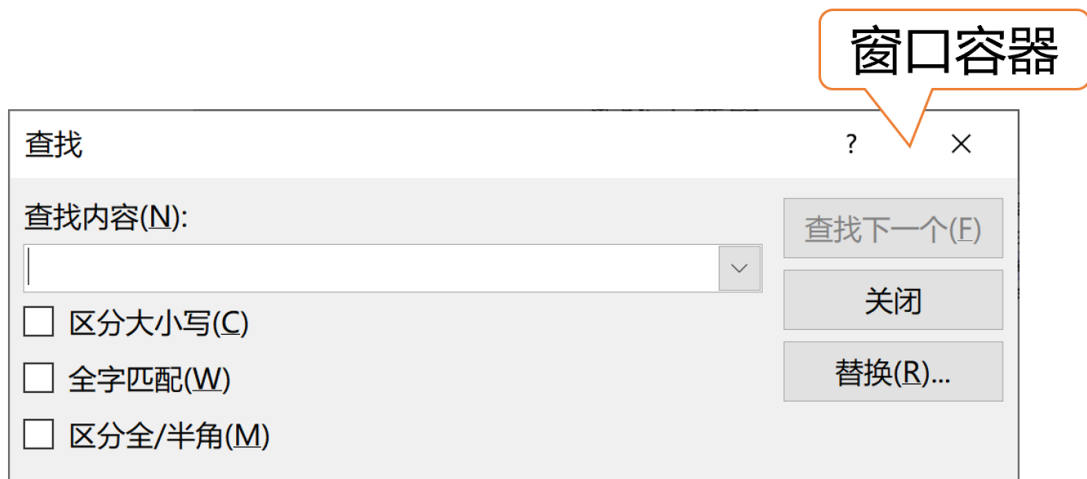
那么，根据我们学习 Java 的经验，Java 的类库中肯定也提供了现成的类型，我们只需要创建对象就可以轻松构建界面了。Java 提供哪些容器和组件类型呢？如下图，awt 和 swing 两大类型的框架，框架里的类，就是我们创建图形化界面的基础。



2) 学习创建容器

首先我们要学习如何创建容器，容器是可以存放组件的区域，可在容器上进行绘制和着色，如上图 java.awt 包中的 Container 类可直接或间接派生出两个常用容器：框架（Frame 类）和面板（Panel 类）

✧ 框架是一个带有边框的独立的窗口，如下图



✧ 面板是包含在窗口中的一个不带边框的区域，如下图

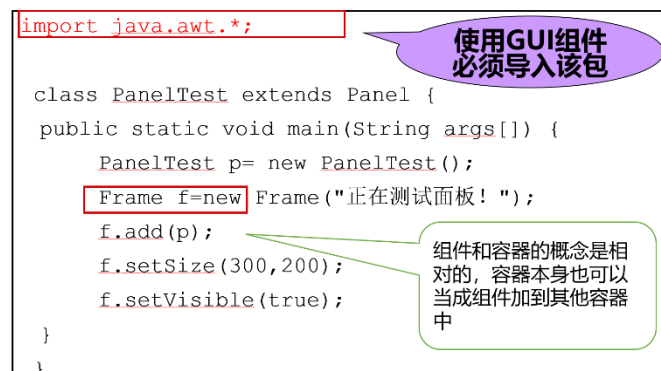


框架是独立于 Applet 和浏览器的一个窗口。可以通过以下任一构造函数来创建：

- ✧ `Frame()`: 创建一个不含标题的标准窗口
- ✧ `Frame(String Title)`: 创建一个含有标题的窗口，这个标题是由参数 `title` 指定的。

当一个 `Frame` 窗口被创建以后，需要调用 `setSize()` 方法来设置窗口的大小，并调用 `setVisible()` 来显示窗口。

【小练习】大家可以试一试这段代码，完成一个窗口的示例



3) 理解布局

会创建容器，就会创建组件，都是 new 一个对象。关键是，创建出来的组件在容器中如何摆放呢？Java 也提供好了布局管理的类型，我们只要创建相应的布局方式，应用到容器身上，组件就会按照预定的方式来摆放了。

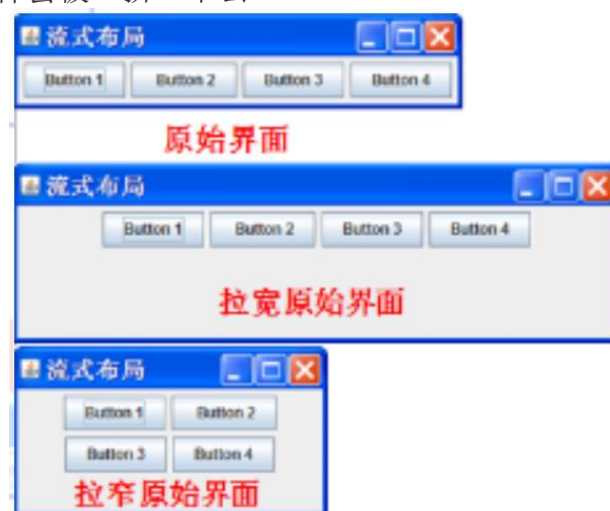
Java 中的基本布局类型包括以下几种：

- ✧ FlowLayout（流式布局）
- ✧ BorderLayout（边界布局）
- ✧ GridLayout（网格布局）
- ✧ GridBagLayout（网格包布局）
- ✧ CardLayout（卡片布局）

我们先看看 FlowLayout 如何工作，FlowLayout 的构造函数有：

- ✧ FlowLayout():生成一个默认的流式布局
- ✧ FlowLayout(int alignment):可以设定每一行组件的对齐方式
- ✧ FlowLayout(int alignment,int horz,int vert):可以设定组件间的水平和垂直距离

FlowLayout 设置在容器上之后，容器的组件会从左到右，从上到下排列，如果空间不够，组件会被“挤”下去



我们再看看 GridLayout 如何工作，FlowLayout 的构造函数有：

- ✧ GridLayout():生成一个单列的网格布局
- ✧ GridLayout(int row,int col):生成一个设定行数和列数的网格布局
- ✧ GridLayout(int row,int col,int horz,int vert): 可以设置组件之间的水平和垂直间隔

GridLayout 设置在容器上之后，容器被切成了行列式表格，组件就一个萝卜一个坑的放进去了，所以这种布局下，组件的相对位置是不会变的，如下图：

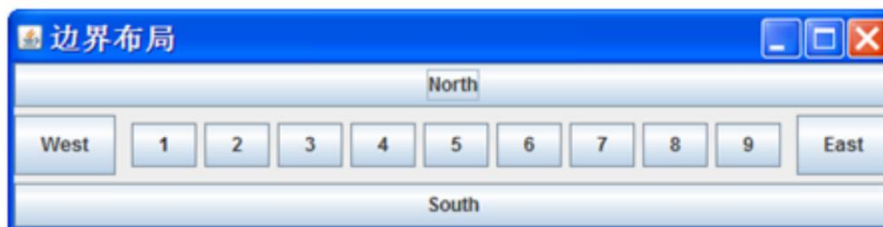


最后我们看看 BorderLayout 如何工作，BorderLayout 的构造函数有：

✧ BorderLayout():生成默认的边界布局

✧ BorderLayout(int horz,int vert): 可以设定组件间的水平和垂直距离

BorderLayout 设置在容器上之后，容器被切成了东南西北中五个区域，所以这种布局特别适合做应用程序的主界面布局，如下图：



程序初始界面

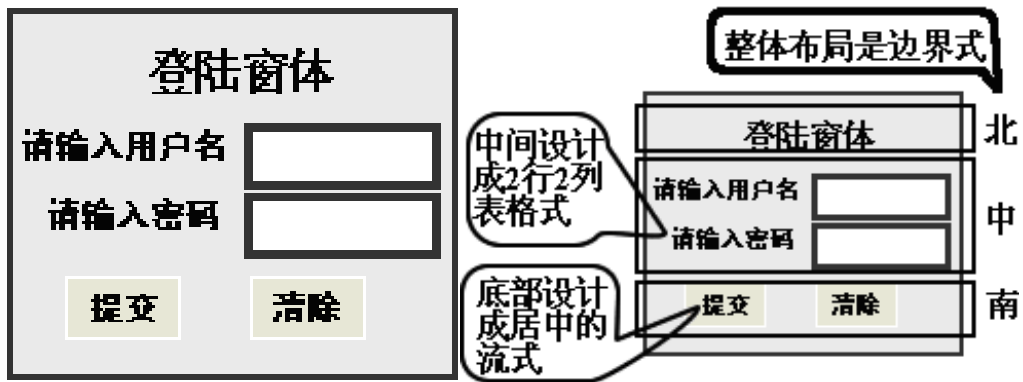


拉窄窗口后的界面

3、技能训练：

根据上面所学的内容，完成如下训练任务。

1) 完成一个登录窗口的设计，如下图



2) 完成一个通讯录管理界面



3) 完成打字母小游戏的界面设计任务（先不考虑字母哪里来的）

首先分析这个界面的布局采用何种方式



然后分析需要哪些容器和组件



最后完成代码的测试

