搭建Java桌面应用程序原型(二) PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/253/2021_2022__E6_90_AD_E5_BB_BAJava_c97_253059.htm 设置系统外观 下面的setSystemLookAndFeel()方法调用了Javax.swing.UIManager类的setLookAndFeel()方法： 它要求Swing从默认的Metal外观转换为本地外观：private void setSystemLookAndFeel() { try { UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName()). } catch (UnsupportedLookAndFeelException x) { log(x). } catch (ClassNotFoundException x) { log(x). } catch (IllegalACCESSException x) { log(x). } catch (InstantiationException x) { log(x). }} 通常，因为setLookAndFeel()参数都有一个可用值所以不会抛出异常。然而用标准日志API任何异常都可以作为严重错误信息被记录：private static void log(Exception x) { Logger.global.severe(x.getMessage()).} 原型安例中用全局日志是可以的，但是一个产品应该用它自己的日志，保存错误信息在文件中。 创建并显示主要窗口 createFrame()方法创建一个MainFrame实例，并且加载了图片：private void createFrame() { mainFrame = new MainFrame(). mainPanel = mainFrame.getMainPanel(). mainPanel.0updateSize(). mainFrame.pack(). loadImage().} 0updateSize()设置了由getMainPanel()获得的主要面板的合理大小。pack()方法使得主框架调整大小从而让主面板和应用程序工具栏调整到合适的大小。注意到getMainPanel()和0updateSize()方法

是MainFrame和MainPanel类实现的应用方法。pack()方法是从Java.awt.Window中继承下来的。 showFrame()方法显示应用程序的主框架并且调用主panel的requestFocus()方法。没有调用requestFocus()，焦点将被工具栏中是缩放下拉框获得，这个组件不是框架的主要组件。当应用程序开始的时候，它的主要组件应该获得焦点，即使主要panel没有处理任何键盘事件。 在窗口关闭的时候调用setDefaultCloseOperation()，禁用这个方法的默认动作而是传递DO_NOTHING_ON_CLOSE作为参数。showFrame()方法注册自己拥有的窗口监听器以便处理窗口关闭事件。当用户关闭主要框架，监听器保存一个做过注释的图片，释放框架所占用的资源并且用System.exit(0)结束应用程序的执行。private void showFrame() { mainFrame.setDefaultCloseOperation(MainFrame.DO_NOTHING_ON_CLOSE). mainFrame.addWindowListener(new WindowAdapter() { public void windowClosing(WindowEvent e) { saveImage(). mainFrame.dispose(). System.exit(0). } }). mainFrame.show(). mainPanel.requestFocus().} 加载和保存图片 一个完成的产品将用文件对话框去加载一个源图片并且保存一个注释过的图片。在观念上，“ 文件打开 ”对话框将让拥护预览图片，“ 文件保存 ”对话框将允许他们去提供不同的参数，例如保存图片的压缩质量。Swing的标准文件对话框是基于组件JFileChooser，这个组件能够通过setACCESSory()方法进行自定义，让你在文件对话框上加载你的组件。 在原型安例中，注意力应该在主要功能上。因此，原型通过从命令行获得加载和保存路径代替用自定义的文件对话框

。Javax.imageio.ImageIO类简单的read()和write()方法被用于加载和保存图片。注意，Image IO API让你知道哪种图象格式是支持的，并且你能设置例如压缩质量的参数。对于自定义文件对话框也将需要这些性质。 loadImage()方法读取一个图片文件，路径是由命令行第一个参数提供的，并且设置主要panel的背景图片：private void loadImage() {if (args.length >= 1) try {File file = new File(args[0]).BufferedImage image = ImageIO.read(file).mainPanel.getPaintView().getModel().setBackImage(image).} catch (IOException x) {log(x).}} saveImage()方法获得一个主要panel的注释过的图片，并且把这个图片保存到一个文件里，路径是由命令行提供的第二个参数给出的：private void saveImage() { if (args.length >= 2) try { File file = new File(args[1]). String name = file.getName(). int k = name.lastIndexOf(.) 1. String ext = name.substring(k). BufferedImage image= mainPanel.getAnnotatedImage(). ImageIO.write(image, ext, file). } catch (IOException x) { log(x). }}