ＪＡＶＡ进阶：Java打印程序设计全攻略 PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/144/2021_2022__EF_BC_AA_EF_BC_A1_EF_BC_B6_EF_c104_144646.htm 前言 在我们的实际工作中，经常需要实现打印功能。但由于历史原因，Java提供的打印功能一直都比较弱。实际上最初的jdk根本不支持打印，直到jdk1.1才引入了很轻量的打印支持。所以，在以前用Java/Applet/JSP/Servlet设计的程序中，较复杂的打印都是通过调用ActiveX/OCX控件或者VB/VC程序来实现的，非常麻烦。实际上，SUN公司也一直致力于Java打印功能的完善，而Java2平台则终于有了一个健壮的打印模式的开端，该打印模式与Java2D图形包充分结合成一体。更令人鼓舞的是，新发布的jdk1.4则提供了一套完整的"Java 打印服务 API" (Java Print Service API)，它对已有的打印功能是积极的补充。利用它，我们可以实现大部分实际应用需求，包括打印文字、图形、文件及打印预览等等。本文将通过一个具体的程序实例来说明如何设计Java打印程序以实现这些功能，并对不同版本的实现方法进行分析比较,希望大家能从中获取一些有益的提示。 Java中的打印 1、Java的打印API Java的打印API主要存在于java.awt.print包中。而jdk1.4新增的类则主要存在于javax.print包及其相应的子包javax.print.event和javax.print.attribute中。其中javax.print包中主要包含打印服务的相关类，而javax.print.event则包含打印事件的相关定义，javax.print.attribute则包括打印服务的可用属性列表等。 2、如何实现打印 要产生一个打印，至少需要考虑两条： 需要一个打印服务对象。这可通过三种方式实现：在jdk1.4之前的版

本，必须要实现java.awt.print.Printable接口或通过Toolkit.getDefaultToolkit().getPrintJob来获取打印服务对象；在jdk1.4中则可以通过javax.print.PrintSerivceLookup来查找定位一个打印服务对象。 需要开始一个打印工作。这也有几种实现方法：在jdk1.4之前可以通过java.awt.print.PrintJob(jdk1.1提供的，现在已经很少用了)调用print或printAll方法开始打印工作；也可以通过java.awt.print.PrinterJob的printDialog显示打印对话框，然后通过print方法开始打印；在jdk1.4中则可以通过javax.print.ServiceUI的printDialog显示打印对话框，然后调用print方法开始一个打印工作。 3、打印机对话框 3.1 Printable的打印对话框 开始打印工作之前，可以通过PrinterJob.printDialog来显示一个打印对话框。它给用户一个机会以选择应该打印的页码范围，并可供用户改变打印设置。它是一个本地对话框。 事实上，当从一个Printable对象进行一个打印工作时，打印对象并不知道需要打印多少页。它只是不停地调用print方法。只要print方法返回Printable.PAGE_EXISTS值，打印工作就不停地产生打印页，直到print方法返回Printable.NO_SUCH_PAGE时，打印工作才停止。 由于打印工作只有在打印完成后才进行准确的页数计算，所以在对话框上的页码范围是尚未初始化的[1,9999]。我们可以通过构建一个java.awt.print.Book对象传递给打印对象；也可以通过指定的格式计算需要打印的页数并传递给打印对象，使其准确地知道要打印多少页。 3.2 ServiceUI的打印对话框 与Printable的对话框不同的是，在jdk1.4提供ServiceUI的打印机对话框的缺省行为已经用新的 API 更改了：缺省情况下对话框不显示。我们必须使用ServiceUI类调用printDialog方

法创建如下所示的打印对话框。Java打印程序设计实例 1、打印文本 1.1 应用场景 假设我们需要打印一个窗体的某个文本编辑域(可能只有几行，也可能包含多页)的内容，并且每页最多打印54行，如何实现呢？ 1.2 解决方法 基本思路如下：首先我们需要实现Printable接口，然后按照每页最多54行的格式计算共需要打印多少页，当打印文本的按钮被点击时，执行相应的打印动作。打印文本的具体操作可通过Graphics2D的drawString方法来实现。 1) 实现Printable接口/*Graphic指明打印的图形环境；PageFormat指明打印页格式(页面大小以点为计量单位，1点为1英寸的1/72，1英寸为25.4毫米。A4纸大致为595×842点)；page指明页号*/public int print(Graphics g, PageFormat pf, int page) throws PrinterException { Graphics2D g2 = (Graphics2D)g. g2.setPaint(Color.black). //设置打印颜色为黑色 if (page >= PAGES) //当打印页号大于需要打印的总页数时，打印工作结束 return Printable.NO_SUCH_PAGE. g2.translate(pf.getImageableX(), pf.getImageableY()).//转换坐标，确定打印边界 drawCurrentPageText(g2, pf, page). //打印当前页文本 return Printable.PAGE_EXISTS. //存在打印页时，继续打印工作}/*打印指定页号的具体文本内容*/private void drawCurrentPageText(Graphics2D g2, PageFormat pf, int page) { String s = getDrawText(printStr)[page]. //获取当前页的待打印文本内容 //获取默认字体及相应的尺寸 FontRenderContext context = g2.getFontRenderContext(). Font f = area.getFont(). String drawText. float ascent = 16. //给定字符点阵 int k, i = f.getSize(), lines = 0. while(s.length() > 0 amp. lines { k = s.indexOf(\). //获取每一个回车符的位置 if (k != -1) //存在回车

符 { lines = 1. //计算行数 drawText = s.substring(0, k). //获取每一行文本 g2.drawString(drawText, 0, ascent). //具体打印每一行文本，同时走纸移位 if (s.substring(k 1).length() > 0) { s = s.substring(k 1). //截取尚未打印的文本 ascent = i. } } else //不存在回车符 { lines = 1. //计算行数 drawText = s. //获取每一行文本 g2.drawString(drawText, 0, ascent). //具体打印每一行文本，同时走纸移位 s = "". //文本已结束 } }} 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com