

Java Printing

Es gibt verschiedene APIs zum Drucken

Zentrales Objekt ist ein Objekt der Klasse `PrintJob`, das man erhält über

```
getPrintJob(Frame frame, String jobtitle, Properties props);
```

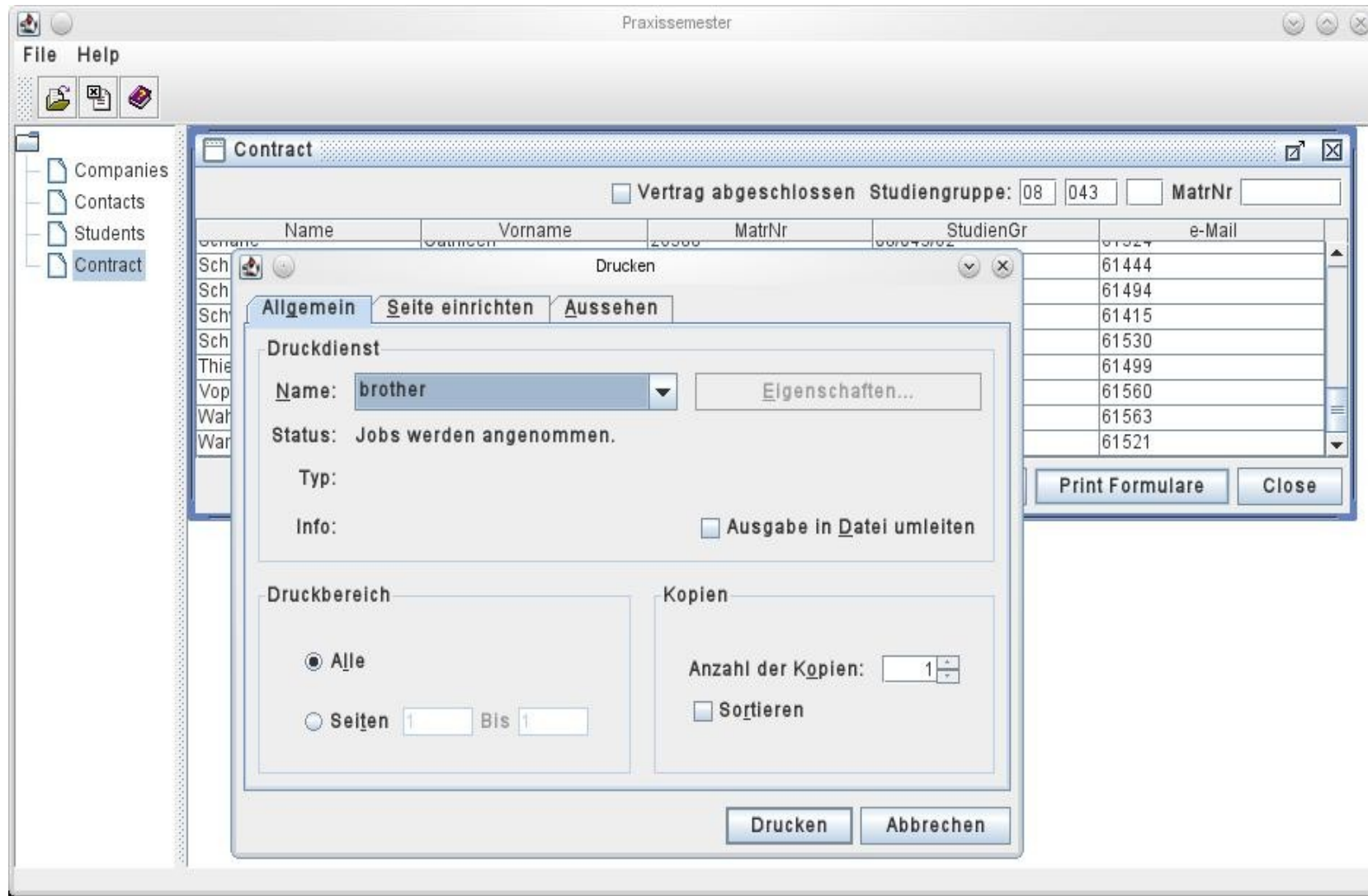
`getPrintJob` öffnet einen plattformabhängigen `PrintDialog`.

Ausgabe auf den Drucker erfolgt in der Auflösung des Bildschirms, die hier vorgestellte API ist somit nicht für hochauflösende Druckaufgaben geeignet.

Alle Ausgaben erfolgen über ein `Graphics` Objekt.

PrintJob

```
Properties Props=new Properties();  
Props.setProperty("awt.print.paperSize", "A4");  
PrintJob Pj=Toolkit.getDefaultToolkit().  
    getPrintJob(theBoss, "Drucken", Props);  
if (Pj!=null)  
{  
    . . .  
}
```



Vorbereitung

```
int ScrRes=Toolkit.getDefaultToolkit()  
                .getScreenResolution();  
  
int TopMargin =ScrRes;  
int LeftMargin=ScrRes/2;  
Dimension PageSize=new  
Dimension(Pj.getPageDimension());  
PageSize.width -=2*LeftMargin;  
PageSize.height-=2*TopMargin;  
Graphics g=null;  
FontMetrics Fm=null;  
int fontHeight=0;  
int fontAscent=0;  
int Line=0;  
int Col2=0;  
int MaxLine=0;  
Font theFont=new Font("SanSerif",Font.PLAIN,12);
```

Graphics vorbereiten



Graphics vom
PrintJob!

```
if (g==null)
  g=Pj.getGraphics();
  g.translate(LeftMargin,TopMargin);
  g.clipRect(0,0,PageSize.width,PageSize.height);
  g.setFont(theFont);
  Fm=g.getFontMetrics();
  fontHeight=Fm.getHeight();
  fontAscent=Fm.getAscent();
  Line= fontAscent;
  Col2=0;
  MaxLine=PageSize.height - fontHeight;
  Line=fontAscent;
}
```

Druckausgabe

```
g.drawString("HTW Dresden (FH)",0,Line);
GregorianCalendar Cal=new GregorianCalendar();
String Datum="" +Cal.get(Calendar.DAY_OF_MONTH)+'.'
              +(Cal.get(Calendar.MONTH)+1)+'.'
              +Cal.get(Calendar.YEAR);
System.out.println(Datum);
g.drawString(Datum,PageSize.width-Fm.stringWidth(Datum),Line);
Line+=fontHeight;
g.drawString("Fachbereich Informatik/Mathematik",0,Line);
Line+=fontHeight;
g.drawString("- Der Praktikumsverantwortliche -",0,Line);
Line+=fontHeight;
g.drawLine(0,Line,PageSize.width,Line);
Line+=fontHeight*3;
Font Ueberschrift=new Font("SanSerif",Font.PLAIN,16);
String S1="Meldung an das Prüfungsamt";
String S2="Erfolgreiches Praxissemester";
g.setFont(Ueberschrift);
FontMetrics FmUeb=g.getFontMetrics();
int PosX1=(PageSize.width-FmUeb.stringWidth(S1))/2;
g.drawString(S1,PosX1,Line);
Line+=FmUeb.getHeight();
```

„Pixelzeile“ muss zu jeder Zeile ausgerechnet werden.

zentrierte Ausgabe ausgerechnet

Seitenende

```
g.dispose();  
g=null;  
...  
}  
Pj.end(); // Ende des Druckens
```

Ausgabe der Seite erfolgt durch dispose.

Danach muss das Graphicsobjekt zerstört werden.

Für jede neue Druckseite wird ein neues Graphicsobjekt benötigt.

Resultat:

HTW Dresden (FH)
Fachbereich Informatik/Mathematik
– Der Praktikumsverantwortliche –

31.1.2011

Meldung an das Prüfungsamt Erfolgreiches Praxissemester

Student:

Andreas, Daniel

Matrikel Nummer:

205 1474

Studiengruppe:

020204/0202

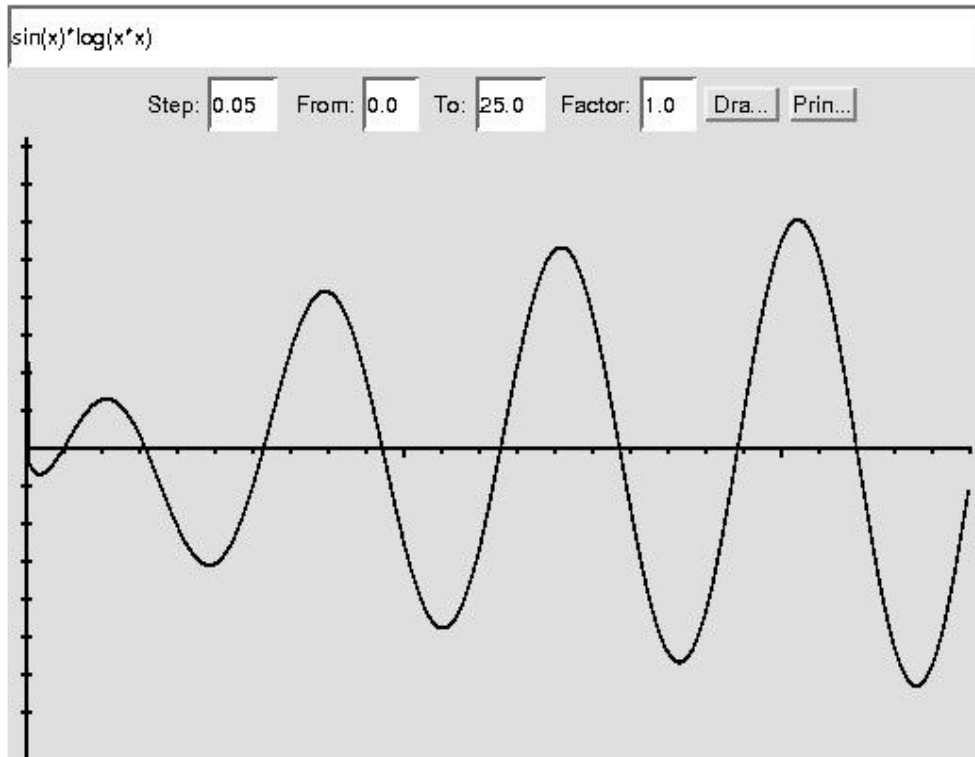
Prof. A. Beck
– Der Praktikumsverantwortliche –

Drucken einer Component

```
void doPrint()  
{  
    Properties Props=new Properties();  
    Props.setProperty("awt.print.paperSize","A4");  
    PrintJob Pj=Toolkit.getDefaultToolkit()  
        .getPrintJob((Frame)getParent(),"Print", Props);  
    if (Pj!=null)  
    {  
        Graphics g=Pj.getGraphics();  
        if (g!=null)  
        {  
            int ScrRes=Toolkit.getDefaultToolkit().getScreenResolution();  
            int TopMargin =ScrRes;  
            int LeftMargin=ScrRes/2;  
            g.translate(LeftMargin,TopMargin);  
            Panel C=this;  
            C.printAll(g);  
            g.dispose();  
        }  
        else System.out.println("G=null");  
        Pj.end();  
    }  
    else System.out.println("Pj=null");  
}
```

Verschiebung des Ursprunges
ist wegen des Druckrandes nötig

Zum Drucken wird intern die
Paintmethode der
Component
verwendet.
Hier printAll, da print nur die
Graphcomponent druckt.



Das Ergebnis