23.6.2 Drucken – Bilder

Das vorgestellte Projekt beschäftigt sich mit der Anzeige (Vorschau) und dem Ausdruck von Bildern, die in einer Bild-Datei gespeichert sind. In diesem Projekt werden viele der Eigenschaften, Methoden und Ereignisse der Klasse *Printer* eingesetzt, was untrennbar mit der Verwendung der Klassen Paint und Draw verbunden ist. Das Programm druckt Bilder aus einer Bilddatei in Farbe oder in Graustufen. Die Ränder für den linken (20mm) und oben Bildrand sind gegenwärtig fest vorgegeben und die Bild-Größen werden nur dann für den Druck auf ein A4-Blatt (Standard) skaliert, wenn das erforderlich ist. Da es für das Drucken von Bildern unter Linux ausgereifte Programme gibt, dient das vorgestellte Projekt vor allem dazu, Ihnen geeignete Prozeduren für das Drucken von Bildern aus einer Gambas-Anwendung heraus vorzustellen. Das Programm orientiert sich am Artikel 'How To Print' auf der Website *http://gambasdoc.org/help/howto/print* und setzt die dort gegebenen Hinweise um.

Abbildung 23.6.2.1: Oberfläche des Druckprogramms – Bildvorschau

Das Ausdrucken erfolgt entweder sofort auf dem ausgewählten Drucker oder in eine Datei – wahlweise im PDF-Format oder im PS-Format, wobei das Drucken in eine PostScript-Datei vorteilhaft ist:

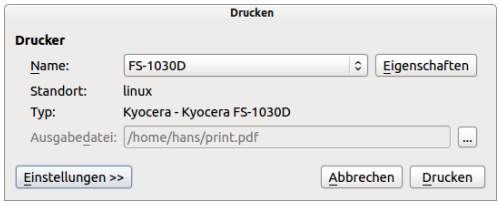

Abbildung 23.6.2.2: Ausdruck auf den Drucker FS-1030D

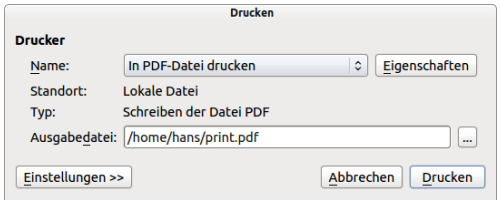

Abbildung 23.6.2.3: Ausdruck in eine PDF-Datei

Der Quelltext wird vollständig angegeben und anschließend für ausgewählte Abschnitte kommentiert, um die internen Kommentare zu ergänzen:

```
[1]  ' Gambas class file
[2]
[3]  Private imgUploaded As Image
[4]  Private fPaperNonPrintingLeft As Float
[5]  Private fPaperNonPrintingTop As Float
```

```
[6]  Private fDesktopResolutionProInch As Float = (Desktop.Resolution / 25.4)
[7]
[8]  Public Sub Form_Open()
[9]
[10]   FMain.Center
[11]   FMain.Arrangement = Arrange.Vertical
[12]   HBox.Spacing = True
[13]   btnOpenImage.AutoResize = True
[14]   btnOpenImage.Expand = False
[15]   btnPrintImage.AutoResize = True
[16]   btnPrintImage.Expand = False
[17]   btnPrintImage.Enabled = False
[18]   lblImageResolution.AutoResize = True
[19]   lblImageResolution.Expand = False
[20]   lblImageResolution.Visible = False
[21]   pSpace.Expand = True ' variabler Zwischenraum
[22]   ScrollArea1.Background = &HEFFFDF&
[23]   ScrollArea1.Expand = True
[24]   togbtnColorModus.Picture = Picture["icon:/16/fill"]
[25]   togbtnColorModus.Tooltip = "FarbModus: Farbdruck"
[26]   printerImage.GrayScale = False
[27]   printerImage.FullPage = False ' Default
[28]
[29] ' Für Testzwecke günstig und erprobt
[30]   printerImage.OutputFile = User.Home &/ "img_print.pdf"
[31]
[32]   GetNonPrintableArea()
[33]
[34] End ' Form_Open()
[35]
[36] Public Sub btnOpenImage_Click()
[37]
[38]   Dialog.Title = "Wählen Sie ein Bild aus..."
[39]   Dialog.Filter = ["*.png;*.jpg;*.jpeg;*.gif;*.xpm", "Bild-Dateien: "]
[40]   If Dialog.OpenFile() Then Return
[41]
[42]   Try imgUploaded = Image.Load(Dialog.Path)
[43]   If Error Then
[44]      Message.Error("Fehler beim Öffnen der Bild-Datei!")
[45]      Return
[46]   Endif ' ERROR?
[47]
[48]   ScrollArea1.ResizeContents(imgUploaded.W, imgUploaded.H)
[49]   ScrollArea1.Refresh
[50]   Wait
[51]   btnPrintImage.Enabled = True
[52]   Label1.Visible = True
[53]   Label1.Text = "Bildgröße: " & imgUploaded.W & " x " & imgUploaded.H & " Pixel"
[54]
[55] End ' btnOpenImage
[56]
[57] Public Sub ScrollArea1_Draw()
[58]   If imgUploaded Then Draw.Image(imgUploaded, - ScrollArea1.ScrollX, - ScrollArea1.ScrollY)
[59] End ' ScrollArea1_Draw()
[60]
[61] Public Sub printerImage_Draw()
[62]
[63]   Dim iMarginLeft, iMarginTop As Float
[64]   Dim iDruckBreite, iDruckHoehe, iDruckrandLinks, iDruckrandOben As Integer
[65]   Dim imgPrint As Image
[66]
[67] ' Festlegung der Druckränder links und oben mit festen Werten
[68]   iDruckrandLinks = 20 ' Millimeter
[69]   iDruckrandOben = 15 ' Millimeter
[70]
[71]   If Not imgUploaded Then Return
[72]   imgPrint = imgUploaded
[73]
[74] ' Umschaltung von Querformat auf Hochformat, wenn Bild.W > Bild.H
[75]   If imgPrint.Width > imgPrint.Height Then imgPrint = imgPrint.Rotate(Pi(0.5))
[76]
[77] ' Umstellung von Einheit Punkt auf Einheit Millimeter
[78]   Paint.Scale(Paint.Width / printerImage.PaperWidth, Paint.Height / printerImage.PaperHeight)
[79]
[80]   iMarginLeft = iDruckrandLinks - fPaperNonPrintingLeft ' Einheit Millimeter
[81]   iMarginTop = iDruckrandOben - fPaperNonPrintingTop ' Einheit Millimeter
[82]
[83] ' Die benutzten Werte 170mm (Weite) und 260mm (Höhe) sind erprobt
[84]   If PixelToMillimeter(imgPrint.W) > 170 Then
[85]      iDruckBreite = 170
[86]      iDruckHoehe = CInt(iDruckBreite * (imgPrint.Height / imgPrint.Width))
[87]      If iDruckHoehe > 260 Then
[88]         iDruckHoehe = 260
[89]         iDruckBreite = CInt(iDruckHoehe / (imgPrint.Height / imgPrint.Width))
```

```
[90]      Endif ' iDruckHoehe > 260?
[91]      Paint.DrawImage(imgPrint, iMarginLeft, iMarginTop, iDruckBreite, iDruckHoehe)
[92]      Return
[93]    Endif ' PixelToMillimeter(imgPrint.W) > 170?
[94]
[95]    If PixelToMillimeter(imgPrint.H) > 260 Then
[96]      iDruckHoehe = 260
[97]      iDruckBreite = CInt(iDruckHoehe / (imgPrint.Height / imgPrint.Width))
[98]      If iDruckBreite > 170 Then
[99]         iDruckBreite = 170
[100]          iDruckHoehe = CInt(iDruckBreite / (imgPrint.Height / imgPrint.Width))
[101]       Endif ' iDruckBreite > 170?
[102]       Paint.DrawImage(imgPrint, iMarginLeft, iMarginTop, iDruckBreite, iDruckHoehe)
[103]       Return
[104]     Endif ' PixelToMillimeter(imgPrint.H) > 260?
[105]
[106]     iDruckBreite = PixelToMillimeter(imgPrint.W)
[107]     iDruckHoehe = CInt(iDruckBreite * (imgPrint.Height / imgPrint.Width))
[108]
[109]     Paint.DrawImage(imgPrint, iMarginLeft, iMarginTop, iDruckBreite, iDruckHoehe)
[110]
[111]   End '  printerImage_Draw()
[112]
[113]   Public Sub btnPrintImage_Click()
[114]
[115]     If printerImage.Configure() Then Return
[116]
[117]     Me.Enabled = False          ' Das Formular wird deaktiviert
[118]        Inc Application.Busy     ' Das Programm nimmt keine Eingaben mehr entgegen ...
[119]            printerImage.Print   ' Der Druck wird gestartet
[120]        Dec Application.Busy     ' Das Programm nimmt wieder Eingaben entgegen...
[121]     Me.Enabled = True           ' Das Formular wird aktiviert
[122]
[123]   End ' btnPrintImage_Click()
[124]
[125]   Public Sub togbtnColorModus_Click()
[126]
[127]     If togbtnColorModus.Picture = Picture["icon:/16/fill"] Then
[128]        togbtnColorModus.Picture = Picture["icon:/16/properties"]
[129]        togbtnColorModus.Tooltip = "FarbModus: Graustufen"
[130]        printerImage.GrayScale = True
[131]     Else
[132]        togbtnColorModus.Tooltip = "FarbModus: Farbe"
[133]        togbtnColorModus.Picture = Picture["icon:/16/fill"]
[134]        printerImage.GrayScale = False
[135]     Endif
[136]
[137]   End ' togbtnColorModus_Click()
[138]
[139]   Public Sub GetNonPrintableArea()
[140]
[141]     Dim sResultPath, sResultRow, sResultString As String
[142]     Dim aMatrix As String[]
[143]     Dim iPosition As Integer
[144]
[145]     Shell "locate -b *.ppd | grep " & printerImage.Name To sResultPath
[146]     Shell "grep '*ImageableArea A4/A4' " & sResultPath To sResultRow
[147]
[148]     iPosition = InStr(sResultRow, Chr(34))
[149]     sResultString = String.Mid(sResultRow, iPosition + 1, -2)
[150]     aMatrix = Split(sResultString, Chr(32))
[151]   ' nicht bedruckbarer oberer Papierrand
[152]     fPaperNonPrintingTop = Val(aMatrix[1]) * (25.4 / 72)
[153]   ' nicht bedruckbarer linker Papierrand
[154]     fPaperNonPrintingLeft = Val(aMatrix[0]) * (25.4 / 72)
[155]
[156]   End ' GetNonPrintableArea()
[157]
[158]   Public Function PixelToMillimeter(iPixel As Integer) As Float
[159]     Return iPixel / fDesktopResolutionProInch
[160]   End ' PixelToMillimeter(iPixel As Integer) As Float
[161]
```

23.6.2.1 Ergänzende Kommentare:

- In der Zeile 30 werden sowohl der Pfad als auch der Dateiname für die Datei festgelegt, wenn man in diese Datei drucken will. Im Druck-Dialog – vergleichen Sie mit Abbildung 23.6.2.3 – werden dann der (Pseudo-)Drucker und der Datei-Pfad angezeigt.
- In der Prozedur zum Öffnen der Bilddatei wird in der Zeile 39 ein Dateifilter für ausgewählte Bild-Formate eingesetzt.
- Die verwendete Komponente *ScrollArea* – als DrawingArea mit einem Scroll-Bereich – erhält in

der Zeile 48 die Werte für den Scroll-Bereich aus den Abmessungen des geöffneten Bildes.
- Die Bildvorschau wird über das Ereignis *ScrollArea1_Draw()* realisiert. Es folgt in der Zeile 58 eine Verschiebung des Koordinatenursprungs $O(x_0,y_0)$ um den *negativen* Wert, der an den (Scroll-)Schiebereglern (verborgen) eingestellt ist. Der Koordinatenursprung $O(x_0,y_0)$ verschiebt sich nach so nach links oben; man sieht mehr vom Bild rechts unten.
- In der Prozedur *GetNonPrintableArea()* werden aus dem Druckerfilter – gespeichert in einer Datei im PPD-Format – die Werte für den oberen und linken Rand ermittelt, der nicht bedruckt werden kann.
- Für die Umrechnung von Bildgrößen (Pixel) in die Einheit Millimeter wird die Funktion *PixelToMillimeter(iPixel As Integer)* genutzt.
- Das Drucken wird über die Anweisungen in den Zeilen 61 bis 111 realisiert, wenn das Ereignis *printerImage_Draw()* ausgelöst wird.
- Dem Objekt *imgPrint* vom Typ Image wird in der Zeile 72 der Bildinhalt des geöffneten Bildes zugewiesen und nur mit diesem weitergearbeitet.
- In der Zeile 75 wird das Bild um 90° gedreht, wenn die Bildweite größer als die Bildhöhe ist.
- Das Bild wird in den Zeilen 84 bis 104 so skaliert, dass dieses Bild auf das Blatt A4-Papier passt, wenn die Breite oder die Höhe des Bildes nicht in den Druckbereich passen.
- Das eigentliche Drucken oder besser Zeichnen auf der Druckfläche übernimmt die Methode *Paint.ImagePrint(Image As Image, x As Float, y As Float, Width As Float, Height As Float)* in den Zeilen 91, 102 und 109.
- Der Ausdruck des Bildes startet in der Zeile 119, nachdem *zuvor* in der Zeile 115 ein Druck-Dialog aufgerufen wird, in dem ausgewählte Eigenschaften gesetzt werden können.

Mit besonderer Sorgfalt sind die Eigenschaften des Formulars und der eingesetzten Komponenten zu setzen, damit Sie genau das Design der Programm-Oberfläche erhalten, das Sie in der Abbildung 23.6.2.1 sehen. Im Projekt werden bestimmten Eigenschaften von ausgewählten Komponenten (Zeilen 11-25) zur Laufzeit passende Werte zugewiesen:

Abbildung 23.6.2.1.1: Container HBox mit 5 Komponenten

Beachtung sollten Sie der Komponente mit der roten 1 schenken. Es ist ein einfaches, randloses Panel in der Hintergrundfarbe der HBox, das über die Eigenschaft *pSpace.Expand* = *True* zur Laufzeit des Programms einen variablen Zwischenraum zwischen den ersten beiden und den letzten Komponenten schafft, nachdem sich bei den anderen 4 Komponenten – außer dem Button zur Wahl des Druckmodus (Farbe oder Graustufen) – die Komponenten-Weite entsprechend der Länge der Beschriftung (plus Weite des eingefügten Icon) über die Eigenschaft *Komponente.AutoResize* = *True* automatisch angepasst hat.

23.6.2.2 Programm-Erweiterung

Als Erweiterung können Sie dem Formular 2 SpinBoxen (*sboxMarginLeft*, *sboxMarginTop*) mit vernünftigen Startwerten sowie gut gewählten Minimum- und Maximum-Werten spendieren und in den Container *HBox* einfügen, um den linken und rechten Druckrand einstellen zu können. Folgende Änderungen im Quelltext sind dann notwendig:

```
Public Sub Form_Open()
...
  sboxMarginLeft.MinValue = 15
  sboxMarginLeft.MaxValue = 50
  sboxMarginLeft.Value = 20 ' Startwert; Einheit Millimeter
  sboxMarginTop.MinValue = 10
  sboxMarginTop.MaxValue = 25
  sboxMarginTop.Value = 15 ' Startwert; Einheit Millimeter
...
End ' Form_Open()

Public Sub printerImage_Draw()
...
  iDruckrandLinks = sboxMarginLeft.Value
  iDruckrandOben = sboxMarginTop.Value
...
End ' printerImage_Draw()
```