

```

package itextdemo;

import com.itextpdf.text.*;
import com.itextpdf.text.pdf.*;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;

public class Test9 {

    public static final int fontSize = 16;
    public static Font Arial = FontFactory.getFont("c:/windows/fonts/arial.ttf",
        BaseFont.IDENTITY_H, fontSize);
    public static Font SimplifiedArabic =
FontFactory.getFont("c:/windows/fonts/simpo.ttf",
        BaseFont.IDENTITY_H, fontSize);
    public static Font Andalus =
FontFactory.getFont("c:/windows/fonts/andlso.ttf",
        BaseFont.IDENTITY_H, fontSize);
    public static Font TraditionalArabic =
FontFactory.getFont("c:/windows/fonts/trado.ttf",
        BaseFont.IDENTITY_H, fontSize);
    public static Font Tahoma =
FontFactory.getFont("c:/windows/fonts/tahoma.ttf",
        BaseFont.IDENTITY_H, fontSize);
    public static final String tashkeelText =
"هَلْ تَعْلَمُ أَنَّ هُنَاكَ أَكْثَرَ مِنْ ٢٠٠ مِليُونٍ"
; " شَخْصٍ يَدْرُسُونَ اللُّغَةَ العَرَبِيَّةَ وَهِيَ لَيْسَتْ لِعَتْمِ الأُمِّ؟ " +
    public static final String normalText =
"هل تعلم أن هناك أكثر من ٢٠٠ مليون"
; " شخص يدرسون اللغة العربية وهي ليست لغتهم الأم؟ " +
    public static String activeText = tashkeelText;

    public static void example1() {
        try {
            String pd = System.getProperty("user.dir") + "/";
            Document document = new Document();
            PdfWriter writer = PdfWriter.getInstance(document,
                new FileOutputStream(pd + "example1.pdf"));
            document.open();
            PdfPTable table = new PdfPTable(2);
            table.setRunDirection(PdfWriter.RUN_DIRECTION_RTL);
            PdfPCell cell;
            cell = new PdfPCell(new Phrase("Invoice No", Arial));
            table.addCell(cell);

```

```

    cell = new PdfPCell(new Phrase("رقم الفاتورة", Arial));
    table.addCell(cell);

    document.add(table);
    document.close();
} catch (Exception e) {
    e.printStackTrace();
}
}

public static void example2() {
    try {
        String pd = System.getProperty("user.dir") + "/";
        Document document = new Document();
        PdfWriter writer = PdfWriter.getInstance(document,
            new FileOutputStream(pd + "example2.pdf"));
        document.open();
        PdfPTable table = new PdfPTable(1);
        table.setRunDirection(PdfWriter.RUN_DIRECTION_RTL);
        PdfPCell cell = new PdfPCell();
        Paragraph p = new Paragraph("رقم الفاتورة", Arial);
        p.setAlignment(PdfPCell.ALIGN_LEFT);
        cell.addElement(p);
        cell.setBorder(Rectangle.NO_BORDER);
        table.addCell(cell);

        document.add(table);
        document.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}

public static void example3() {
    try {
        BaseFont TahomaBase = BaseFont.createFont(
            "c:/windows/fonts/tahoma.ttf", BaseFont.IDENTITY_H, true);
        Font TahomaFont = new Font(TahomaBase, fontSize);

        BaseFont ArialBase = BaseFont.createFont(
            "c:/windows/fonts/arial.ttf", BaseFont.IDENTITY_H, true);
        Font ArialFont = new Font(ArialBase, fontSize);

        BaseFont AndaluslBase = BaseFont.createFont(

```

```

        "c:/windows/fonts/andlso.ttf", BaseFont.IDENTITY_H, true);
Font AndalusFont = new Font(Andalus1Base, fontSize);

BaseFont TraditionalArabicBase = BaseFont.createFont(
    "c:/windows/fonts/trado.ttf", BaseFont.IDENTITY_H, true);
Font TraditionalArabicFont = new Font(TraditionalArabicBase, fontSize);

BaseFont SimplifiedArabicBase = BaseFont.createFont(
    "c:/windows/fonts/simpo.ttf", BaseFont.IDENTITY_H, true);
Font SimplifiedArabicFont = new Font(SimplifiedArabicBase, fontSize);

String pd = System.getProperty("user.dir") + "/";
Document document = new Document();
PdfWriter writer = PdfWriter.getInstance(document,
    new FileOutputStream(pd + "example3.pdf"));
document.open();
PdfPTable table = new PdfPTable(1);
table.setRunDirection(PdfWriter.RUN_DIRECTION_RTL);
PdfPCell cell = new PdfPCell();
cell.setBorder(Rectangle.NO_BORDER);
Paragraph p;

p = new Paragraph("using basefont Tahoma\n" + activeText, TahomaFont);
p.setAlignment(PdfPCell.ALIGN_LEFT);
cell.addElement(p);

p = new Paragraph("using fontfactory Tahoma\n" + activeText, Tahoma);
p.setAlignment(PdfPCell.ALIGN_LEFT);
cell.addElement(p);

p = new Paragraph("using fontfactory Arial\n" + activeText, Arial);
p.setAlignment(PdfPCell.ALIGN_LEFT);
cell.addElement(p);

p = new Paragraph("using basefont Arial\n" + activeText, ArialFont);
p.setAlignment(PdfPCell.ALIGN_LEFT);
cell.addElement(p);

p = new Paragraph("using fontfactory Andalus\n" + activeText, Andalus);
p.setAlignment(PdfPCell.ALIGN_LEFT);
cell.addElement(p);

p = new Paragraph("using basefont Andalus\n" + activeText, AndalusFont);
p.setAlignment(PdfPCell.ALIGN_LEFT);

```

```

cell.addElement(p);

    p = new Paragraph("using fontfactory TraditionalArabic\n" + activeText,
TraditionalArabic);
    p.setAlignment(PdfPCell.ALIGN_LEFT);
    cell.addElement(p);

    p = new Paragraph("using basefont TraditionalArabic\n" + activeText,
TraditionalArabicFont);
    p.setAlignment(PdfPCell.ALIGN_LEFT);
    cell.addElement(p);

    p = new Paragraph("using fontfactory SimplifiedArabic\n" + activeText,
SimplifiedArabic);
    p.setAlignment(PdfPCell.ALIGN_LEFT);
    cell.addElement(p);

    p = new Paragraph("using basefont SimplifiedArabic\n" + activeText,
SimplifiedArabicFont);
    p.setAlignment(PdfPCell.ALIGN_LEFT);
    cell.addElement(p);

table.addCell(cell);

document.add(table);
document.close();
} catch (Exception e) {
    e.printStackTrace();
}

}

public static void example4() throws DocumentException,
FileNotFoundException {
    String pd = System.getProperty("user.dir") + "/";
    Document document = new Document();
    PdfWriter writer = PdfWriter.getInstance(document,
        new FileOutputStream(pd + "example4.pdf"));
    document.open();
    PdfPTable table = new PdfPTable(1);
    table.setRunDirection(PdfWriter.RUN_DIRECTION_RTL);
    PdfPCell cell = new PdfPCell();
    cell.setBorder(Rectangle.NO_BORDER);
    Paragraph p;

```

```
p = new Paragraph("using basefont Arial\n" + activeText, Arial);
p.setAlignment(PdfPCell.ALIGN_LEFT);
cell.addElement(p);

p = new Paragraph("using basefont Andalus\n" + activeText, Andalus);
p.setAlignment(PdfPCell.ALIGN_LEFT);
cell.addElement(p);
table.addCell(cell);

document.add(table);
document.close();

}

public static void main(String[] args) throws DocumentException,
FileNotFoundException {
    example1();
    example2();
    example3();
    example4();
}
}
```