Java 核心技术(进阶)

第四章 高级文件处理

第五节 条形码和二维码简介及解析

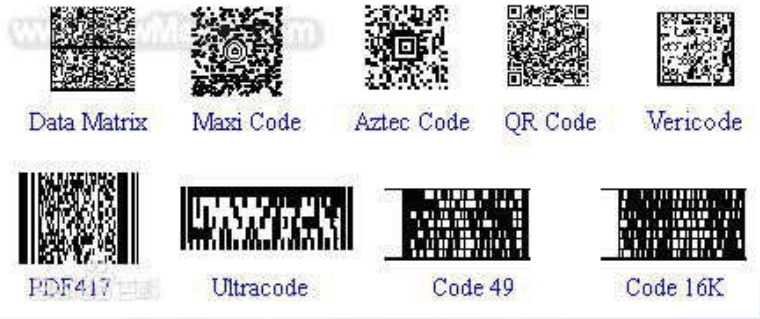
华东师范大学 陈良育

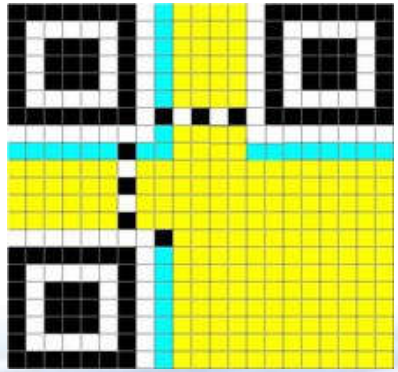
条形码

- 条形码(barcode)
  - 将宽度不等的多个黑条和空白，按照一定的编码规则排列，用以表达一组信息的图形标识符
  - 上个世纪40年代发明的
  - 通常代表一串数字/字母，每一位有特殊含义
  - 一般数据容量30个数字/字母
  - 专门机构管理：中国物品编码中心

二维码

- 二维码，二维条形码
  - 用某种特定的几何图形按一定规律在平面（二维方向上）分布的黑白相间的图形记录数据符号信息
  - 比一维条形码能存更多信息，表示更多数据类型
  - 能够存储数字/字母/汉字/图片等信息
  - 字符集128个字符
  - 可存储几百到几十KB字符
  - 抗损坏





- Zxing(Zebra Crossing)
  - Google 出品
  - 支持1D和2D的Barcode
  - 主要类
    - BitMatrix 位图矩阵
    - MultiFormatWriter 位图编写器
    - MatrixToImageWriter 写入图片

1D product	1D industrial	2D
UPC-A	Code 39	QR Code
UPC-E	Code 93	Data Matrix
EAN-8	Code 128	Aztec (beta)
EAN-13	Codabar	PDF 417 (beta)
	ITF	MaxiCode
	RSS-14	
	RSS-Expanded	

Barcode4J

- Barcode4J
  - http://barcode4j.sourceforge.net/
  - 纯Java实现的条形码生成
  - 只负责生成，不负责解析
  - 主要类
    - BarcodeUtil
    - BarcodeGenerator
    - DefaultConfiguration

- 1D barcode implementations [examples] [xml-format]:
  - Interleaved 2 of 5
  - ITF-14
  - Code 39
  - Code 128
  - EAN-128, GS1-128 (based on Code 128)
  - Codabar
  - UPC-A and UPC-E (with supplementals)
  - EAN-13 and EAN-8 (with supplementals)
  - POSTNET
  - Royal Mail Customer Barcode (Four State)
  - USPS Intelligent Mail (4-State Customer Barcode)
- 2D barcode implementations [examples] [xml-format]:
  - PDF 417 (ISO/IEC 15438:2001(E))
  - DataMatrix (ISO/IEC 16022:2000(E))
  - QR Code (ISO/IEC 18004:2006(E)) (requires ZXing)

总结

- 总结
  - 注意条形码种类
  - 高并发的时候，注意产生图片的速度
  - API很多，需要多查询、多练习

代码(1) zxing/BarCodeTest.java


```
package zxing;

import com.google.zxing.BarcodeFormat;

public class BarCodeTest {
    /**
     * generateCode 根据code生成相应的一维码
     * @param file 一维码目标文件
     * @param code 一维码内容
     * @param width 图片宽度
     * @param height 图片高度
     */
```

代码(2) zxing/BarCodeTest.java


```
public static void generateCode(File file, String code, int width, int height) {
    //定义位图矩阵BitMatrix
    BitMatrix matrix = null;
    try {
        // 使用code_128格式进行编码生成100*25的条形码
        MultiFormatWriter writer = new MultiFormatWriter();

        matrix = writer.encode(code,BarcodeFormat.CODE_128, width, height, null);
        //matrix = writer.encode(code,BarcodeFormat.EAN_13, width, height, null);
    } catch (WriterException e) {
        e.printStackTrace();
    }

    //将位图矩阵BitMatrix保存为图片
    try (FileOutputStream outStream = new FileOutputStream(file)) {
        ImageIO.write(MatrixToImageWriter.toBufferedImage(matrix), "png",
                outStream);
        outStream.flush();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

代码(3) zxing/BarCodeTest.java


```
    public static void readCode(File file){
        try {
            BufferedImage image = ImageIO.read(file);
            if (image == null) {
                return;
            }
            LuminanceSource source = new BufferedImageLuminanceSource(image);
            BinaryBitmap bitmap = new BinaryBitmap(new HybridBinarizer(source));

            Map<DecodeHintType, Object> hints = new HashMap<>();
            hints.put(DecodeHintType.CHARACTER_SET, "GBK");
            hints.put(DecodeHintType.PURE_BARCODE, Boolean.TRUE);
            hints.put(DecodeHintType.TRY_HARDER, Boolean.TRUE);

            Result result = new MultiFormatReader().decode(bitmap, hints);
            System.out.println("条形码内容: "+result.getText());
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public static void main(String[] args) throws Exception {
        //generateCode(new File("1dcode.png"), "123456789012", 500, 250);
        readCode(new File("1dcode.png"));
    }
}
```

代码(4) QRCodeTest.java


```
package zxing;

import com.google.zxing.*;

public class QRCodeTest {
    /*
     * 定义二维码的宽高
     */
    private static int WIDTH = 300;
    private static int HEIGHT = 300;
    private static String FORMAT = "png";//二维码格式
```

代码(5) QRCodeTest.java


```java
//生成二维码
public static void generateQRCode(File file, String content) {
    //定义二维码参数
    Map<EncodeHintType, Object> hints = new HashMap<>();

    hints.put(EncodeHintType.CHARACTER_SET, "utf-8");//设置编码
    hints.put(EncodeHintType.ERROR_CORRECTION, ErrorCorrectionLevel.M);//设置容错等级
    hints.put(EncodeHintType.MARGIN, 2);//设置边距默认是5

    try {
        BitMatrix bitMatrix = new MultiFormatWriter().encode(content, BarcodeFormat.QR_CODE, WIDTH, HEIGHT, hints);
        Path path = file.toPath();
        MatrixToImageWriter.writeToPath(bitMatrix, FORMAT, path);//写到指定路径下

    } catch (Exception e) {
        e.printStackTrace();
    }

}
```

代码(6) QRCodeTest.java


```
    //读取二维码
    public static void readQrCode(File file) {
        MultiFormatReader reader = new MultiFormatReader();
        try {
            BufferedImage image = ImageIO.read(file);
            BinaryBitmap binaryBitmap = new BinaryBitmap(new HybridBinarizer(new BufferedImageLuminanceSource(image)));
            Map<DecodeHintType, Object> hints = new HashMap<>();
            hints.put(DecodeHintType.CHARACTER_SET, "utf-8");//设置编码
            Result result = reader.decode(binaryBitmap, hints);
            System.out.println("解析结果:" + result.toString());
            System.out.println("二维码格式:" + result.getBarcodeFormat());
            System.out.println("二维码文本内容:" + result.getText());
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public static void main(String[] args) {
        generateQRCode(new File("2dcode.png"), "https://www.baidu.com");
        readQrCode(new File("2dcode.png"));
        //readQrCode(new File("2dcode.jpg"));
    }
}
```

代码(7) barcode4j/BarcodeTest.java


```
package barcode4j;

import java.awt.image.BufferedImage;


public class BarCodeTest {

    public static void main(String[] args) {
        String msg = "123456789012";
        String path = "1dcode.png";
        generateFile(msg, path);
    }

    public static void generateFile(String msg, String path) {
        File file = new File(path);
        try {
            Code39Bean bean = new Code39Bean();
            //EAN13Bean bean = new EAN13Bean();
```

代码(8) barcode4j/BarcodeTest.java


```
            // dpi精度
            final int dpi = 150;
            // module宽度
            //bean.setModuleWidth(0.2);
            final double width = UnitConv.in2mm(2.0f / dpi);
            bean.setWideFactor(3);
            bean.setModuleWidth(width);
            bean.doQuietZone(false);

            String format = "image/png";
            // 输出到流
            BitmapCanvasProvider canvas = new BitmapCanvasProvider(new FileOutputStream(file), format, dpi,
                    BufferedImage.TYPE_BYTE_BINARY, false, 0);

            // 生成条形码
            bean.generateBarcode(canvas, msg);

            // 结束绘制
            canvas.finish();

        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

代码(9) DataMatrixCodeTest.java


```
package barcode4j;

import java.awt.image.BufferedImage;

public class DataMatrixCodeTest {

    public static void main(String[] args) throws Exception {

        BarcodeUtil util = BarcodeUtil.getInstance();
        BarcodeGenerator gen = util.createBarcodeGenerator(buildCfg("datamatrix"));

        OutputStream fout = new FileOutputStream("2dcode.png");
        int resolution = 300;
        BitmapCanvasProvider canvas = new BitmapCanvasProvider(fout, "image/png", resolution,
                BufferedImage.TYPE_BYTE_BINARY, false, 0);

        gen.generateBarcode(canvas, "be the coder");
        canvas.finish();
    }
```

代码(10) DataMatrixCodeTest.java


```
    private static Configuration buildCfg(String type) {
        DefaultConfiguration cfg = new DefaultConfiguration("barcode");

        // Bar code type
        DefaultConfiguration child = new DefaultConfiguration(type);
        cfg.addChild(child);

        // Human readable text position
        DefaultConfiguration attr = new DefaultConfiguration("human-readable");
//      DefaultConfiguration subAttr = new DefaultConfiguration("placement");
//      subAttr.setValue("bottom");
//      attr.addChild(subAttr);
//      child.addChild(attr);
//      datamatrix code has no human-readable part
//      see http://barcode4j.sourceforge.net/2.1/symbol-datamatrix.html

        attr = new DefaultConfiguration("height");
        attr.setValue(50);
        child.addChild(attr);
        attr = new DefaultConfiguration("module-width");
        attr.setValue("0.6");
        child.addChild(attr);
        return cfg;
    }
}
```

谢 谢！