

Die Perlen iText und JAI

# Java, Visitenkarten und QR-Code

Michael Niedermair, E-Mail: m.g.n@gmx.de

Auf diversen Plattformen wie beispielsweise Sourceforge oder GitHub wird eine Menge Code verwaltet. Dabei finden sich Schätze, die für einen Entwickler sehr hilfreich sein können, aber nicht immer bekannt sind.

► Bekommt man viele Visitenkarten, so hat man meist viel Handarbeit, die Daten in den Computer bzw. das Smartphone zu bekommen. Eleganter wäre es, wenn man die Daten abfotografieren könnte und dann automatisch einen neuen Kontakteintrag hätte. Für die Generierung eines entsprechenden QR-Codes findet eine Suchmaschine schnell eine Lösung. Allerdings ist die Webseitenhandarbeit nicht wirklich eine gute Lösung, wenn man viele Visitenkarten für die Kollegen erstellen will. Java bietet hier eine schnelle und einfache Lösung, indem man die Bibliotheken iText (für QR-Code und PDF) und JAI (Java Advanced Imaging für Bildformate wie PNG, ...) kombiniert.

Das QR-Code-Format für Visitenkarten trägt den Namen MECARD. Tabelle 1 zeigt das Format. Je nach Smartphone und Kontaktdatenbank werden nicht alle Schlüssel verwendet. Manche akzeptieren auch weitere Schlüssel aus dem vCard-Format, jedoch ist darauf kein Verlass. Tests mit verschiedenen Systemen zeigen aber, dass unbekannte Schlüssel einfach ignoriert werden.

Listing 1 zeigt den einfachen Java-Code zur Erstellung des QR-Codes. Dabei werden zuerst die ganzen Felder in einem `StringBuilder` gesammelt und dann an die Klasse `BarcodeQRCode` übergeben. Die Größe wird dabei über `SIZE` festgelegt, hier 150 Pixel. Wird als Größe 1 verwendet, so wird entsprechend skaliert. Anschließend wird der QR-Code in einem PDF gespeichert, damit man diesen als Vektorformat hat und somit ohne Verlust beliebig skalieren kann. Danach wird das Bild im PNG-Format zusätzlich gespeichert (für Programme, die keine PDF-Bilder verarbeiten können).

Scannt man anschließend den QR-Code mit dem Smartphone, so wird der Kontakt dem Telefonbuch hinzugefügt (s. Abb. 2). Einzig die Art der Telefonnummer (Privat, Firma, Fax, ...) und der E-Mail muss von Hand angepasst werden, da der Typ von MECARD nicht unterstützt wird.

Verwendet man die in [Nie14] vorgestellte `CsvJdbc`-Bibliothek, so lassen sich die Daten sehr einfach aus einer CSV-Datei laden und somit viele QR-Codes einfach erzeugen.

```
// Größe: 1 -> nicht skaliert
private static final int SIZE = 150;
public static void main(final String[] args) {
    try {
        StringBuilder buf = new StringBuilder(512);
        buf.append("MECARD:");
        buf.append("N:").append("Niedermair,Michael").append(",");
        buf.append("ORG:").append(
            "Städt. Berufsschule für Informationstechnik").append(",");
        buf.append("TEL:").append("+49 89 233 85200").append(",");
        buf.append("TEL:").append("+49 89 233 85201").append(",");
        buf.append("URL:").append("https://www.bsinfo.eu/").append(",");
        buf.append("EMAIL:").append("mgn.schule@gmx.de").append(",");
        buf.append("EMAIL:").append("m.g.n@gmx.de").append(",");
        buf.append("ADR:").append("Riesstr. 34,München,80992,Deutschland")
            .append(",");
        buf.append("NOTE:").append("Kordinator, Studiendirektor")
            .append(",");
        // QR-Code
        BarcodeQRCode qrcode =
            new BarcodeQRCode(buf.toString(), SIZE, SIZE, null);
        /* PDF */
        Document document = new Document(PageSize.A5);
        PdfWriter.getInstance(document,
            new FileOutputStream(QRCodeVCard.FILE_PDF));
        document.open();
        document.add(qrcode.getImage());
        document.close();
        System.out.println(QRCodeVCard.FILE_PDF + " erstellt...");
        /* PNG */
        Image png = qrcode.createAwtImage(Color.BLACK, Color.WHITE);
        ImageIO.write(AWTImageDescriptor.create(png, null), "png",
            new File(QRCodeVCard.FILE_PNG));
        System.out.println(QRCodeVCard.FILE_PNG + " erstellt...");
        // ...
    }
}
```

Listing 1: Erstellung des QR-Codes

Schlüssel	Beschreibung
N:	Der Nachname und Vorname (werden durch ein Komma getrennt)
TEL:	Telefonnummer (1 bis 24 Ziffern) Kann auch mehrmals verwendet werden
TEL-AV:	Videotelefonnummer (1 bis 24 Ziffern)
EMAIL:	Die E-Mail-Adresse Kann auch mehrmals verwendet werden
NOTE:	Zusätzliche Bemerkung
ADR:	Die Adresse mit folgender Bedeutung (durch Komma getrennt): PO box, room number, house number, city, prefecture, zip code and country. Mehrals möglich, jedoch einige Kontaktdatenbanken im Smartphone mögen keine zwei Adressen pro Kontakt
URL:	URL der Webseite (inkl. http://)
ORG:	Organisation/Unternehmen
BDAY:	Geburtstag (im Format YYYYMMDD)
NICKNAME:	Spitzname

Tabelle 1: MECARD-Datenformat

## Links

- [CsvJdbc] Java database driver for reading comma-separated-value files, <http://csvjdbc.sourceforge.net/>
- [iText] <http://sourceforge.net/projects/itext/files/>
- [JAI] Java Advanced Imaging, <http://www.oracle.com/technetwork/java/current-142188.html>
- [MECARD] <http://theqrplace.wordpress.com/2011/05/02/qr-code-tech-info-mecard-format/> und <https://www.nttdocomo.co.jp/english/service/developer/make/content/barcode/function/application/addressbook/index.html>
- [Nie14] M. Niedermair, Programmierperlen: Strukturierte Ein- und Ausgabe, in: JavaSPEKTRUM, 1/2014

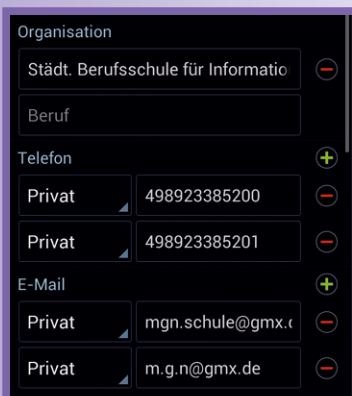


Abb. 1: Kontakteintrag im Smartphone



Abb. 2: Erzeugter QR-Code