

Outil de génération de fiches produit

On souhaite automatiser la production de fiches et d'étiquettes pour des produits dans un système de gestion de stock.

Le projet sera réalisé en plusieurs étapes. Ce document décrit la première version du système à réaliser.

Objectifs

La première version du système se limitera à un utilitaire en ligne de commande permettant la génération de fiches produit PDF et de planches d'étiquettes. Cet utilitaire sera développé en Java.

À travers la réalisation de cet utilitaire, vous devrez mettre en pratique des principes de la programmation objet.

Principales fonctionnalités

Le système accepte en entrée un fichier au format [CSV](#) contenant la liste des produits. Un produit est défini par un code, un nom, une description, une catégorie et un prix à l'unité HT (hors taxe).

Selon les paramètres de lancement, le système produit :

- un document PDF qui contient une fiche produit par page
- un document PDF qui contient des planches d'étiquettes

Exemple de fichier d'entrée

Le document d'entrée est un fichier [CSV](#) sans en-têtes (c'est-à-dire que la première ligne ne contient pas le nom des colonnes mais directement le premier produit). Tous les champs d'un produit sont obligatoires et doivent être renseignés. Ci-dessous un exemple de fichier :

```
XV13948573;Mysterium;jeu d'enquête coopératif;jeu;30,21
VT66556555;The Witcher 3;Incarnes un tueur de monstres et vivez son épopée !;jeu video;32,12
GF55556566;Wonder Woman DVD;L'adaptation cinéma du DC Comics;film;16,55
```

Le fichier définit pour chaque ligne : le code, le nom, la description, la catégorie et le prix HT.

Exemple de fichiers de sortie

















Les fichiers produits sont au format PDF. Selon les paramètres passés au lancement, il est possible de créer des fiches produits ou des planches d'étiquettes.

Un document de fiches produit contient une fiche par page. Sur chaque fiche produit figure l'ensemble des informations du fichier d'entrée avec en plus le montant de la TVA et le prix TTC.

Ci-dessous un exemple de fiche produit :

Code: XV13948573	
Mysterium	Catégorie : jeu
Description	
<div style="border: 1px solid black; padding: 10px; min-height: 150px;"> jeu d'enquête coopératif </div>	
Montant HT : 30,21 € TVA : 6,04 € Montant TTC: 36,25 €	

Un document de planches d'étiquettes contient 16 étiquettes par page sur deux colonnes. Chaque étiquette contient le code produit, le nom du produit, la prix HT, le prix TTC et un QRCode correspondant au code produit. Ci-dessous un exemple de planche d'étiquettes :

 XV13948573 Mysterium 36,25€ TTC (30,21€ HT)	 XV13948573 Mysterium 36,25€ TTC (30,21€ HT)
 XV13948573 Mysterium 36,25€ TTC (30,21€ HT)	 XV13948573 Mysterium 36,25€ TTC (30,21€ HT)
 XV13948573 Mysterium 36,25€ TTC (30,21€ HT)	 XV13948573 Mysterium 36,25€ TTC (30,21€ HT)
 XV13948573 Mysterium 36,25€ TTC (30,21€ HT)	 XV13948573 Mysterium 36,25€ TTC (30,21€ HT)
 XV13948573 Mysterium 36,25€ TTC (30,21€ HT)	 XV13948573 Mysterium 36,25€ TTC (30,21€ HT)
 XV13948573 Mysterium 30,21€ TTC (30,21€ HT)	 XV13948573 Mysterium 36,25€ TTC (30,21€ HT)
 XV13948573 Mysterium 36,25€ TTC (30,21€ HT)	 XV13948573 Mysterium 36,25€ TTC (30,21€ HT)
 XV13948573 Mysterium 36,25€ TTC (30,21€ HT)	 XV13948573 Mysterium 36,25€ TTC (30,21€ HT)

Utilisation de la ligne de commandes

Le système se limite à un utilitaire que l'on peut lancer en ligne de commande. Comme il s'agit d'un programme Java, la ligne de commande sera de la forme :

```
java -classpath <CLASSPATH> NomDeLaClasse <PARAMETRES>
```

Les paramètres possibles sont les suivants :

- h pour afficher l'aide
- i suivi du chemin du fichier CSV d'entrée
- categorie suivi du nom de la catégorie pour filtrer les produits
- etiquette suivi du chemin du fichier PDF qui contiendra les étiquettes produit
- fiche suivi du chemin du fichier PDF contenant les fiches produit
- tva montant de la TVA à utiliser pour calculer le prix TTC (par défaut 20%)

Pour produire un fichier des étiquettes :

```
java -classpath <CLASSPATH> NomDeLaClasse -i produits.csv -etiquette out/etiquettes.pdf
```

Pour produire un fichier des fiches produit :

```
java -classpath <CLASSPATH> NomDeLaClasse -i produits.csv --fiche out/fiches.pdf
```

Pour produire un fichier des étiquettes **ET** un fichier des fiches produit :

```
java -classpath <CLASSPATH> NomDeLaClasse -i produits.csv -etiquette etiquettes.pdf -fiche fiches.pdf
```

Pour produire un fichier des étiquettes uniquement pour la catégorie « film » :

```
java -classpath <CLASSPATH> NomDeLaClasse -i produits.csv -categorie film -etiquette out/etiquettes.pdf
```

Pour produire un fichier des étiquettes en utilisant une TVA à 5 % :

```
java -classpath <CLASSPATH> NomDeLaClasse -i produits.csv -etiquette out/etiquettes.pdf -tva 5.0
```

Structure du projet et livrable

Vous devrez fournir les sources du projet et indiquer **le nom de la classe principale** permettant d'exécuter l'utilitaire. La structure de votre projet doit **obligatoirement** contenir deux répertoires :

- src** : ce répertoire contient les packages et les sources de votre application. Les sources doivent être compatibles avec le JDK 1.8 de Oracle
- lib** : ce répertoire contient les fichiers JAR des bibliothèques tierces que vous utiliserez.

Aucune documentation n'est requise par contre le programme doit afficher une aide complète lorsqu'il est lancé avec l'option -h.

Pour l'évaluation, une attention particulière sera donnée à la façon dont vous aurez conçu votre application et dans la mise en œuvre des concepts de la programmation objet. Vous devez notamment vous assurer que votre application est conçue de façon à isoler et à assurer un couplage minimal entre les différents sous-systèmes : lecture du fichier CSV, production des différents fichiers, génération PDF, calcul des données pour chaque produit (montant de la TVA et montant TTC).

Pensez également que votre application doit pouvoir gérer des volumes importants de données en entrée. Évitez de charger l'intégralité du fichier CSV en mémoire et essayez de concevoir un traitement en flux.

Bibliothèques tierces

Pour réaliser ce projet, vous allez avoir besoin de bibliothèques Java. Elles sont disponibles au format JAR et vous pouvez les incorporer à votre projet Eclipse. Vous êtes libres d'utiliser les bibliothèques qui vous paraissent les plus utiles. Néanmoins, je vous conseille d'utiliser les suivantes :

iText

iText permet de produire des fichiers PDF. Pour l'utiliser, vous devez télécharger et ajouter dans votre projet les trois fichiers JAR suivants :

<http://central.maven.org/maven2/com/itextpdf/itextpdf/5.5.12/itextpdf-5.5.12.jar>
<http://central.maven.org/maven2/org/bouncycastle/bcprov-jdk15on/1.58/bcprov-jdk15on-1.58.jar>
<http://central.maven.org/maven2/org/bouncycastle/bcpkix-jdk15on/1.58/bcpkix-jdk15on-1.58.jar>

Il existe de nombreux articles sur le Web décrivant l'utilisation de iText, je vous recommande néanmoins ce site : <http://tutorials.jenkov.com/java-itext/index.html>

zxing

zxing est une bibliothèque pour produire des codes-barres et plus particulièrement des QRcodes. Pour l'utiliser, vous devez télécharger et ajouter dans votre projet les deux fichiers JAR suivants :

<http://central.maven.org/maven2/com/google/zxing/core/3.3.1/core-3.3.1.jar>
<http://central.maven.org/maven2/com/google/zxing/javase/3.3.1/javase-3.3.1.jar>

Pour l'intégrer dans votre application, vous pouvez vous inspirer du code ci-dessous :

```
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.IOException;

import javax.imageio.ImageIO;

import com.google.zxing.BarcodeFormat;
import com.google.zxing.WriterException;
import com.google.zxing.client.j2se.MatrixToImageWriter;
import com.google.zxing.common.BitMatrix;
import com.google.zxing.qrcode.QRCodeWriter;

public class QRCodeGenerator {

    public static void main(String[] args) throws IOException, WriterException {
        String content = "http://test.com";
        String filename = "qrcode.png";

        BufferedImage image = generate(content, 150);
        ImageIO.write(image, "PNG", new File(filename));
    }

    private static BufferedImage generate(String content, int sizeInPixels) throws WriterException {
        QRCodeWriter qrWriter = new QRCodeWriter();
        Object matrix = qrWriter.encode(content, BarcodeFormat.QR_CODE, sizeInPixels, sizeInPixels);
        return MatrixToImageWriter.toBufferedImage((BitMatrix) matrix);
    }
}
```

commons-cli

commons-cli est une bibliothèque pour interpréter les paramètres passés à votre application (cli est la contraction de *command line interface*). Pour l'utiliser, vous devez télécharger et ajouter dans votre projet le fichier JAR suivant :

<http://central.maven.org/maven2/commons-cli/commons-cli/1.4/commons-cli-1.4.jar>

Pour plus d'informations, vous pouvez vous reporter à la documentation officielle :

<https://commons.apache.org/proper/commons-cli/usage.html>