**Документация**

BoerseDataConvert

Изготвили: Деян Делчев и Димитър Бялков

# Изисквания

Програмата се изпълнява чрез команден ред и изисква тези параметри:

-i или --input – Показват на програмата къде се намира входящият ZIP архив

-d или --directory – Указва директория в която програмата ще съхранява временни файлове. **ДИРЕКТОРИЯТА СЕ ПОЧИСТВА РЕКУРСИВНО АКО НЕ Е ПРАЗНА**

-o или --output – Задава директория, в която се извеждат конвертираните данни

-t или --tags – Указва мястото на файла tags.txt, в който се съдържат изискванията за таговете – техния номер, тяхното име и техните ограничения, ако има такива. Незадължително, ако програмата и tags.txt са в една директория.

-h или --help – Отпечатва информация за програмата, версия, автори и изисквани параметри

# Структура на програмата:

Прогарамата състои от:

* Един модел – Record
* Една таблица от тагове - TagsTable
* Един контролер – RecordController
* Един четец – Reader
* Един записващ клас – Writer

## Модел – Record

Record съхранява информацията от един ред от текстовите файлове, под формата на лист от KeyValuePairs, чиито ключове са таговете, а стойности са информацията към конкретния таг.

Пример:

Реда:

900#01|204#J|008#Raiffeisen Schweiz Genossenschaft|205#130355|206#111205|460#Raiffeisen Schweiz Genossenschaft|207#Raiffeisen Schweiz Genossenschaft#Raiffeisenplatz 4##St. Gallen, CH-SG 9001#Switzerland|208#socialmedia@raiffeisen.ch|209#41 71 225 88 88|210#www.raiffeisen.ch|451#5299006GIHQ1ELISCV48

ще бъде записан в речника:

Dictionary<string, string> TagsValues

900:01

204:J

206: 111205

460:Raiffeisen Schweiz Genossenschaft

. . .

## Една таблица от тагове – TagsTable

При създаване на класа се прочита текстов файл с тагове, чиято се информация се запазва в речник. TagsTable съдържа метода GetTagValue(), който полува таг, като аргумент и връща информацията за него.Ако тага не съществъва хвърля KeyNotFoundException.

## Контролер – RecordController

RecordController съдържа метод ConvertToXml(), който получава като аргумент Record и връща XML кода в string формат. При неправилен таг или неправилна стойност се хвърля подходящ Exception, който посочва точното място на грешката.

Пример: Record съхранява следната информация

Dictionary<string, string> TagsValues

900:01

204:J

008:HSH Nordbank AG

205:150215

206:111138

460:HSH Nordbank

207:HSH Nordbank AG#Gerhart-Hauptmann-Platz 50#20095 Hamburg#Deutschland

208:info@hsh-nordbank.com

209:040 33330

210:https://www.hsh-nordbank.de

451:TUKDD90GPC79G1KOE162

И ще бъде превърната във вида:

<record id=”1”>

<Source>01</Source>

<Active\_status>J</Active\_status>

<Name\_of\_the\_Issuer>HSH Nordbank AG</Name\_of\_the\_Issuer>

<Issuer\_ID>150215</Issuer\_ID>

<Issuer\_Group\_ID>111138</Issuer\_Group\_ID>

<Issuer\_Group\_Name>HSH Nordbank</Issuer\_Group\_Name>

<Issuer\_address>HSH Nordbank AG#Gerhart-Hauptmann-Platz 50#20095 Hamburg#Deutschland</Issuer\_address>

<Issuer\_e\_mail\_address>info@hsh-nordbank.com</Issuer\_e\_mail\_address>

<Issuer\_hotline>040 33330</Issuer\_hotline>

<Issuer\_URL>https://www.hsh-nordbank.de</Issuer\_URL>

<LEI>TUKDD90GPC79G1KOE162</LEI>

</record>

## Четец – Reader

Reader обхожда всички файлове и превръща всеки ред (с изключение на първия и последния) в Record.

## Записващ клас – Writer

Writer създава XML файловете и записва в тях върнатите от RecordController.ConvertToXml() стрингове.