Eric Rossetto (134020), Stefano Travasci (133014)

[rossetto.eric@spes.uniud.it](mailto:rossetto.eric@spes.uniud.it), [travasci.stefano@spes.uniud.it](mailto:travasci.stefano@spes.uniud.it)

Relazione di progetto del corso di Informatica Medica

# Problema e requisiti

La richiesta per il progetto è la seguente:

*creare un programma di traduzione di certificati di morte per poter estrarre un sottoinsieme di certificati tradotti in maniera immediata.*

I certificati di morte fornitoci contenevano codici nel formato standard ICD-10 da tradurre attraverso un mapping predefinito in codici ICD-11. In aggiunta a ciò, vi si richiedeva di fare una statistica sull’insieme dei certificati di morte tradotti.

# Soluzione

La soluzione presentata è basata sugli oggetti e le classi definite nel package strumentiConvertiCertificati:

* **Code**: questa classe rappresenta una patologia secondo i formati ICD-10 e ICD-11. Tale oggetto possiede 3 campi: un codice ICD-10, il rispettivo codice ICD-11 (se presente) e il tipo di mapping che sussiste fra di essi. Le tipologie di mapping sono: NotConvertedYet, Equivalent, SubClass, SubClassGood, NoMapping.
* **Certificate**: questa classe rappresenta un certificato di morte. In quanto tale contiene i campi per l’anno del decesso, età e sesso del deceduto e infine tutti i codici di tipo Code che rappresentano le varie cause di morte. L’ultimo codice, UCOD – Underlined Cause Of Death, ci dice quale sia stata la causa principale del decesso.
* **CodeConverter**: questa classe si occupa di convertire i singoli codici ICD10 in ICD11, specificando anche il tipo di conversione. Per eseguire la conversione rapidamente, usa un albero di ricerca contente tutti i codici ICD10 e il loro eventuale corrispondente in ICD11, creato a partire da un file di mapping. È importante sottolineare che, per rendere la creazione dell’albero più rapida, è stata fatta una supposizione sulla struttura del file di mapping: è necessario infatti che nel file di mapping i codici ICD10 appaiano sempre in ordine dal più generico al più specifico, ovvero che classificazioni più specifiche non appaiano prima di classificazioni meno specifiche.
* **MapNode**: questa classe implementa l’albero di ricerca utilizzato da CodeConverter.
* **CertificateConverter**: questa classe rappresenta un convertitore di certificati, possiede un campo privato contenente il convertitore di codici che userà per convertire i singoli certificati.

Nel definire l’oggetto Code abbiamo fatto riferimento a delle diverse tipologie di mapping che vengono in parte definire nel file di mapping. Tali tipologie sono definite nel seguente modo:

* NotConvertedYet
* Equivalent
* SubClass
* SubClassGood
* NoMapping

# Analisi statistica

La prima statistica vede il confronto tra il numero di certificati che sono stati tradotti *correttamente*, ossia che presentano un match di un tipo **Equivalent** o **SubClassGood** o **SubClass**, e il numero di certificati che invece sono stati scartati per **NoMapping** oppure perché il codice ICD10 inserito non ha avuto una corrispondenza nel file di mapping (si presume per un qualche errore di battitura).

|  |  |  |
| --- | --- | --- |
| Certificati totali | 100000 |  |
| **Certificati tradotti correttamente** | 99788 | 99.788% sul totale |
| **Certificati scartati** | 212 | 0.212% sul totale |
| Certificati scartati per NoMapping | 185 | 87.264% sul totale dei certificati scartati |
| Certificati scartati per codice inesistente | 27 | 12.735% sul totale dei certificati scartati |

Un ulteriore dato che potrebbe risultare interessante è il numero di certificati convertiti correttamente suddivisi per tipologia di conversione.

|  |  |  |
| --- | --- | --- |
| Certificati Equivalent | 16868 | 19.903% |
| Certificati SubClassGood | 628 | 0.629% |
| Certifcati | 82292 | 82.467% |

Molti dei campi all’interno dei certificati non sono valorizzati, è possibile quindi analizzare quale sia la dimensione media dei certificati, anche in base alla tipologia di conversione.

# Compilazione ed esecuzione del progetto

Sia sistemi Windows che Unix e Unix-like è possibile compilare con il comando javac. Spostandosi sulla directory /src, digitare:

javac ConvertiCertificatiInICD11.java

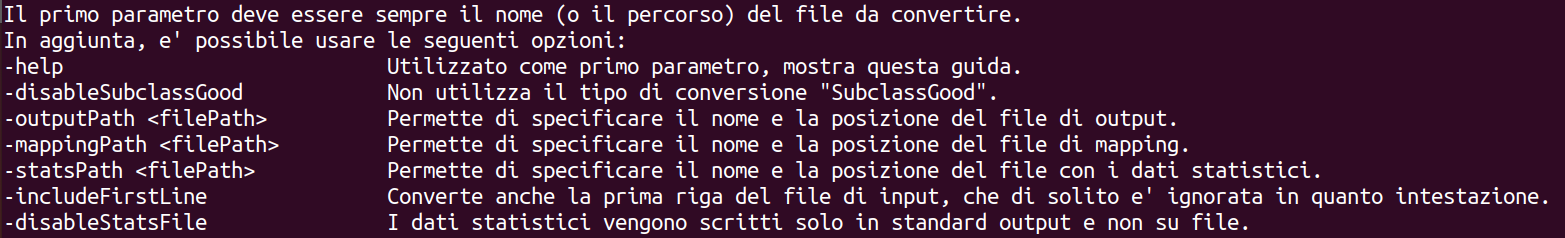
Per l’esecuzione del progetto è necessario conoscere la posizione del file da convertire, nello stesso formato fornitoci in cert2017100K.txt:

java ConvertiCertificatiInICD11 <pathFileDaConvertire>

L’esecuzione di tale commando produrrà due file il cui nome è generato proceduralmente:

1. out\_*LocalTime*.txt, che contiene l’insieme dei certificati di morte tradotti in ICD11;
2. stats\_*LocalTime*.txt, che contiene tutti i dati relativi all’analisi statistica, che compare anche come output al terminale.

Il comportamento del programma può essere modificato mediante l’utilizzo di alcune opzioni digitabili via linea di comando. Per mostrare una guida di aiuto digitare:

java ConvertiCertificatiInICD11 -help

Le opzioni sono esplicative ma per completezza ne chiariamo l’uso. Oltre al comando -help, abbiamo:

* -disableSubClassGood, in riferimento a quanto detto precedentemente sui tipi di conversione, tale comando disabilita la suddivisione tra i SubClass;
* -outputPath <filePath>, permette di specificare il nome e la posizione del file dove andranno i certificati convertiti in ICD11;
* -mappingPath <filePath>, permette di specificare il nome e la posizione del file utilizzato per il mapping;
* -statsPath <filePath>, permette di specificare il nome e la posizione del file dove andranno i risultati dell’analisi statistica effettuata;
* -includeFirstLine, legge anche la prima riga del file di input, che altrimenti è ingorata in quanto contenente l’intestazione;
* -disableStatsFile, disabilita la scrittura dell’analisi statistica su file limitandosi a stamparla su stdout.