

Fondamenti di Informatica 1 - anno accademico 200? - 200?
Prova di Programmazione, appello simulato

Si vuole realizzare un traduttore inglese-italiano tramite un dizionario che contiene coppie di tipo “parola traduzioni”. Il campo “parola” è una stringa che ha la funzione di chiave per il dizionario, mentre il campo “traduzioni” è un array di stringhe che contiene una o più traduzioni associate a ciascuna parola. Il dizionario può inoltre essere aggiornato *tramite un apposito metodo update*.

Il candidato ha a disposizione il file **TraduttoreTester.java**, il quale contiene classi e interfacce da utilizzare e/o completare secondo quanto qui indicato:

- **DictionaryUD**. Interfaccia che rappresenta il tipo di dati astratto “dizionario con update”. Contiene firme e specifiche di funzionamento dei metodi. In particolare contiene firma e specifiche di funzionamento del metodo update. *Non modificare*.
- **DictionaryItemNotFoundException**. Eccezione lanciata in caso di mancato ritrovamento di una coppia nel dizionario. *Non modificare*.
- **Traduttore**. Classe che implementa l’interfaccia DictionaryUD. *Completare la classe* scrivendone la parte privata e realizzandone i metodi pubblici. Il traduttore conterrà coppie di tipo “parola traduzioni” appartenenti alla classe **WordPair**, realizzata come classe interna alla classe **Traduttore** e il cui codice *non va modificato*. Si richiede inoltre di *realizzare un metodo toString per la classe Traduttore*, che restituisca una stringa contenente gli elementi del traduttore secondo il seguente formato: (1) ogni coppia “parola traduzioni” viene scritta su una riga diversa, e (2) all’interno di ogni riga la coppia viene scritta seguendo il formato specificato dal metodo toString della classe WordPair (si veda il corpo di tale metodo).
- **TraduttoreTester**. Classe che collauda il traduttore. *Completare la classe* in modo che il suo metodo main realizzi le seguenti specifiche:
 - Due nomi di file di testo, *file1* e *file2*, vengono passati come argomenti sulla riga di comando.
 - Viene creato un primo esemplare della classe **Traduttore**, in cui si inseriscono elementi letti dal file *file1* (scritto nello stesso formato specificato sopra per il metodo toString di Traduttore).
 - Viene creato un secondo esemplare della classe **Traduttore**, in cui vengono inseriti elementi letti dall’input standard (nello stesso formato specificato più sopra per il metodo toString di Traduttore). La fine dell’inserimento di elementi viene segnalata dalla terminazione dell’input standard <CTRL>+D.
 - Tramite invocazione del metodo update, il primo Traduttore viene aggiornato con il contenuto del secondo Traduttore e successivamente stampato sul file *file2* (nello stesso formato specificato più sopra per il metodo toString).

Nello svolgimento si possono usare, se ritenute utili, solo le seguenti classi della libreria standard:

- tutte le classi dei package java.lang e java.io;
- le classi Scanner, StringTokenizer e NoSuchElementException del package java.util.

Usando i file **eng2ita.txt** e **input.txt** allegati, il programma può essere provato con il comando
java TraduttoreTester eng2ita.txt eng2itaUD.txt < input.txt

Al termine della prova lasciare nella directory di lavoro tutti i file. La prima riga del file .java dovrà contenere un commento con nome/cognome del candidato, matricola, data, numero postazione. Prima di uscire dall’aula consegnare questa scheda compilata.

Cognome e Nome:	Corso di Laurea:
Matricola:	Postazione: ADT
Consegno l’elaborato.	Non consegno l’elaborato e mi ritiro dall’esame.
Firma	Firma