

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

Si vuole gestire una rubrica telefonica realizzando un dizionario che contiene coppie di tipo “nome numero”. Il campo “nome” è una stringa che ha la funzione di chiave per il dizionario (dunque la rubrica non potrà contenere omonimi) mentre il campo “numero” è un numero intero in formato long che rappresenta i recapiti telefonici associati a ciascun nome.

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

- **Dictionary**. Interfaccia che rappresenta il tipo di dati astratto dizionario, contiene le firme dei metodi e le specifiche del loro funzionamento. *Non modificare*.
- **DictionaryItemNotFoundException**. Eccezione lanciata in caso di mancato ritrovamento di una coppia nel dizionario. *Non modificare*.
- **Rubrica**. Classe che implementa l'interfaccia Dictionary. *Completare la classe* scrivendone la parte privata e realizzandone i metodi pubblici. La rubrica conterrà coppie di tipo “nome numero” appartenenti alla classe **Pair**, realizzata come classe interna alla classe **Rubrica** e il cui codice *non va modificato*. Si richiede che *il metodo find() abbia prestazioni $O(\log n)$* . Si richiede inoltre di *realizzare un metodo toString per la classe Rubrica*, che restituisca una stringa contenente gli elementi della rubrica secondo il seguente formato: (1) ogni coppia “nome numero” 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 Pair (si veda il corpo di tale metodo).
- **RubricaTester**. Classe che collauda la rubrica. *Completare la classe* in modo che il suo metodo main realizzi il seguente comportamento:
 - Due nomi di file di testo, *file1* e *file2*, vengono passati come argomenti sulla riga di comando.
 - Vengono creati due oggetti di tipo **Rubrica**. Nella prima rubrica si inseriscono elementi letti dal file *file1* (scritto nello stesso formato specificato sopra per il metodo toString di Rubrica).
 - Viene richiesta l'immissione di un nome da input standard. Il nome immesso viene cercato nella prima rubrica e la corrispondente coppia “nome numero” viene (se trovata) spostata dalla prima alla seconda rubrica (ovvero rimossa dalla prima e inserita nella seconda rubrica).
 - L'operazione descritta al punto precedente può essere ripetuta un numero non prefissato di volte. Il ciclo viene terminato tramite inserimento del carattere “Q” da input standard.
 - Al termine delle ricerche effettuate dall'utente, il contenuto della seconda rubrica viene stampato sul file *file2* (nello stesso formato specificato sopra per il metodo toString di Rubrica).

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 **paperopoli.txt** e **input.txt** allegati, il programma può essere provato con il comando
java RubricaTester paperopoli.txt topolinia.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