

Laboratorio di Programmazione

(Corso di Laurea in Informatica)

ESAME del 12 Febbraio 2018 - PROVA FILTRO B

Avvertenze

- SARANNO AMMESSI AL RESTO DELLA PROVA SOLO COLORO CHE CONSEGNANO UN ESERCIZIO FILTRO CHE FUNZIONA PERFETTAMENTE
 - Il programma realizzato DEVE rispettare le specifiche funzionali fornite (input/output).
 - Nello svolgimento dell'elaborato è possibile usare qualunque classe delle librerie standard di Java.
 - Si raccomanda di salvare, compilare e fare upload delle soluzioni con una certa regolarità.
 - Per la procedura di **consegna** si veda in fondo al documento.
 - ATTENZIONE! Rispettare le specifiche dell'output! La correzione avviene in automatico mediante script che verificano la forma dell'output, aggiungere messaggistica ulteriore o utilizzare termini e/o spaziatura differenti significa invalidare il proprio lavoro.
-

Filtro B

Scrivere un programma (la classe si DEVE chiamare “FiltroB” ed essere pubblica) che legga da linea di comando il nome di un file e per ogni riga di testo del file visualizzi su standard output:

- la riga di testo stessa
- il numero di parole di LUNGHEZZA DISPARI presenti su quella riga
- il rapporto fra il numero di parole di LUNGHEZZA DISPARI e il numero totale di parole su quella riga

Infine deve visualizzare su una linea l'andamento del numero di parole di LUNGHEZZA DISPARI (partendo dalla seconda riga), indicato da :

- ‘+’ se il numero cresce rispetto alla riga precedente,
- ‘=’ se è uguale a quello della riga precedente,
- ‘-’ se decresce rispetto alla riga precedente.

L'output deve essere una sequenza di tre righe, per ogni riga del testo, nel seguente formato:

Riga: <riga di testo>

Dispari: <numero intero>

Rapporto: <numero double>

seguita da una riga composta con i soli caratteri ‘+’, ‘=’, ‘-’.

Attenzione a non aggiungere spazi o altri caratteri non previsti nella specifica. Per un esempio, vedere sotto.

Nota Bene 1: nel caso una riga di testo sia vuota, il numero di stringhe di lunghezza dispari sarà zero e come rapporto deve essere visualizzato “NaN”.

Nota Bene 2: le stringhe su una riga possono essere separate una dall'altra da un numero qualsiasi di *whitespace* (spazi o tab).

Nota Bene 3: la lettura da file può sollevare un'eccezione controllata, per la quale il compilatore si aspetta indicazioni esplicite. Basta aggiungere all'intestazione del main la clausola `throws FileNotFoundException`:

```
public static void main(String[] args) throws FileNotFoundException
```

Vincoli

Non ci sono vincoli sul numero di righe del file, né sul numero di stringhe in una riga.

Esempio

Sul seguente input:

```
tutteminuscole  
TUTTEMAIUSCOLE
```

```
paperin12 prova pari dispari cavoloneso
```

```
minuscolecon1numero
```

```
12345 pippo=pluto
```

```
PAP)(
```

```
wqeuiyq!dkajh
```

```
<kjhaks  !92837127
```

il programma deve emettere il seguente output:

```
Riga: tutteminuscole  
Dispari:0  
Rapporto: 0.0  
Riga: TUTTEMAIUSCOLE  
Dispari:0  
Rapporto: 0.0  
Riga:  
Dispari:0  
Rapporto: NaN  
Riga: paperin12 prova pari dispari cavoloneso  
Dispari:3  
Rapporto: 0.6  
Riga:  
Dispari:0  
Rapporto: NaN
```

```
Riga: minuscolecon1numero
Dispari:1
Rapporto: 1.0
Riga:
Dispari:0
Rapporto: NaN
Riga: 12345 pippo=pluto
Dispari:2
Rapporto: 1.0
Riga:
Dispari:0
Rapporto: NaN
Riga: PAP)(
Dispari:1
Rapporto: 1.0
Riga:
Dispari:0
Rapporto: NaN
Riga: wqeuiyq!dkajh
Dispari:1
Rapporto: 1.0
Riga:
Dispari:0
Rapporto: NaN
Riga: <kjhaks !92837127
Dispari:2
Rapporto: 1.0
==+--+--+--+--+
```

Consegna

Si ricorda che le classi devono essere tutte *public* e che vanno consegnati tutti (e soli) i file *.java* prodotti. NON vanno consegnati i *.class*. Per la consegna, eseguite l'upload dei SINGOLI file sorgente (NON un file archivio!) dalla pagina web: <http://upload.di.unimi.it> nella sessione che vi è stata indicata.

Per ritirarsi fare l'upload di un file vuoto di nome `ritirato.txt`.
