

Esercizi per il Corso di Metodologie di Programmazione

Capitolo 11

Università degli Studi di Roma "La Sapienza"

Disclaimer: le consegne degli esercizi 1 – 9 non contengono nessun riferimento alla gestione delle eccezioni. Ciò nonostante, la loro applicazione è contemplata, compatibilmente con la consegna di ciascun esercizio.

Esercizio 1

Eseguire i seguenti passaggi:

- Creare a mano un file di nome *hello.txt* e scrivere al suo interno la stringa "Hello, World!"
- Chiudere il file
- Aprire nuovamente lo stesso file, questa volta tramite codice
- Leggere il contenuto del file, memorizzarlo in una stringa e stamparlo a video

Esercizio 2

Scrivere un programma che legga un file, richiesto da input all'utente, e che elimini le eventuali righe vuote, memorizzando invece quelle valorizzate. Successivamente, riaprire tale file e scrivere ciascuna riga precedentemente memorizzata, sovrascrivendo eventualmente il contenuto.

Esempio: contenuto nella cartella 2. All'interno di tale cartella sarà possibile trovare *input_2.txt* e *input_2_after.txt*, contenenti rispettivamente l'input e l'output dell'esercizio.

Esercizio 3

Scrivere un programma che simuli la funzione *trim* su file. Esso dovrà modificare il contenuto di un file richiesto in input dall'utente, eliminando le eventuali righe vuote all'inizio e alla fine.

Esempio: contenuto nella cartella 3. All'interno di tale cartella sarà possibile trovare il file di input *lines.txt* e *lines_after.txt*, contenenti rispettivamente l'input e l'output dell'esercizio.

Esercizio 4

Scrivere un programma che chieda all'utente il percorso del file di input e di output. Successivamente, il programma leggerà ciascuna riga del file di input e la scriverà in quello di output, in modo tale che ciascuna riga cominci con un riferimento alla propria posizione. Se il file letto è:

*Mary had a little lamb
Whose fleece was white as snow.
And everywhere that Mary went,
The lamb was sure to go!*

il programma dovrà creare il seguente file di output:

/ 1 */ Mary had a little lamb
/* 2 */ Whose fleece was white as snow.
/* 3 */ And everywhere that Mary went,
/* 4 */ The lamb was sure to go!*

Esercizio 5

Risolvere nuovamente l'esercizio precedente, consentendo all'utente di specificare il nome dei file di input e output da riga di comando o valorizzando direttamente le variabili corrispondenti all'interno del codice. In caso di parametro mancante o di variabile senza contenuto, il programma dovrà chiedere il nome del file mancante durante l'esecuzione.

Esercizio 6

Scrivere un programma che legga un file, il cui nome è richiesto in input dall'utente, contenente due colonne di numeri in virgola mobile, separate da spazi. Visualizzare il valore medio di ciascuna colonna.

Consiglio: la cartella 6 contiene un file chiamato *input_6.txt* da utilizzare come esempio di input.

Esercizio 7

Scrivere un programma che chieda all'utente il nome di un file la cui struttura è formata da un insieme di righe contenenti dei numeri in virgola mobile, separati da uno spazio. Successivamente, il programma dovrà leggere ciascuna riga, stampandone il valore medio. In caso di riga vuota, la media sarà pari a 0.

Consiglio: la cartella 7 contiene un file chiamato *data.in* e *output_7.txt* che rappresentano rispettivamente il file di input, da trattare come nell'esercizio precedente, e il relativo output.

Esercizio 8

Scrivere un programma che legga un file, una riga per volta, e scriva le righe in un altro file in ordine inverso. Ipotizziamo che il file *input.txt* abbia questo contenuto:

*Mary had a little lamb
Whose fleece was white as snow
And everywhere that Mary went
The lamb was sure to go.*

Eseguendo il programma in questo modo:

```
java ReverseLines input.txt output.txt
```

il file *output.txt* dovrà avere questo contenuto:

*The lamb was sure to go.
And everywhere that Mary went
Whose fleece was white as snow
Mary had a little lamb*

Consiglio: la cartella 8 contiene due file chiamati *input.txt* e *output.txt* con l'esempio di input e output del programma.

Esercizio 9

Scrivere un programma che inverta i caratteri di ciascuna riga all'interno di un file passato come parametro dall'utente. Suppongo che *input.txt* abbia questo contenuto:

*Mary had a little lamb
Whose fleece was white as snow
And everywhere that Mary went
The lamb was sure to go.*

Eseguendo il programma in questo modo:

```
java Reverse input.txt
```

il file *input.txt* dovrà avere questo contenuto:

*,bmal elttil a dah yraM
,bmal elttil ,bmal elttil
.wons sa etihw saw eceelf sti ,bmal elttil a dah yraM
,tnew yraM taht erehwyreve dnA
,tnew yraM ,tnew yraM
.og ot erus saw bmal eht ,tnew yraM taht erehwyreve dna*

Consiglio: la cartella 9 contiene due file chiamati *input.txt* e *outfile.txt* con l'esempio di input e output del programma.

Esercizio 10

Modificare la classe *BankAccount* in modo che lanci l'eccezione *IllegalArgumentException* quando viene costruito un oggetto il cui conto ha saldo negativo, quando viene versata una quantità di denaro negativa o quando viene prelevata una somma che non sia compresa tra 0 e il saldo del conto. Scrivere un programma di test che provochi il lancio di tutte e tre le eccezioni, catturandole.

Consiglio: la cartella 10 conterrà il file *BankAccount.java*, da cui partire per la modifica del codice, e *output_10.txt* con un esempio di output.

Esercizio 11

Ripetere l'esercizio precedente, ma lanciare eccezioni di tre tipi definiti manualmente.

Esempio: contenuto nel file *output_11.txt* all'interno della cartella 11.

Esercizio 12

Scrivere un programma che chieda all'utente di inserire un insieme di valori in virgola mobile. Nel caso in cui l'utente inserisca un valore non numerico, il programma dovrà dargli una seconda possibilità di inserimento per tale valore. In caso di due tentativi di inserimento consecutivi errati, il programma dovrà terminare. Nel caso invece in cui l'inserimento sia andato a buon fine, visualizzare il totale dei valori inseriti. Usare la gestione delle eccezioni per identificare i valori di ingresso non validi.

Esempio: contenuto nel file *output_12.txt* all'interno della cartella 12.