# Esercitazione 4



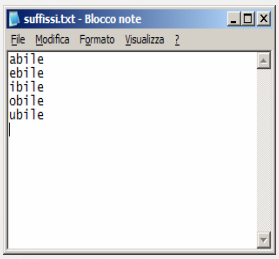
## Esercitazione

- In ***Psicolinguistica*** vengono studiate le modalità secondo le quali alcuni aspetti sublessicali dei termini possono influenzare alcuni processi cognitivi (*abilità di lettura*, *scrittura*, *parlato*, *memoria*, etc.) legati al linguaggio.
- Ci sono due tipi di lettura:
  - **Lettura lessicale**: influenzata dagli aspetti lessicali dei termini, tipo la ***frequenza***, la ***lunghezza***, la ***concretezza***, la ***familiarità***, l'***interpretabilità***, etc.
  - **Lettura sublessicale**: influenzata dagli aspetti sublessicali dei termini, tipo il ***fonema iniziale***, la ***frequenza dei bigrammi*** che compongono il termine, la ***struttura sillabica***, la ***difficoltà ortografica***, etc.
- Per eseguire degli esperimenti in maniera corretta, è quindi necessario essere in grado di misurare e correlare le precedenti caratteristiche (ad esempio, la ***frequenza*** degli aspetti ortografici sublessicali) in modo da avere una misura concreta della categoria sintattica di appartenenza di un termine.
- A questo scopo è stato preparato un file contenente una ***lista di termini*** (**lemmi.txt**) nonché la loro ***frequenza***, calcolata in base al numero di volte in cui questi sono presenti in un *corpus* di 1.500.000.000 parole prese da riviste, giornali e letteratura varia.

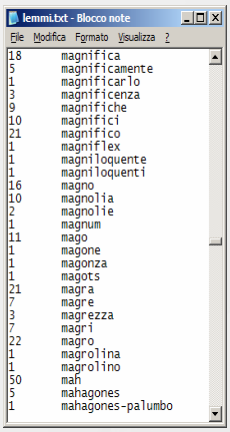
Planning & Scheduling Team
ISTC-CNR

# Esercitazione

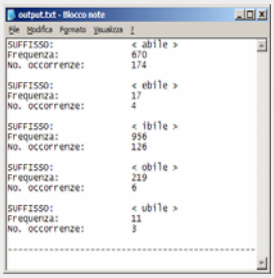
- Scrivere un programma che calcoli la frequenza con cui una serie di *suffissi* (presenti in un file che possiamo chiamare **suffissi.txt**), risultino presenti nell'insieme di termini presenti nel file **lemmi.txt**. Il risultato dell'analisi dovrà essere scritto in un file che chiameremo **output.txt**.
- Esempio:




**suffissi.txt**




**lemmi.txt**




**output.txt**



# Esercitazione

- Il programma si aspetta che il file **lemmi.txt** sia composto da righe che abbiano tutte il seguente fomat:

**<int> <'\t'> <String>**

- La lettura da file avverrà attraverso il metodo ***readLine()*** della classe **BufferedReader**, il quale restituisce in formato **String** una intera riga da un file di testo:

| CLASSI | METODI |
|---|---|
| ***BufferedReader*** | readLine() |

```
BufferedReader inFile = new BufferedReader(new FileReader(nomeFile));
```

oppure:

```
FileReader fin = new FileReader(nomeFile);
BufferedReader inFile = new BufferedReader(fin);
```

## Esercitazione

- La scrittura su file avverrà attraverso il metodo ***println()*** della classe **PrintWriter**.

| CLASSI | METODI |
|---|---|
| ***PrintWriter*** | println() |
| ***BufferedReader*** | readLine() |

```
PrintWriter ps = new PrintWriter(new FileOutputStream(nomeFile));
```

oppure:

```
FileOutputStream fout = new FileOutputStream(nomeFile);
PrintWriter ps = new PrintWriter(fout);
```



## Esercitazione

Processo di trasformazione da **String** a **int**:

da **String** freq_s = “123”
a **int** freq = 123;

1. Si istanzia un oggetto della classe **Integer** nel seguente modo:

```
Integer freqI = new Integer(freq_s);
```

2. Si utilizza il metodo ***intValue()*** messo a disposizione dalla classe **Integer**:

```
freq = freqI.intValue();
```

## Esercitazione

Utilizzo del metodo ***indexOf()*** della classe **String**:

Il metodo **stringa.*indexOf(char c)*** restituisce l'indice relativo alla posizione della prima occorrenza del carattere ***c*** all'interno dell'oggetto **stringa**.

Esempio:

```
String riga = "riga di prova";
int pos;

pos = riga.indexOf('p');
```

le precedenti linee di codice restituiranno **pos = 8**, in quanto:

| riga = | r | i | g | a | | d | i | | p | r | o | v | a |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | **8** | 9 | 10 | 11 | 12 |

Planning & Scheduling Team
ISTC-CNR

## Esercitazione

Utilizzo del metodo ***substring()*** della classe **String**:

Il metodo **stringa.*substring(int inizio, int fine)*** restituisce la sottostringa della stringa **stringa** estratta dalla posizione ***inizio*** alla posizione ***fine***.

Esempio:

```
String riga = "particolarmente";
String sottoriga;
int inizio = 4;
int fine = 8;

sottoriga = riga.substring(inizio, fine);
```

le precedenti linee di codice restituiranno **sottoriga = "icola"**, in quanto:

| riga = | p | a | r | t | **i** | **c** | **o** | **l** | **a** | r | m | e | n | t | e |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | **4** | 5 | 6 | 7 | **8** | 9 | 10 | 11 | 12 | 13 | 14 |
| | | | | | inizio | | | | fine | | | | | | |

Planning & Scheduling Team
ISTC-CNR

## Esercitazione

```
import java.io.*;

class suffixProject
{
          public static void main(String args[]) throws IOException
          {
                    int frequenza, count;
                    int freq=0;

                    FileOutputStream fout = new FileOutputStream("output.txt");
                    PrintWriter ps = new PrintWriter(fout);

                    //Attenzione! Il file deve terminare con un RETURN!
                    FileReader fin1 = new FileReader("suffissi.txt");
                    BufferedReader inFile1 = new BufferedReader(fin1);

                    //legge l'intera riga del file dei suffissi
                    String suffisso = inFile1.readLine();
                    while(!suffisso.equals(""))
                    {
                              //Attenzione! Il file deve terminare con un RETURN!
                              FileReader fin2 = new FileReader("lemmi.txt");
                              BufferedReader inFile2 = new BufferedReader(fin2);

                              frequenza = 0; //riazzeriamo la frequenza
                              count = 0; //riazzeriamo il contatore di occorrenze del suffisso

                    //CONTINUA…
```



## Esercitazione

```
…CONTINUA
//legge l'intera riga del file dei lemmi
String riga = inFile2.readLine();
while(!riga.equals(""))
{
          String freq_s = ""; //questa stringa conterrà la frequenza
          String lemma = ""; //questa stringa conterrà il lemma
          //estraiamo la sottostringa relativa alla frequenza
          freq_s = riga.substring(0,riga.indexOf('\t'));
          freq_s = freq_s.trim(); //ed eliminiamo gli spazi superflui
          //convertiamo la stringa frequenza in un intero
          Integer freqI = new Integer(freq_s);
          freq = freqI.intValue();
          //estraiamo la sottostringa relativa al lemma
          lemma = riga.substring(riga.indexOf('\t')+1);
          lemma = lemma.trim(); //ed eliminiamo gli spazi superflui

          //Ricerca di 'suffisso' alla fine di 'lemma'
          int dimsuffix = suffisso.length();
          int dimlemma = lemma.length();
          //se la stringa 'suffisso' è effettivamente un suffisso della stringa 'lemma'...
          if((dimsuffix <= dimlemma)&&
                    lemma.substring((dimlemma-dimsuffix),dimlemma)).equals(suffisso))
          {
                    ++count; //aumentiamo il contatore di occorrenze del suffisso
                    frequenza+=freq; //incrementiamo il valore della frequenza
          }
          riga = inFile2.readLine();
}
…CONTINUA
```

# Esercitazione

```
…CONTINUA

                        System.out.println("SUFFISSO: \t\t< " + suffisso + " >");
                        System.out.println("Frequenza: \t\t" + frequenza);
                        System.out.println("No. occorrenze: \t" + count + "\n");
                        ps.println("SUFFISSO: \t\t< " + suffisso + " >");
                        ps.println("Frequenza: \t\t" + frequenza);
                        ps.println("No. occorrenze: \t" + count);
                        ps.println();
                        inFile2.close();

                        suffisso = inFile1.readLine();
                }

                inFile1.close();
                ps.close();
        }
}
```