

FONDAMENTI DI INFORMATICA II (Prof. E. Lamma) – 22 Luglio 2010
PROVA PRATICA – LINGUAGGIO JAVA

NOTA PER LA CONSEGNA DEL COMPITO

Il compito deve essere stato **compilato ed eseguito**. E' importante che tutti i file e le classi si chiamino come richiesto. **Nel testo .java (come commento) va indicato il nome dello studente.**

Esercizio (da realizzare in un UNICO FILE .java)

Nella soluzione, prediligere il maggior riutilizzo di codice e la maggiore protezione possibile.

- Si realizzi una classe **Studente** che rappresenta uno studente. Tale classe ha tre attributi: **nome** e **cognome** di tipo **String** e **matricola** intero. La classe codifica un metodo costruttore a tre argomenti e definisce un metodo **toString()** che restituisce una stringa contenente il nome, il cognome e la matricola. La classe inoltre implementa l'interfaccia **Comparable** e definisce il metodo **compareTo(Object x)** come il seguente:

```
public int compareTo(Object x) {  
    Studente p = (Studente) x;  
    return cognome.compareTo(p.cognome);  
}
```

- Per convertire la stringa della matricola voto in un intero utilizzare il metodo **Integer.parseInt(stringa)**. Tale metodo restituisce l'intero convertito dalla stringa.
- Si realizzi un metodo **main** in una classe **Prova** che legga da input i dati di alcuni studenti, fino a quando non viene inserita una stringa vuota nel cognome e li memorizzi in una lista ordinata sul campo cognome.
- Si scelga il tipo di componente più adatto della JCF per la realizzazione di tale lista.
- Stampi poi su file di testo OUTPUT.TXT il contenuto di tale lista.

Esempio di input:

```
Mario Rossi      12345  
Elena Bianchi   23123  
Giovanni Rana   23456
```

Esempio di output nel file OUTPUT.TXT:

```
Bianchi Elena matr. 23123  
Rana Giovanni matr. 23456  
Rossi Mario matr. 12345
```

```

// Prova.java
import java.util.*;
import java.io.*;

public class provagiugno {

    public static void main(String args[])throws IOException {

        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

        ArrayList lista = new ArrayList();
        System.out.println("inserire cognome");
        String cognome = br.readLine();
        while(!cognome.equals("")) {
            System.out.println("inserire nome");
            String nome = br.readLine();
            System.out.println("inserire matricola");
            String matricola = br.readLine();
            int matr = Integer.parseInt(matricola);
            Studente s = new Studente(cognome, nome, matr);
            lista.add(s);
            System.out.println("inserire cognome");
            cognome = br.readLine();
        }
        br.close();
        Collections.sort(lista);
        PrintWriter pw = new PrintWriter(new FileWriter("output.txt"));
        for(Iterator i = lista.iterator(); i.hasNext();) {
            pw.println(i.next());
        }
        pw.close();
    }
}

class Studente implements Comparable {
    private String nome;
    private String cognome;
    private int matricola;

    public Studente(String c, String n, int m) {
        this.cognome = c;
        this.nome = n;
        this.matricola = m;
    }

    public String toString() {
        return (cognome + " " + nome + " matr. " + matricola);
    }

    public int compareTo(Object x) {
        Studente p = (Studente) x;
        return cognome.compareTo(p.cognome);
    }
}

```