L’interfaccia Comparable

Una List contiene oggetti che possono essere ordinati secondo un ordine naturale utilizzando il medodo statico Collections.sort(List list) se la classe degli oggetti implementa l’interfaccia Comparable.
Il solo metodo richiesto dall’interfaccia Comparable è compareTo che ha come argomento un oggetto dello stesso tipo di quelli che si intendono ordinare e come valore restituito un intero.
Quindi la chiamata al metodo è:
int result = x.compareTo(y)
con x e y oggetti dello stesso tipo di quelli nella lista e result un intero che assume valore:

- `<0 se x<y`
- `==0 se x==y`
- `>0 se x>y`

Osserva il seguente esempio:

Main.java

```
import java.util.ArrayList;
import java.util.Collections;
import java.util.List;

public class Main {

    public static void main(String[] args) {

        // FIRST SECTION

        List<String> myStringList = new ArrayList<String>();

        myStringList.add(new String("Germany"));
        myStringList.add(new String("French"));
        myStringList.add(new String("United Kingdom"));
        myStringList.add(new String("Italy"));
        myStringList.add(new String("Spain"));

        System.out.println("UNSORTED LIST:");
        for (String item : myStringList) {
            System.out.println(item);
        }

        // SECOND SECTION

        System.out.println("\nSORTED LIST:");
        Collections.sort(myStringList);
```

```
        for (String item : myStringList) {
            System.out.println(item);
        }

        // THIRD SECTION

        List<Country> myCountryList = new ArrayList<Country>();

        Country germany = new Country("Germany", 357021);
        Country french = new Country("French", 547030);
        Country unitedKingdom = new Country("United Kingdom", 244820);
        Country italy = new Country("Italy", 301230);
        Country spain = new Country("Spain", 504851);

        myCountryList.add(germany);
        myCountryList.add(french);
        myCountryList.add(unitedKingdom);
        myCountryList.add(italy);
        myCountryList.add(spain);

        System.out.println("\nSORTED LIST BY AREA:");
        Collections.sort(myCountryList);

        for (Country item : myCountryList) {
            System.out.println(item.getName());
        }
    }
}
```

Country.java

```
public class Country implements Comparable<Country> {

    private String name;
    private int area;

    public Country(String name, int area) {
      this.name = name;
      this.area = area;
    }

    public String getName() {
      return name;
    }

    public void setName(String name) {
      this.name = name;
    }
```

```
    public int getArea() {
      return area;
    }

    public void setArea(int area) {
      this.area = area;
    }

    @Override
    public int compareTo(Country o) {
      int result;
      result = area - o.getArea();
      return result;
    }
}
```

L’output è il seguente:

```
UNSORTED LIST:
Germany
French
United Kingdom
Italy
Spain

SORTED LIST:
French
Germany
Italy
Spain
United Kingdom

SORTED LIST BY AREA:
United Kingdom
Italy
Germany
Spain
French
```

Nella prima sezione l’elenco è stampata nell’ordine in cui gli elementi sono stati aggiunti alla lista.
Nella seconda sezione l’ordinamento con il metodo Collections.sort è possibile perchè la classe String implementa l’interfaccia Comparable è l’ordinamento è quello alfabetico.
Nella terza sezione la lista è composta da oggetti (Country) che implementano

Comparable è quindi l’ordinamento è crescente per area per come è stato implementato il metodo compareTo.