

INTERFACCE

CARATTERISTICHE GENERALI

- Serve per definire una proprietà astratta di un oggetto
- Sintassi:

```
public interface NomeInterfaccia {  
    firme di metodi NON statici  
}
```

- Caratteristiche interfaccia
 - NO costruttori
 - NO variabili di istanza
 - contiene le firme ma **non** l'implementazione del metodo
- Per implementare una interfaccia in una classe: operatore `implements`
 - si possono implementare + interfacce (separate da una virgola)
- Una stessa interfaccia può estendere altre interfacce
- È possibile definire una variabile il cui tipo è un'interfaccia
 - si può fare il cast da tipo interfaccia a classe
 - se si sbaglia: `ClassCastException`

Polimorfismo e interfacce

Solo in esecuzione la JVM determina il tipo di oggetto a cui una variabile si riferisce e chiama metodo corrispondente

ORDINAMENTO DI OGGETTI

COMPARABLE

- Il tutti gli oggetti che implementano comparable posso applicare `compareTo()`
- Il metodo `compareTo` deve definire una **relazione di ordine totale**
 - proprietà **antisimmetrica**
 - proprietà **riflessiva**
 - proprietà ****transitiva**

```
public interface Comparable {
    int compareTo(Object object);
}

public class MyClass implements Comparable {
    public int compareTo(Object object) {
        ...
    }
}

Comparable myObject = new MyClass();
```

```
if (a < b)
if (a > b)
if (a == b)
if (a != b)
```



```
if (a.compareTo(b) < 0)
if (a.compareTo(b) > 0)
if (a.compareTo(b) == 0)
if (a.compareTo(b) != 0)
```

SELECTION SORT PER OGGETTI COMPARABLE

```
public static void selectionSort(Comparable[] v, int vSize);
```

Best practice

- Implementare `equals()` se si implementa `compareTo()`

COMPARABLE CON TIPI DI DATI NUMERICI

- Basta usare delle **classi involucro** per ordinare numeri (Integer, Double, etc)