**Classe Random ed Ordinamento**

La classe ***Random*** della libreria Java permette di generare numeri casuali, cioè produce numeri arbitrari.   
Contiene i seguenti metodi:

* ***nextInt(n)***  restituisce un numero intero casuale compreso fra 0 (incluso) e n (escluso);
* ***nextDouble()*** restituisce un numero in virgola mobile casuale compreso fra 0 (incluso) e 1 (escluso).

**Esempio**:

*Random generatore = new Random();   
int d = 1+ generatore.nextInt(6);*

produce un numero casuale compreso tra zero e cinque inclusi.

**Esempio:**

import java.util.Random;

public class Dice   
{  public static void main(String[] args)   
 {  Random generator = new Random();   
      // getta il dado dieci volte

      for (int i = 1; i <= 10; i++)   
      {  int d = 1 + generator.nextInt(6);   
         System.out.print(d + ” ");   
      }   
      System.out.println();   
   }   
}

[*https://www.tutorialspoint.com/java/util/java\_util\_random.htm#*](https://www.tutorialspoint.com/java/util/java_util_random.htm)

È possibile ordinare un ***ArrayList*** utilizzando il metodo **sort ()** della classe **Collections.** Questo metodo accetta un oggetto list come parametro e ne ordina il contenuto in ordine crescente.

**Esempio**:

import java.util.ArrayList e java.util.Collections;

public class TestSort {

public static void main(String[] args){

ArrayList<String> list = new ArrayList<String>();

list.add("ghi");

list.add("def");

list.add("abc");

System.out.println("\n Non ordinato : " + list);

Collections.sort(list);

System.out.println("\n Ordinato : " + list

}

}

Per ordinare liste di oggetti in Java sono a disposizione 2 interfacce:

1. **java.lang.Comparable:**

L’interfaccia ***java.lang.Comparable*** di solito si utilizza per definire ***l’ordinamento “naturale” di un oggetto*** e consta di un metodo con firma:

**int compareTo(T o),** che paragona l’oggetto corrente (this) con l’oggetto fornito in input al metodo (o).

Il risultato sarà:

* un intero positivo se this>o
* 0 se sono uguali
* un intero negativo se this<0

**Esempio**:

public class Persona ***implements Comparable<***Persona***>*** {

public Persona(String nome,String cognome,int eta){

this.nome=nome;

this.cognome=cognome;

this.eta=eta;

}

private String nome;

private String cognome;

private int eta;

***@Override***

**public int compareTo(Persona o) {**

**return this.cognome.compareTo(o.cognome);**

**}**

}

1. **java.util.Comparator:**

L’interfaccia ***java.util.Comparator*** si utilizza invece quando si vogliono definire ***ordinamenti alternativi*** dell’oggetto. Si ***crea*** dunque una ***classe a parte*** che implementa l’interfaccia con il seguente metodo:

**public int compare(T a,T b);**

Il risultato sarà:

* un intero positivo se a>b
* 0 se sono uguali
* un intero negativo se a<b

**Esempio**:

import java.util.Comparator;

public class PersonaEtaComparator implements Comparator<Persona> {

***@Override***

**public int compare(Persona p1, Persona p2) {**

**int retVal=0;**

**if(p1.getEta()>p2.getEta()){**

**retVal=1;**

**}**

**else if(p1.getEta()<p2.getEta()){**

**retVal=-1;**

**}**

**return retVal;**

**}**

}