**LIBRERIE:**

<system>

System.out.println(“string”); // cout + endl

System.out.print(“string”); // cout

System.exit(1);

System.arraycopy(Object src, int srcPos, Object dest, int destPos, int length);

System.currentTimeMillis(); // restituisce un long dalla epoca Unix

System.gc(); // Garbage collector force start

System.runFinalization(); // Chiamato dopo il gc per finalize()

<Math>

Math.PI <- costante

Math.random() <- numero tra 0 e 1

<Varie>

Integer.parseint(string); // Parsing di interi e float in stringhe

Float.parseFloat(string);

String.charAt(0); // Indexing di stringhe

**UTILIZZO DA RIGA DI COMANDO:**

Compilazione:

javac -verbose -sourcepath "src/" -d "build/classes/" "src/it/unitn/prog2/HelloWorldApp.java"

Run:

java -verbose -classpath build/classes it.unitn.prog2.HelloWorldApp args

Jar:

jar cfe HelloWorldApp.jar it.unitn.prog2.HelloWorldApp -C build/classes/ it/unitn/prog2/HelloWorldApp.class

ALTRO / da ricatalogare:

import javax.swing.JOptionPane;

JOptionPane.showInputDialog(“string”);

AssertionError e <- quando si usa assert

e.printStackTrace();

**USO DEGLI ITERATORI:**

Iterator i=x.iterator();

while (i.hasNext()) {

if (!cond(i.next()))

i.remove();

}

**METODI DA OVERRIDARE:**

public String toString() {

return nome + " " + cognome;

}

public int compareTo(Persona a) {

int res;

if (cognome.compareTo(a.cognome) != 0)

res = cognome.compareTo(a.cognome);

else

res = nome.compareTo(a.nome);

return res;

}

public boolean equals(Object a) { // Equals deve implementare una relazione

boolean res; // di uguaglianza

if (a == null)

res = false; // Richiesto dalla definizione!

else if (!(a instanceof Persona)) // a.getClass() != this.getClass()

res = false;

else

res = cognome.equals(((Persona)a).cognome); // altro implementabile

return res;

}

public int hashCode() {

int result = 7;

result = 31 \* result + nome.hashCode();

result = 31 \* result + cognome.hashCode();

return result;

}