

I membri della classe System.

Costanti pubbliche statiche:

- `java.io.PrintStream err`
- `java.io.InputStream in`
- `java.io.PrintStream out`

Metodi pubblici statici:

- `void arraycopy(Object src, int srcPos, Object dest, int destPos, int length)`
- `long currentTimeMillis()`
- `void exit(int status)`
- `void gc()`
- `java.util.Properties getProperties()`
- `String getProperty(String key)`
- `String getProperty(String key, String default)`
- `SecurityManager getSecurityManager()`
- `void runFinalization()`
- `void setErr(java.io.PrintStream err)`
- `void setIn(java.io.InputStream in)`
- `void setOut(java.io.PrintStream out)`
- `void setProperties(java.util.Properties properties)`
- `String setProperty(String key, String value)`
- `void setSecurityManager(SecurityManager s)`

Classe System

La classe System ha il compito di interfacciare il programma Java con il sistema operativo sul quale sussiste la virtual machine.

Tutto ciò che esiste nella classe System è dichiarato statico.

VARIABILI

Iniziamo subito esaminando tre attributi statici, che rappresentano i flussi (stream) di informazioni scambiati con la console (standard input, standard output, standard error):

```
static PrintStream out  
  
static PrintStream err  
  
static InputStream in
```

- Ciascuno di questi tre attributi è un oggetto e sfrutta i metodi della classe relativa.
- l'oggetto out è di tipo PrintStream e viene usato per indicare l'output di default del sistema.

- l'oggetto `err`, anch'esso di tipo `PrintStream`, che viene usato per segnalare gli errori che avvengono durante l'esecuzione del programma.
- l'oggetto `in` è di tipo `InputStream`: serve per ricevere il flusso di informazioni dallo standard input, `p.es` la tastiera.
- E' possibile modificare il puntamento di queste tre variabili verso altre fonti di input o di output usando i metodi statici `setOut()`, `setErr()` e `setIn()`.

METODI Principali

- il metodo `arraycopy()` permette di copiare il contenuto di un array in un altro.
- il metodo `exit(int code)` che consente di bloccare istantaneamente l'esecuzione del programma.

```
if (continua == false) {  
    System.err.println("Si è verificato un problema!");  
    System.exit(0);  
}
```

Altri metodi interessanti:

```
setProperty(String key, String value) e  
getProperty(String key) che servono rispettivamente ad impostare le  
proprietà del sistema e a recuperare informazioni sulle proprietà del  
sistema.
```

Gli oggetti di tipo `Properties` sono specializzazioni di tabelle hash di Java, semplici coppie chiave-valore.

Per esempio:

```
System.out.print("Versione Java Runtime Environment (JRE): ");  
System.out.println(System.getProperty("java.version"));  
*  
System.out.print("Java è installato su: ");  
System.out.println(System.getProperty("java.home"));
```

impostare una nuova proprietà mediante il codice:

```
System.setProperty("User.lastName", "Verdi");
```

Le proprietà automaticamente disponibili nell'ambiente Java.

Chiave	Valore
java.version	La versione di Java in uso.
java.vendor	Il produttore della versione di Java in uso.
java.vendor.url	L'URL del produttore della versione di Java in uso.
java.home	La directory di installazione di Java.
java.vm.specification.version	La versione delle specifiche della macchina virtuale in uso.
java.vm.specification.vendor	Il produttore delle specifiche della macchina virtuale in uso.
java.vm.specification.name	Il nome delle specifiche della macchina virtuale in uso.
java.vm.version	La versione della macchina virtuale in uso.
java.vm.vendor	Il produttore della macchina virtuale in uso.
java.vm.name	Il nome della macchina virtuale in uso.
java.specification.version	La versione delle specifiche di Java in uso.
java.specification.vendor	Il produttore delle specifiche di Java in uso.
java.specification.name	Il nome delle specifiche di Java in uso.
java.class.version	La versione delle classi di Java.
java.class.path	Il percorso delle classi di Java.
java.library.path	Il percorso delle librerie di Java.
java.io.tmpdir	Il percorso della directory dei file temporanei.
java.ext.dirs	I percorsi delle directory che contengono le estensioni di Java.
os.name	Il nome del sistema operativo in uso.
os.arch	L'architettura del sistema operativo in uso.
os.version	La versione del sistema operativo in uso.
file.separator	La sequenza per la separazione degli elementi dei percorsi nel sistema in uso.
path.separator	La sequenza per la separazione dei percorsi nel sistema in uso.
line.separator	La sequenza impiegata dal sistema in uso per esprimere il ritorno a capo.
user.name	Il nome dell'utente che sta usando l'applicazione.
user.home	La home directory dell'utente che sta usando l'applicazione.
user.dir	L'attuale cartella di lavoro dell'utente che sta usando l'applicazione.