

Javap: IL DISASSEMBLATORE

Javap è il disassemblatore contenuto nel JDK

- Dato un file `.class`, esso lo analizza ed estrae tutte le informazioni sulla classe (o interfaccia)
 - Per default mostra *solo l'interfaccia* della classe ma, con apposita opzione, può mostrare il disassemblato *di ogni metodo e funzione statica*
 - Con apposita opzione, può mostrare anche tutti i dati e metodi *privati*
 - Se la classe è stata compilata con l'opzione `-g`, può mostrare anche l'elenco delle *variabili locali*
 - L'output è su video, ma può essere ridiretto su file

SINTASSI

Javap [*opzioni*] *classfile*

Principali opzioni:

- public** mostra solo classi, dati e funzioni pubbliche
- protected** mostra solo classi, dati e funzioni pubbliche e protected
- package** (*default*) mostra solo classi, dati e funzioni pubbliche, protected, e con visibilità package
- private** mostra tutte le classi, i dati e le funzioni
- c** mostra anche il disassemblato di metodi e funzioni
- l** mostra anche le variabili locali (se la classe è stata compilata con l'apposita opzione **-g**)
- verbose** mostra anche la dimensione sullo stack, il numero delle variabili locali e degli argomenti di ciascun metodo

UNA CLASSE DI PROVA

```
public class Item {  
    protected int val;  
    private boolean status = true;  
    public Item() { val=7; }  
    public void print() { System.out.println(val); }  
    public int getValue(){ return val;}  
  
    protected static char flag = 'A';  
  
    public static void main(String a[]){  
        Item x = new Item();  
        x.print();  
        flag = 'z';  
        x.print();  
    }  
}
```

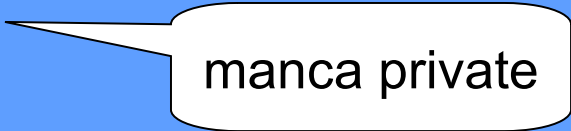
JAVAP ALL'OPERA - Caso base

Invocazione base: **javap Item**

Output:

Compiled from Item.java

```
public class Item extends java.lang.Object {  
    protected int val;  
    protected static char flag;  
    public Item();  
    public void print();  
    public int getValue();  
    public static void main(java.lang.String[]);  
    static {};  
}
```



manca private

JAVAP ALL'OPERA - Variante

Invocazione base: **javap -private Item**

Output:

Compiled from Item.java

```
public class Item extends java.lang.Object {  
    protected int val;  
    private boolean status;  
    protected static char flag;  
    public Item();  
    public void print();  
    public int getValue();  
    public static void main(java.lang.String[]);  
    static {};  
}
```

JAVAP ALL'OPERA - disass. (1/3)

Con disassemblato: **javap -c Item**

Output *(oltre all'interfaccia precedente):*

Method Item()

```
0 aload_0
1 invokespecial #1 <Method java.lang.Object()>
4 aload_0
5 bipush 7
7 putfield #2 <Field int val>
10 return
```

Method void print()

```
0 getstatic #3 <Field java.io.PrintStream out>
3 aload_0
4 getfield #2 <Field int val>
7 invokevirtual #4 <Method void println(int)>
10 return
```

- segue -

JAVAP ALL'OPERA - disass. (2/3)

Method int getValue()

```
0  aload_0
1  getfield #2 <Field int val>
4  ireturn
```

Method void main(java.lang.String[])

```
0  new #5 <Class Item>
3  dup
4  invokespecial #6 <Method Item()>
7  astore_1
8  aload_1
9  invokevirtual #7 <Method void print()>
12 bipush 122
14 putstatic #8 <Field char flag>
17 aload_1
18 invokevirtual #7 <Method void print()>
21 return
```

JAVAP ALL'OPERA - disass. (3/3)

```
Method static {}  
  0 bipush 65  
  2 putstatic #8 <Field char flag>  
  5 return
```

JAVAP ALL'OPERA - variabili locali

Se Item.java è stato compilato con -g

Invocazione: **javap -c -l Item**

Output (*oltre alle informazioni precedenti*):

```
Local variables for method Item()
```

```
Item this  pc=0, length=16, slot=0
```

```
Local variables for method void print()
```

```
Item this  pc=0, length=11, slot=0
```

```
Local variables for method int getValue()
```

```
Item this  pc=0, length=5, slot=0
```

```
Local variables for method void main(java.lang.String[])
```

```
java.lang.String[] a  pc=0, length=22, slot=0
```

```
Item x  pc=8, length=13, slot=1
```

RIFERIMENTI sul sito Sun

- **Javap:**

`http://java.sun.com/j2se/1.3/docs/tooldocs/
win32/javap.html#Options`

- **Mini tutorial:**

`http://developer.java.sun.com/developer/TechTips/
txtarchive/Aug00_Stu.txt`