

## L'array `args[]` per i valori passati da linea di comando

Le applicazioni Java possono ricevere in input i valori che l'utente indica da linea di comando. In pratica, i parametri vengono aggiunti al comando che invoca l'applicazione, subito dopo il nome della classe che deve essere eseguita. Successivamente l'interprete Java passa l'elenco dei parametri al metodo `main` dell'applicazione tramite un array di stringhe.

Nel seguito vengono presentate due applicazioni che eseguono un'elaborazione basata sui parametri ricevuti da linea di comando.

### ESEMPIO

**Realizzare un'applicazione che riceve da linea di comando due numeri e restituisce come risultato il loro prodotto.**

L'applicazione verifica se il numero di parametri è corretto e, in caso contrario, mostra la sintassi da usare e interrompe l'esecuzione.

**PROGRAMMA JAVA** (*Prodotto.java*)

```
class Prodotto
{
    public static void main(String args[])
    {
        if (args.length != 2)
        {
            System.out.println("Sintassi:");
            System.out.println(" java Prodotto <num1> <num2>");
            return;
        }

        try
        {
            int num1 = Integer.valueOf(args[0]).intValue();
            int num2 = Integer.valueOf(args[1]).intValue();

            System.out.println(num1*num2);
        }
        catch(Exception e)
        {
            System.out.println("Numeri non corretti");
            return;
        }
    }
}
```

Se si esegue l'applicazione da linea comandi con i seguenti parametri:

```
java Prodotto 10 4
```

si ottiene come risultato il numero 40.

## ESEMPIO

**Realizzare un'applicazione che riceve da linea di comando una sequenza di numeri ed estrae il valore minimo o massimo a seconda del parametro inserito.**

Il primo parametro dell'applicazione stabilisce se il numero da estrarre deve essere il minimo (*-min*) oppure il massimo (*-max*). Se il primo parametro non contiene uno dei precedenti valori, l'applicazione mostra la sintassi da utilizzare e interrompe l'esecuzione. Dal secondo parametro in poi viene inserita la sequenza di numeri tra cui dovrà essere cercato il valore minimo oppure il valore massimo.

### PROGRAMMA JAVA (*Estrazione.java*)

```
class Estrazione
{
    public static void main(String args[])
    {
        String help = "Sintassi:\n"
            + "java Estrazione <p> <numeri>\n"
            + "<p> -min : sceglie il minimo\n"
            + "    -max : sceglie il massimo\n"
            + "<numeri> : elenco di numeri";

        if (args.length < 2)
        {
            System.out.println(help);
            return;
        }

        String tipo = args[0];
        int valore;

        if (tipo.equals("-min"))
        {
            int minimo = Integer.valueOf(args[1]).intValue();

            for(int i=2; i<args.length; i++)
            {
                valore = Integer.valueOf(args[i]).intValue();
                if (valore < minimo)
                {
                    minimo = valore;
                }
            }

            System.out.println("Minimo = " + minimo);
        }
        else if (tipo.equals("-max"))
        {
            int massimo = Integer.valueOf(args[1]).intValue();
```

```

for(int i=2; i<args.length; i++)
{
    valore = Integer.valueOf(args[i]).intValue();
    if (valore > massimo)
    {
        massimo = valore;
    }
}

System.out.println("Massimo = " + massimo);
}
else
{
    System.out.println(help);
    return;
}
}
}

```

Si noti che per valutare il primo parametro è stato usato il comando

```
tipo.equals("-min")
```

che esegue il confronto tra la stringa contenuta nella variabile *tipo* e la stringa indicata tra parentesi.

Se si esegue l'applicazione da linea comandi con i seguenti parametri:

```
java Estrazione -max 4 5 91 50
```

si ottiene come risultato:

```
Massimo = 91
```