

Letture dei dati da tastiera con la classe Console

La lettura dei dati da tastiera può essere implementata usando le classi contenute nel package **java.io**. Nel libro di testo si è partiti dalla classe che descrive lo standard input *System.in* e, tramite la concatenazione di stream, si è ottenuto un oggetto *tastiera* su cui poter eseguire l'operazione di lettura *readLine*.

Il seguente frammento di codice mostra le istruzioni per la lettura di una stringa di testo.

```
InputStreamReader input = new InputStreamReader(System.in);
BufferedReader tastiera = new BufferedReader(input);

String nome;
try
{
    nome = tastiera.readLine();
}
catch(Exception e) {}
```

Oltre alla lettura dei dati tramite standard input, le librerie di Java offrono un oggetto predefinito, di classe **Console**, che svolge le stesse operazioni e, in aggiunta, ha dei metodi per la lettura mascherata delle password e per la scrittura formattata dei testi.

La classe *Console* è contenuta nel package *java.io* e, per poterla utilizzare, si deve richiamare il metodo **System.console()** nel seguente modo.

```
Console cons = System.console();
```

Tramite il nuovo oggetto *cons* di classe *Console* si possono utilizzare due metodi per la lettura dei dati:

- **readLine**, legge una riga di testo,
- **readPassword**, legge una password mascherando le lettere digitate.

Il metodo *readLine* restituisce un oggetto di classe *String*, mentre il metodo *readPassword* restituisce un array di *char*. Entrambi restituiscono il valore *null* se l'utente ha inserito il valore *ctrl-Z*.

Oltre ai metodi di lettura, la classe *Console* permette la stampa a video di una stringa di testo formattata usando il metodo **printf**.

Per esempio, il valore delle variabili *articolo* di tipo *String* e *prezzo* di tipo *double* può essere stampato all'interno di un messaggio di testo nel seguente modo:

```
cons.printf("Il prezzo di %s è %7.2f", articolo, prezzo);
```

Il primo parametro del metodo **printf** è il testo da stampare, mentre i successivi parametri sono i valori da inserire nel testo. L'inserimento dei parametri nel testo avviene nei punti in cui sono posizionati i caratteri %, seguiti dal tipo di formattazione. La sequenza **%s** indica che il primo valore da formattare è una stringa mentre la sequenza **%7.2f** indica che il secondo valore è un numero a cui vanno riservati 7 spazi per le cifre intere e 2 per quelle decimali. La formattazione di un numero intero è gestita dalla sequenza **%d**.

ESEMPIO

Consentire l'inserimento di una password composta da almeno otto caratteri e contenente almeno un numero.

Il programma chiede all'utente di inserire un nome e una password. I due valori vengono letti con i metodi `readLine` e `readPassword` della classe `Console` e vengono salvati rispettivamente nelle seguenti variabili:

```
String nome;  
char[] pwd;
```

Il controllo sulla lunghezza della password viene eseguito confrontando il valore di `pwd.length`, mentre la presenza di un valore numerico viene verificata con un ciclo su tutti gli elementi dell'array di `char`.

Per la stampa dei messaggi a video viene usato il metodo generico `System.out.println` e il metodo `cons.printf` con i parametri di formattazione. Il codice completo del programma è riportato di seguito.

PROGRAMMA JAVA (*ProgPassword.java*)

```
import java.io.*;  
  
class ProgPassword  
{  
    public static void main(String args[])  
    {  
        String nome;  
        char[] pwd;  
        boolean pwdValida = true;  
  
        Console cons = System.console();  
  
        System.out.print("Nome: ");  
        nome = cons.readLine();  
  
        System.out.print("Password: ");  
        pwd = cons.readPassword();  
  
        if (pwd.length < 8)  
        {  
            cons.printf("ERRORE: lunghezza=%d.", pwd.length);  
            System.out.println("La lunghezza minima deve essere 8.");  
  
            pwdValida = false;  
        }  
        else  
        {  
            boolean contieneNumero = false;
```

```
for(int i=0; i<pwd.length; i++)
{
    if (pwd[i] >= '0' && pwd[i] <= '9')
    {
        contieneNumero = true;
    }
}

if (!contieneNumero)
{
    System.out.println("ERRORE: deve contenere almeno un numero.");
    pwdValida = false;
}

if (pwdValida)
{
    cons.printf("La password per %s è valida.", nome);
}
}
```