

Programmazione 1 - Modulo C

Java: Istruzioni per input da tastiera

Marco Beccuti

Università degli Studi di Torino

Dipartimento di Informatica

Ottobre 2020





classe SavitchIn

classe SavitchIn

- Fino ad ora se volevamo valutare una qualunque espressione con variabili al suo interno dovevamo dichiarare le variabili, assegnare dei valori alle variabili, compilare ed eseguire;
- Se volevamo modificare i valori delle variabili bisognava ancora ricompilare ed eseguire;
- Oggi vedremo come utilizzare la classe **SavitchIn** che ci permette di richiedere all'utente l'input da tastiera.

Per utilizzare la classe SavitchIn

- Per utilizzare la classe **SavitchIn** dobbiamo avere il bytecode (file .class) nella directory in cui eseguiamo il programma
- Una volta ottenuto il file **SavitchIn.java (SIn.java)** basta compilarlo e tenere il file **SavitchIn.class** nella stessa directory in cui normalmente eseguite i programmi java.

Metodi per l'input della classe SavitchIn

- La classe **SavitchIn.java** mette a disposizione diversi metodi, ognuno per un diverso tipo di dati;
- Quando l'interprete Java incontra uno di questi metodi il programma si blocca in attesa che l'utente del programma inserisca un valore tramite tastiera;

Nome	Cosa Ritorna
<code>readLineInt()</code>	Un intero
<code>readLineLong()</code>	Un long
<code>readLineFloat()</code>	Un float
<code>readLineDouble()</code>	Un double
<code>readLineBoolean()</code>	Un boolean
<code>readLineChar()</code>	Un char
<code>readLineNonwhiteChar()</code>	Un char - esclusi i caratteri "invisibili"
<code>readLine()</code>	Una String che è l'intera linea inserita
<code>readLineWord()</code>	Una String che è la prima parola della linea

Esempio di uso della classe SavitchIn

- Una classe che legge da terminale un numero intero:

```
class ProvaSavitch {  
  
    public static void main(String[] args) {  
        System.out.print("Inserisci un numero: ");  
        int a = SavitchIn.readLineInt();  
        System.out.println("Hai inserito " + a);  
    }  
}
```

Esempio d'uso:

Inserisci un numero: 42

Hai inserito 42

Esempio di uso della classe SavitchIn

- Una classe che legge da terminale due numeri interi e esegue la divisione intera:

```
class ProvaSavitch2 {  
  
    public static void main(String[] args) {  
        System.out.print("Inserisci il numeratore: ");  
        int num = SavitchIn.readLineInt();  
        System.out.print("Inserisci il denominatore: ");  
        int den = SavitchIn.readLineInt();  
        if (den == 0) {  
            System.out.println("Non posso dividere "  
                + "per zero!");  
        } else {  
            System.out.println(num + " diviso " + den  
                + " fa " + (num / den)  
                + " con resto " + (num % den));  
        }  
    }  
}
```

Esercizi

- Scrivere un programma che richieda in input due numeri interi e stampi la loro media (puó non essere intera);
- Scrivere un programma che richieda di inserire due valori booleani a e b e valuti l'espressione (b and not a)
- Scrivere un programma che calcoli le soluzioni reali di una equazione di secondo grado. Si utilizzi il metodo `Math.sqrt(x)` per la radice quadrata e `Math.pow(x,2)` per calcolare il quadrato.

equazione completa $ax^2 + bx + c = 0$

$$\Delta > 0 \quad \longrightarrow \quad x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

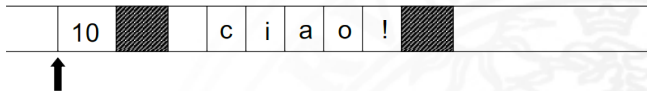
$$\Delta = 0 \quad \longrightarrow \quad x_1 = x_2 = -\frac{b}{2a}$$

$$\Delta < 0 \quad \longrightarrow \quad \text{equazione impossibile}$$

Un altro Esempio

	10		c	i	a	o	!	
--	----	--	---	---	---	---	---	--

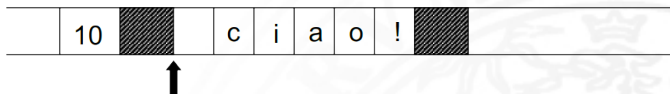
Un altro Esempio



```
int x = SavitchIn.readLineInt();  
char c = SavitchIn.readChar();  
char c = SavitchIn.readChar();  
char c = SavitchIn.readChar();  
char c = SavitchIn.readChar();  
char c = SavitchIn.readChar();  
String s = SavitchIn.readLine();
```

<input type="text"/>	x
<input type="text"/>	c
<input type="text"/>	s

Un altro Esempio



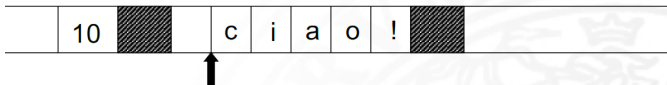
```
int x = SavitchIn.readLineInt();  
char c = SavitchIn.readChar();  
char c = SavitchIn.readChar();  
char c = SavitchIn.readChar();  
char c = SavitchIn.readChar();  
char c = SavitchIn.readChar();  
String s = SavitchIn.readLine();
```


10 x

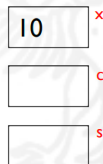
c

s

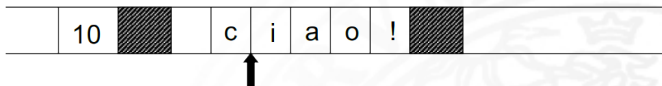
Un altro Esempio



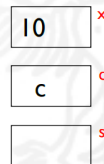
 `int x = SavitchIn.readLineInt();`
`char c = SavitchIn.readChar();`
`char c = SavitchIn.readChar();`
`char c = SavitchIn.readChar();`
`char c = SavitchIn.readChar();`
`char c = SavitchIn.readChar();`
`String s = SavitchIn.readLine();`



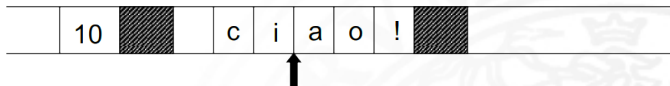
Un altro Esempio



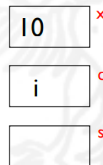
```
int x = SavitchIn.readLineInt();  
char c = SavitchIn.readChar();  
char c = SavitchIn.readChar();  
☛ char c = SavitchIn.readChar();  
char c = SavitchIn.readChar();  
char c = SavitchIn.readChar();  
String s = SavitchIn.readLine();
```



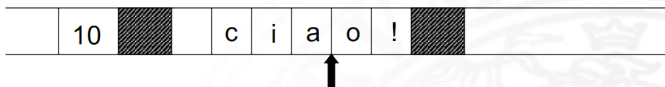
Un altro Esempio



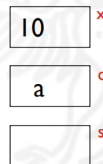
```
int x = SavitchIn.readLineInt();  
char c = SavitchIn.readChar();  
char c = SavitchIn.readChar();  
char c = SavitchIn.readChar();  
☛ char c = SavitchIn.readChar();  
char c = SavitchIn.readChar();  
String s = SavitchIn.readLine();
```



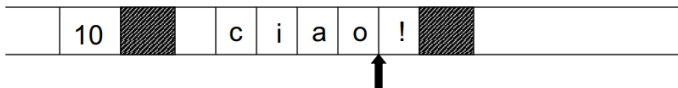
Un altro Esempio



```
int x = SavitchIn.readLineInt();  
char c = SavitchIn.readChar();  
char c = SavitchIn.readChar();  
char c = SavitchIn.readChar();  
char c = SavitchIn.readChar();  
☛ char c = SavitchIn.readChar();  
String s = SavitchIn.readLine();
```



Un altro Esempio



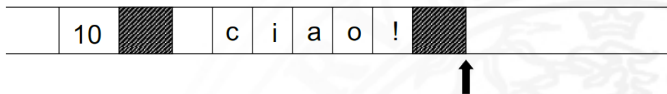
```
int x = SavitchIn.readLineInt();  
char c = SavitchIn.readChar();  
char c = SavitchIn.readChar();  
char c = SavitchIn.readChar();  
char c = SavitchIn.readChar();  
char c = SavitchIn.readChar();  
String s = SavitchIn.readLine();
```

10 x

o c

s

Un altro Esempio



```
int x = SavitchIn.readLineInt();  
char c = SavitchIn.readChar();  
char c = SavitchIn.readChar();  
char c = SavitchIn.readChar();  
char c = SavitchIn.readChar();  
char c = SavitchIn.readChar();  
String s = SavitchIn.readLine();
```

10 x

o c

! s