Stringhe e input/output

Istruzioni di input: la classe SavitchIn (= SIn del laboratorio)

public static String readLine()
/**Legge una riga di testo e la restituisce come valore di tipo String*/

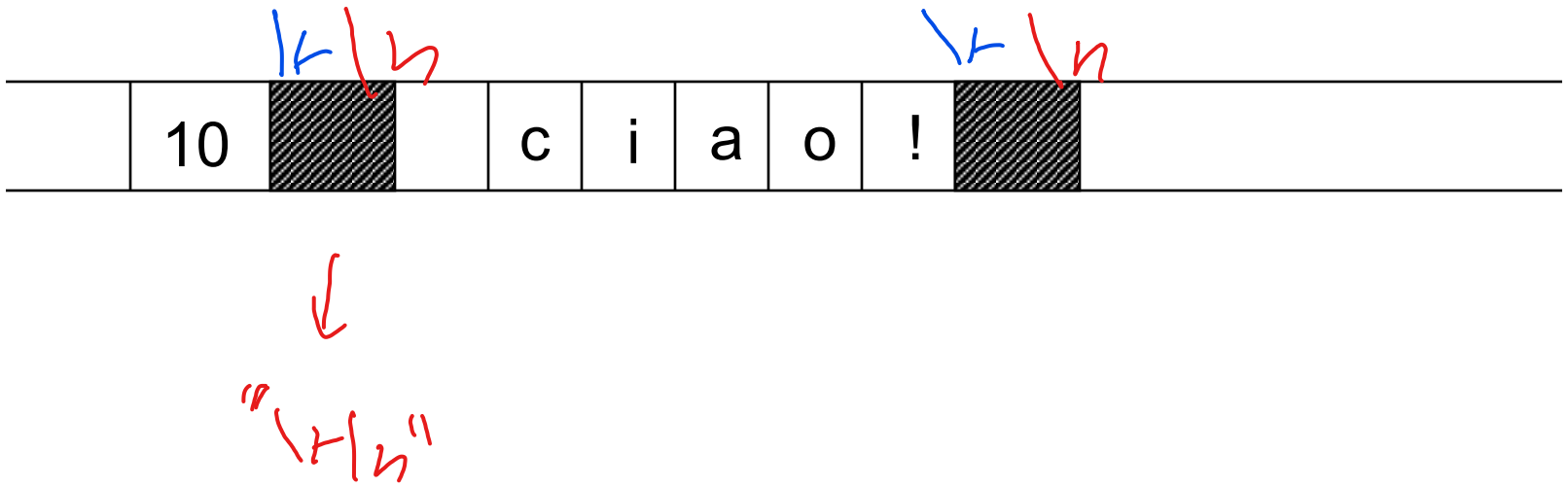
public static int readLineInt()
/**Legge un intero (preceduto o seguito a spazi) scritto su una riga e lo restituisce come valore di tipo Int*/

public static char readChar()
/**Legge il prossimo carattere, e lo restituisce come valore di tipo Char*/

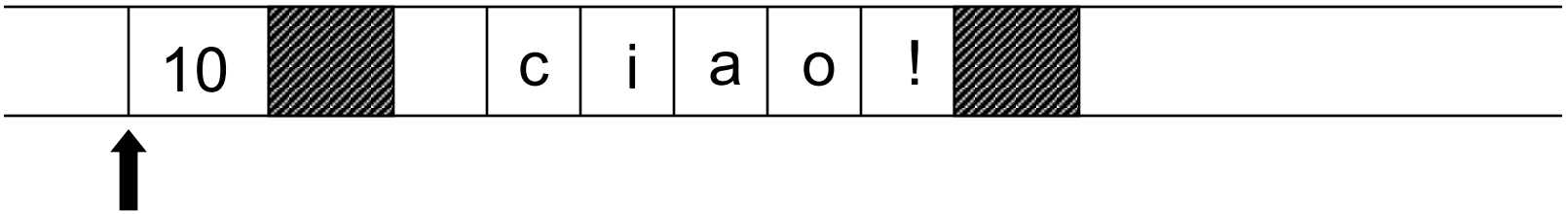
Ciascuno di questi metodi viene invocato in un programma scrivendo:

SavitchIn.<nome del metodo>

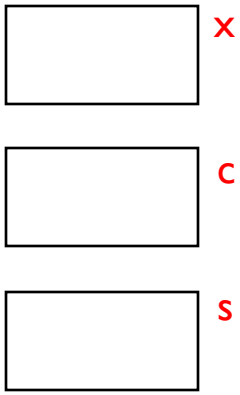
Il file di input



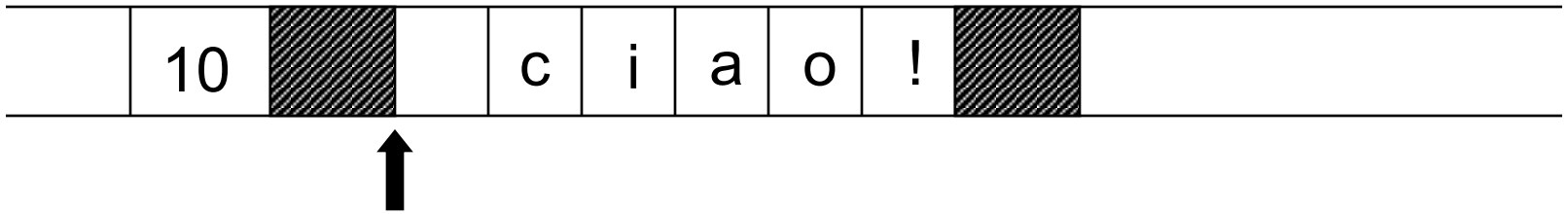
Il file di input



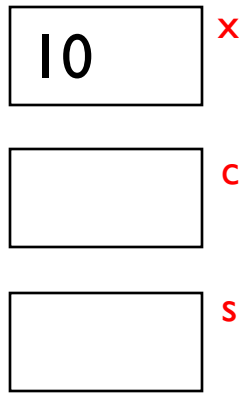

```
int x = SavitchIn.readLineInt();
char c = SavitchIn.readChar();
char c = SavitchIn.readChar();
char c = SavitchIn.readChar();
char c = SavitchIn.readChar();
char c = SavitchIn.readChar();
String s = SavitchIn.readLine();
```



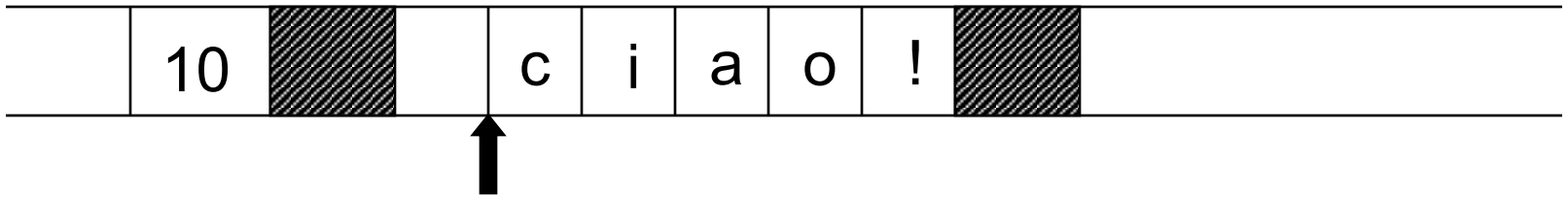
Il file di input



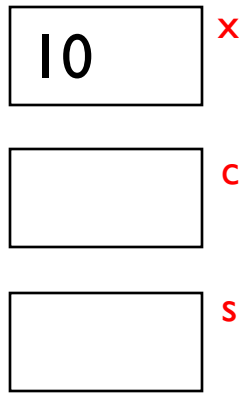

```
int x = SavitchIn.readLineInt();
char c = SavitchIn.readChar();
char c = SavitchIn.readChar();
char c = SavitchIn.readChar();
char c = SavitchIn.readChar();
char c = SavitchIn.readChar();
String s = SavitchIn.readLine();
```



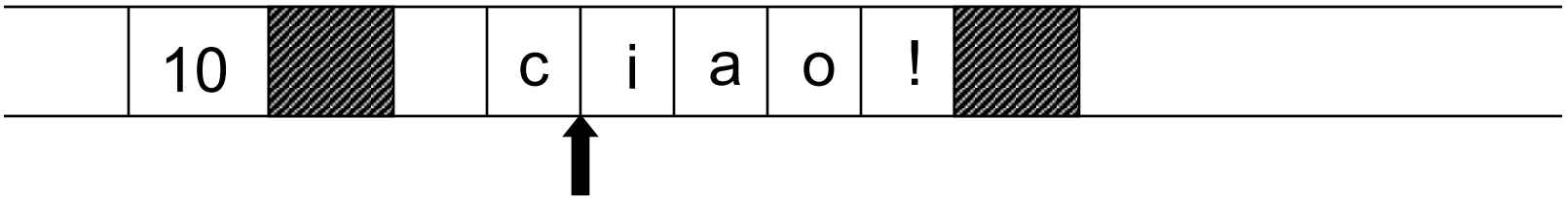
Il file di input



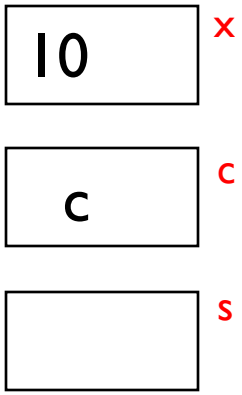

```
int x = SavitchIn.readLineInt();
char c = SavitchIn.readChar();
char c = SavitchIn.readChar();
char c = SavitchIn.readChar();
char c = SavitchIn.readChar();
char c = SavitchIn.readChar();
String s = SavitchIn.readLine();
```



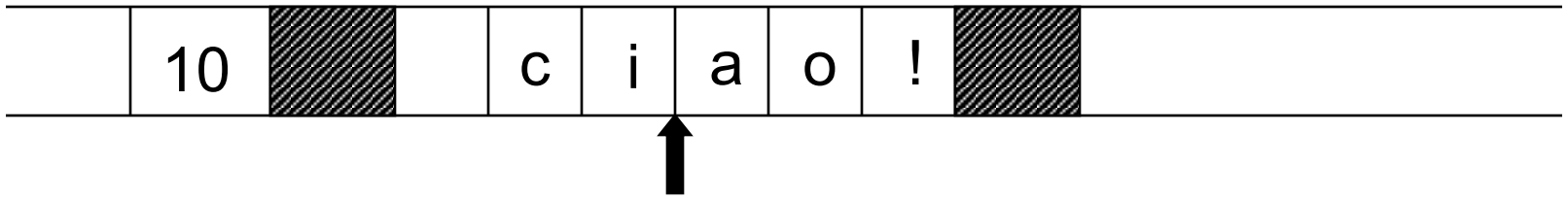
Il file di input




```
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();
```

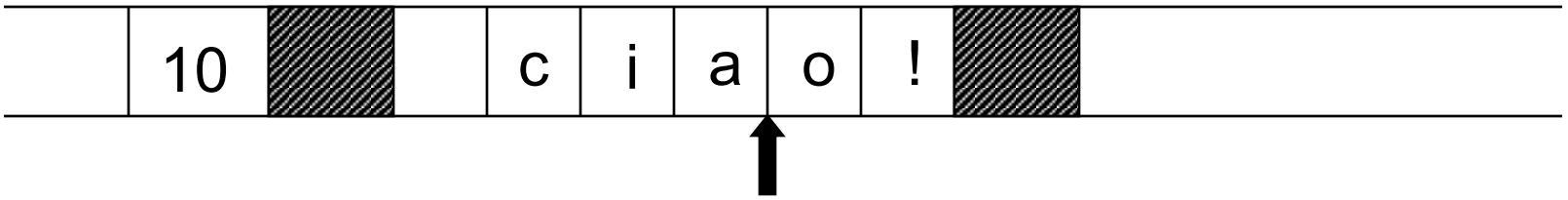



Il file di input



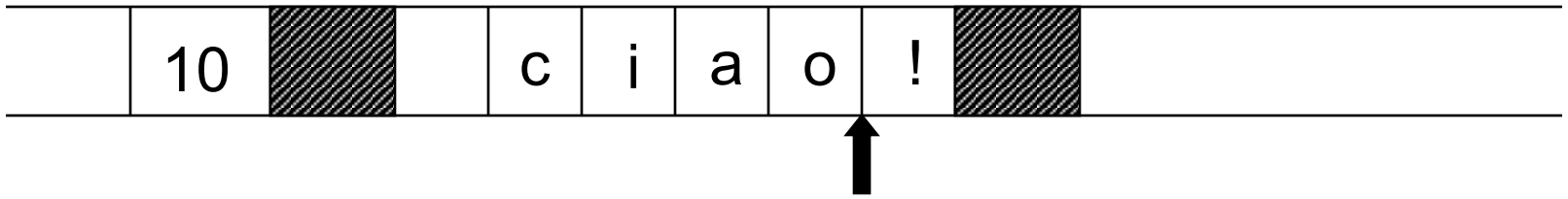

```
int x = SavitchIn.readLineInt();
char c = SavitchIn.readChar();
char c = SavitchIn.readChar();
char c = SavitchIn.readChar();
char c = SavitchIn.readChar();
char c = SavitchIn.readChar();
String s = SavitchIn.readLine();
```

Il file di input

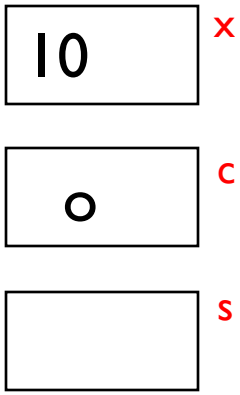



```
int x = SavitchIn.readLineInt();
char c = SavitchIn.readChar();
char c = SavitchIn.readChar();
char c = SavitchIn.readChar();
char c = SavitchIn.readChar();
char c = SavitchIn.readChar();
String s = SavitchIn.readLine();
```

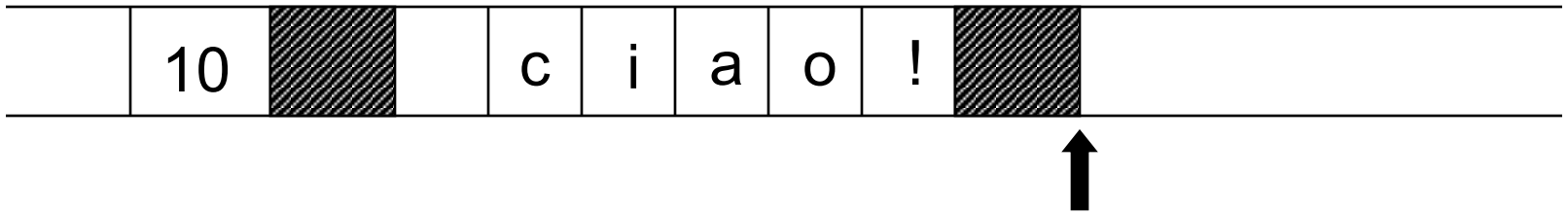
Il file di input



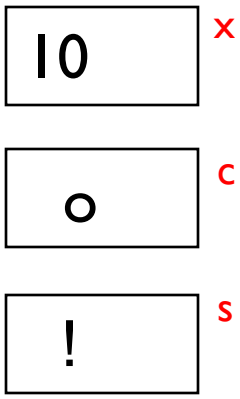

```
int x = SavitchIn.readLineInt();
char c = SavitchIn.readChar();
char c = SavitchIn.readChar();
char c = SavitchIn.readChar();
char c = SavitchIn.readChar();
char c = SavitchIn.readChar();
String s = SavitchIn.readLine();
```



Il file di input




```
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();
```



Stringhe

Il tipo String rappresenta il tipo delle sequenze di caratteri. Non è un tipo primitivo: in effetti è una classe.

Una costante di tipo String è una sequenza di caratteri racchiusa tra doppi apici, per esempio “pippo”. Una stringa di n caratteri ha I caratteri in posizioni 0,…,n – 1.

La notazione delle espressioni di tipo String dipende dal fatto che String sia una classe, e impiega i metodi della classe String.

Esempi

lunghezza di una stringa w:

w.length() (espressione di tipo int)

carattere in posizione i di una stringa w:

w.charAt(i) (espressione di tipo char, per i < w.length())

uguaglianza di stringhe w,w’:

w.equals(w’) (espressione di tipo boolean)

concatenazione di stringhe w,w’:

w + w’ (espressione di tipo String)