
Introduzione all'uso degli oggetti in Java (parte II)

Walter Didimo

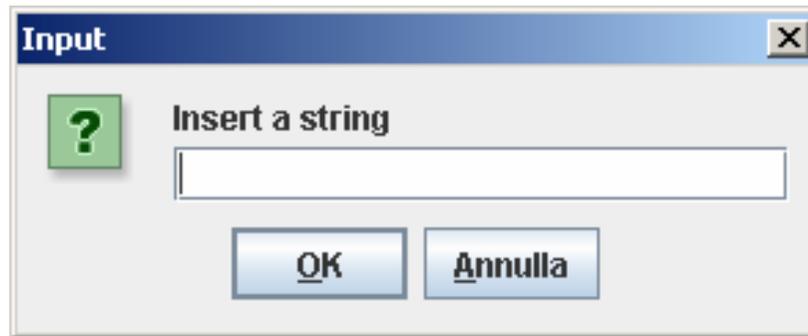
Esercizio all'uso di oggetti

Nel corso viene fornita una classe i cui oggetti permettono di leggere dati immessi dalla tastiera – la classe *InputWindow*

- vogliamo imparare ad usare gli oggetti di tipo *InputWindow*
- vogliamo anche esercitarci ad usare oggetti di altre classi già definite

La classe `InputDialog`

Un oggetto di tipo `InputDialog` permette di acquisire dati immessi da tastiera attraverso un piccolo pannello grafico



Creazione di oggetti `InputWindow`

La classe `InputWindow` mette a disposizione un solo costruttore per creare i suoi oggetti

`InputWindow ()`

Per creare un oggetto di tipo `InputWindow` occorre dunque scrivere

`new InputWindow ()`

Lettura di dati con InputWindow

Ad un oggetto di tipo *InputWindow* possiamo chiedere di leggere dati di vario tipo immessi da tastiera: numeri interi (int), numeri reali (double), stringhe (String), ecc.

La trattazione sistematica dei *tipi di dato* in Java sarà oggetto di una prossima lezione; per ora procediamo con degli esempi

Metodi di InputWindow

Per ogni tipo di dato che un oggetto *InputWindow* può acquisire da tastiera, esiste un metodo apposito che acquisisce tale dato e lo restituisce

int readInt ();

double readDouble ();

String readString ();

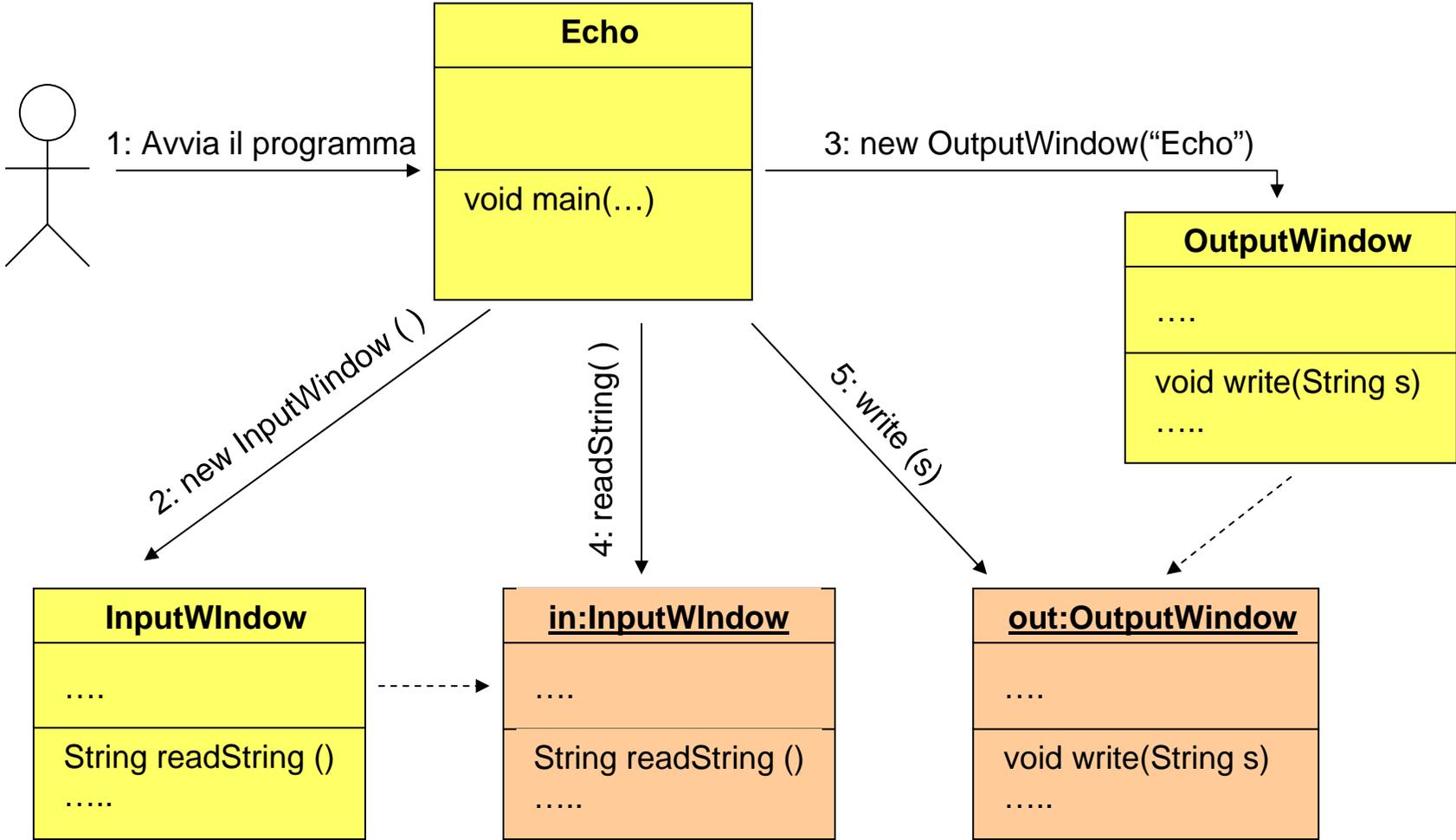
.....

Il semplice programma “Echo”

Come primo esempio, realizziamo un semplice programma che acquisisce una stringa immessa da tastiera dall'utente, e che visualizza tale stringa in una finestra sullo schermo

- Il programma sarà formato dalle classi *InputWindow* ed *OutputWindow* (già definite), e dalla classe *Echo*, che dobbiamo definire
- La classe *Echo* conterrà il solo metodo *main* ed utilizzerà oggetti di *InputWindow* ed *OutputWindow*

Schema del programma



Il codice della classe Echo

```
class Echo{  
    /* Visualizza una stringa inserita dall'utente*/  
    public static void main (String[] args){  
        InputWindow in = new InputWindow ();  
        OutputWindow out = new OutputWindow ("Echo");  
        String str = in.readString();  
        out.write(str);  
    }  
}
```

Discutiamo nel seguito le varie linee di codice del metodo *main* della classe *Echo*

Creazione di un oggetto InputWindow

```
class Echo{  
    /* Visualizza una stringa inserita dall'utente*/  
    public static void main (String[] args){  
        InputWindow in = new InputWindow ();  
        OutputWindow out = new OutputWindow ("Echo");  
        String str = in.readString();  
        out.write(str);  
    }  
}
```

Utilizza il *new* ed un costruttore per creare un oggetto di tipo *InputWindow*; il riferimento viene memorizzato nella variabile *in*

Creazione di un oggetto `OutputWindow`

```
class Echo{  
    /* Visualizza una stringa inserita dall'utente*/  
    public static void main (String[] args){  
        InputWindow in = new InputWindow ();  
        OutputWindow out = new OutputWindow ("Echo");  
        String str = in.readString();  
        out.write(str);  
    }  
}
```

Analogamente viene creato un oggetto *OutputWindow* ed il suo riferimento è memorizzato nella variabile *out*, la finestra di output avrà il titolo "Echo"

Leggere un dato da tastiera

```
class Echo{  
    /* Visualizza una stringa inserita dall'utente*/  
    public static void main (String[] args){  
        InputWindow in = new InputWindow ();  
        OutputWindow out = new OutputWindow ("Echo");  
        String str = in.readString();  
        out.write(str);  
    }  
}
```

Questa istruzione è una istruzione composta: legge una stringa immessa da tastiera e la memorizza in una variabile riferimento opportuna

Leggere un dato da tastiera

in.readString()

Quando si invoca il metodo *readString()* sull'oggetto *in*, tale metodo apre una finestra sullo schermo ...



... ed attende che l'utente immetta una stringa e prema OK



... quindi il metodo termina, ed all'espressione *in.readString()* rimane associata la stringa inserita

Stringhe ed oggetti String

Una stringa (sequenza di caratteri) viene rappresentata in Java tramite un oggetto, la cui classe è la classe *String*

- La classe *String* è predefinita nella API di Java ed offre molte funzionalità utili (le studieremo in seguito)
- Il metodo *readString()* restituisce il riferimento ad un oggetto di tipo *String*, che rappresenta la stringa immessa dall'utente

Memorizzare una stringa

Possiamo memorizzare il riferimento restituito da *readString()* in una opportuna variabile di tipo String, utilizzando l'operatore di assegnamento “=”

```
String str = in.readString()
```

La variabile che stiamo usando ha il nome *str*

Visualizzare la stringa

```
class Echo{  
    /* Visualizza una stringa inserita dall'utente*/  
    public static void main (String[] args){  
        InputWindow in = new InputWindow ();  
        OutputWindow out = new OutputWindow ("Echo");  
        String str = in.readString();  
        out.write(str);  
    }  
}
```

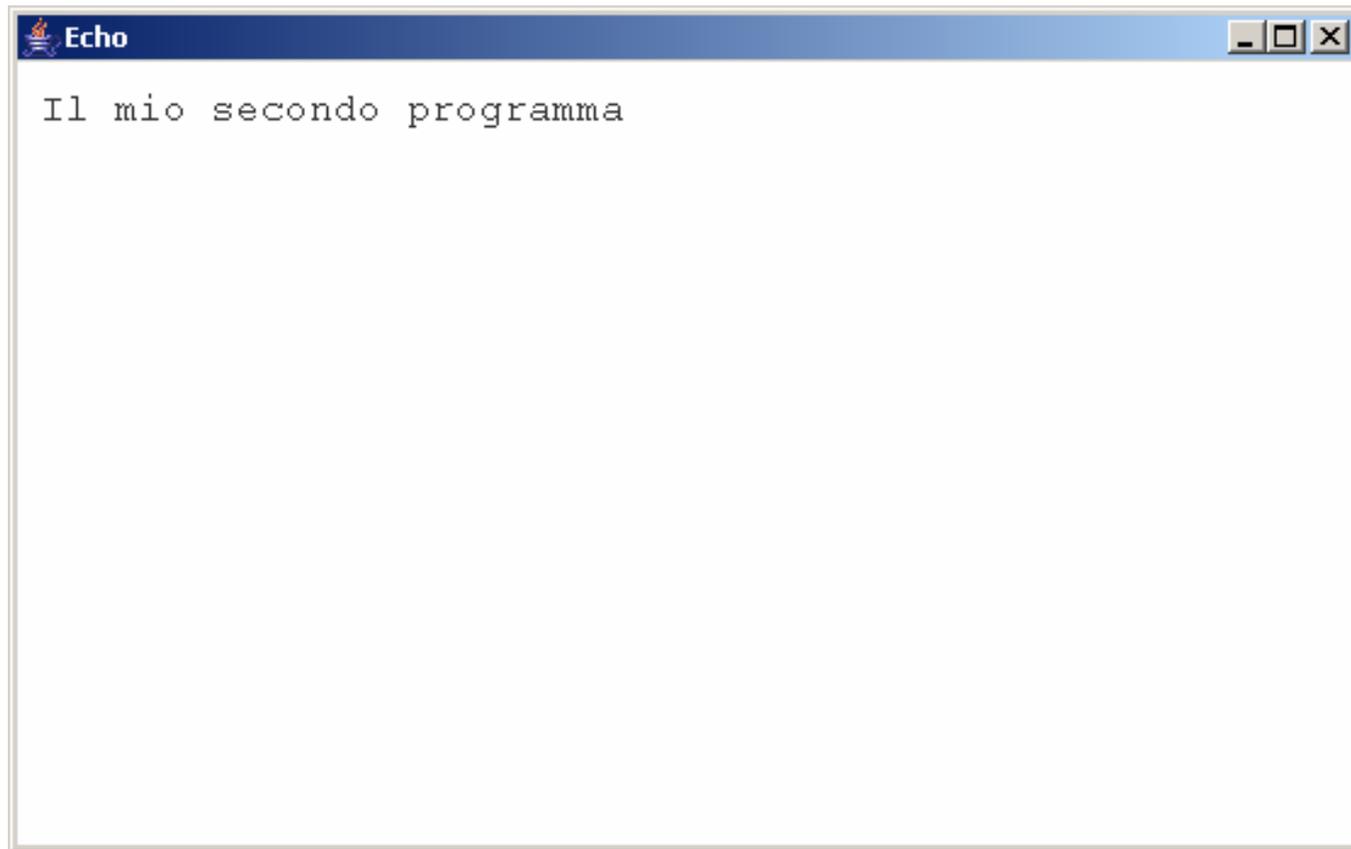
Invocando il metodo *write* sull'oggetto *out*, visualizziamo nella finestra dal titolo "Echo" la stringa inserita dall'utente, cioè quella memorizzata in *str*

Visualizzare la stringa

```
class Echo{  
    /* Visualizza una stringa inserita dall'utente*/  
    public static void main (String[] args){  
        InputWindow in = new InputWindow ();  
        OutputWindow out = new OutputWindow ("Echo");  
        String str = in.readString();  
        out.write(str);  
    }  
}
```

Il metodo *write* accetta una stringa come parametro; questo significa che possiamo passare ad esso anche riferimenti ad oggetti *String*, oltre che letterali stringa

Il risultato



The image shows a screenshot of a window titled "Echo". The window has a blue title bar with standard Windows window controls (minimize, maximize, close) on the right. The main content area of the window is white and contains the text "Il mio secondo programma" in a monospaced font.

```
Il mio secondo programma
```

Ancora sui metodi di `InputWindow`

Il metodo `readString()` di un oggetto di tipo `InputWindow` apre un pannello con un messaggio (“Insert a string”) che chiede di inserire una stringa



Esiste anche una variante, `readString (String s)`, che permette di cambiare il messaggio “Insert a string” con il messaggio `s` passato come parametro

`in.readString (“Inserisci una stringa”)`



Un altro programma

Vogliamo scrivere ora un programma che risolve equazioni di secondo grado

- Il programma utilizzerà una classe, *EquazioneDiSecondoGrado*, che immaginiamo sia già stata definita
- Un oggetto di tipo *EquazioneDiSecondoGrado* è in grado di rappresentare un'equazione di secondo grado nella forma $ax^2 + bx + c = 0$ e di restituirci le sue soluzioni reali

I metodi di EquazioneDiSecondoGrado

/ il costruttore – permette di creare un oggetto della classe, che rappresenta l'equazione $ax^2 + bx + c = 0$ */*

EquazioneDiSecondoGrado (double a, double b, double c)

/ visualizza le soluzioni (reali) sullo standard output (finestra non grafica) */*

void stampaSoluzioni ()

/ visualizza le soluzioni (reali) sulla finestra grafica rappresentata dall'oggetto OutputWindow passato come parametro */*

void stampaSoluzioni (OutputWindow out)

Overloading di metodi

- La classe *EquazioneDiSecondoGrado* ha due metodi con lo stesso nome, i metodi *stampaSoluzioni()* e *stampaSoluzioni(OutputWindow out)*
- Essi hanno però *diversa signature*
- Una classe può avere molti metodi con lo stesso nome ([overloading di metodi](#)) ma non due metodi con la stessa signature!!

Specifiche del programma

Il programma chiederà all'utente di inserire i coefficienti dell'equazione da calcolare e poi visualizzerà le soluzioni, sia su finestra grafica sia sullo standard output

L'equazione si assume nella forma $ax^2 + bx + c = 0$, dove ciascuno dei coefficienti a , b e c può essere un numero reale a piacere

Struttura del programma

Il programma si comporrà delle seguenti classi:

- *EquazioneDiSecondoGrado* (già scritta)
- *InputWindow* (già scritta)
- *OutputWindow* (già scritta)
- *CalcolaEquazione* (da scrivere) – contiene il solo metodo *main*

Il codice della classe CalcolaEquazione

```
class CalcolaEquazioni{

    public static void main (String[] args){
        InputWindow in = new InputWindow ();
        double a = in.readDouble ("Inserisci il coeff. a");
        double b = in.readDouble ("Inserisci il coeff. b");
        double c = in.readDouble ("Inserisci il coeff. c");
        EquazioneDiSecondoGrado eq = new EquazioneDiSecondoGrado(a,b,c);
        eq.stampaSoluzioni ();
        OutputWindow out = new OutputWindow();
        eq.stampaSoluzioni (out);
    }
}
```