



UNIVERSITÀ DEGLI STUDI DI SALERNO

Fondamenti di Informatica

Introduzione ad AlgoBuild

Prof. Arcangelo Castiglione

A.A. 2016/17



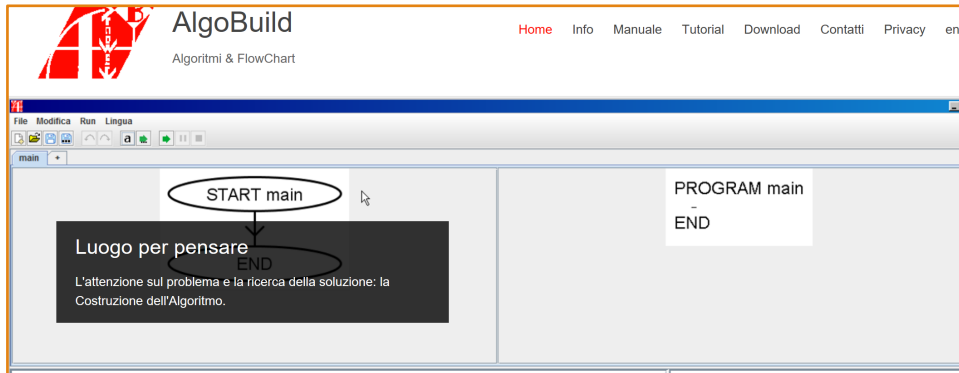
AlgoBuild

Introduzione ad AlgoBuild: OUTLINE

- Caratteristiche
- Come si presenta
- Utilizzo del blocco di output
 - *Esempio*
- Utilizzo dei blocchi di input, output ed esecuzione
 - *Esempio*



Caratteristiche – 1/3



Crea Diagrammi di Flusso, Pseudo Codice, Programmi

Partenza... VIA!

Disegna i **programmi** con **AlgoBuild**, l'ambiente didattico per lo studio della programmazione e degli algoritmi.

Semplice e veloce permette di apprendere le nozioni base della **programmazione strutturata** per mezzo di **diagrammi di flusso** (flowchart) e **pseudo codice** (pseudocode).

E' Divertente e facile da usare ma basato su una sintassi formale grafica strutturata.

- Fonte
- <https://algorithmbuild.com/it/index.html>



Caratteristiche – 2/3

- Con AlgoBuild è possibile disegnare in maniera semplice ed efficace **diagrammi di flusso**
- AlgoBuild permette anche di tradurre i diagrammi di flusso in **pseudo-codice**
- *Maggiori informazioni*
 - <https://algorithmbuild.com/it/index.html>

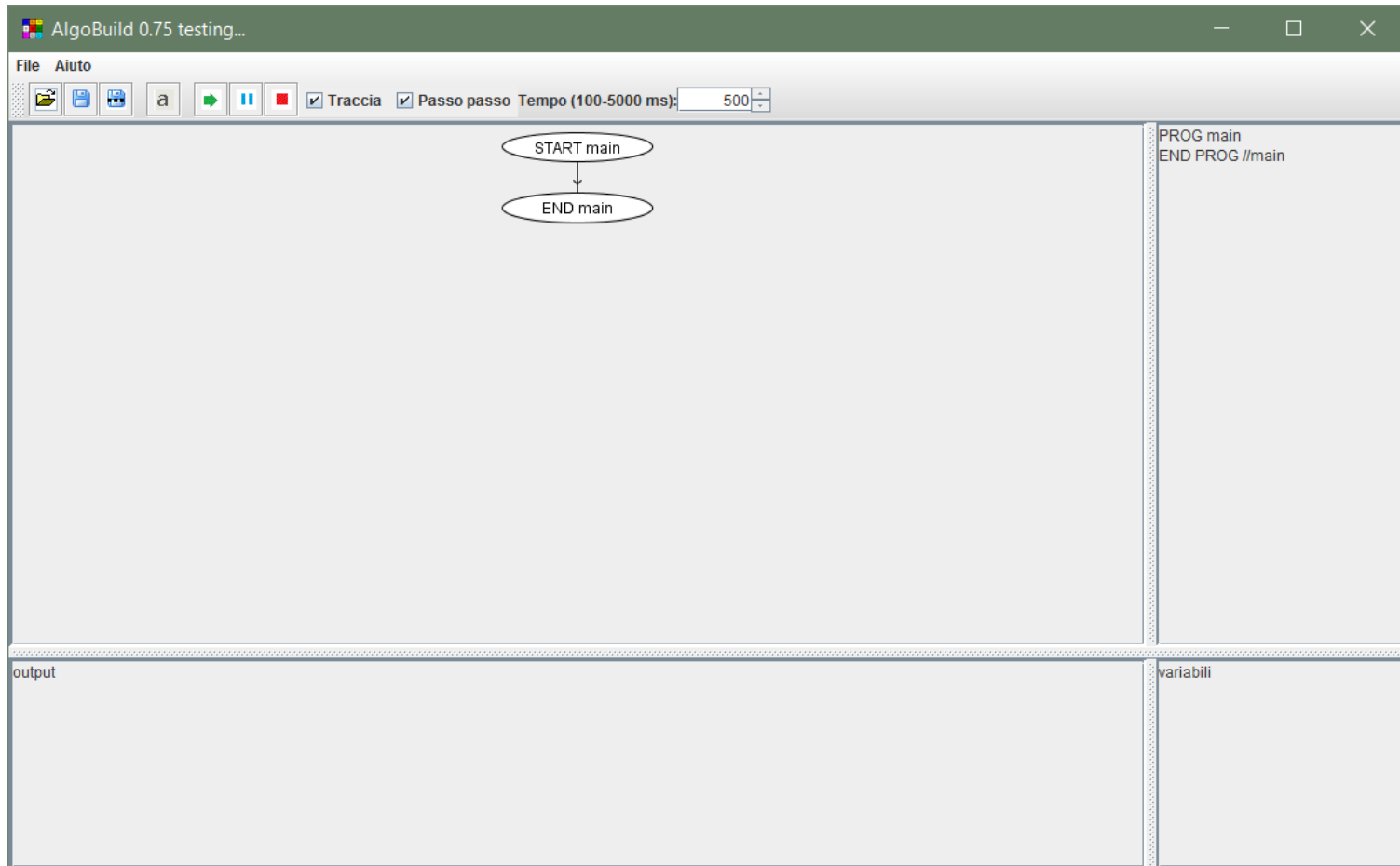


Caratteristiche – 3/3

- Dove reperirlo?
 - AlgoBuild è scaricabile gratuitamente
 - L'indirizzo da cui può essere scaricato è
 - <https://algotool.com/it/download.html>
- La versione *stabile* attualmente è la **0.75**



Come si Presenta



Come si Presenta

The screenshot displays the AlgoBuild 0.75 testing... application window. The interface includes a menu bar with 'File' and 'Aiuto', a toolbar with icons for file operations and execution, and a control panel with checkboxes for 'Traccia' and 'Passo passo', and a 'Tempo (100-5000 ms):' field set to 500. The main workspace contains a flowchart with two nodes: 'START main' and 'END main', connected by a downward arrow. The text 'Diagramma di flusso' is centered in the workspace. On the right side, there are two panels: the top one contains the code 'PROG main' and 'END PROG //main', and the bottom one is labeled 'variabili'. At the bottom left, there is an 'output' panel.

```
graph TD; A([START main]) --> B([END main]);
```

output

variabili

PROG main
END PROG //main

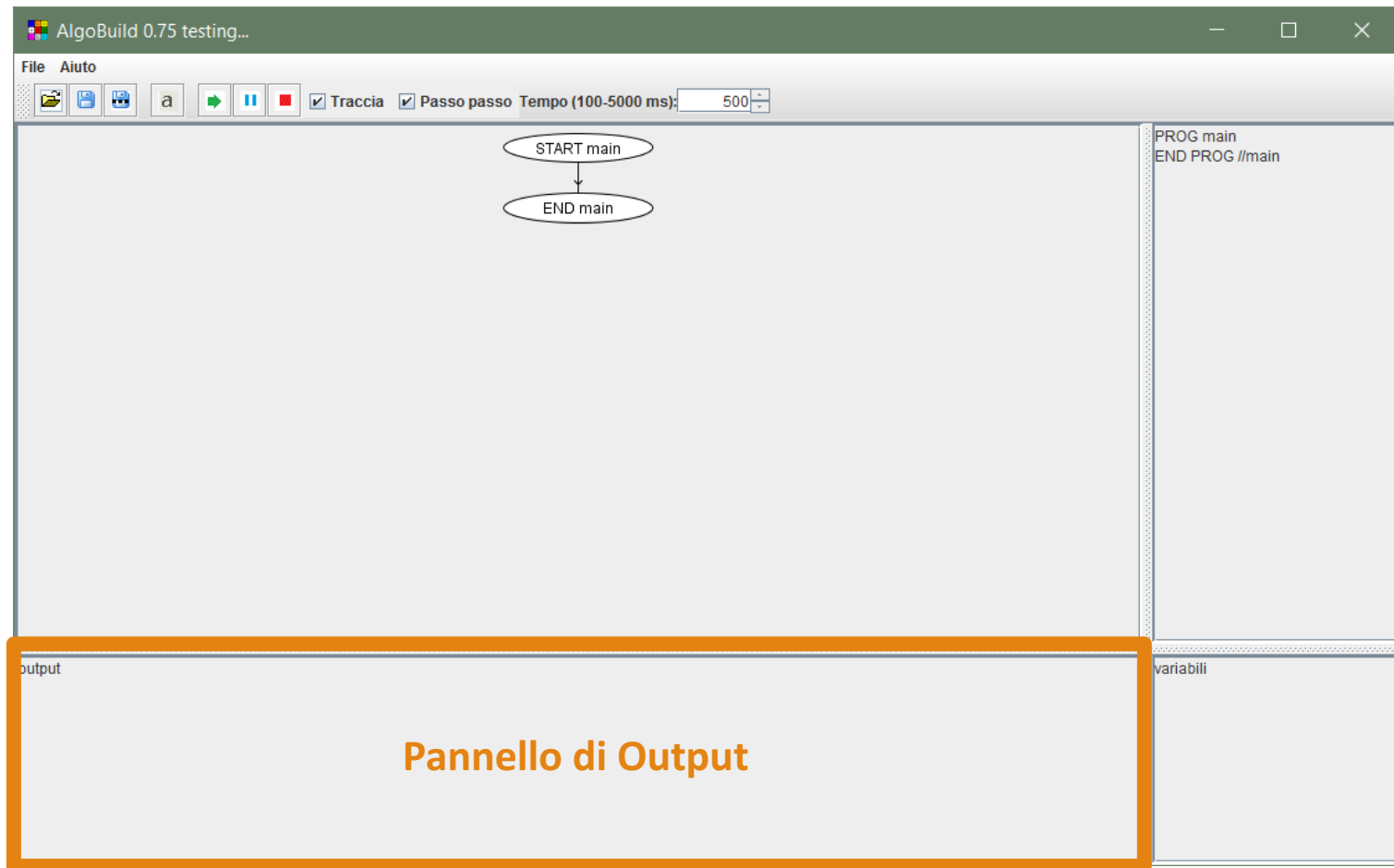
Come si Presenta

The screenshot displays the AlgoBuild 0.75 testing environment. The interface is divided into several sections:

- Toolbar:** Located at the top, it includes icons for file operations (File, Aiuto), a search icon (a), and execution controls (run, pause, stop). It also features checkboxes for "Traccia" and "Passo passo", and a "Tempo (100-5000 ms):" field set to 500.
- Flowchart:** The central workspace contains a simple flowchart with two ovals: "START main" at the top and "END main" at the bottom, connected by a downward-pointing arrow.
- Pseudo-codice:** A panel on the right side, highlighted with an orange border, contains the text:

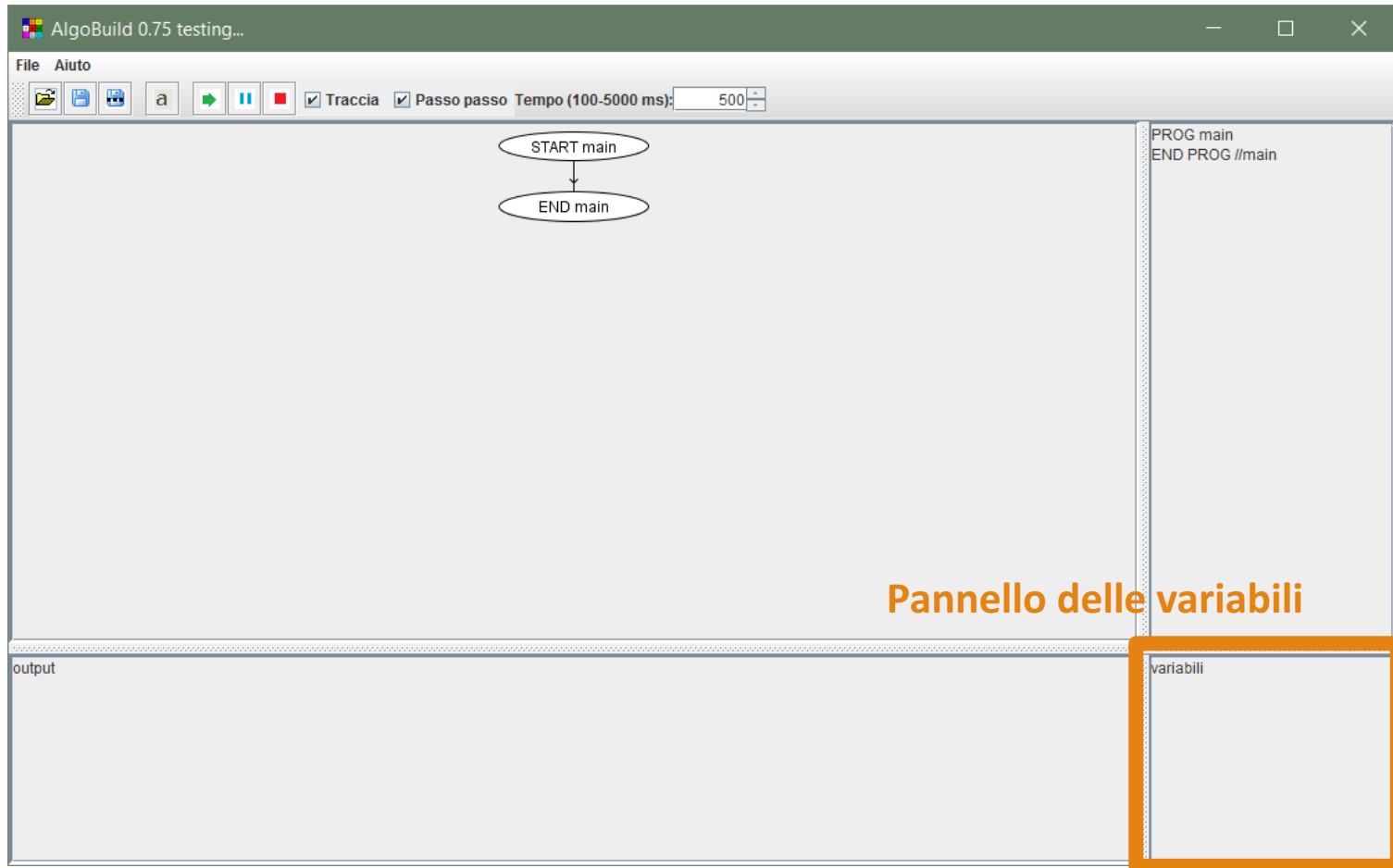
```
PROG main  
END PROG //main
```
- Output and Variables:** At the bottom, there are two empty panels labeled "output" and "variabili".

Come si Presenta



Pannello di Output

Come si Presenta



Pannello delle variabili

Come si Presenta



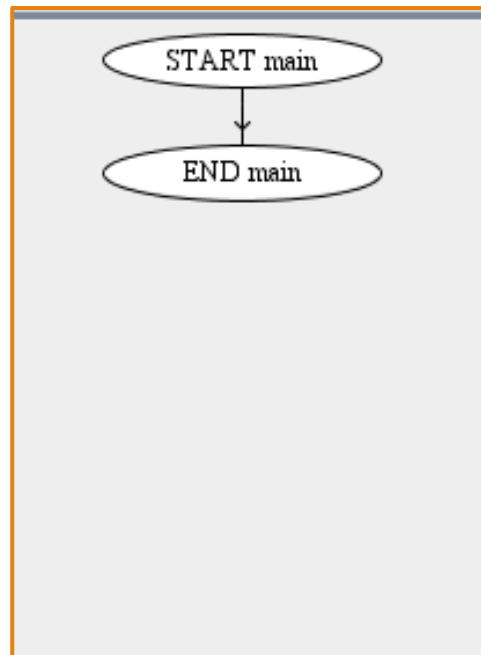
Esempio 1: “Hello, World!” – 1/13

- Iniziamo ad utilizzare AlgoBuild con l’esempio *Hello, World!*
- *Hello, World!* mostra semplicemente la stringa
 - “Ciao, Mondo!”
- Storicamente, molti manuali di programmazione usano l’esempio “Hello, world!” per mostrare lessico, sintassi e semantica basilare di un dato linguaggio di programmazione



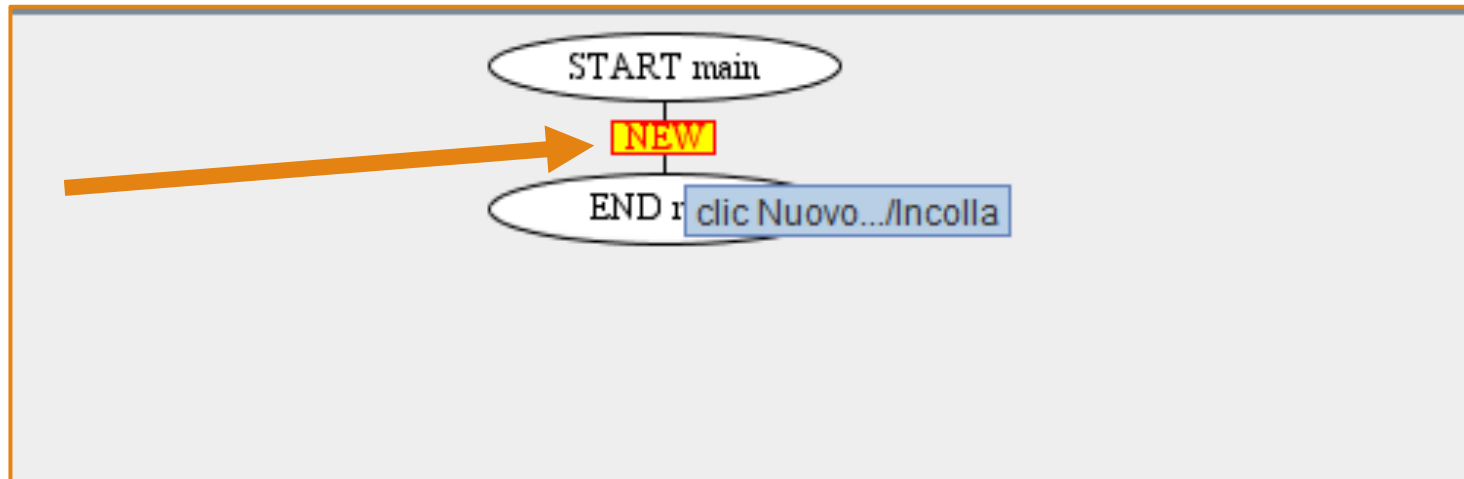
Esempio 1: “Hello, World!” – 2/13

- Nell’area del diagramma di flusso possiamo notare i due blocchi di inizio (*START*) e fine (*END*)
- Sono inseriti **automaticamente** da AlgoBuild



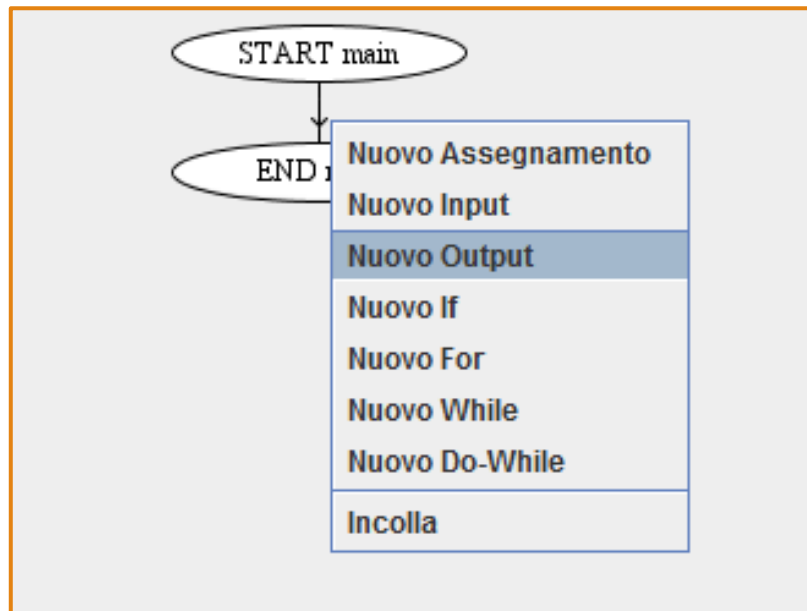
Esempio 1: “Hello, World!” – 3/13

- Posizioniamoci con il mouse **sulla freccia** che collega lo *START* e l'*END* del nostro diagramma di flusso



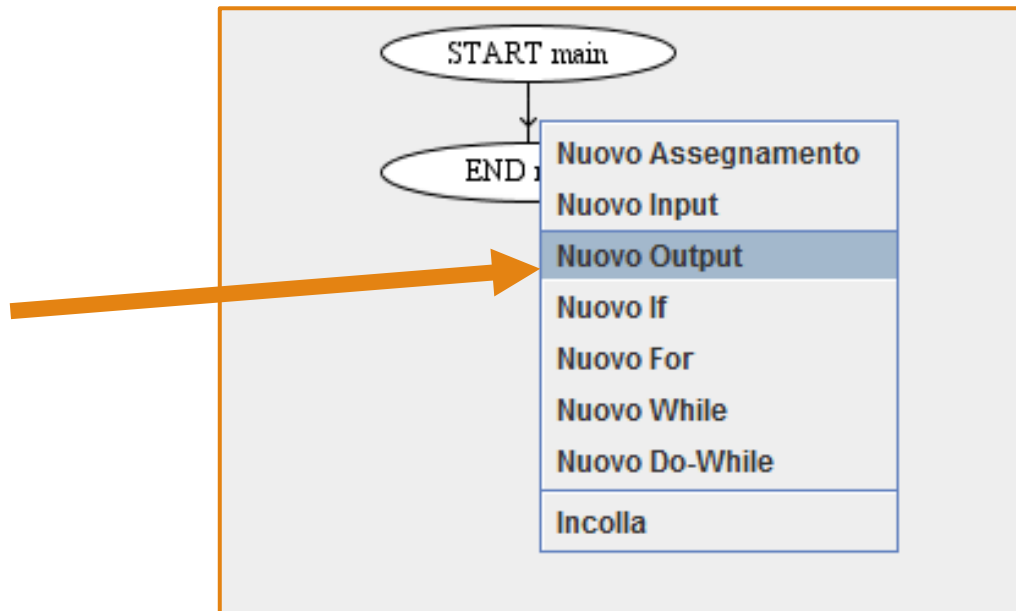
Esempio 1: “Hello, World!” – 4/13

- Cliccando su «**NEW**», ci verranno proposte diverse alternative per l’inserimento di un nuovo blocco



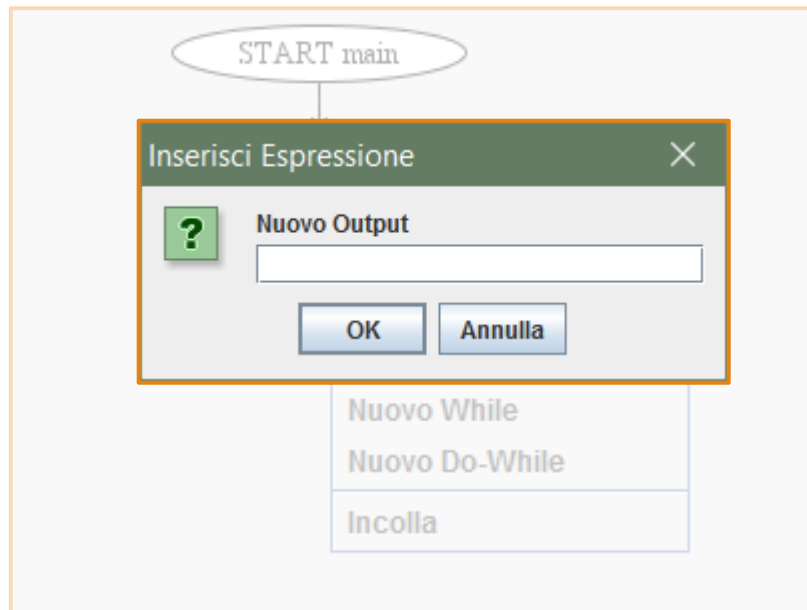
Esempio 1: “Hello, World!” – 5/13

- Selezioniamo **Nuovo Output**



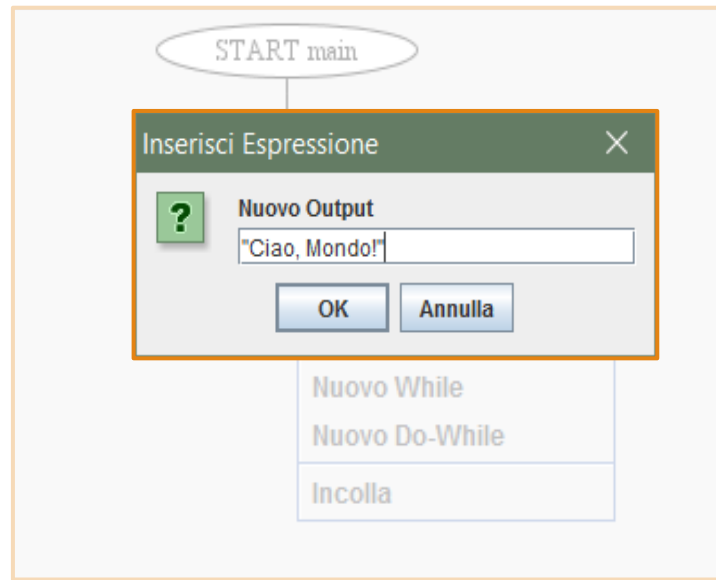
Esempio 1: “Hello, World!” – 6/13

- Selezioniamo **Nuovo Output**
 - Ci verrà richiesto qual è l’output che vogliamo mostrare



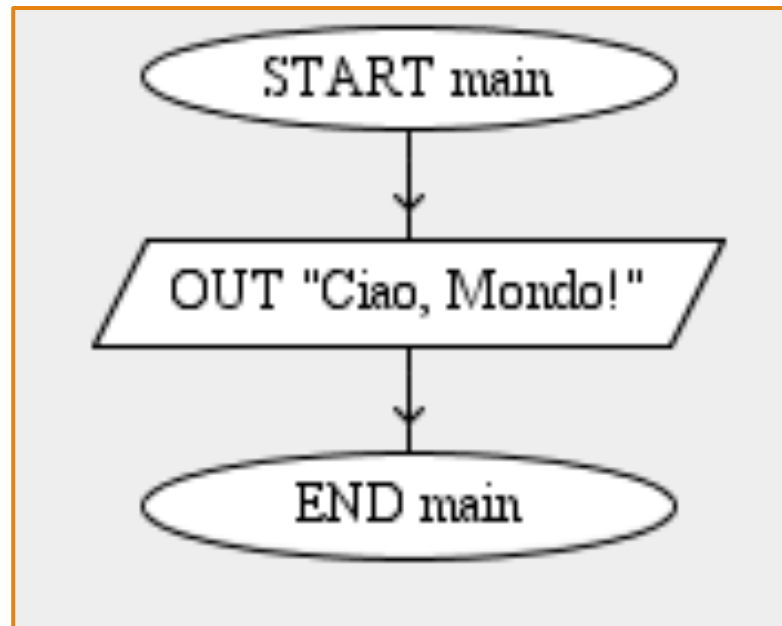
Esempio 1: “Hello, World!” – 7/13

- Scriviamo la stringa “Ciao, Mondo!”, poi
 - Clicchiamo su **OK**
 - Oppure premiamo il tasto **Invio** della tastiera



Esempio 1: “Hello, World!” – 8/13

- Ecco il nostro **diagramma di flusso**



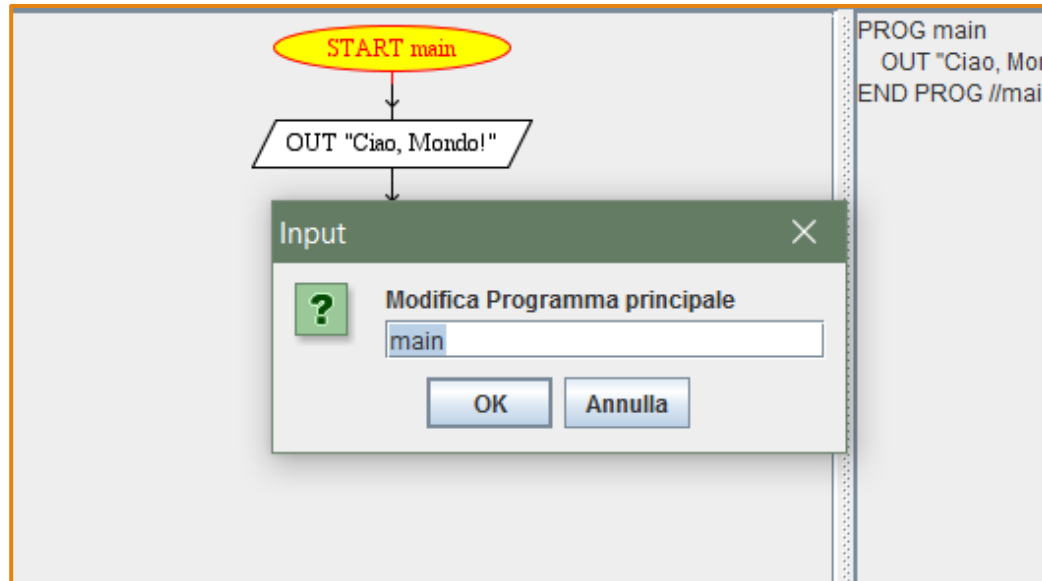
Esempio 1: “Hello, World!” – 9/13

- AlgoBuild ha contestualmente generato anche lo pseudo-codice
- Ecco cosa ci presenterà l’area preposta

```
PROG main  
  OUT "Ciao, Mondo!"  
END PROG //main
```

Esempio 1: “Hello, World!” – 10/13

- Possiamo anche modificare il nome del diagramma di flusso
 - Cliccando sul blocco *START* oppure *END*
 - Scrivendo il nome che vogliamo assegnare al diagramma



Esempio 1: “Hello, World!” – 11/13

- Possiamo modificare anche il nome del diagramma di flusso
 - Cliccando sul blocco *START* oppure *END*
 - Scrivendo il nome che vogliamo assegnare al diagramma
 - Ad esempio, lo chiamiamo **helloworld**



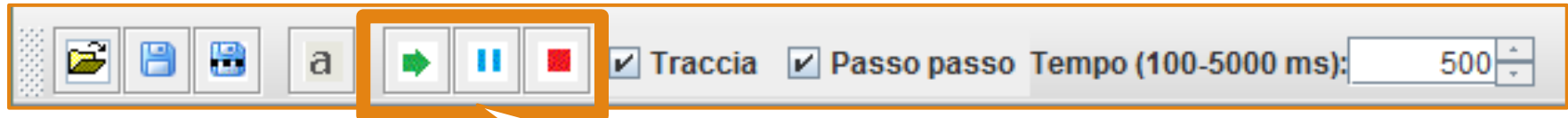
Esempio 1: “Hello, World!” – 12/13

- **NOTA IMPORTANTE**

- Per inserire un **nuovo blocco** dobbiamo sempre **clickare sulla freccia** che collega i **due blocchi** tra i quali vogliamo inserire un nuovo blocco

Esempio 1: “Hello, World!” – 13/13

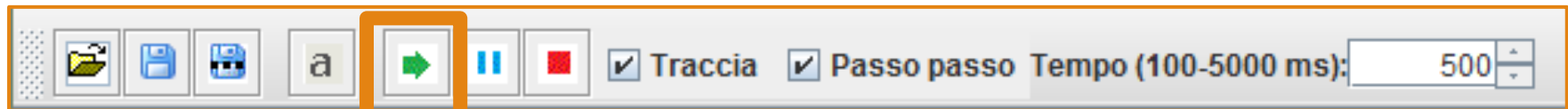
- L'esecuzione...
 - Ora che il nostro diagramma è stato generato, possiamo simulare la sua esecuzione tramite AlgoBuild



Comandi per la gestione dell'esecuzione

Esempio 1: “Hello, World!” – 13/13

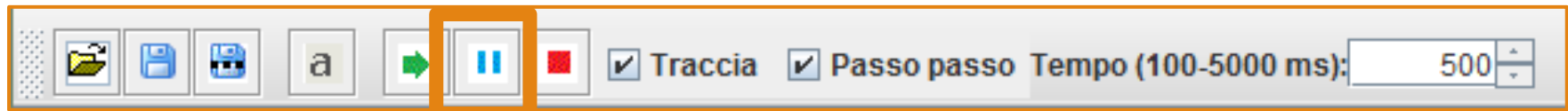
- L'esecuzione...
 - Ora che il nostro diagramma è stato generato, possiamo simulare la sua esecuzione tramite AlgoBuild



Serve ad avviare l'esecuzione delle istruzioni

Esempio 1: “Hello, World!” – 13/13

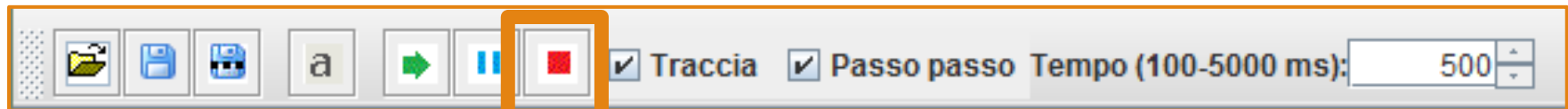
- L'esecuzione...
 - Ora che il nostro diagramma è stato generato, possiamo simulare la sua esecuzione tramite AlgoBuild



Serve a mettere in pausa
l'esecuzione


Esempio 1: “Hello, World!” – 13/13

- L'esecuzione...
 - Ora che il nostro diagramma è stato generato, possiamo simulare la sua esecuzione tramite AlgoBuild



Serve ad interrompere
l'esecuzione

Esempio 1: “Hello, World!” – 13/13

- L'esecuzione...
 - Ora che il nostro diagramma è stato generato, possiamo simulare la sua esecuzione tramite AlgoBuild
- AlgoBuild permette di simulare l'esecuzione anche *passo passo*
 - In questo caso sarà necessario cliccare ogni volta su  per eseguire l'istruzione successiva



Esempio 1: “Hello, World!” – 13/13

- L'esecuzione...
 - Ora che il nostro diagramma è stato generato, possiamo simulare la sua esecuzione tramite AlgoBuild
- AlgoBuild permette di simulazione l'esecuzione, anche passo passo
- Possiamo anche decidere il tempo (in millisecondi) che intercorre tra ogni istruzione eseguita



DEMO Esecuzione «Hello, World!» (Tempo passo: 5000ms, ovvero 5 secondi)

File Aiuto

Traccia Passo passo Tempo (100-5000 ms): 500

```
graph TD; A([START helloworld]) --> B[/OUT "Ciao, Mondo!"/]; B --> C([END helloworld]);
```

```
PROG helloworld  
  OUT "Ciao, Mondo!"  
END PROG //helloworld
```

output

variabili

AlgoBuild:

Operatori Aritmetici, Relazionali e Logici

- **Operatori Aritmetici**

Operatore	Descrizione
+	Addizione
-	Sottrazione
*	Moltiplicazione
/	Divisione
%	Resto della divisione intera

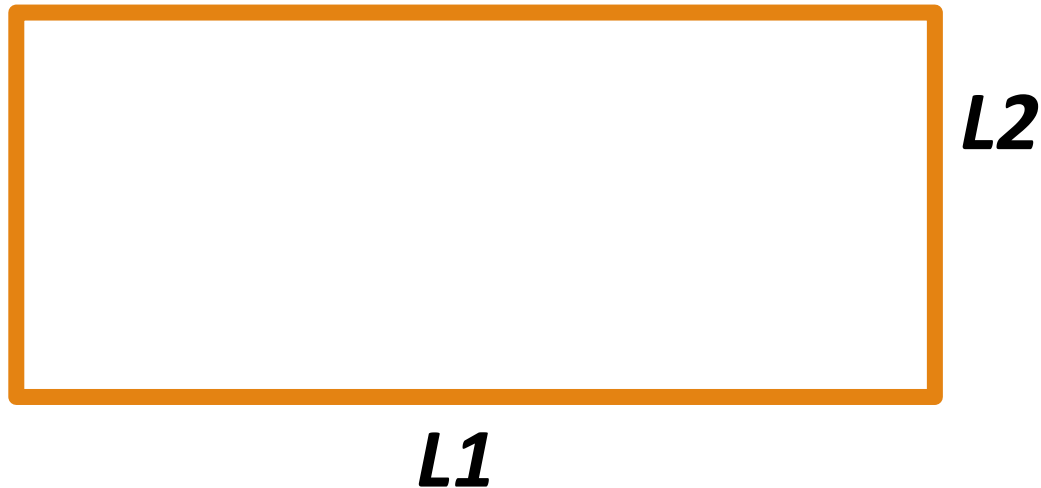
- **Operatori Logici**

Operatore	Descrizione
&&	AND
	OR
!	NOT

- **Operatori Relazionali**

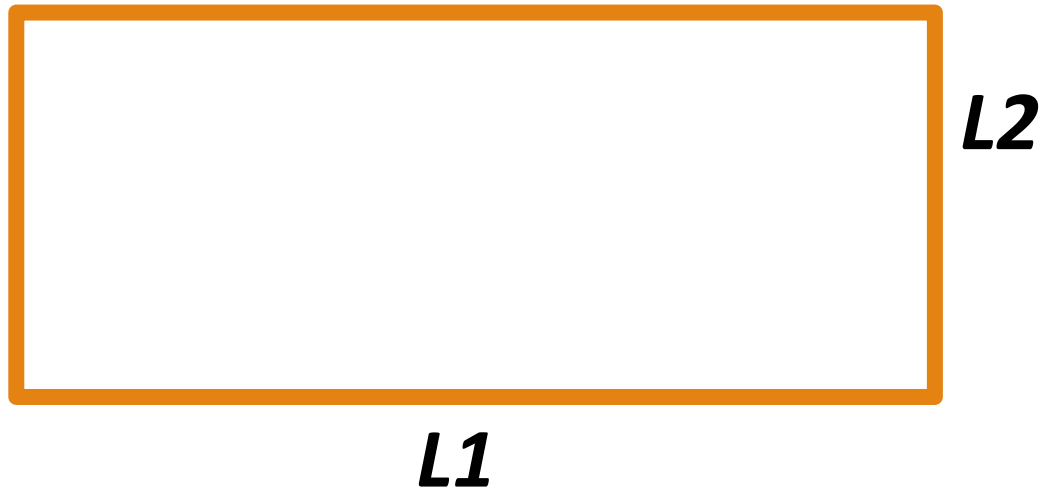
Operatore	Descrizione
<	Minore
<=	Minore o uguale
>	Maggiore
>=	Maggiore o uguale
==	Uguale
!=	Diverso

Esempio 2: Perimetro Rettangolo – 1/5



$$P = 2 * (L1 + L2)$$

Esempio 2: Perimetro Rettangolo – 1/5



$$P = 2 * (L1 + L2)$$

L1 → *input*
L2 → *input*
P → *output*

Esempio 2: Perimetro Rettangolo – 2/5

The screenshot shows the AlgoBuild 0.75 testing interface. The window title is "AlgoBuild 0.75 testing...". The menu bar includes "File" and "Aiuto". The toolbar contains icons for file operations (open, save, print), a search icon, and execution controls (run, pause, stop). The execution options are checked: "Traccia" and "Passo passo". The "Tempo (100-5000 ms):" field is set to 500.

The main workspace displays a flowchart with two ovals: "START perimetro_ret" at the top and "END perimetro_ret" at the bottom, connected by a downward arrow.

On the right side, there is a code editor with the following text:

```
PROG perimetro_ret  
END PROG //perimetro
```

At the bottom of the interface, there are two empty panels labeled "output" and "variabili".

Esempio 2: Perimetro Rettangolo – 2/5

AlgoBuild 0.75 testing...

File Aiuto

Traccia Passo passo Tempo (100-5000 ms): 500

START perimetro_ret

END perim

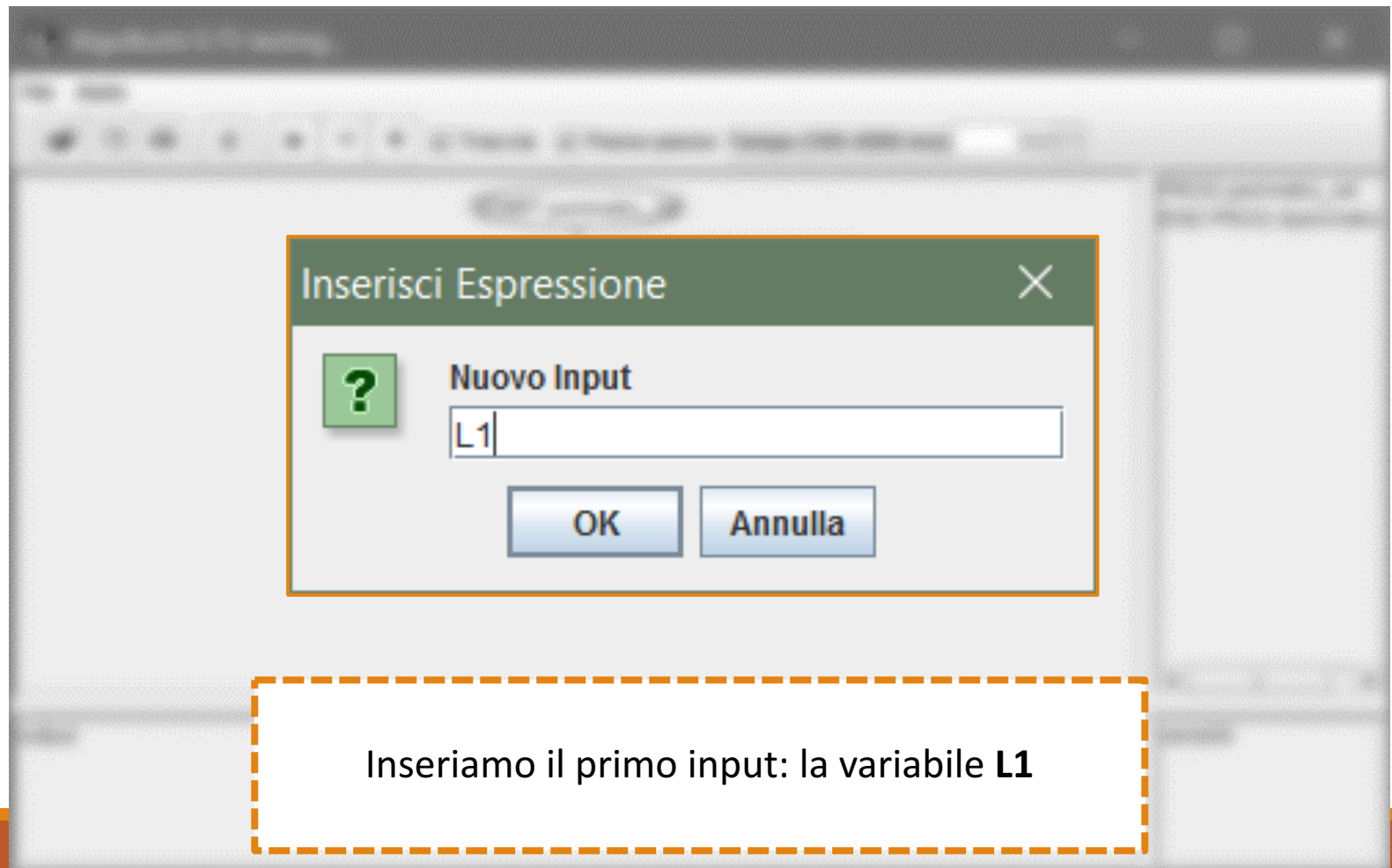
- Nuovo Assegnamento
- Nuovo Input**
- Nuovo Output
- Nuovo If
- Nuovo For
- Nuovo While
- Nuovo Do-While
- Incolla

PROG perimetro_ret
END PROG //perimetro

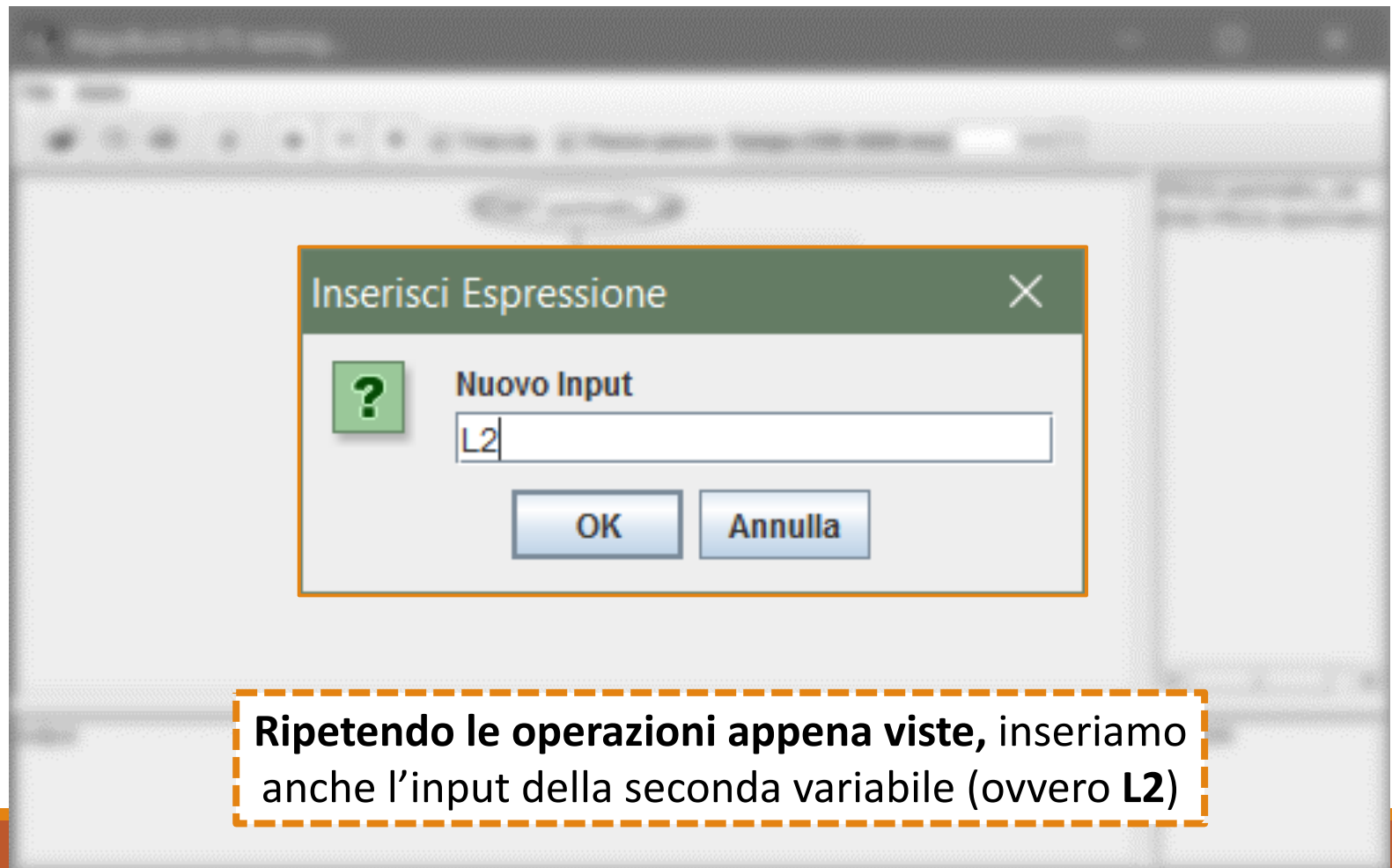
output

variabili

Esempio 2: Perimetro Rettangolo – 2/5



Esempio 2: Perimetro Rettangolo – 2/5



Ripetendo le operazioni appena viste, inseriamo anche l'input della seconda variabile (ovvero L2)

Esempio 2: Perimetro Rettangolo – 2/5

AlgoBuild 0.75 testing...

File Aiuto

Traccia Passo passo Tempo (100-5000 ms): 500

```
graph TD; Start([START perimetro_ret]) --> InL1[/IN L1/]; InL1 --> InL2[/IN L2/]; InL2 --> End([END perimetro_ret]);
```

PROG perimetro_ret
IN L1
IN L2
END PROG //perimetro

output

variabili

Come si può notare, tale diagramma di flusso presenta due blocchi di input, denotati dalle variabili **L1** ed **L2**

Esempio 2: Perimetro Rettangolo – 2/5

AlgoBuild 0.75 testing...

File Aiuto

Traccia Passo passo Tempo (100-5000 ms): 500

```
graph TD; Start([START perimetro_ret]) --> InL1[/IN L1/]; InL1 --> InL2[/IN L2/]; InL2 --> End([END perimetro_ret]);
```

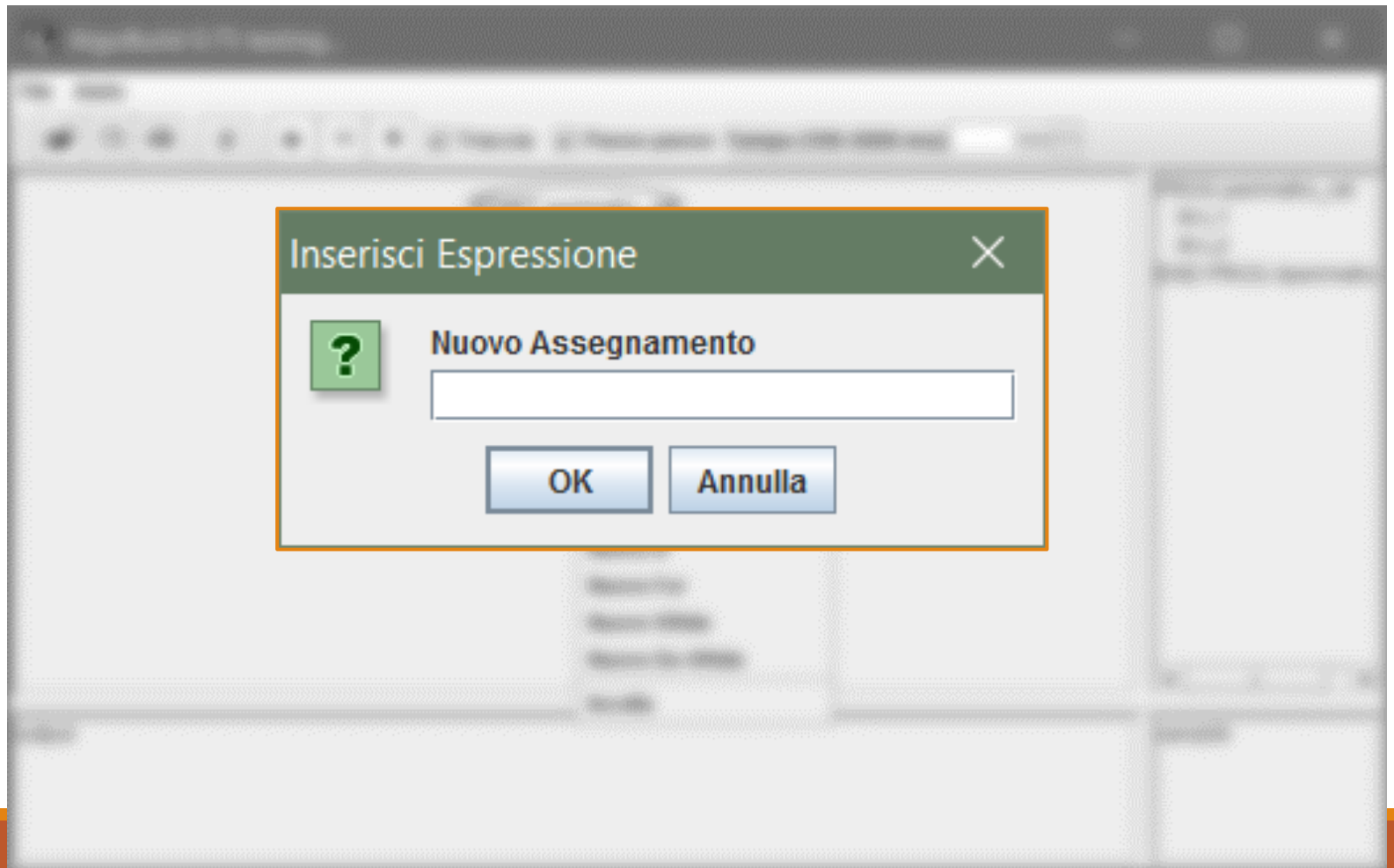
Nuovo Assegnamento
Nuovo Input
Nuovo Output
Nuovo If
Nuovo For
Nuovo While
Nuovo Do-While
Incolla

PROG perimetro_ret
IN L1
IN L2
END PROG //perimetro

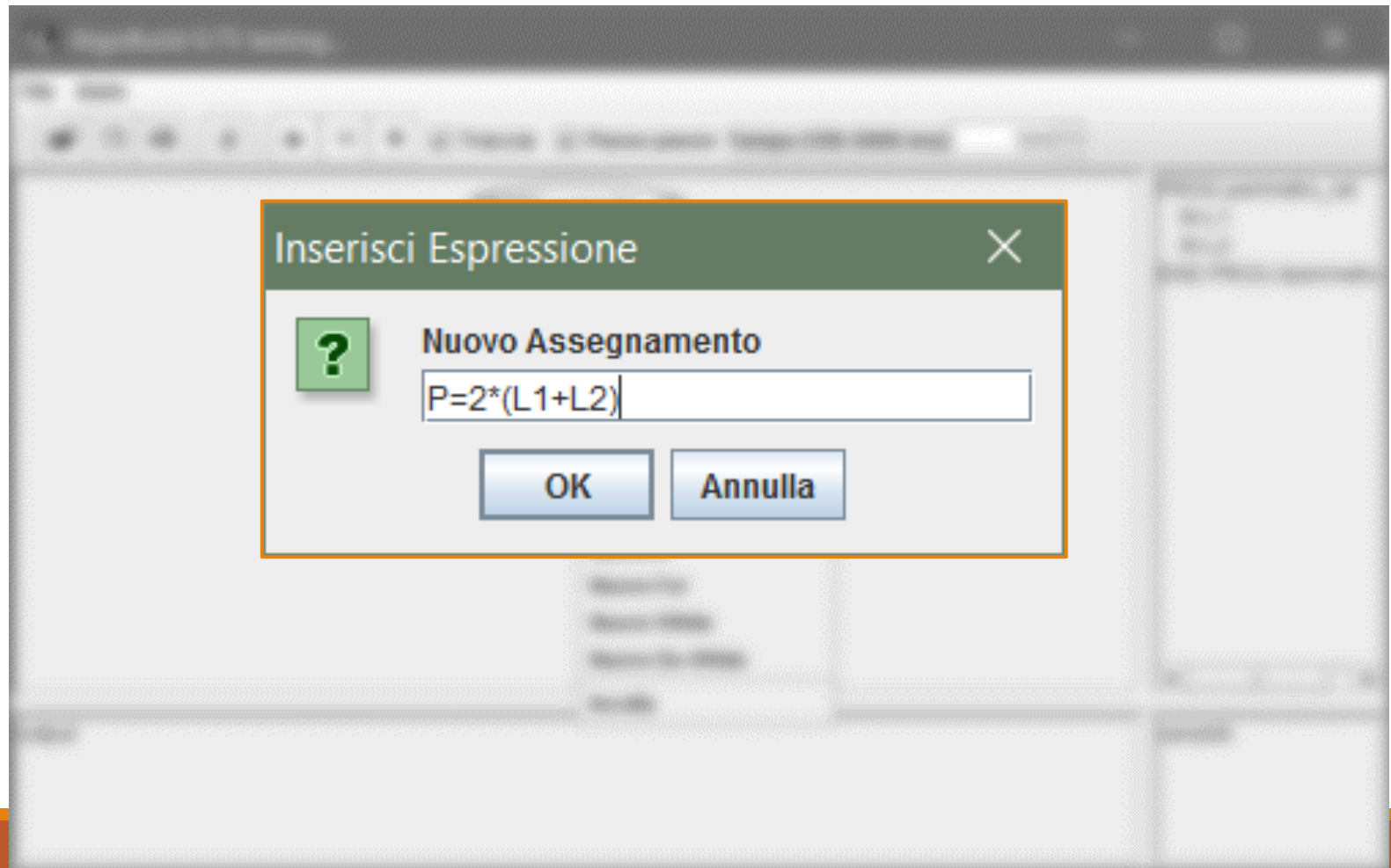
output

variabili

Esempio 2: Perimetro Rettangolo – 2/5



Esempio 2: Perimetro Rettangolo – 2/5



Esempio 2: Perimetro Rettangolo – 2/5

The screenshot displays the AlgoBuild 0.75 testing environment. The main workspace contains a flowchart for calculating the perimeter of a rectangle. The flowchart consists of the following steps:

- START perimetro_ret (Start node)
- IN L1 (Input node)
- IN L2 (Input node)
- P=2*(L1+L2) (Assignment node)
- END perimetro_ret (End node)

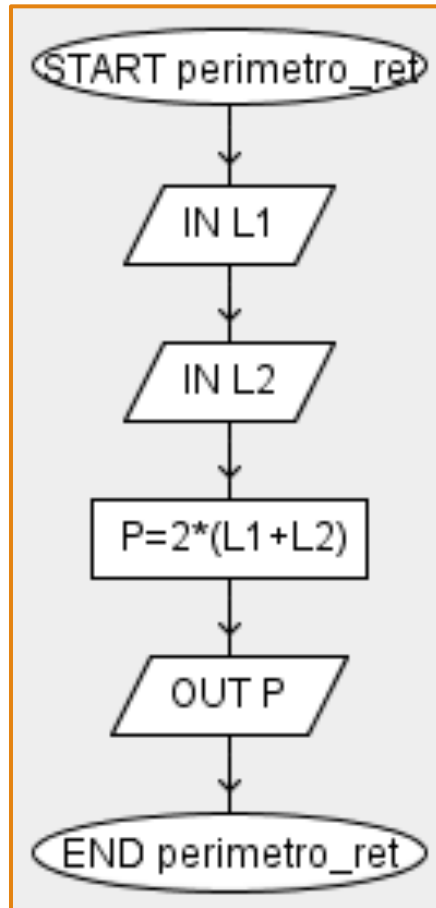
The right-hand panel shows the corresponding code:

```
PROG perimetro_ret
IN L1
IN L2
ASSIGN P=2*(L1+L2)
END PROG //perimetro
```

The bottom panel is divided into two sections: 'output' and 'variabili'. A text box is overlaid on the 'output' section, stating:

L'ultima operazione è quella di mostrare a video il valore della variabile risultato **P** (dopo l'operazione di assegnamento)

Esempio 2: Perimetro Rettangolo – 3/5



PSEUDO-CODICE

```
PROG perimetro_ret  
  IN L1  
  IN L2  
  ASSIGN P=2*(L1+L2)  
  OUT P  
END PROG //perimetro_ret
```

DEMO Esecuzione *Perimetro Rettangolo* (Tempo passo: 5000ms, ovvero 5 secondi)

The screenshot displays the AlgoBuild 0.75 testing environment. The main workspace contains a flowchart for a program named "perimetro_ret". The flowchart starts with a green oval labeled "START perimetro_ret", followed by two input boxes labeled "IN L1" and "IN L2", a process box containing the formula $P=2*(L1+L2)$, an output box labeled "OUT P", and finally an oval labeled "END perimetro_ret".

On the right side, a code editor shows the following program code:

```
PROG perimetro_ret
IN L1
IN L2
ASSIGN P=2*(L1+L2)
OUT P
END PROG //perimetro_ret
```

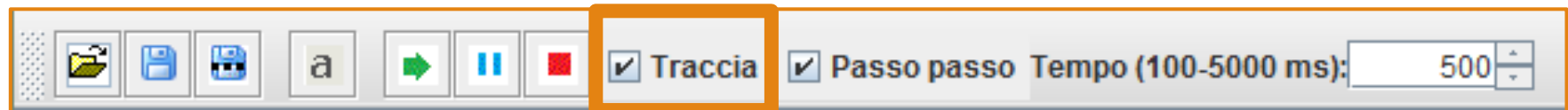
At the bottom left, an output window displays the text: "output *** PROGRAMMA perimetro_ret inizia."

At the bottom right, a panel labeled "variabili" is visible but currently empty.

The top toolbar includes a green play button (execution) which is highlighted by a mouse cursor, a blue pause button, a red stop button, and a text input field with the value "5.000" and a step size control.

Esempio 2: Perimetro Rettangolo – 4/5

- Selezionando l'opzione «**Traccia**» verranno fornite (nel Pannello di Output) ulteriori informazioni riguardanti il flusso di esecuzione
 - Oltre ad *eventuali errori che possono intercorrere*



Esempio 2: Perimetro Rettangolo – 5/5

Selezionando l'opzione «Traccia»

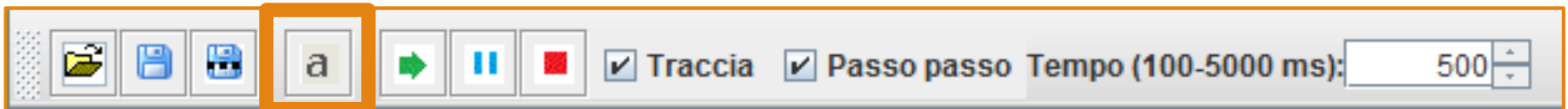
```
output
*** PROGRAMMA L2 inizia.
10
8
36.0
*** PROGRAMMA L2 termina.
```


Non selezionando l'opzione «Traccia»

```
output
*** PROGRAMMA L2 inizia.
INPUT: L1=10
var: | L1=10.0 |
INPUT: L2=8
var: | L1=10.0 | L2=8.0 |
ASSEGNA: P <- 36.0
var: | L1=10.0 | L2=8.0 | P=36.0 |
OUTPUT P: 36.0
*** PROGRAMMA L2 termina.
```

Altre Opzioni AlgoBuild – 1/3

- AlgoBuild permette di modificare le opzioni di visualizzazione del diagramma di flusso, permettendo di cambiare
 - Tipo di carattere
 - Dimensioni del carattere
 - Dimensioni del carattere nel pannello di output
 - Spessore delle linee relative al contorno dei blocchi ed agli archi orientati che collegano i blocchi (freccie)




- Cliccando sul tasto  apparirà una finestra di dialogo che ci permetterà di vedere e modificare le suddette caratteristiche di visualizzazione

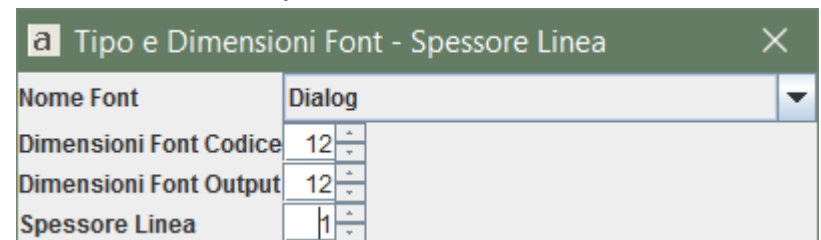
Altre Opzioni AlgoBuild – 1/3

- AlgoBuild permette di modificare le opzioni di visualizzazione del diagramma di flusso, permettendo di cambiare
 - Tipo di carattere
 - Dimensioni del carattere
 - Dimensioni del carattere nel pannello di output
 - Spessore delle linee relative al contorno dei blocchi ed agli archi orientati che collegano i blocchi (freccette)



- Cliccando sul tasto  apparirà una finestra di dialogo che permetterà di vedere e modificare le opzioni di visualizzazione

Finestra di Dialogo Modifica opzioni di visualizzazione



Altre Opzioni AlgoBuild – 2/3

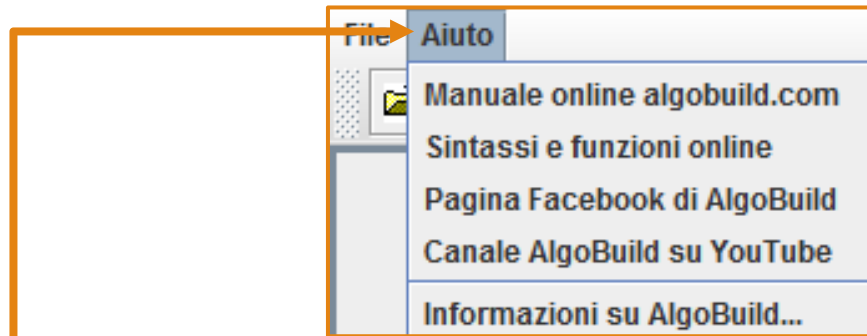
- È anche possibile **salvare** un diagramma di flusso, per poi **aprirlo** successivamente (ad esempio, per fare modifiche, per simulare altre esecuzioni, etc)
 - Le opzioni di salvataggio ed apertura sono presenti nella barra strumenti



- Le opzioni di salvataggio/apertura sono presenti anche nel menu «**File**»
 - In particolare, sono presenti anche opzioni per
 - Stampare il diagramma di flusso
 - Copiare il diagramma di flusso negli appunti
 - Il diagramma verrà copiato come immagine
 - Salvare il diagramma come immagine (*JPG, PNG, GIF*)



Altre Opzioni AlgoBuild – 3/3



- Il menu «**Aiuto**» permette di accedere rapidamente a
 - Manuale AlgoBuild (online)
 - Sintassi e funzioni (online)
 - Pagina *Facebook ufficiale* e canale *Youtube* (con suggerimenti e video)
 - Informazioni sulla versione di AlgoBuild

Riepilogo

- Primo approccio ad AlgoBuild
- Utilizzo dei blocchi di
 - *Input*
 - *Output*
 - *Esecuzione/Assegnamento*
- Simulazione di esecuzione, mediante AlgoBuild

