



Manuale Flowgorithm

Versione 2.20

Roberto Atzori

**« L'informatica non riguarda i computer più
di quanto l'astronomia riguardi i telescopi. »**

(Edsger Wybe Dijkstra)

Sommario

Flowgorithm	4
Caratteristiche e funzionalità	5
L'interfaccia	5
Supporto Multilingua.....	5
Barra del menù	7
Voce di menù "File"	7
Stampa	8
Voce di menù "Modifica"	9
Stile diagramma.....	9
Modifica Stile Diagramma	10
Modifica schema colori	11
Voce di menù "Programma"	13
Velocità di esecuzione	13
Gestione funzioni utente	14
Informazioni programma	14
Voce di menù "Strumenti"	15
Dimensionamento	15
Esporta immagine in un file.....	16
Esporta immagine in formato vettoriale	17
Finestra "Esecuzione"	18
Finestra "Codifica"	19
Finestra "Variabili"	20
Disposizione finestre	21
Voce di menù "Aiuto"	21
Barra degli strumenti standard	22
Menù contestuale.....	23
Specifiche.....	24
E ancora.....	24
Il mio primo diagramma di flusso	25
Tipi di dato.....	27
Integer (Intero)	27
Real (Reale).....	27
String (Stringa).....	27
Boolean (Booleano o Logico).....	27
Identificatori	28

Parole chiave	28
Operatori	28
Precedenze	29
Esempi	29
Funzioni predefinite	30
Matematiche	30
Stringa.....	30
Conversione tipo di dati	30
Altre	31
Costanti predefinite.....	31
Formato File.....	32
Esempio	32
Istruzioni	38
Assegnazione	38
Chiamata di Funzione o Procedura	39
Commento	39
Dichiarazione	40
Do	40
For.....	41
If.....	42
Input	42
Output	43
While.....	45
Per gli Amministratori di Sistema	46
Chiavi Registro	46
Contatto.....	47
Informazioni.....	47
Licenza	47
Revision History	48
Additional Information	63

Flowgorithm

Flowgorithm è un'applicazione gratuita che consente di creare programmi utilizzando diagrammi di flusso.

In [informatica](#) il **diagramma di flusso** (in inglese detto anche **flowchart**) rappresenta un [linguaggio di modellazione](#) grafico per rappresentare il flusso di controllo ed esecuzione di [algoritmi](#), procedure, istruzioni operative (in senso lato).

Esso consente di descrivere in modo schematico ovvero grafico:

- le operazioni da compiere, rappresentate mediante sagome convenzionali (rettangoli, rombi, esagoni, parallelogrammi, rettangoli smussati...), ciascuna con un preciso significato logico e all'interno delle quali un'indicazione testuale descrive tipicamente l'attività da svolgere
- la sequenza nella quale devono essere compiute sono rappresentate con frecce di collegamento.

Per questa loro connotazione topologica i diagrammi di flusso possono essere ricondotti alla classe più ampia dei [diagrammi a blocchi](#), che a loro volta rientrano nell'ancora più vasta categoria delle [mappe concettuali](#), utilizzate soprattutto per la descrizione e la rappresentazione delle informazioni e della conoscenza.

In genere, i programmi sono scritti utilizzando un editor di testo. A seconda del linguaggio di programmazione, questo può risultare facile o molto difficile per un programmatore alle prime armi. Molti linguaggi richiedono la stesura di diverse righe di codice solo per arrivare alla visualizzazione di una semplice frase come "Ciao, mondo!".

Tuttavia, utilizzando Flowgorithm, è possibile utilizzare dei simboli per rappresentare le diverse azioni che si desidera che il programma esegua. Quindi, ci si può concentrare su un algoritmo (**Pensiero Computazionale**) piuttosto che su tutte le sfumature e i dettagli propri di un tipico linguaggio di programmazione (**Coding**).

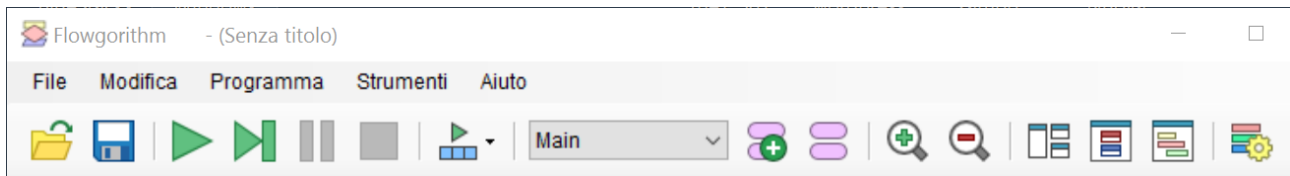
È possibile eseguire i programmi direttamente in Flowgorithm. Ma, se si vuole imparare un linguaggio di programmazione ad alto livello, Flowgorithm può convertire il diagramma di flusso in molti linguaggi popolari. Questi includono: C#, C++, Delphi/Pascal, Java, JavaScript, Lua, Perl, Python, QBasic, Ruby, Swift 2, Visual Basic .NET e Visual Basic for Applications (utilizzato in Microsoft Office), TypeScript, Scala.

Una importante collezione di algoritmi di vario genere, famosi o meno, creati con il software Flowgorithm, possono essere visionati e/o scaricati dal mio eBook presente nel mio sito all'indirizzo <http://robatz.altervista.org/blog/flowcharts-book/>.

Caratteristiche e funzionalità

L'interfaccia

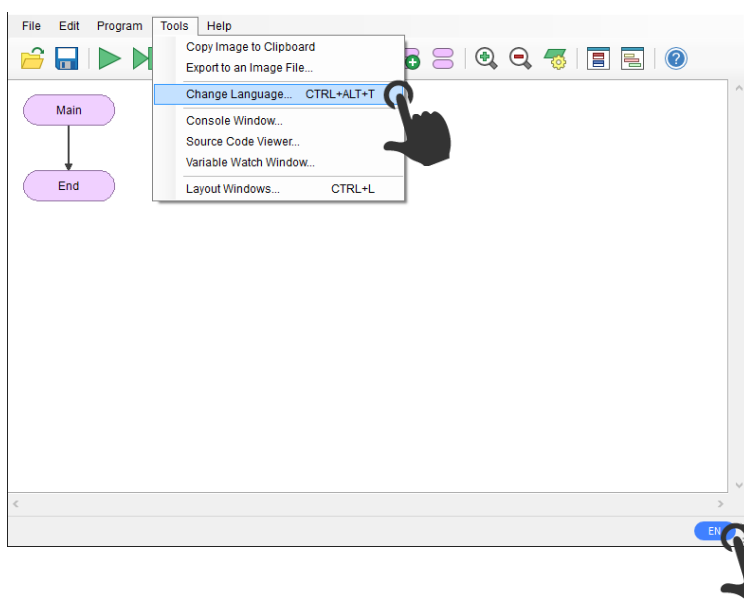
Flowgorithm si presenta con una interfaccia molto semplice ed intuitiva. La zona menù è corredata di tutte le funzionalità fornite dal software. La zona pulsanti è munita delle principali funzionalità per una facile implementazione dell'algoritmo, una facile esecuzione e un facile controllo dello stesso.



Supporto Multilingua

Flowgorithm è tradotto in molti linguaggi:

- Cinese (Semplificato)
- Ceco
- Inglese - U.S. e British
- Francese
- Galiziano
- Tedesco
- Ungaro
- Italiano
- Giapponese
- Portoghese
- Spagnolo – Messicano e Castigliano
-

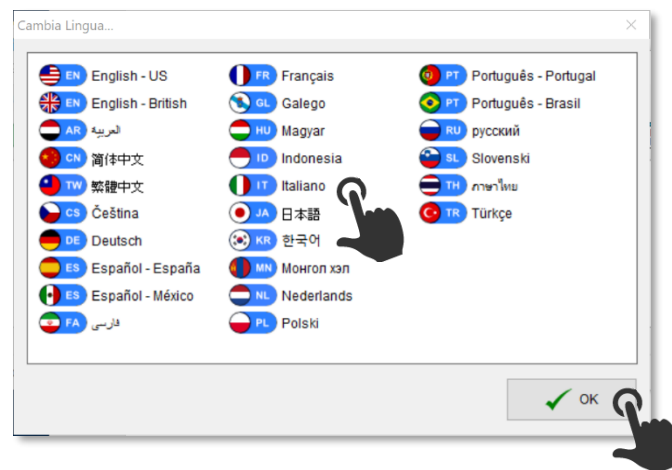


Per cambiare linguaggio (inizialmente) possiamo cliccare la voce di menù **Tools** → **Change Language**

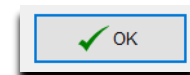
Oppure fare doppio click col tasto sinistro del mouse in basso a destra della finestra in corrispondenza della figura



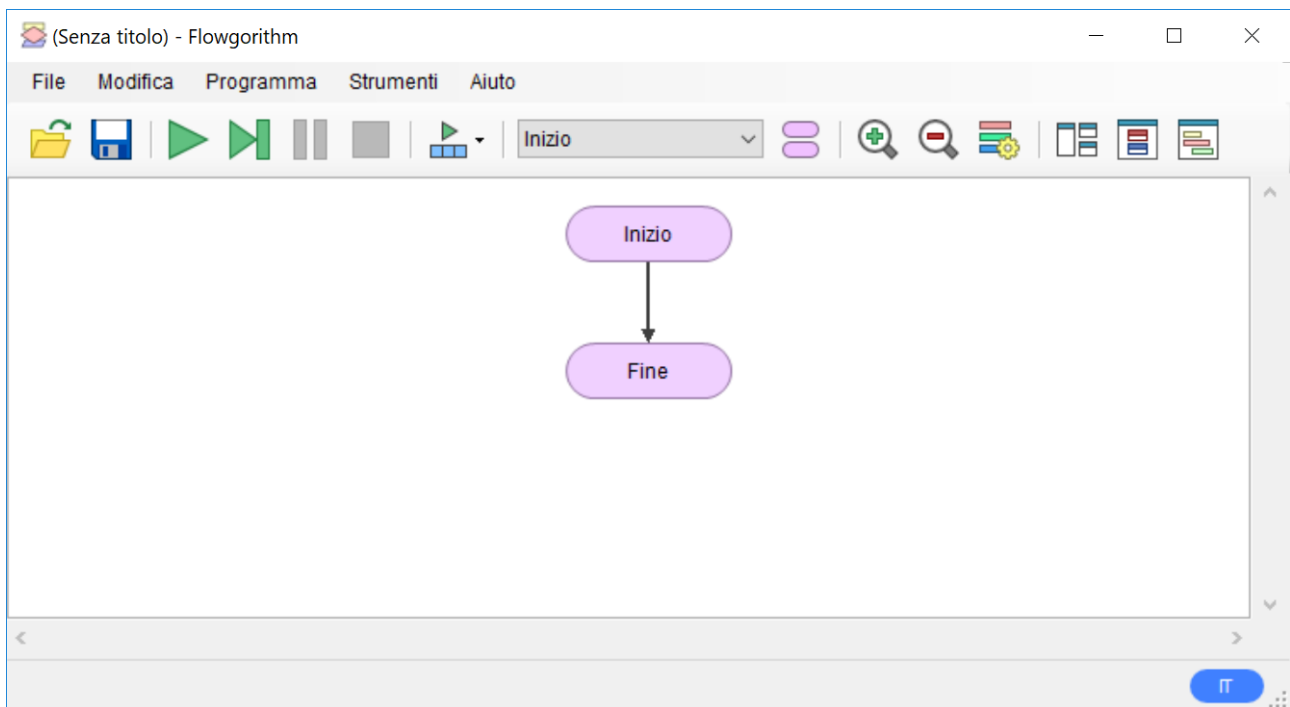
Verrà aperta la finestra di selezione nuovo linguaggio:



Selezionare la voce "Italiano" e quindi confermare cliccando sul pulsante



A questo punto l'intera interfaccia di Flowgorithm userà l'idioma italiano.



e il simbolo **EN** cambierà in **IT** ad indicare la lingua correntemente in uso.

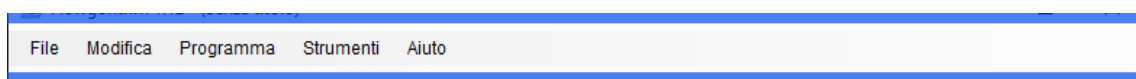
Barra del menù

La **barra dei menu** presenta la maggior parte dei comandi disponibili nel programma, suddivisi per menu (elenchi di scelte). Tramite tastiera è possibile consultare la barra dei menu premendo il tasto **Alt** e spostandosi con la **Freccia Destra** o **Sinistra** per leggere le intestazioni di ogni menu. Una volta individuato quello di interesse, è possibile leggere le voci contenute al suo interno utilizzando i comandi **Freccia Giù** o **Su**. Per attivare una voce è sufficiente, dopo averla individuata, premere il tasto Invio.

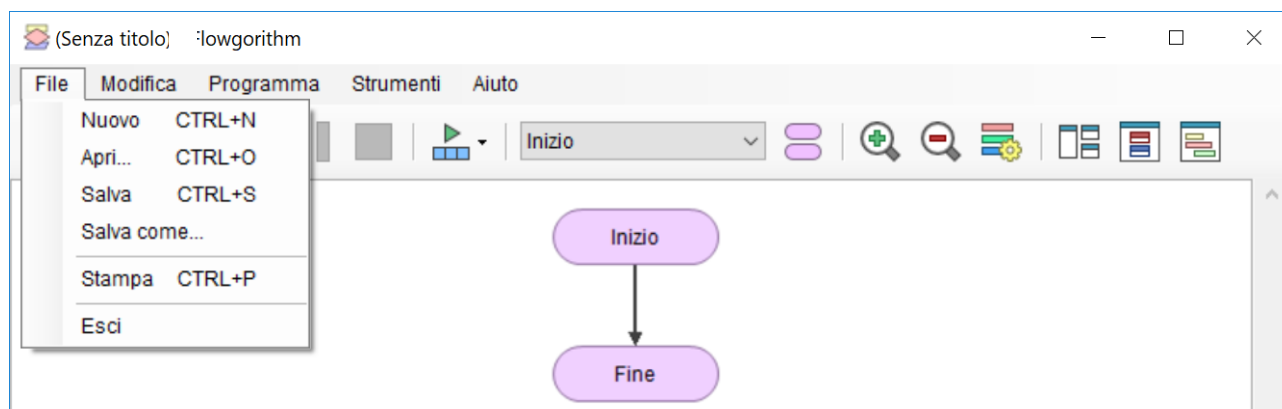
Per aprire ciascuno dei menu disponibili, si possono utilizzare anche le seguenti combinazioni di tasti, dette anche tasti caldi (hot):

- **Alt più F** per il menu **File**;
- **Alt più M** per il menu **Modifica**;
- **Alt più P** per il menu **Programma**;
- **Alt più S** per il menu **Strumenti**;
- **Alt più A** per il menu **Aiuto**.

Per chiudere un menu è sufficiente selezionare un comando del menu, attivare un altro menu o premere il tasto **Alt**.



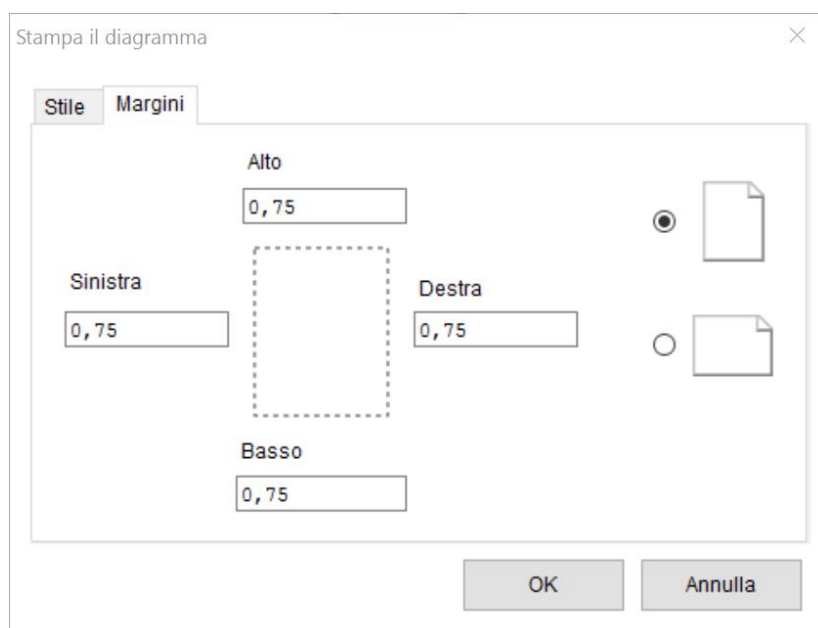
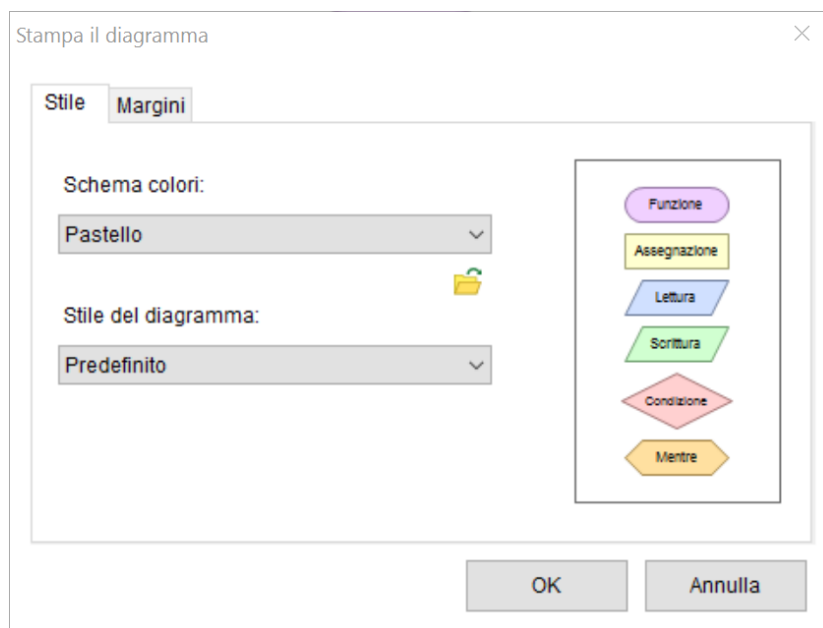
Voce di menù "File"



Voce menù	Descrizione
Nuovo	Crea un nuovo flowchart
Apri	Carica un flowchart precedentemente creato e salvato
Salva	Salva il flowchart
Salva come...	Salva il flowchart con la possibilità di cambiare nome allo stesso e cartella dove salvarlo
Stampa	Per stampare la zona correntemente visualizzata
Esci	Uscire dal programma

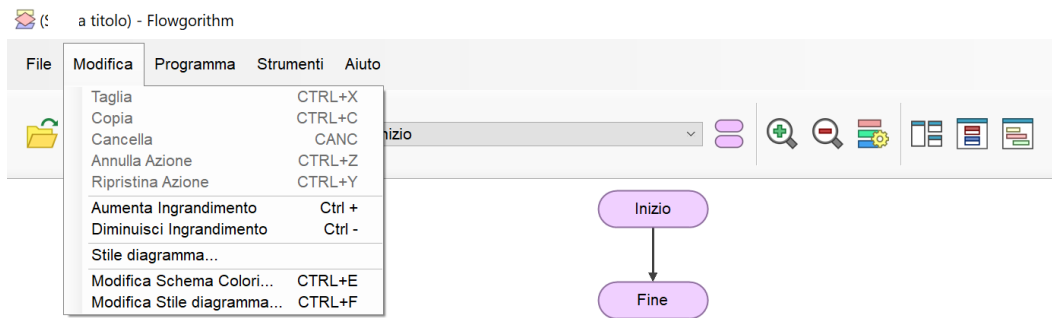
Stampa

Selezionando la voce di menù “Stampa” si aprirà la seguente finestra:



attraverso la quale è possibile inserire delle impostazioni personalizzate per la stampa.

Voce di menù “Modifica”



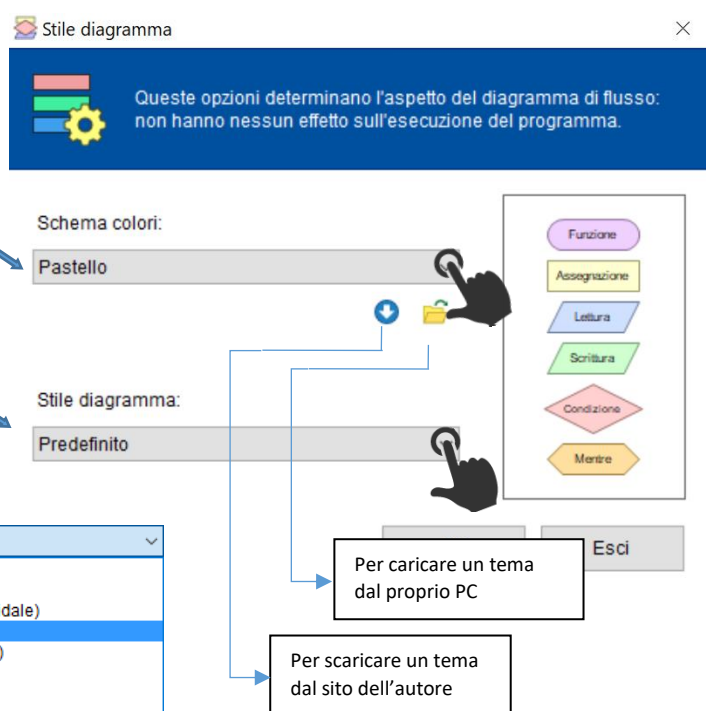
Voce menù	Descrizione
Taglia	Cancellate l'elemento selezionato copiandolo in memoria
Copia	Copia in memoria l'elemento selezionato
Cancella	Cancella l'elemento selezionato
Annulla Azione	Annulla operazione inserimento forme, digitazione, ...
Ripristina Azione	Ripristina operazione annullata
Aumenta ingrandimento	Ingrandisce la zona
Diminuisci ingrandimento	Rimpicciolisce la zona
Stile diagramma ...	Seleziona lo stile del diagramma
Modifica Schema Colori ...	Modifica colorazione temi
Modifica Stile diagramma ...	Modifica lo stile del diagramma

Stile diagramma

Particolare importanza riveste la scelta della selezione “Stile del diagramma” che porta all’apertura di una nuova finestra:

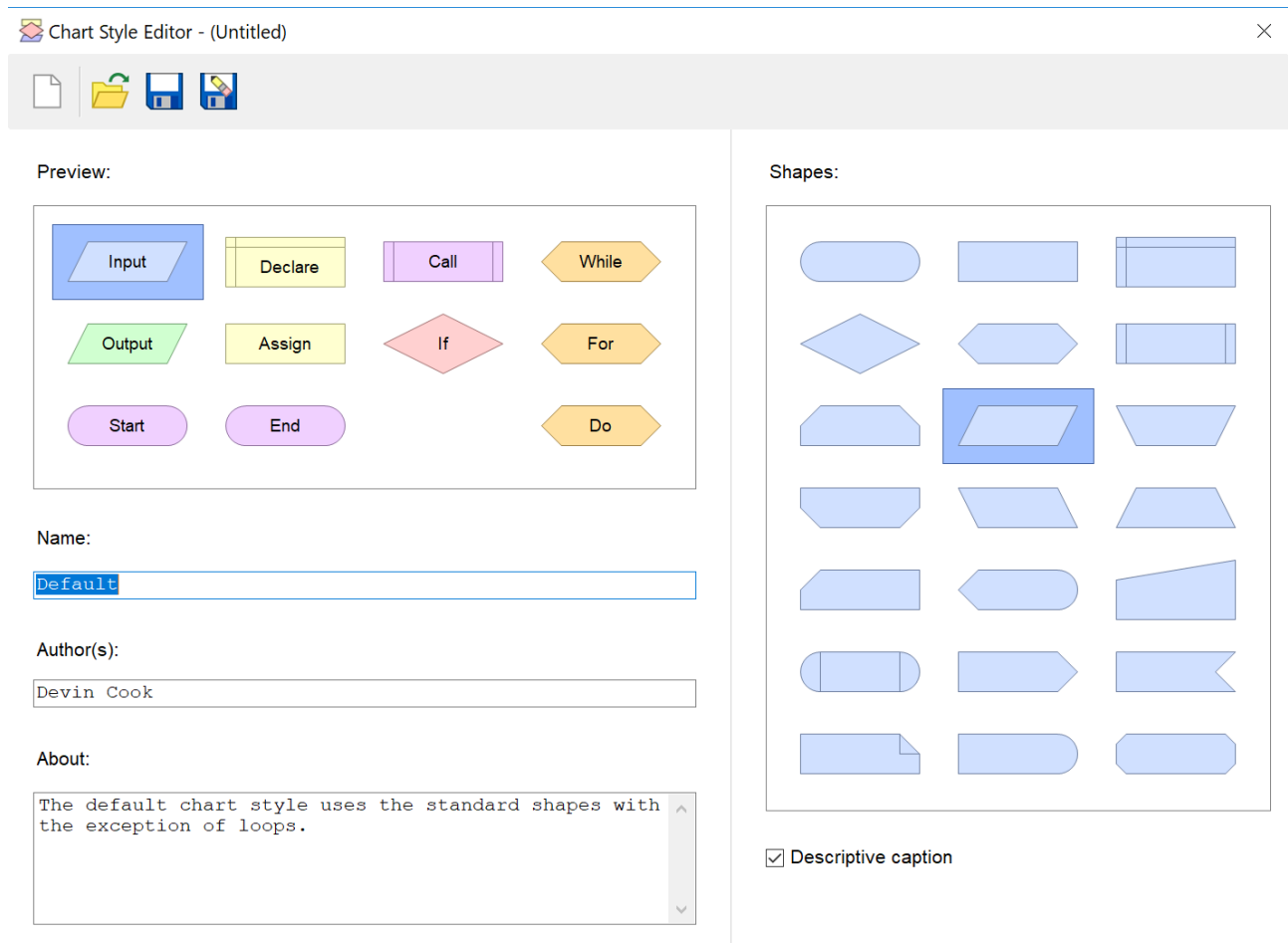
Da questa finestra è possibile utilizzare un **Pannello di colori** diverso da quello preimpostato cliccando sulla finestrella

oppure cambiare **Stile diagramma**



Modifica Stile Diagramma

Selezionando questa voce, è possibile modificare la grafica dei simboli predefiniti con altri simboli alternativi



Questo ti permetterà di creare qualsiasi stile (entro certi limiti) del diagramma di flusso di cui hai bisogno. Attualmente sono supportate 20 forme, ma in futuro potrebbero esserne aggiunte altre.

Modifica schema colori

Selezionando la voce di menù “Modifica schema colori” si aprirà la seguente finestra:

The screenshot shows the 'Modifica Schema Colori' window. A red box highlights the 'Proprietà del blocco 'Output'' panel, which contains a grid of flowchart symbols: Breakpoint, Input, Output, Incom..., Error, Intestazione, Dichiarazio..., Assegnazione, Librarsi, Call, If, Execu..., and Function. To the right of this panel is the 'Modifica Colore: Pagina' section, which includes three color selection tools: 'Colore di riempimento' (fill color), 'Colore del bordo' (border color), and 'Colore testo' (text color). Each tool has a color picker and a 'Preleva Colore' (pick color) button. Blue arrows point from these buttons to the 'Guida Immagine' (Image Guide) area, which is a large rectangular area for selecting colors from an image. A yellow box labeled 'Zona visualizzazione immagine dalla quale prelevare il colore con lo strumento 'Preleva Colore'' points to this area. Another yellow box labeled 'Titolo del Tema' points to the 'Nome:' field. A third yellow box labeled 'Autore/i' points to the 'Autore/i:' field. A fourth yellow box labeled 'Area cliccabile col mouse per selezionare l'oggetto cui cambiare le proprietà colore' points to the 'Proprietà del blocco 'Output'' panel.

Anteprima:

Nome:

Autore/i:

Informazioni:

Guida Immagine:

Modifica Colore: Pagina

☐ Colore di riempimento

☐ Colore del bordo

☐ Colore testo

Strumento Seleziona Colore

Strumento Preleva Colore dall'area "Guida Immagine"

Strumento Preleva Colore che copierà tutti gli attributi (testo, riempimento e contorno) da un'altro Form selezionato

Zona visualizzazione immagine dalla quale prelevare il colore con lo strumento 'Preleva Colore'

Titolo del Tema

Autore/i

Area cliccabile col mouse per selezionare l'oggetto cui cambiare le proprietà colore

All'indirizzo

<http://flowgorithm.org/download/schemes.htm>

si possono trovare gli schemi preinstallati ed altri nuovi, la cui colorazione può essere modificata dall'utente.



[Main](#) [About](#) [Download](#) [Schemes](#) [Documentation](#) [Resources](#)

Color Schemes

Seasonal

- Christmas**
The colors of a snowy day
- Christmas Tree**
- Easter**
- Fall**
- Halloween**
- Disneyland's Haunted Mansion**
The colors of the creepy wallpaper

Science Fiction

- Doctor Who**
The colors of the TARDIS
- LCARS**
The colors from Star Trek: The Next Generation
- The Orville**
The new Star Trek inspired show
- Mad Science**
Mystery Science Theater 3000 inspired

Movies & Television

- Hoplitte**
Based on the movie "300"
- Napoleon Dynamite**
The color of the colors from the classic film
- Robbie Rotten**
Larry Town's greatest villain. We are #1
- Tea Party**
Alice in Wonderland. Happy un-birthday!
- Mr. Meeseeks**
Colors based on the popular character from Rick and Morty

Harry Potter

These color schemes play homage to the fantastic Harry Potter books by J.K. Rowling.

- Gryffindor**
- Hufflepuff**
- Ravenclaw**
- Slytherin**

Computer Science

- DOB**
- Solarized - Solid**
Popular color scheme used by computer programmers.
- Solarized - Outline**
Outline version of the scheme
- Windows 8**
The classic colors from the much-maligned Start Screen.
- Windows 8 - Border**
The same scheme, but the shapes have a black border for contrast.

Computer Hardware

- Circuit Board**
- HP Calculator**
The classic orange, blue and brown.
- Liquid Crystal**
For that creepy electronics look
- Terminal**
The classic green computer screen.

Classic Video Games

- Bard's Tale III**
A classic RPG game on the Commodore 64
- Battle Chess**
The colors of the classic 1980's game.
- Super Mario Brothers**
Nintendo Entertainment System (NES)
- Super Mario Brothers - Castle Level**
Nintendo Entertainment System (NES)
- Super Mario Brothers 3**
Nintendo Entertainment System (NES)

Classic Computer Systems

- Commodore Amiga Workbench**
Colors of a very early GUI
- Commodore 64**
- CGA - Palette 1 - Low**
Black, cyan, magenta, gray
- CGA - Palette 1 - High**
Black, light cyan, light magenta, white
- CGA - Palette 2 - Low**
Black, green, red, amber
- CGA - Palette 2 - High**
Black, light green, pink, yellow
- GEOS**
An early GUI on the Commodore 64.
- Windows 3.1 Hotdog Stand**
(Yes, this was real).

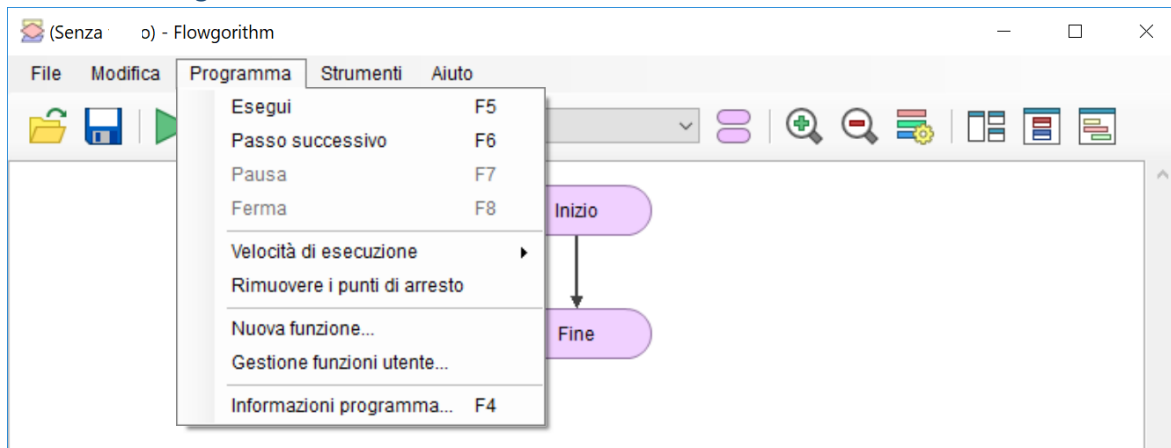
Miscellaneous

- 1970's - Brown**
- 1970's - Green**
- Chalkboard - Black**
- Chalkboard - Green**
- City Lights**
- Flag of California**
- Medical Clinic**
- Traffic - At Night**
This was the original traffic color scheme.

Created A Scheme?

If you created a color scheme, and would like to see it on the site, please send it to Devin @ Flowgorithm.org.

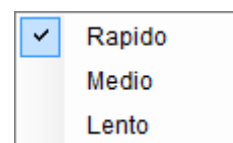
Voce di menù “Programma”



Voce menù	Descrizione
Esegui	Esegue il flowchart (interpretazione)
Passo successivo	Esegue passo passo il flowchart
Pausa	Pausa esecuzione
Ferma	Interrompe l'esecuzione
Velocità di esecuzione	Selezione della velocità di esecuzione in modalità automatica
Rimuovere i punti di arresto	Rimuove tutti i punti di interruzione inseriti
Nuova funzione	Aggiunge una nuova funzione
Gestione funzioni utente ...	Gestione delle funzioni
Informazioni programma ...	Proprietà del programma

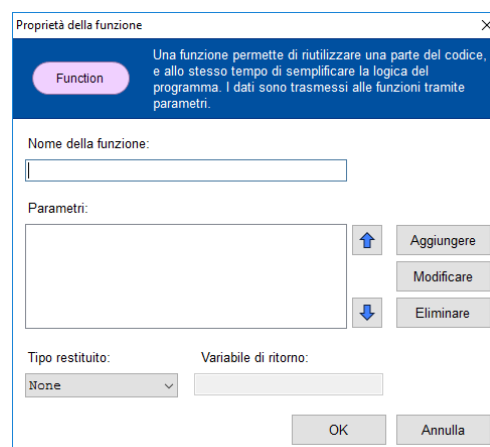
Velocità di esecuzione

Selezionando la voce di menù **Velocità di esecuzione** si aprirà un altro menù attraverso il quale sarà possibile individuare la velocità di esecuzione in automatico delle istruzioni.



Selezionando la voce di menù **Aggiungi una funzione** si aprirà un'altra finestra tramite la quale sarà possibile creare procedure e funzioni accessori e utili al funzionamento dell'algoritmo stesso.

Ovviamente questa zona necessita di conoscenze di programmazione avanzata per poterla sfruttare appieno.



I pulsanti freccia servono per modificare l'ordine dei parametri della funzione. Modificando l'ordine dei parametri si deve prestare attenzione a quanto scritto nel flowchart!!

Gestione funzioni utente

Selezionando la voce di menù **Gestione funzioni utente** si accede alla lista delle funzioni create per l'algoritmo corrente e tramite la quale è possibile modificare le funzioni create, duplicarle (copia e incolla) o cancellare quelle che non servono più.

Taglia

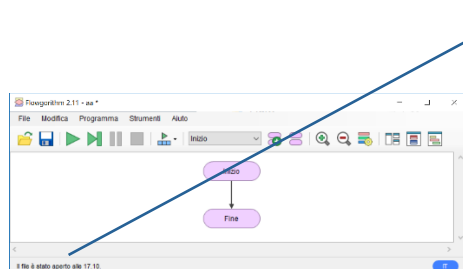
Incolla

Copia

Informazioni programma

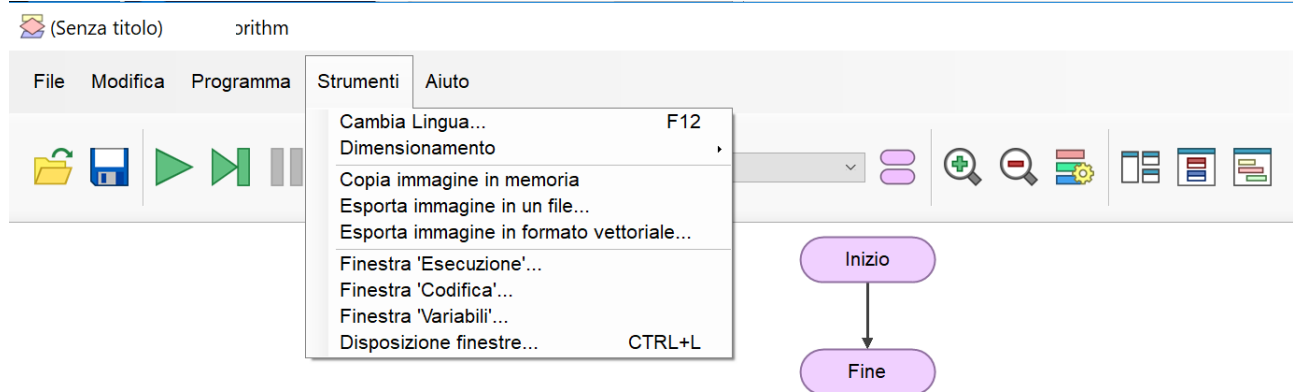
Selezionando la voce di menù **Informazioni programma** (oppure cliccando sul tasto funzione **F4**) viene attivata un'altra finestra che offre la possibilità di poter inserire ulteriori informazioni sul flowchart.

Queste informazioni, comunque, non modificano il funzionamento del flowchart.



Informazioni
ultimo salvataggio
(solo lettura)

Voce di menù “Strumenti”



Voce menù	Descrizione
Cambia Lingua...	Cambia lingua
Dimensionamento	Dimensionamento interfaccia grafica
Copia immagine in memoria	Copia l'immagine in memoria
Esporta immagine in un file ...	Esporta l'immagine area attiva in un file PNG
Esporta immagine in formato vettoriale	Esporta l'immagine area attiva in un file in formato vettoriale .SVG (Scalable Vector Graphics) o .EMF (Enhanced Metafile Format)
Finestra 'Esecuzione' ...	Abilita la finestra per monitorare l'esecuzione del programma e per inserire i valori da tastiera
Finestra 'Codifica' ...	Abilita la finestra per visualizzare la codifica automatica nei linguaggi previsti da Flowgorithm
Finestra 'Variabili' ...	Abilita la finestra per il controllo delle variabili utilizzate nel programma
Disposizione finestre ...	Disposizione delle finestre nel desktop

Dimensionamento

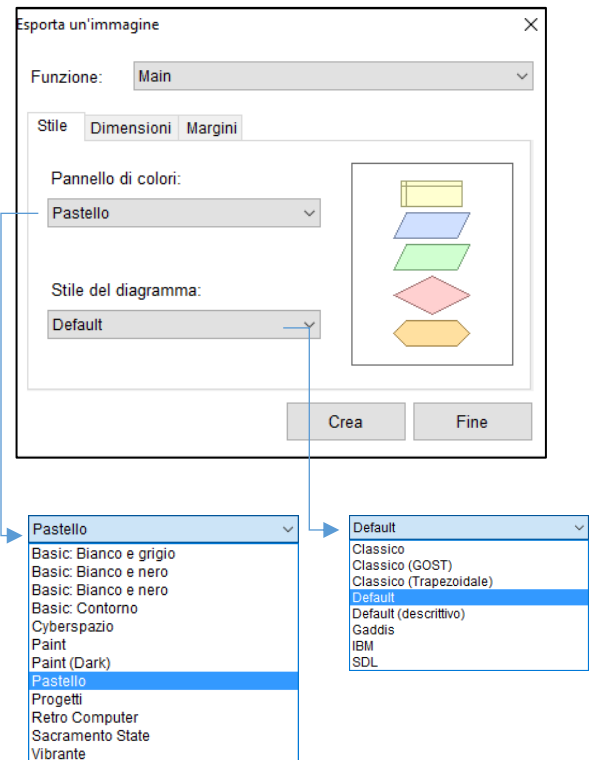
L'applicazione ora è "DPI-aware". Ciò significa che la finestra si "ridimensionerà" su schermi più grandi (con grande risoluzione) utilizzando un fattore di scala. Normalmente questo è impostato al 125% o 150%. L'utente può anche modificare il fattore di scala corrente dal menu Strumenti → Dimensionamento. Prima di questo cambiamento, Windows avrebbe semplicemente, su schermi più grandi come ad esempio un 4K, rimpicciolito tutta la grafica rendendola poco leggibile.

Esporta immagine in un file

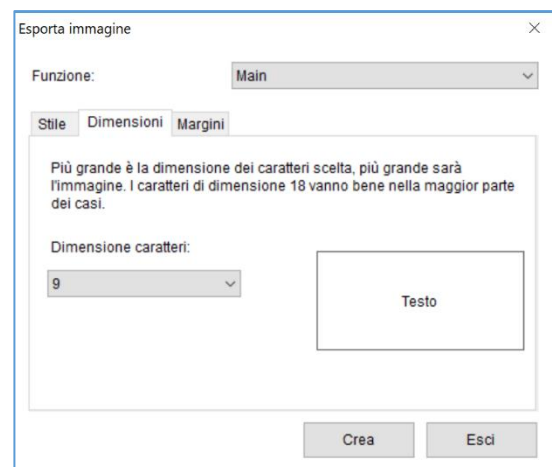
Selezionando la voce di menù **Esporta immagine in un file** viene aperta la seguente finestra. Se lasciamo tutto così, come ci viene proposto, e clicchiamo sul pulsante **Crea**, ci verrà proposto di dare un nome al file (con estensione PNG) e di selezionare in quale cartella salvarlo.

All'interno della finestra ci vengono proposte tre schede per la selezione di ulteriori impostazioni prima della creazione.

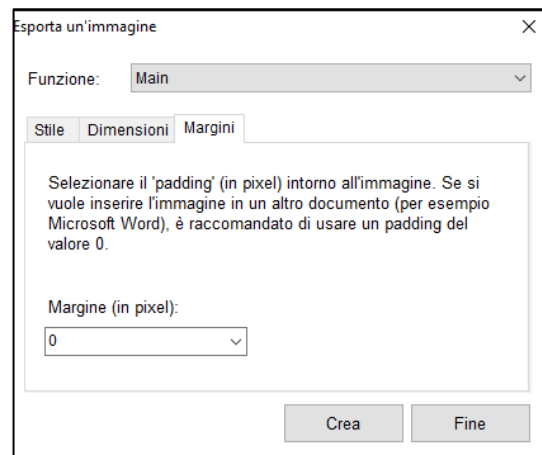
La prima scheda **Stile** ci ricorda quella che viene attivata dalla sequenza del menù **Modifica** → **Scegli** lo stile del diagramma vista precedentemente.



La seconda scheda **Dimensioni** permette di impostare la grandezza del carattere. Un valore di 18 è considerato un valore adatto nella maggior parte dei casi.



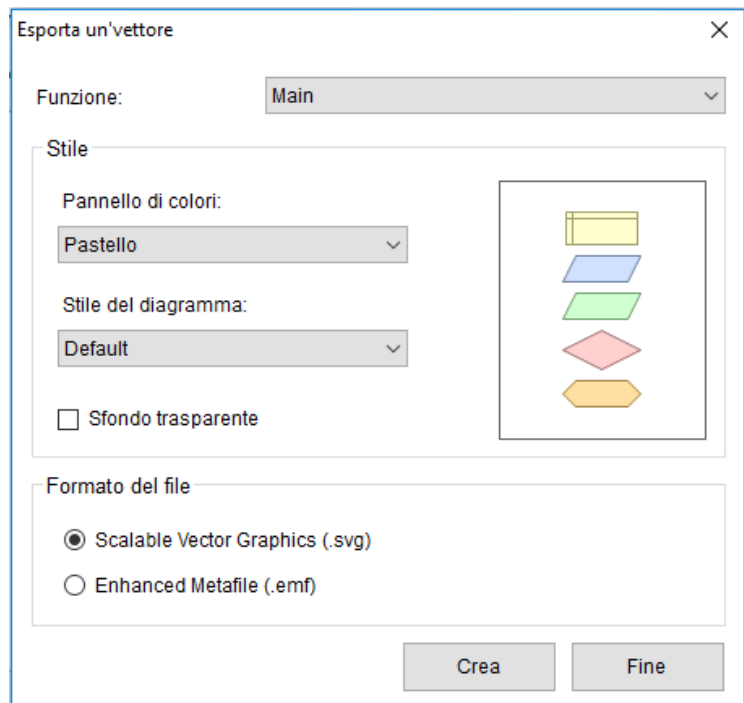
L'ultima scheda **Margini** consente di impostare lo spazio in pixels intorno all'immagine (padding). Per Word è consigliato impostare tale spazio a 0.



Esporta immagine in formato vettoriale

Selezionando la voce di menù **Esporta immagine in formato vettoriale** viene aperta la seguente finestra.

È possibile salvare l'immagine della funzione scelta in un file in formato .SVG (Scalable Vector Graphics) o in formato .EMF (Enhanced Meta File). Inoltre è possibile salvare le immagini anche con sfondo trasparente.




Finestra "Esecuzione"

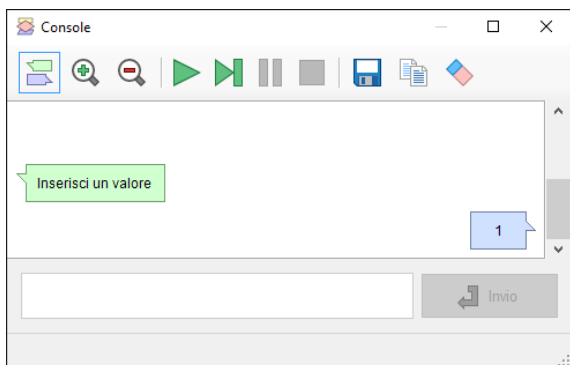
Selezionando la voce di menù **Finestra "Esecuzione"** viene attivata una finestra utilizzata per monitorare l'esecuzione dell'algoritmo e per inserire da tastiera i valori necessari.

Il metodo classico per interagire con il computer è utilizzare la modalità "Console". Usando questo approccio, il programma visualizza in uscita su uno schermo solo testo e l'utente immette i dati utilizzando la tastiera. A volte la console permette al testo di cambiare i colori, ma, per la maggior parte, è testo bianco su sfondo nero.

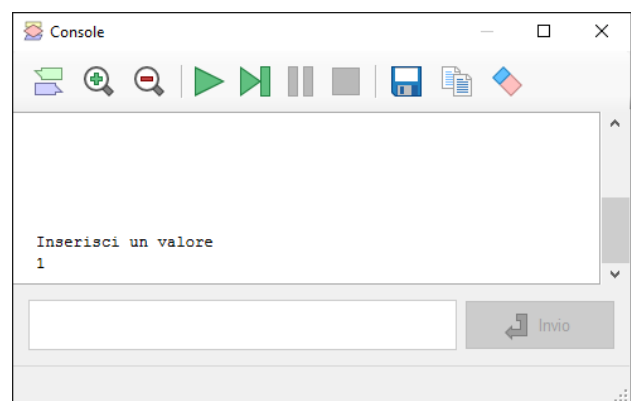
Mentre la Console funziona, la sua interfaccia semplice può rendere difficile distinguere l'input dell'utente e l'output del programma. Così, invece di usare la schermata di solo testo, Flowgorithm trasforma il tutto in una tipica finestra di messaggio. Dal punto di vista dei programmatori, sono messaggi diretti e testuali.



I messaggi sono colorati in abbinamento alle forme di ingresso e di uscita utilizzati nel diagramma di flusso. Nella schermata a destra, l'input dell'utente è visualizzato in blu, mentre l'output del programma viene visualizzato in verde. Nella sua **barra degli strumenti** si possono notare delle icone che non compaiono nella barra degli strumenti della finestra principale.

Il pulsante  permette di visualizzare o meno i valori sotto forma di riquadri di dialogo (visione grafica)




Se viene disabilitato possiamo notare questa differenza di tipo grafica:



I pulsanti   hanno la funzionalità di ingrandire o rimpicciolire il contenuto della finestra.

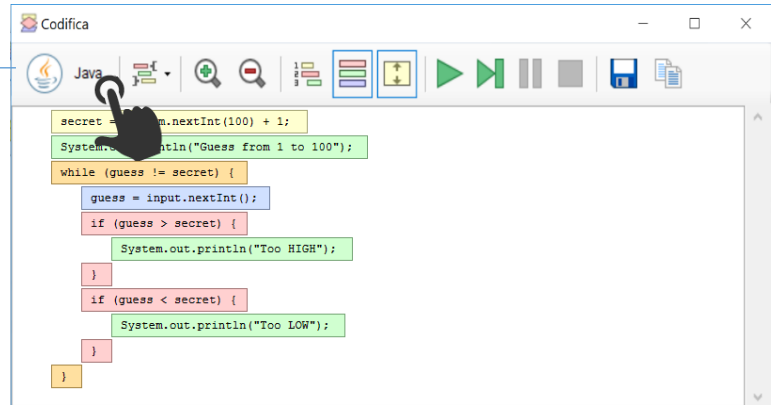
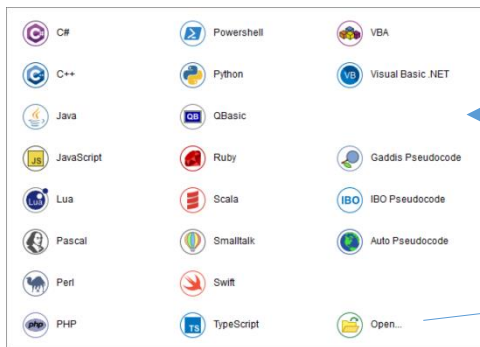
Il pulsante  permette di salvare il contenuto della finestra in un file testuale (.TXT)

Il pulsante  permette di copiare in memoria il contenuto della finestra

Il pulsante  permette di cancellare il contenuto della finestra


Finestra "Codifica"

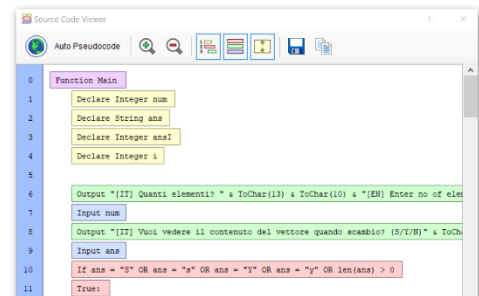
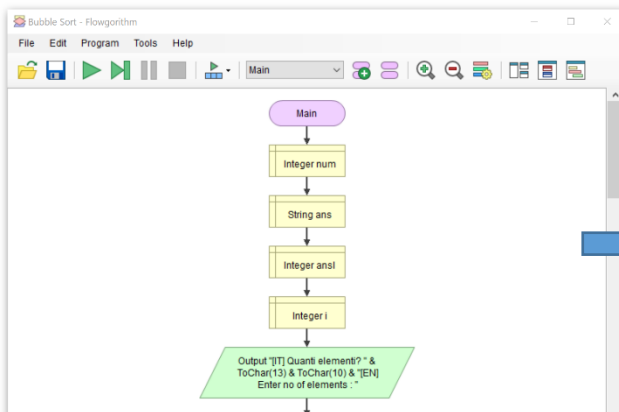
Selezionando la voce di menù **Finestra "Codifica"** si aprirà la finestra tramite la quale sarà possibile vedere la trasformazione del flowchart creato, codificato **in automatico** in uno dei tanti linguaggi di programmazione predisposti da Flowgorithm.



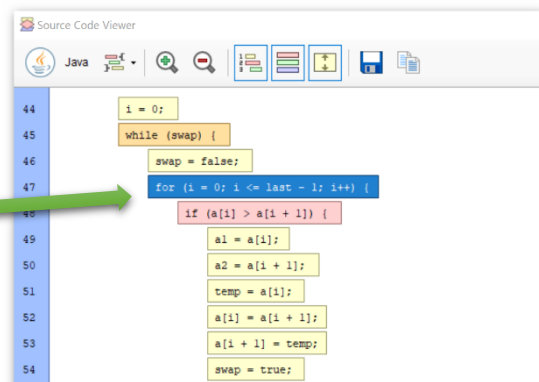
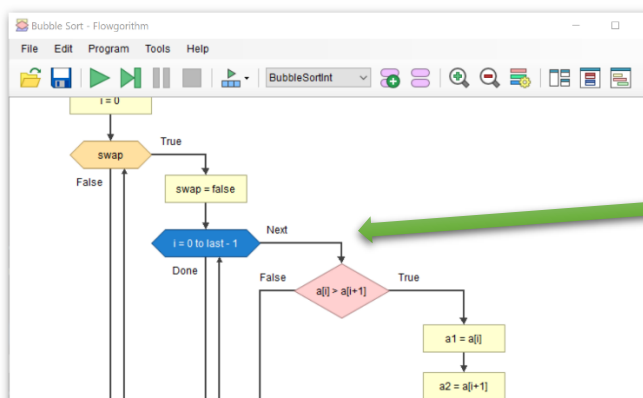
Apertura file **.fpgt** per utilizzo di modelli di programma definiti dall'utente


Oltre i pulsanti che abbiamo già incontrato, nella barra degli strumenti di questa finestra ne troviamo altri nuovi.

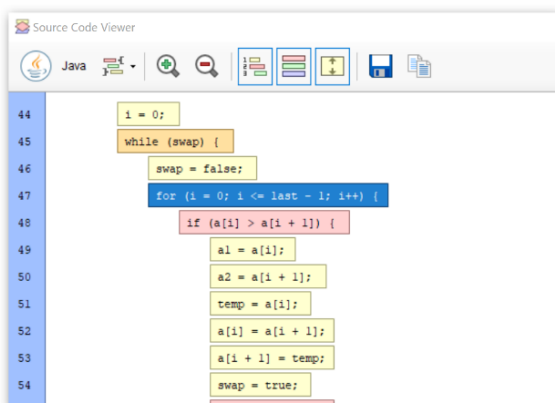
Il pulsante  permette di disabilitare o abilitare l'evidenziazione delle istruzioni



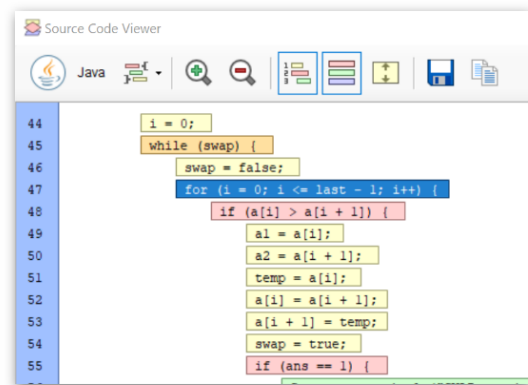
Da sottolineare il fatto che questa funzionalità permette di individuare in maniera interattiva l'elemento del diagramma direttamente nella finestra codice. Infatti selezionando, ad esempio, una struttura del diagramma di flusso, verrà evidenziata la stessa all'interno della finestra codice di riferimento.




Il pulsante  permette di rimuovere o aggiungere lo spazio tra le righe per una migliore lettura

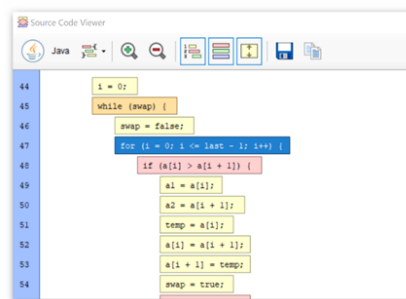


(attivato)

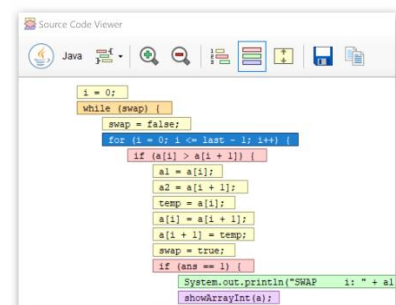


(disattivato)

Il pulsante  permette di disabilitare o abilitare la numerazione delle righe



(attivato)



(disattivato)

Se avete intenzione di imparare un nuovo linguaggio di alto livello, questa caratteristica è di grande aiuto. Il codice viene evidenziato utilizzando lo stesso colore utilizzato per le forme corrispondenti utilizzate nel diagramma di flusso. Questo permette, visivamente, di vedere la correlazione tra il codice e il diagramma di flusso. Inoltre, se si evidenzia una forma nel diagramma di flusso, o fate un passo con il vostro programma, il codice sorgente verrà evidenziato nello stesso punto.

Finestra "Variabili"

Selezionando la voce di menù **Finestra "Variabili"** si aprirà la finestra. La finestra di controllo delle variabili viene utilizzata per tenere traccia di come le variabili cambiano man mano che il programma viene eseguito. Quindi, se si è arrivati ad un punto del programma, la finestra mostrerà ogni variabile e il suo valore corrente. Questo non è limitato solo a variabili fondamentali. Anche gli array verranno visualizzati.

Ogni variabile ha un colore codificato in base al suo tipo di dati. A prima vista, si può dire esattamente che tipo di dati vengono memorizzati e focalizzarsi su un determinato tipo di dato.

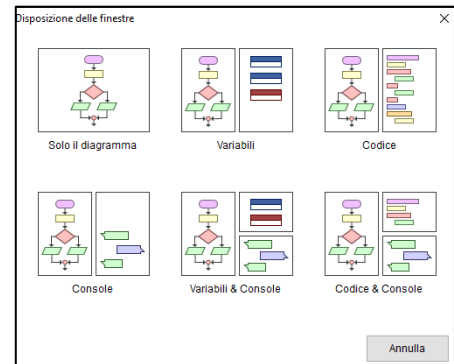
year
1947
ratio
1.618
name
Sacramento State
done
True

- Gli Interi sono visualizzati in blu.
- I numeri Reali sono visualizzati in viola.
- Le variabili di tipo Stringa sono visualizzate in rosso.
- Le variabili Booleane sono visualizzate in verde scuro.

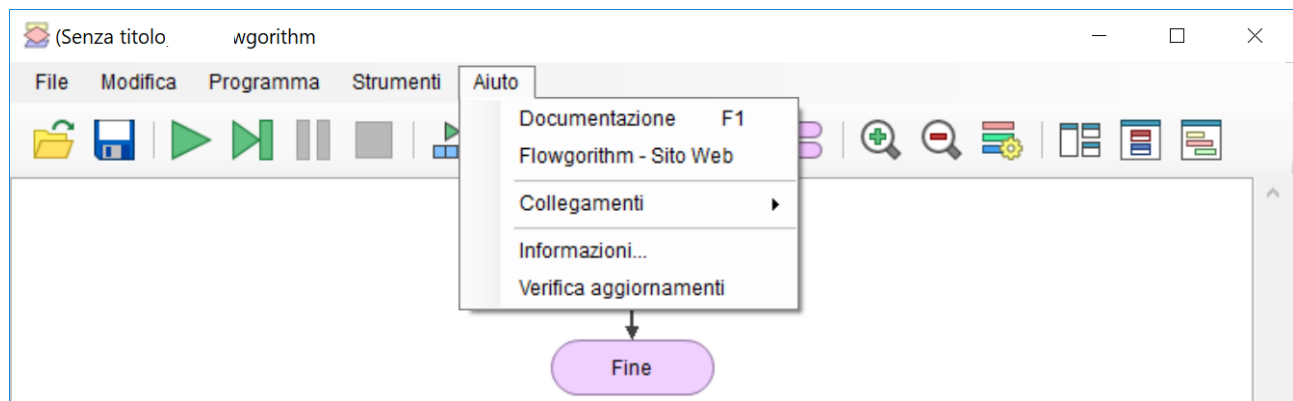
Fibonacci	
0	0
1	1
2	1
3	2
4	3
5	5
6	8
7	13

Disposizione finestre

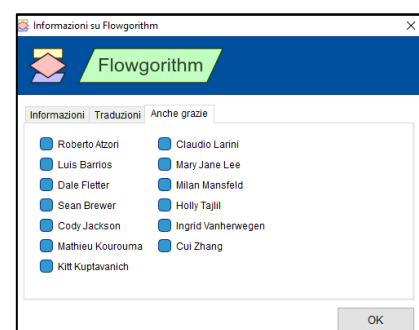
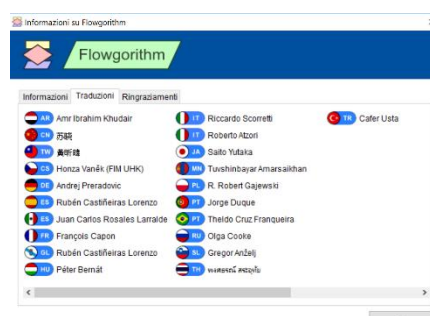
Selezionando la voce di menù **Disposizione finestre** si aprirà la finestra per dare la possibilità di una impostazione delle finestre definita dall'utente secondo le proprie necessità.



Voce di menù "Aiuto"



Voce menù	Descrizione
Documentazione	Aprire la pagina online relativa alla documentazione http://www.flowgorithm.org/documentation/
Flowgorithm – Sito Web	Aprire la pagina iniziale online relativa al sito Flowgorithm http://www.flowgorithm.org
Collegamenti	Aprire un sottomenu di collegamenti all'Università di appartenenza dell'autore di Flowgorithm
Informazioni	Aprire finestra informazioni sul software, traduzioni e ringraziamenti a chi ha partecipato all'evoluzione di questo software
Verifica aggiornamenti	Permette di controllare se esistono nuove versioni del software



Barra degli strumenti standard

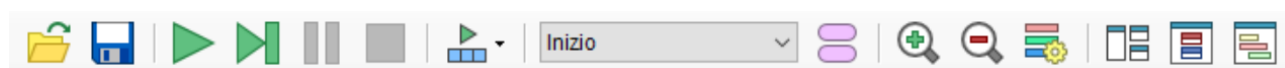
Immediatamente sotto la barra dei menu è presente la **barra degli strumenti standard**, che contiene i pulsanti dei comandi utilizzati più di frequente (tasti di scelta rapida) per la gestione dei file e per l'esecuzione delle operazioni più importanti.

Attenzione: i pulsanti della barra degli strumenti standard corrispondono alle voci più utilizzate presenti nella barra dei menu; la barra degli strumenti standard, perciò, ha solo la funzione di rendere tali voci più accessibili. Per esempio, se si vuole salvare il documento su cui si sta scrivendo è sufficiente utilizzare l'apposito pulsante presente sulla barra degli strumenti senza dover ogni volta aprire il menu **File** e selezionare la voce **Salva**.

Per selezionare i pulsanti della barra degli strumenti standard tramite tastiera è possibile:

- posizionarsi sulla barra dei menu tramite il tasto **Alt**;
- utilizzare la combinazione di tasti **Control** più **Tab**.

In questo modo viene selezionato il primo pulsante della barra; per spostarsi fra i pulsanti è sufficiente utilizzare il tasto **Tab** per muoversi da sinistra verso destra e la combinazione di tasti **Shift** più **Tab** per muoversi da destra verso sinistra. Una volta individuato il pulsante d'interesse occorre premere il pulsante **Invio** per attivarlo.

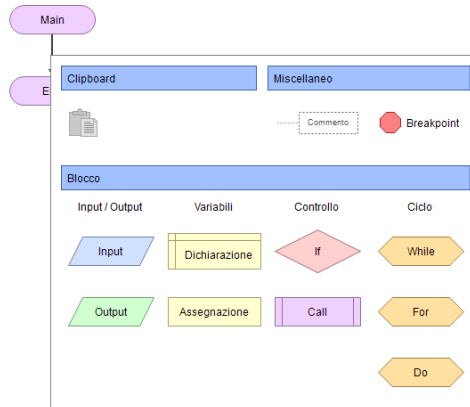
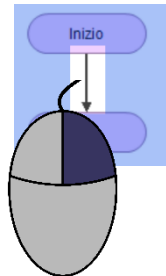


Pulsante	Funzione
	Carica un flowchart precedentemente creato e salvato
	Salva il flowchart
	Esegui (interpreta) il flowchart
	Singola iterazione (passo passo)
	Pausa
	Stop
	Velocità di esecuzione (automatica) del flowchart
	Selezioni Funzioni o Procedure
	Aggiungi una funzione utente
	Gestione delle Funzioni utente
	Zoom avanti (ingrandimento visualizzazione)
	Zoom indietro (rimpicciolimento visualizzazione)
	Scegli lo stile del diagramma
	Disposizione finestre nel desktop
	Osserva le variabili
	Osserva il codice sorgente

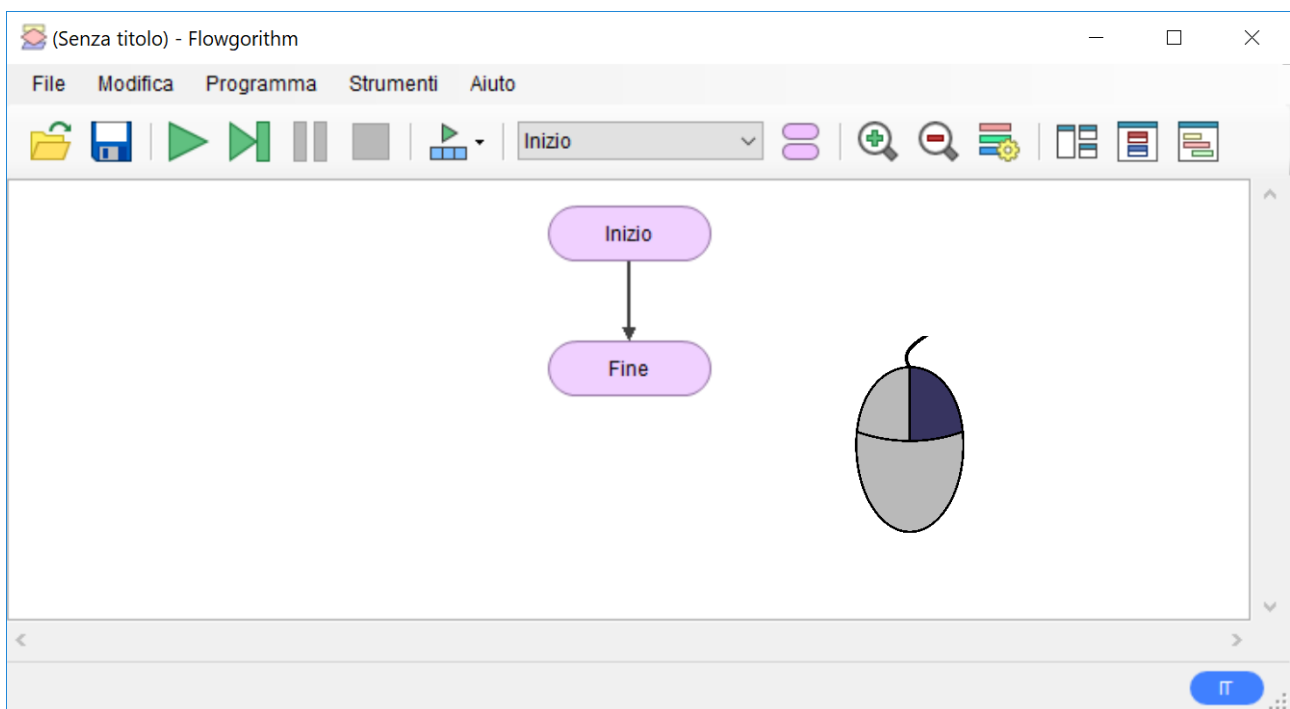
Menù contestuale

Il menù contestuale può essere attivato utilizzando il tasto destro del mouse su specifiche posizioni.

- Se si clicca, col tasto destro del mouse, sulla freccia di flusso apparirà un menu pop-up mostrando tutte le figure che possono essere inserite



- Se si clicca, col tasto destro del mouse, sulla parte vuota della finestra di editazione



si attiverà il menù pop-up a destra attraverso il quale è possibile passare a menù già trattati precedentemente.



Specifiche

- Array mono dimensionali
- Cicli con controllo in testa e in coda
- Cicli enumerativi (FOR)
- Dichiarazione esplicita delle variabili. I diagrammi di flusso generalmente usano dichiarazione implicita (essendo abbastanza minimalisti). Tuttavia, praticamente tutti i linguaggi di programmazione utilizzano dichiarazioni esplicite. Quindi, è una buona idea per gli studenti impararlo fin dall'inizio.
- Ricorsione controllata. Flowgorithm è fornito di uno stack interno (piuttosto che lo stack del sistema). Se lo studente crea accidentalmente una chiamata ricorsiva infinita, il programma non si bloccherà. Si riceverà, al contrario, un messaggio di errore.
- Supporta operatori sia della famiglia dei linguaggi di programmazione come C che della famiglia del BASIC. Si possono utilizzare qualunque di questi operatori indipendentemente dal linguaggio di programmazione che si vuole imparare.
- Sono supportate funzioni e procedure create dall'utente che possono o meno restituire un valore.
- Oltre 20 funzioni predefinite.

E ancora...

- Sono supportati diversi stili di rappresentazione come il tipo classico, IBM, SDL, e altri...
- I diagrammi di flusso possono essere stampati sia a colori che in bianco e nero.
- I diagrammi di flusso possono essere salvati in un file PNG sia a colori che in bianco e nero.
- L'editor possiede una caratteristica chiamata "Layout Windows". È progettato per aiutare a visualizzare il contenuto senza alcun intervento manuale sulla finestra.
- I programmi sono salvati in XML.

Il mio primo diagramma di flusso

Questo breve tutorial, serve creare un semplice programma chiamato "Ciao Mondo". Si tratta di un programma tradizionale per neofiti che visualizza "Ciao, mondo!" sullo schermo.

Quando si avvia un nuovo diagramma di flusso, vedrete due rettangoli arrotondati chiamati simboli "terminali". Questi simboli rappresentano l'inizio e la fine del programma.

Molti diagrammi iniziano il flusso visualizzando il testo "Start". Flowgorithm, invece, usa il testo "Main". La maggior parte dei linguaggi di programmazione iniziano con "Main" e Flowgorithm ... altrettanto.

Tutto, in un diagramma di flusso, è rappresentato da una figura geometrica. Potrai aggiungere le figure tra la figura principale e quella finale.

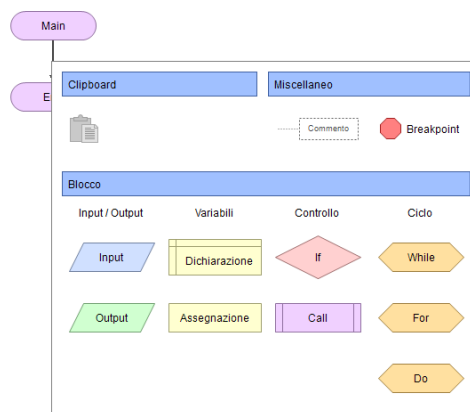
Per aggiungere una figura, spostare il puntatore del mouse su una linea. Se è possibile aggiungere una figura, la linea diventa di colore arancione.

Fare doppio clic o clicca col pulsante destro del mouse per aggiungere una figura.

Un menu pop-up apparirà mostrando tutte le figure che possono essere inserite.

Nei diagrammi di flusso, ogni azione che il computer può eseguire è rappresentata da una figura diversa. Ad esempio, le figure di ingresso e di uscita sono rappresentate con parallelogrammi.

Si può utilizzare anche il "copia e incolla". Nell'immagine qui sotto è disegnata in grigio perché non si è ancora copiato niente in memoria.

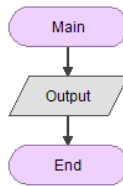


Il programma "Ciao Mondo" necessita di una figura per l'output. Ciò permetterà di visualizzare l'informazione nello schermo del computer.

Spostare il mouse sopra il parallelogramma verde con la scritta interna "Output": ora verrà evidenziata con un rettangolo blu.

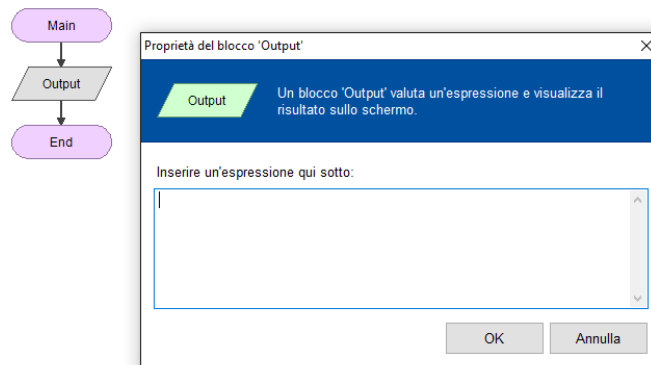
Cliccare sulla figura col pulsante sinistro del mouse e la figura verrà inserita nel diagramma.

Appena inserita, la figura apparirà colorata di grigio. In Flowgorithm, il colore grigio indica che la figura è "incompleta". Ciò significa che la figura ha necessita di altra informazione per poter funzionare.



È necessario indicare a Flowgorithm cosa volete che visualizzi. Ciò può essere qualsiasi cosa: da un risultato di un calcolo ad un messaggio.

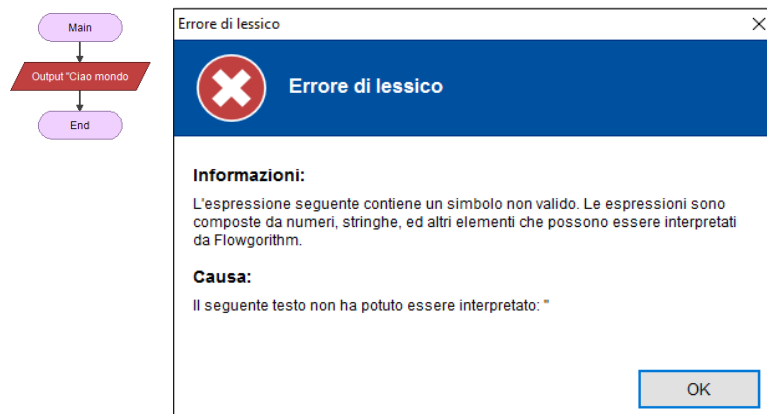
Eseguendo un doppio click sulla figura si entra nella modalità modifica.



Apparirà una finestra intitolata "Output Properties". Con questa finestra sarà possibile indicare cosa far apparire nello schermo del computer.


Nella finestra, scrivere "Ciao mondo!". Attenzione: è necessario aggiungere anche gli apici. Nei linguaggi di programmazione ciò viene chiamata String (stringa).

Cliccare sul pulsante OK per continuare. Se viene commesso un errore, Flowgorithm ci avvisa subito. Doppio click sulla figura per correggere l'errore.



La colorazione in verde della figura di output indicherà la corretta creazione.

Poiché tutte le figure sono colorate nella maniera corretta, ora è possibile eseguire quanto creato.

Sulla barra principale clicca sul pulsante  per eseguire il programma.

Apparirà una nuova finestra chiamata "Console Screen" usata per visualizzare le informazioni e per inserire i dati.

Hai realizzato il tuo primo programma. Congratulazioni!

Tipi di dato

Integer (Intero)

Il tipo di dati Integer è uno dei tipi più comunemente utilizzati nella programmazione. Una variabile di tipo integer può memorizzare un numero intero positivo o negativo, ma non può memorizzare valori frazionari. Così, può memorizzare valori quali 5, 42, 1947, ma non può memorizzare numeri come 3.2, 4.5, ecc ...

La finestra di controllo variabili visualizza i numeri interi in blu.

Integer
1947

Real (Reale)

Il tipo di dati Real può memorizzare un numero qualsiasi, sia i numeri interi che quelli frazionari. In molte lingue, questo è chiamato un Double dopo lo standard implementazione noto come "virgola mobile a doppia precisione".

La finestra di controllo variabili visualizza i reali in viola.

Real
1.618

String (Stringa)

Il tipo di dati String viene utilizzata per memorizzare tutti i dati testuali. Questo include parole, lettere, o qualsiasi altra cosa si potrebbe inviare con un messaggio di testo. Nella programmazione, il testo viene delimitato da doppi apici. Ad esempio: "CSU, Sacramento", "computer", e "Anno 1947" sono tutte delle stringhe.

La finestra di controllo variabile visualizza le stringhe in rosso.

String
Sacramento State

Boolean (Booleano o Logico)

Il tipo di dati booleano è in grado di memorizzare sia "vero" o "falso". Questi sono la base del processo decisionale in un programma per computer.

La finestra di controllo variabile mostra booleani in grigio verde.

Boolean
True

Identificatori

Ogni volta che si definisce una funzione o una variabile, viene dato un nome univoco chiamato "identificatore". Per evitare che gli identificatori possano essere confusi con altri elementi in una espressione, esse devono seguire una convenzione. Ogni linguaggio di programmazione ne possiede una e possono differire da un linguaggio all'altro.

In Flowgorithm, gli identificatori devono seguire le seguenti regole:

- Devono iniziare con una lettera.
- Dopo la prima lettera, l'identificatore può contenere lettere o numeri.
- Non sono permessi gli spazi.
- Non possono essere parole riservate o parole già definite in Flowgorithm.

Altre note:

- I linguaggi di programmazione come il Visual Basic e il C permettono anche l'uso del carattere "_" (underscore). Flowgorithm, al contrario, non lo accetta.
- Gli identificatori non sono case-sensitive, per cui "CASA" è uguale a "casa".

Parole chiave

Molte parole usate nei linguaggi di programmazione si sovrappongono alla convenzione di denominazione utilizzata per gli identificatori. In questi casi, la parola è "riservata" e non può essere utilizzata per gli identificatori. Inoltre, in molti linguaggi di programmazione funzioni e altre costanti sono predefinite. Anche queste non possono essere utilizzate.

Operatori

Le espressioni utilizzano operatori usati nei due principali linguaggi di programmazione: la famiglia dei linguaggi di programmazione afferenti al "BASIC" e la famiglia dei linguaggi di programmazione afferenti al "C" (che include C, Java, C#,...).

Poiché sono supportate entrambe le famiglie, siamo in presenza di un numero ridondante di operatori. Questi sono:

Tabella Operatori

Operatore	C	Basic
Negazione	!	not
Modulo	%	mod
Uguaglianza	==	=
Disuguaglianza	!=	<>
E (logico)	&&	and
O (logico)		or

Flowgorithm fornisce anche la possibilità d'uso di alcuni operatori tipici del Visual Basic poiché sono utili, semanticamente ben definiti:

Operatore Visual Basic	Descrizione
&	Concatenazione tra stringhe
^	potenza

In Java e in C#, l'operatore "+" è utilizzato sia per la concatenazione di stringhe sia come operatore nelle addizioni. Ciò potrebbe trarre in inganno. In Flowgorithm, l'operatore "+" è utilizzato come operatore aritmetico. Per la concatenazione viene usato il simbolo "&" (ampersand) e per l'operazione di esponenziazione, il simbolo "^"

Precedenze

La tabella seguente evidenzia il livello di precedenza degli operatori aritmetici e logici, dal più alto (valutato per prima) al più basso (valutato per ultimo).

Livello	Nome	Operazione	Note
8	Operatore unario	! not	In Visual Basic 'not' precede tutti gli operatori relazionali
7	Esponente (potenza)	^	Il simbolo non è utilizzato in C# o Java
6	Operatori aritmetici	* / % mod	La divisione ritorna sempre un numero reale
5	Operatori aritmetici	+ -	Il simbolo '+' è utilizzato solo con i numeri
4	Concatenazione	&	C# e Java usano in maniera ambigua l'operatore '+'
3	Operatori relazionali	> >= < <= == = != <>	
2	E (logico)	&& and	
1	O (logico)	or	

Esempi

Espressione	Soluzione	Note
1 + 3 ^ 2	10	
10 * 2 + 5 * 6	50	E' come scrivere (10 * 2) + (5 * 6)
7 * (4 - 1)	21	Le parentesi modificano le precedenze
6 / 3 + 2	4	E' come scrivere (6 / 3) + 2
10 MOD 3	1	MOD restituisce il resto della divisione tra interi
10 % 3	1	Come la precedente ma per la famiglia C

Funzioni predefinite

Matematiche

Funzione	Descrizione	Dalla versione..
Abs(n)	Valore assoluto	
Arcsin(n)	Arcoseno di n (radianti)	1.7
Arccos(n)	Arcocoseno di n (radianti)	1.7
Arctan(n)	Arcotangente di n (radianti)	
Cos(n)	Coseno di n (radianti)	
Sin(n)	Seno di n (radianti)	
Tan(n)	Tangente di n (radianti)	
Int(n)	Parte intera di un numero reale	
Log(n)	Logaritmo naturale	
Log10(n)	Logaritmo in base 10	
Sgn(n)	Segno matematico (-1 se n è negativo, 0 se zero, 1 se positivo)	
Sqrt(n)	Radice quadrata di n	

Stringa

Funzione	Descrizione
Len(s)	Restituisce la lunghezza (intero) di una stringa es: Len("casa") → 4
Char(s, i)	Restituisce il carattere alla posizione i della stringa s. La posizione parte da 0. es: Char("casa", 1) → "a"

Conversione tipo di dati

Funzione	Descrizione	Dalla versione..
ToChar(n)	Converte un codice carattere n in un carattere es: ToChar(65) → "A"	1.5
ToCode(c)	Converte un carattere c in un codice carattere (intero) es: ToCode("A") → 65	1.8
ToFixed(r, i)	Converte un numero reale r in una stringa con i cifre decimali es: ToFixed(123.456, 1) → "123.4"	1.8
ToInteger(s)	Converte una stringa con caratteri numerici in un intero es: ToInteger("123") → 123	
ToReal(s)	Converte una stringa con caratteri numerici in un reale es: ToReal("123.5") → 123.5	
ToString(n)	Converte un numero in stringa es: ToString(123.5) → "123.5"	

Altre

Funzione	Descrizione	Dalla versione..
Random(n)	Restituisce un numero intero tra 0 e n - 1	
Size(a)	Restituisce la grandezza (numero degli elementi) di un vettore (array)	1.7

Costanti predefinite

La semantica di Flowgorithm prevede tre costanti predefinite. “True” e “False” spesso utilizzate per inizializzare le variabili booleane e “Pi”, costante normalmente utilizzata in matematica.

Tabella Operatori

Costanti	Note
True	Valore booleano (logico) per Vero
False	Valore booleano (logico) per Falso
Pi	Costante matematica (P greco π = 3.1415)

Formato File

Molte applicazioni salvano i dati in un file binario (di difficile lettura). Ciò ha senso se i dati sono molto complessi e se si vuole minimizzare la grandezza del file attraverso l'uso della compressione.

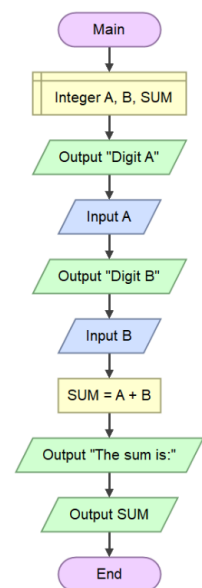
I programmi scritti con Flowgorithm sono molto semplici. Non esistono dati di grande complessità da dover essere compressi. I dati vengono salvati in un file di testo proprio come se fossero scritti in Java, C#, ecc... In questo modo, piuttosto che avere file binari complessi, questi ultimi sono di facile lettura e portabili verso altri sistemi. L'idea è che usando questa applicazione questi file possano essere letti da altri strumenti (tipo editor). Perciò i files vengono salvati in formato XML (Extended Markup Language).

Esempio

Prendiamo come esempio il seguente flowchart corrispondente all'esercizio "Somma tra due numeri":

Quando l'utente salva il suo esercizio con un determinato nome, Flowgorithm genera un file. Il file generato da Flowgorithm (file sorgente) ha estensione .FPRG, ma non è un file proprietario cioè non è un file che può essere letto solo dal software di appartenenza.

Proviamo ad aprire il file .FPRG corrispondente all'esercizio di cui sopra utilizzando non Flowgorithm ma un editor di testo come ad esempio WordPad. Ecco cosa potremo vedere.



```
<?xml version="1.0"?>
<flowgorithm fileversion="2.11">
  <attributes>
    <attribute name="name" value=""/>
    <attribute name="authors" value="Roberto"/>
    <attribute name="about" value=""/>
    <attribute name="saved" value="2019-02-07 08:45:07 "/>
    <attribute name="created"
value="Um9iZXJ0bztERVNLVE9QLVJPQkVSVE87MjAxOS0wMi0wNzsiMDg6NDI
6NDUgIjszMDU2"/>
    <attribute name="edited"
value="Um9iZXJ0bztERVNLVE9QLVJPQkVSVE87MjAxOS0wMi0wNzsiMDg6NDU
6MDcgIjsxOzMxNjU="/>
  </attributes>
  <function name="Main" type="None" variable="">
    <parameters/>
    <body>
      <declare name="A, B, SUM" type="Integer"
array="False" size=""/>
      <output expression="&quot;Digit A&quot;;"
newline="True"/>
      <input variable="A"/>
      <output expression="&quot;Digit B&quot;;"
newline="True"/>
      <input variable="B"/>
      <assign variable="SUM" expression="A + B"/>
      <output expression="&quot;The sum is:&quot;;"
newline="True"/>
      <output expression="SUM" newline="True"/>
    </body>
  </function>
</flowgorithm>
```


Forse a molti questo codice risulterà incomprensibile ma, anche per costoro, potrebbe essere utile sapere alcune informazioni.

Per prima cosa si nota che l'autore, D. Cook, ha utilizzato il linguaggio XML (**eXtended Markup Language**) che è una estensione del linguaggio HTML (**Hyper Text Markup Language**), il linguaggio base che viene utilizzato per la generazione delle pagine web.

Perché l'autore ha utilizzato l'XML? Una prima risposta potrebbe essere perché è intelligibile cioè, nonostante sia un codice sconosciuto ai più, si riesce a interpretare anche se vagamente. E poi, con l'XML è possibile creare nuovi tags seconda dell'intendimento dell'autore.

Vediamolo un po' più nel dettaglio.

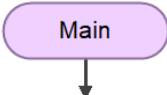
La parte iniziale:

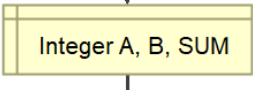
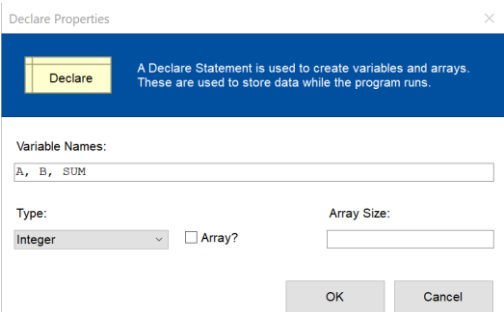
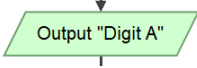
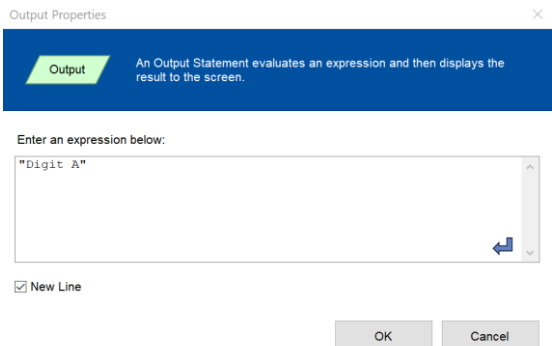
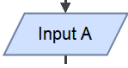
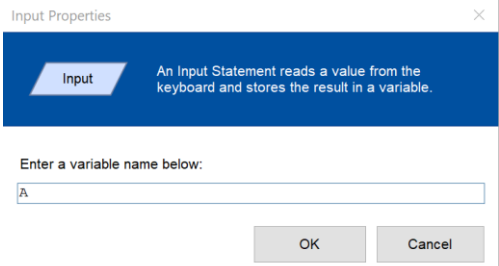
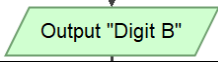
```
<?xml version="1.0"?>
<flowgorithm fileversion="2.11">
  <attributes>
    <attribute name="name" value=""/>
    <attribute name="authors" value="Roberto"/>
    <attribute name="about" value=""/>
    <attribute name="saved" value="2019-02-07 08:45:07 "/>
    <attribute name="created"
value="Um9iZXJ0bztERVNLVE9QLVJPQkVSVE87MjAxOS0wMi0wNzsiMDg6NDI6NDUgIjszMDU2"/>
    <attribute name="edited"
value="Um9iZXJ0bztERVNLVE9QLVJPQkVSVE87MjAxOS0wMi0wNzsiMDg6NDU6MDcgIjsxOzMxNjU="/>
  </attributes>
```

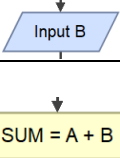
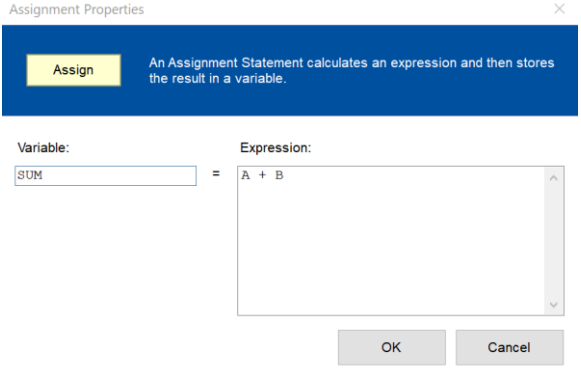
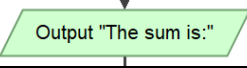
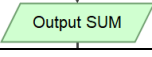
Vediamo più nel dettaglio.

<?xml version="1.0"?>	Indica la versione del linguaggio XML utilizzata
<flowgorithm fileversion="2.11">	Indica la versione di Flowgorithm utilizzata per la creazione del file in oggetto
<attributes> <attribute name="name" value=""/> <attribute name="authors" value="Roberto"/> <attribute name="about" value=""/> <attribute name="saved" value="2019-02-07 08:45:07 "/> <attribute name="created" value="Um9iZXJ0bztERVNLVE9QLVJPQkVSVE87MjAxOS0wMi0wNzsiMDg6NDI6NDUgIjszMDU2"/> <attribute name="edited" value="Um9iZXJ0bztERVNLVE9QLVJPQkVSVE87MjAxOS0wMi0wNzsiMDg6NDU6MDcgIjsxOzMxNjU="/> </attributes>	Questa parte è utilizzata per memorizzare alcune informazioni da associare al file sorgente (.FPRG) come Autore, Data di creazione, e qualche altra informazione utile a Flowgorithm.

A partire fin da ora, le linee di codice avranno un riferimento più "familiare".

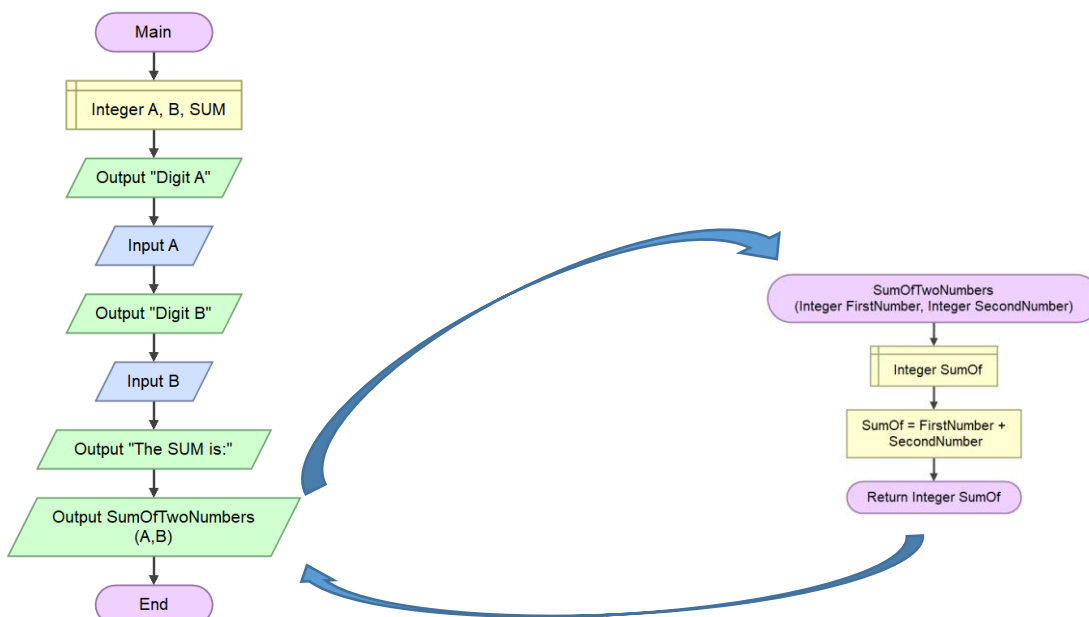
<function name="Main" type="None" variable="">	<p>È l'inizio di una funzione (function) particolare in quanto è la prima funzione e questa corrisponde al</p>  <p>Il tag "<function ..>" ha un suo tag di chiusura </function></p>
--	--

<code><parameters/></code>	La funzione principale "Main" non ha parametri. Questo tag assume forma diversa nel caso di funzioni con parametri.
<code><body></code>	Il corpo della funzione e <code></body></code> il fine tag
<code><declare name="A, B, SUM" type="Integer" array="False" size=""/></code>	<p>La dichiarazione delle variabili</p> <p>Nel flowchart:</p>  <p>In fase di creazione è facile individuare le singole voci di appartenenza all'interno del tag <code><declare ... ></code>:</p> 
<code><output expression="&quot;Digit A&quot;" newline="True"/></code>	<p>Nel flowchart:</p>  <p>In fase di creazione:</p> 
<code><input variable="A"/></code>	<p>Nel flowchart:</p>  <p>In fase di creazione:</p> 
<code><output expression="&quot;Digit B&quot;" newline="True"/></code>	<p>Nel flowchart:</p> 
<code><input variable="B"/></code>	<p>Nel flowchart:</p>

	 <p>Nel flowchart:</p> <p>In fase di creazione:</p> 
<code><assign variable="SUM" expression="A + B"/></code>	
<code><output expression="&quot;The sum is:&quot;" newline="True"/></code>	<p>Nel flowchart:</p> 
<code><output expression="SUM" newline="True"/></code>	<p>Nel flowchart:</p> 
<code></body></code>	Fine tag del tag <code><body></code>
<code></function></code>	Fine tag del tag <code><function></code>
<code></flowgorithm></code>	Fine tag del tag <code><flowgorithm></code> (termine esercizio)

Ovviamente possono essere presenti altri tag diversi da quelli visti fino ad ora e corrispondenti a particolari operazioni / funzioni di Flowgorithm.

Per capire bene qual è l'importanza (per l'utente generico) dell'uso dell'XML, proviamo a vedere il sorgente dell'esercizio "Somma tra due numeri" svolto utilizzando una funzione nuova creata dall'utente che svolge, appunto, la somma tra due numeri. L'esercizio si compone di due parti: una parte principale (MAIN) e una parte secondaria (FUNCTION) richiamata all'interno del MAIN.



Vediamo il file sorgente:

```
<?xml version="1.0"?>
<flowgorithm fileversion="2.11">
  <attributes>
    <attribute name="name" value=""/>
    <attribute name="authors" value="Roberto"/>
    <attribute name="about" value=""/>
    <attribute name="saved" value="2019-02-07 04:04:45 "/>
    <attribute name="created"
value="Um9iZXJ0bztERVNLVE9QLVJPQkVSVE87MjAxOS0wMi0wNzsiMDg6NDI6
NDUgIjsgMDU2"/>
    <attribute name="edited"
value="Um9iZXJ0bztERVNLVE9QLVJPQkVSVE87MjAxOS0wMi0wNzsiMDQ6MDQ6
NDUgIjsgOzMxNjA="/>
  </attributes>
  <function name="Main" type="None" variable="">
    <parameters/>
    <body>
      <declare name="A, B, SUM" type="Integer"
array="False" size=""/>
      <output expression="&quot;Digit A&quot;";
newline="True"/>
      <input variable="A"/>
      <output expression="&quot;Digit B&quot;";
newline="True"/>
      <input variable="B"/>
      <output expression="&quot;The SUM is:&quot;";
newline="True"/>
      <output expression="SumOfTwoNumbers (A,B) "
newline="True"/>
    </body>
  </function>
  <function name="SumOfTwoNumbers" type="Integer"
variable="SumOf">
    <parameters>
      <parameter name="FirstNumber" type="Integer"
array="False"/>
      <parameter name="SecondNumber" type="Integer"
array="False"/>
    </parameters>
    <body>
      <declare name="SumOf" type="Integer" array="False"
size=""/>
      <assign variable="SumOf" expression="FirstNumber +
SecondNumber"/>
    </body>
  </function>
</flowgorithm>
```

Oltre ad aver già spiegato prima parte della sua composizione, all'interno del file .FRPG (in XML) è presente una funzione in più rispetto a quella principale (MAIN): è la funzione "SumOfTwoNumbers" posizionata subito dopo la funzione MAIN.

```
<function name="SumOfTwoNumbers" type="Integer"
variable="SumOf">
  <parameters>
    <parameter name="FirstNumber" type="Integer"
array="False"/>
    <parameter name="SecondNumber" type="Integer"
array="False"/>
  </parameters>
  <body>
    <declare name="SumOf" type="Integer" array="False"
size=""/>
    <assign variable="SumOf" expression="FirstNumber +
SecondNumber"/>
  </body>
</function>
```

Si nota, ora, la presenza di parametri (non presenti nella funzione MAIN).

Questa funzione, così costruita, può essere copiata dal file XML e incollata all'interno di un altro file XML (altro esercizio) che può far uso di questa funzione subito prima del tag <flowgorithm>. Dopo aver salvato il tutto e riaperto il file col software Flowgorithm, la funzione verrà posizionata nella zona funzioni perfettamente funzionante e pronta all'uso all'interno del nostro flowchart.

In questa maniera è possibile costruire una libreria (un file di tipo testuale .TXT) di funzioni da copiare e incollare in quegli esercizi dove fosse necessario il loro utilizzo.

Istruzioni

Assegnazione

Figura associata

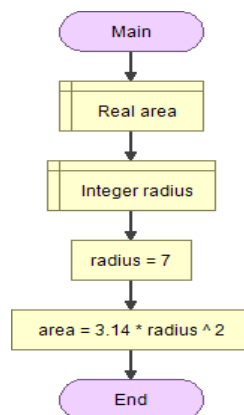
Assignment

Cosa fa

Un comando di assegnazione, calcola una espressione e registra il risultato in una variabile.

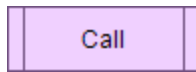
Esempio

Nel seguente esempio vengono dichiarate due variabili: “area” (che conterrà numeri reali) e “radius” (che conterrà numeri interi). Viene poi utilizzato il comando di assegnazione per impostare a 7 la variabile 'radius'. Per finire, nella variabile “area” verrà assegnato il calcolo dell’area di un cerchio.



Chiamata di Funzione o Procedura

Figura associata



Cosa fa

Un comando di chiamata trasferisce il controllo ad una funzione o ad una procedura. Per veicolare le informazioni in una funzione o in una procedura, si usano i parametri associati alla funzione o alla procedura e ciò può essere portato a termine utilizzando o la modalità del passaggio per valore o la modalità del passaggio per riferimento.

Commento

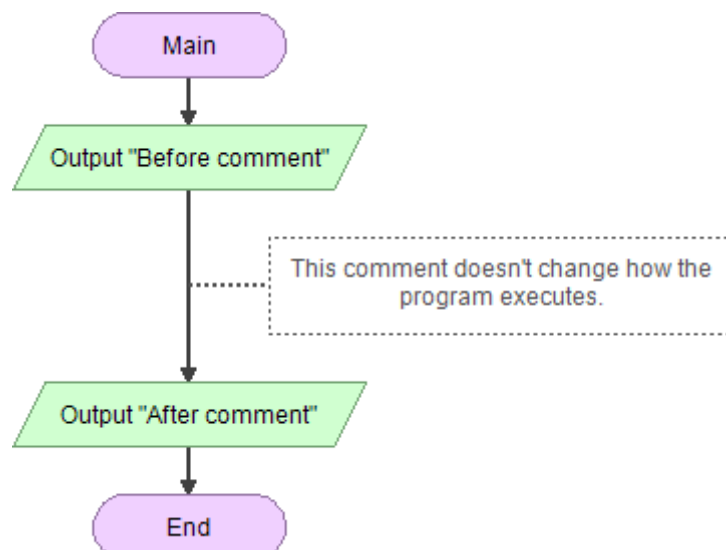
Figura associata



Cosa fa

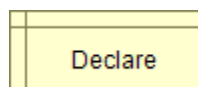
I commenti non interferiscono con l'esecuzione del programma. Sono utilizzati per documentare meglio le azioni intraprese e/o gli intendimenti dell'autore. Questi possono includere: la logica di un ciclo, regole, cambiamenti, ecc...

Esempio



Dichiarazione

Figura associata

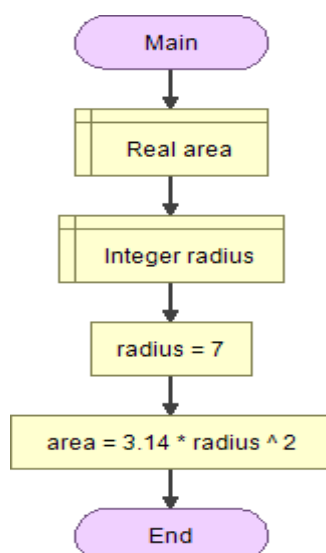


Cosa fa

Il comando di dichiarazione viene usato per creare variabili e vettori (arrays). Sono utilizzati per conservare i dati nel frattempo che il programma è in esecuzione.

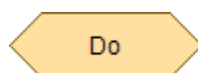
Esempio

Nel seguente esempio vengono dichiarate due variabili: “area” (che conterrà numeri reali) e “radius” (che conterrà numeri interi). Viene poi utilizzato il comando di assegnazione per impostare a 7 la variabile 'radius'. Per finire, nella variabile “area” verrà assegnato il calcolo dell’area di un cerchio.



Do

Figura associata

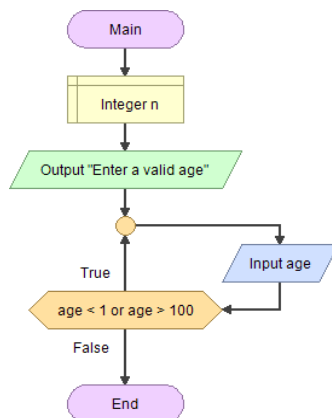


Cosa fa

Un ciclo Do è simile al ciclo While tranne per il fatto che le istruzioni al suo interno vengono eseguite almeno una volta prima di poterne verificare la condizione di ciclo.

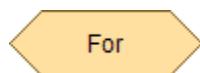
Esempio

Il seguente esempio mostra il ciclo DO che accetta solo un input valido, una età compresa tra 1 e 100. Se $age < 1$ o $age > 100$ ripete il ciclo.



For

Figura associata

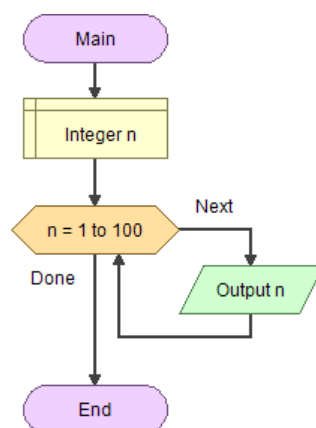


Cosa fa

Il ciclo enumerativo For incrementa una variabile all'interno di un intervallo di valori assegnati. Questa struttura, di solito, rimpiazza la struttura While.

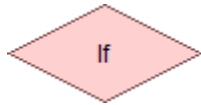
Esempio

Il seguente esempio visualizza a monitor i numeri da 1 a 100.



If

Figura associata



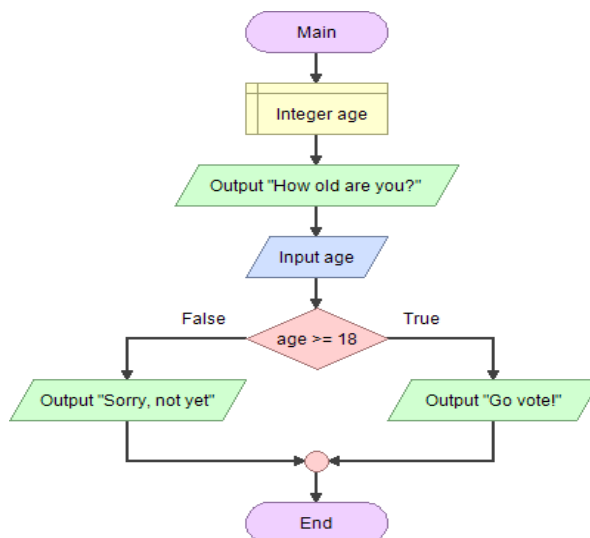
Cosa fa

Il comando If controlla una espressione booleana e passa al ramo del Vero o del Falso a seconda se è verificata o meno la condizione.

Esempio

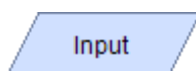
Il seguente esempio dichiara una variabile di tipo intero chiamata 'age'. Viene inserita, quindi, l'età da tastiera per controllare se l'utente può andare a votare o meno.

Alla fine, una struttura If controlla se l'età inserita da tastiera e assegnata alla variabile "age" è maggiore o uguale a 18. Se si è maggiorenni. Il programma visualizzerà la frase "Go vote!" altrimenti visualizzerà la frase "Sorry, not yet".



Input

Figura associata

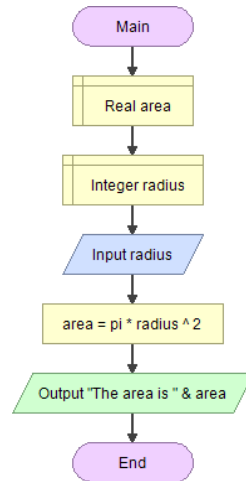


Cosa fa

Un Input legge un valore da tastiera e immagazzina tale valore in una variabile.

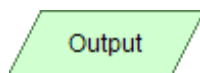
Esempio

Il seguente esempio crea due variabili: 'area' e 'radius'. Utilizza poi un Input per leggere il raggio da tastiera e assegnarlo alla variabile 'radius'. Alla fine verrà visualizzato un Output sullo schermo del risultato del calcolo dell'area.



Output

Figura associata

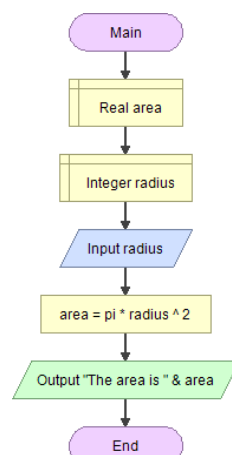


Cosa fa

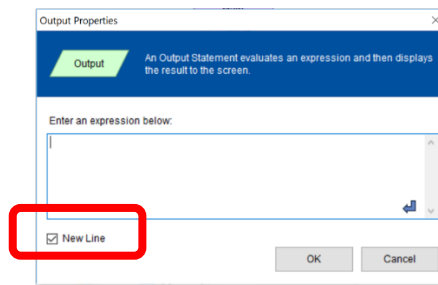
Un comando di Output valuta un'espressione e poi visualizza il risultato sullo schermo.

Esempio

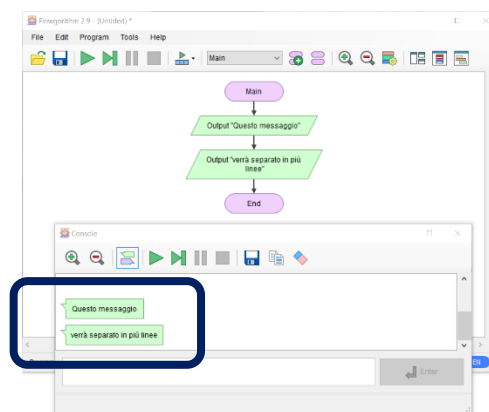
Il seguente esempio crea due variabili: 'area' e 'radius'. Utilizza poi un Input per leggere il raggio da tastiera e assegnarlo alla variabile 'radius'. Alla fine verrà visualizzato un Output del risultato del calcolo dell'area.



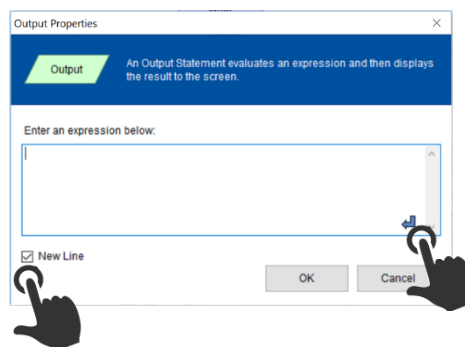
Dalla versione 2.9 è stata inserita, per l'output, la possibilità di utilizzare la funzione "Ritorno a capo" (in inglese: "New Line")



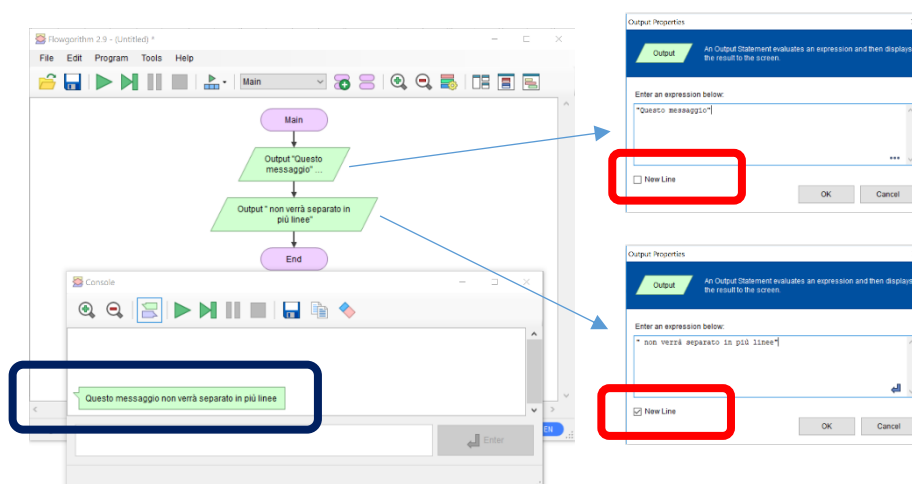
Questa opzione influisce sulla visualizzazione dei messaggi con o senza il ritorno a capo. Se utilizzata Flowgorithm creerà un messaggio diverso per ogni istruzione di Output utilizzata.



Se invece non si imposta questa opzione, cioè si toglie la spunta all'opzione, cliccando col mouse come da figura



si otterrà un unico messaggio con ritorno a capo nell'ultima parte del messaggio:



While

Figura associata

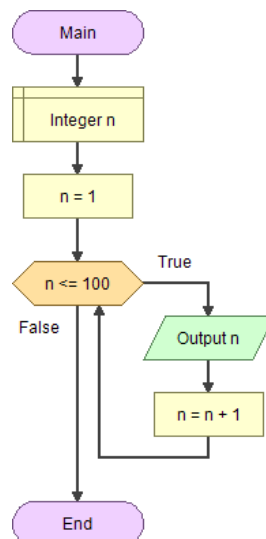


Cosa fa

Il ciclo While valuta una espressione booleana e, se vera, esegue le istruzioni contenute al suo interno. Cicla fino a quando la condizione diventa falsa.

Esempio

Il seguente esempio visualizza i numeri da 1 a 100. Il comando di assegnazione " $n = n + 1$ " incrementa la variabile 'n' ogni iterazione del ciclo.



Per gli Amministratori di Sistema

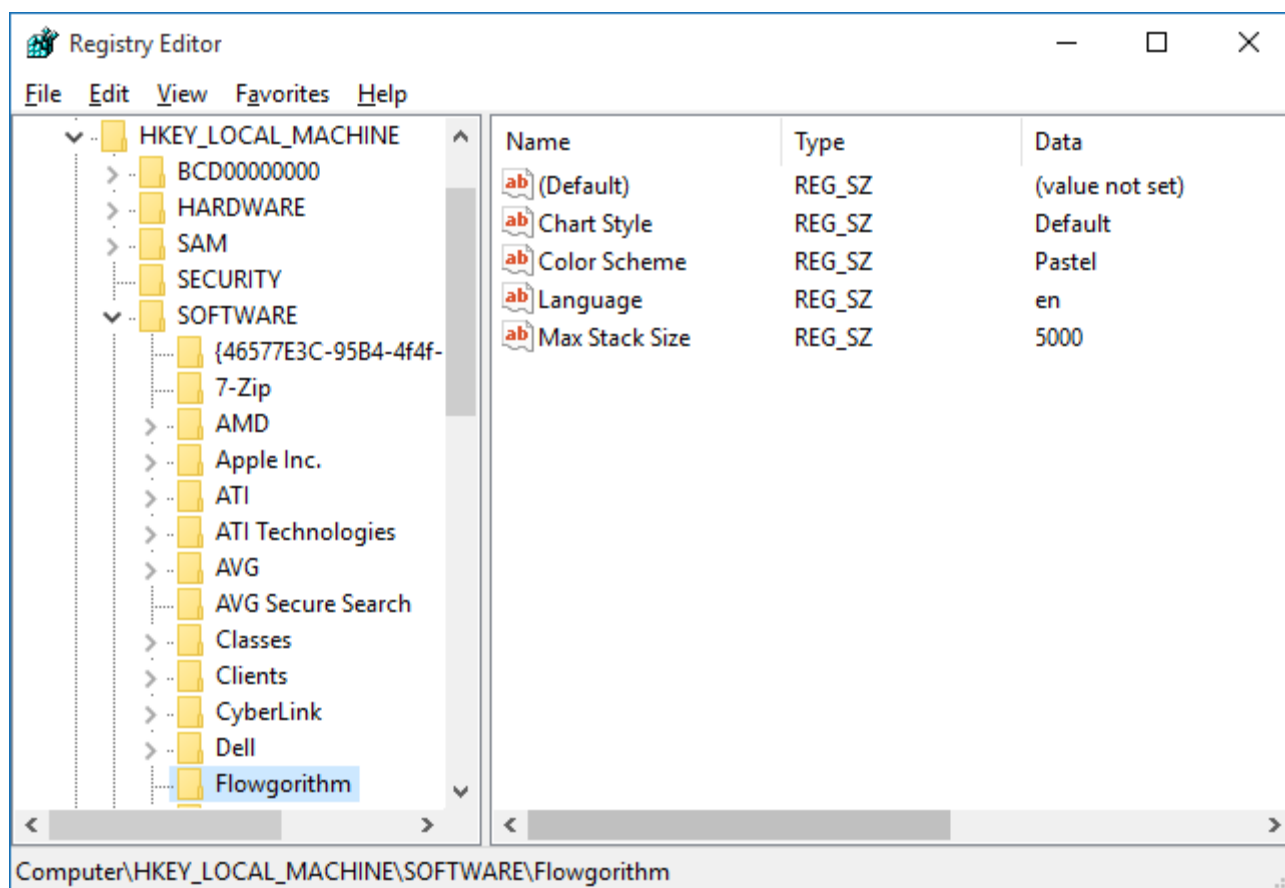


Come Flowgorithm utilizza il file registro di Windows

Flowgorithm usa il file registro di Windows per conservare le impostazioni utente. Sono conservate solo poche impostazioni. Queste includono linguaggio, colori, modello di diagramma e massima grandezza dello stack interno di Flowgorithm.

Quando si esegue Flowgorithm, verrà controllata prima la chiave CURRENT_USER per caricare i valori in essa contenuti. Se non la trova, verrà controllata la chiave LOCAL_MACHINE.

Se l'applicazione viene eseguita in modalità Amministratore, Flowgorithm salverà le sue impostazioni nella chiave di registro LOCAL_MACHINE. Altrimenti la salverà in CURRENT_USER. Se si vuole impostare i settaggi di default per tutti gli utenti del laboratorio, apri l'applicazione in modalità Amministratore e imposta linguaggio, colore, ecc... Ogni nuovo utente (che non possiede impostazioni utente), userà queste.



Chiavi Registro

Chiave	Descrizione
Chart Style	Contiene il nome del modello ("Default", "Classic", ecc...) Se non viene riconosciuto il valore della chiave, l'applicazione utilizzerà il modello "Default".

Color Scheme	Contiene il nome dello schema colori corrente ("Pastel", "Cyberspace", ecc...) Se non viene riconosciuto il valore della chiave, l'applicazione utilizzerà il schema colori "Pastel".
Language	Contiene il codice ISO del linguaggio visualizzato. Se non viene riconosciuto il valore della chiave, l'applicazione utilizzerà il codice lingua "en" (English).
Max Stack Size	Contiene la grandezza massima dello stack. Se l'applicazione eccede questa grandezza, ad esempio nel caso di una chiamata ricorsiva infinita, verrà visualizzata l'informazione "Stack Overflow Error". Questa impostazione non può essere modificata dalla applicazione.

Contatto

e-mail: **Flowgorithm@DevinCook.com**.

Informazioni

Origine del Nome

Il nome è la fusione dei termini "flowchart" e "algorithm".

Scritto in

Flowgorithm è implementato utilizzando Microsoft C#.

Autore

Devin Cook

Licenza



[Download in PDF](#)

Revision History

Version 2.20 (February 12, 2019)

- **Fortran 2003 programming language was added to the Source Code Viewer.**
- **MATLAB programming language was added to the Source Code Viewer.**
- The Italian translation was improved. *Special thanks to Riccardo Scorretti.*
- Many of the generated languages, in the Source Code Viewer, were improved:
 - VBA - The input and output helper functions were changed. Before, both functions used an argument that held the variable's name. This was changed to a more flexible "prompt" argument.
 - JavaScript - The input and output helper functions were replaced with direct calls to Window.alert and Window.prompt.
 - Perl - Whenever an array is passed into a function, the code now uses a "dereferencing" variable for easier to maintain (and readable) code.
 - Swift - The named arguments will now properly match the called functions parameters.
 - Powershell - The code can be now be displayed in aligned or hanging styles.
- A bug in the function manager was fixed. The system would sometimes report that a parameter was a duplicate name, when it was not.

Version 2.19 (January 15, 2019)

- *Added support for Ukrainian - special thanks to Vitalii Shpytiak.*
- Improved GUI support for right-to-left languages - such as Farsi (Persian) and Arabic. The some of the controls were not being mirrored in the windows - in, particular, the GroupBox, TabPage, and Panel. The system now manually mirrors these controls.
- The Change Translation Window was improved. Each translation is now represented with a button (albeit a flat one) rather than an item in the list. The icons now also scale with the application. The button on the bottom now contains 'Cancel' rather than 'OK'.
- The About Window was improved. Each of the contributors, who provided translations, is listed more prominently. The icons now also scale with the application.
- Some defensive code was added to handle drawing errors.

Version 2.18.4 (November 30, 2018)

- In the Color Scheme Editor, the "all" dropper button can now be used with the preview image. When the image is clicked, the text, border, and fill will be set to shades of the clicked pixel.
- Improved the line numbering in the Source Code Viewer.
- Some minor changes were made to the internal scaling logic. The application will now work better on scaled desktops.
- Fixed an odd bug that occurred on the 64-bit version of the application. The bug caused the Chart Style dropdown to malfunction.

Version 2.18.3 (November 20, 2018)

- Updated the Paint (Dark) color scheme.
- Improved the indentation in the Source Code Viewer.

- Increased the size of the color scheme/chart style preview image. A preview of the Declare Shape was also added (which is what justified the increase in size).
- Fixed several built-in program templates. In particular, C# and Visual Basic .NET. *Special thanks to Dr. Petr Vanecek for reporting this.*
- Fixed a very minor bug when changing color schemes. If a custom scheme is created, the Console Window wouldn't update its colors. Reopening the window fixed it.

Version 2.18.2 (November 11, 2018)

- Error messages, caused by a function, were improved. The system now properly identifies the cause.
- Spacing in the Variable Watch Window was improved.
- Fixed a cosmetic bug in the Source Code Viewer. The icon for "Aligned" was not being set correctly.
- Various minor cosmetic tweaks to the Source Code Viewer.

Version 2.18.1 (November 7, 2018)

- The Chart Style Editor and Color Scheme Editors were improved. Beforehand, any time a color scheme or chart style was **simply** opened, the system would treat it being a custom design. Now, the system will only treat as customized if there were, in fact, changes.
- Made some minor revisions to the built-in color themes.
- Made some a minor change to the Korean Translation. In particular, the translation of "Undo" and "Redo" was updated.
- Fixed a few mistakes in the Gaddis and SDL chart styles.

Version 2.18.0 (September 4, 2018)

- **Customizable Chart Styles were added.** You can now use the Chart Style Editor to create change the shapes used in the flowchart. This will allow you to create any style (within some limits) of flowchart you need. Currently, 20 shapes are supported, but additional ones may be added in the future.
- **The application is now "DPI-aware".** This means that the window will "scale" on larger displays if Windows using a scaling factor. Often this is 125% or 150%. The user can also change the current scaling factor from the Tools Menu > Scale. Before this change, Windows would simply strength the application on larger displays - making it become a tad blurry.
- The layout of the Color Scheme Editor was revamped. The current scheme is displayed on the left which includes the preview and the meta fields (name, author, about). The right-hand side includes all the editing options. This layout matches the layout of the Chart Style Editor.
- Browse buttons were added to the Export Bitmap, Export Vector, Print, and Choose Chart Style windows. This buttons allow you to open a customized Chart Style File.
- The Variable Watch Window now follows the current color theme. The current page's border, text, and background color are used. The color of the variable headers are hard coded to their current color's.
- Language dropdown in the Source Code Viewer is a tad bigger.
- Several of the built-in color schemes were updated slightly. For example, in the Pastel color scheme, the "executing" color was change to be a tad more blue (it is still green - just a better hue).
- Bug fixes:
 - The Source Code Viewer was generating incorrect Python code for string concatenation (in some cases). The flags, used by the template, were not being set properly. *Special thanks to Dave Braunschweig for reporting this.*

- The application would crash if an function (with a return type) was declared and then used before saving. *Special thanks to Robert Platt & Dave Braunschweig for reporting this.*
- If a runtime error occurred, the shape was not being highlighted in red. *Special thanks to Jose Mogrovejo Cardenas for reporting this.*
- The Source Code Viewer had a bug in generated Gaddis Pseudocode. *Special thanks to Jaime Fernando Meza Meza for reporting this.*
- The Source Code Viewer had numerous mistakes in C# generated code. *Special thanks to Paul Shaddick for reporting this.*

Version 2.17.1 (August 4, 2018)

- Bug fix: The Source Code Viewer was incorrectly using the syntax for addition for the modulus operator. *Special thanks to T. Cruz for reporting this.*
- Added the Layout Windows button to the Console Window.
- The Console Window buttons were also rearranged slightly.

Version 2.17 (July 31, 2018)

- Bug fix: The Do Shape was not being properly exported with either bitmap file or vector file. The shape wasn't being aligned properly in some cases.
- The Export Vector tool was improved slightly. The file type (EMF or SVG) can be selected from the dialog window rather than on the second tab. The second tab was also eliminated as it was no longer needed.
- The Change Language menu entry can be activated by pressing F12 rather than the convoluted Control+Alt+T. It was also moved to the top of the menu.
- The tooltip for the current translation, in the bottom right of the main window, was changed. It now uses the same text as the Change Language menu item.

Version 2.16 (July 24, 2018) - Major Release

- Flowchart improvements:
 - *If Shapes are now more compact.* The true and false branches are now aligned with the center of If Shape (with some padding). Before, both branches were aligned to the side of the If Shape. This new version will take up less screen space.
 - *While and For Shapes, like If, are now more compact.* The loop branches now align to the right of the center of the header shape.
 - The width of text in If Shapes was increased. Beforehand, the text wrapped at a much smaller width. This became problematic for complex conditional expressions.
 - Function headers will now wrap text. Beforehand, it displayed all parameters on one line.
- Variable Watch Window:
 - *Variable values will now be retain when the program finishes.* Beforehand, any time the program finished, the variable window was cleared.
 - By default size of the Variable Watch Window was changed. When opened, it will take 20% of the screen's width and 100% of the screen's height. It will also be right-aligned on the screen.
- Source Code Viewer:
 - Added a dropdown icon next to the current language name.
 - The icon used for Visual Basic .NET, used in the Source Code Viewer, was changed.
- New installer!

- When a new function is added, and it returns a value, the system automatically adds the return variables declaration.
- The Layout Windows utility was modified slightly. The "Variables" layout will set the Variable Window at 20% of the screen's width. The flowchart will be set to 80%.
- When stepping through the program, the system will now only highlight the function's footer (rather than both the header and footer). This change was designed to emphasize the "return". The Source Code Viewer still highlights both the start and end of the function.
- Added a button to the Console Window that opens the Variable Watch Window.
- The default size of windows was increased slightly.
- The Function Manager window was increased in size.
- When the application is first executed, it will check if the current language, used by Windows, is English. If not, it will prompt the user for a language selection. This prompt will also appear if the application is first loaded in Administrative Mode regardless of language.
- Bug fixes:
 - Program Templates were not setting the *type-type* flags properly. This causes the Source Code Viewer's generated programs to sometimes be incorrect. This affected C, Java, C#.
 - Layout Windows now works on a second screen. Beforehand, the windows were resizing on the main screen (and with incorrect sizes).
 - When a new program is created, the system was not clearing the Console Window.
 - Also, when a new program is created, the system wasn't stopping the current one from executing.
 - The dropdown language menu wasn't always aligning correctly on right-to-left translations (e.g. Arabic, Farsi).
 - In the Source Code Viewer, the line separating the Language and Line Style (if present) wasn't always showing.
 - If the current translation is changed, the Source Code Viewer was resetting to the Registry value rather than keeping the current settings.

Version 2.15.2 (June 15, 2018)

- Color Schemes can be dragged and dropped on the Choose Chart Style window. This window was also increased in size, slightly.
- The various translations were updated to report an "Invalid File" rather than an "Invalid Program File" when an invalid file is dragged and dropped. The original translation dates back to when Flowgorithm only used one file type: fpgt. Now, it also uses color schemes and program templates.
- Added some internal exception-handling code. The code handles situations when links fail to open correctly.

Version 2.15.1 (May 29, 2018)

- **Program Templates were added!** Templates are simple text files that the Source Code Viewer can use to generate code for any pseudocode and/or programming language. This means that Flowgorithm is not limited to the programming languages currently supported. Please see the [documentation](#) for more information.
- Fixed a bug when occurred of Flowgorithm was closed while running. In this case, it was possible for the runtime engine to get stuck in an infinite loop and the application hangs (without a GUI). *Special thanks to Daniel Ivan Escobar.*

- Fixed a bug in Visual Basic .NET generated code. The Source Code Viewer was ending a Module definition with an End Class rather than End Module. There were also some mistakes with generated functions. *Special thanks to Jon Bek.*
- Fixed a bug in Python Generated Code. The Source Code Viewer was displaying {1} rather than the correct operand for concatenation. This was a careless typo. *Special thanks to Dave Braunschweig.*

Version 2.14.1 (April 1, 2018)

- The Italian Translation was updated - *special thanks to Roberto Atzori.*
- The Gaddis Chart Style was updated. The application was displaying the text "Function" at the top of each flowchart - which was a mistake.
- Auto Pseudocode was updated. For If Statements, with an false branch, the pseudocode no longer contains the "True" label line.

Version 2.14 (March 30, 2018)

- *Added Undo and Redo.* The new feature will allow users to undo and redo changes to the flowchart.
- New pop-up menu was added to the main window. The new menu is displayed when the background of the flowchart is right-clicked. It contains "Choose Program Style...", "Program Attributes..." as well as Undo and Redo (if available).
- PowerShell generated code was improved. In particular, how text is used with Write-Host was syntactically simplified.
- The icon used to represent Gaddis Pseudocode in the Source Code Viewer was changed. The update was designed to reflect the release of the 5th edition of his popular book: *Starting Out With Programming Logic*. The fruit on the front of the book changed from an apple to a plum.
- Updated the logic that forces the Console Window to the top of other windows. Now, it will only be forced to the top if either #1 the window was first opened or #2 the user clicks "Run" at full speed.
- Added the Run, Step, Pause, and Stop buttons to the Source Code Viewer.

Version 2.13 (March 8, 2018)

- *Added support for Romanian - special thanks to Vlăduț Ilie.*
- Fixed a bug in Visual Basic .NET generated code. Call Statements weren't putting parenthesis around the arguments.

Version 2.12.2 (March 6, 2018)

- The Source Code Viewer now remembers if you selected line numbers. This setting is saved so that, the next time the application is opened, it will retain its value.
- The Function Manager now checks an function, on the list, after a single click.
- Bug Fixes:
 - Pascal generated code was improved. The generated code was in inserting a semicolon after every statement rather than just between them. In Pascal, the semicolon is a statement separator rather than a delimited. Some compilers allow it to be used as a delimiter, but not all. The generated code was also using StrToFloat rather than FloatToStr for concatenation. *Special thanks to Dr. Richard Lui for his help.*
 - In the Source Code Viewer, VBA (Visual Basic for Applications) was displaying the aligned/hanging option even though it doesn't apply.

- IBO Pseudocode and Gaddis Pseudocode were not displaying the "end". This was careless mistake.

Version 2.12.1 (February 28, 2018)

- Updated the title bar on the main window. Now, the system will display the name of the file, a dash, and then "Flowgorithm". This is consistent with how most applications use the title bar. The version, which used to be displayed as part of the title bar, now is displayed in the footer when the application first loads.
- Made some cosmetic changes to the About Window.
- Fixed a bug in the Edit Function Window. If the user tabs to the Return Type dropdown, the system wasn't enabling the Variable Name field. Special thanks to Nathan Todd.
- Fixed a bug in the VBA (Visual Basic for Applications) generated code. I had failed to add the "As" clause to functions. Thanks to Nic Lightfoot for letting me know.

Version 2.12 (February 21, 2018)

- *Added support for Malay - special thanks to Husnil Khatimi Hussin.*
- The Italian translation was updated - *thanks to Roberto Atzori.*
- When an error occurs, the window will now highlight the shape before displaying the error window.
- Source Code Viewer:
 - *Added "Auto Pseudocode".* This pseudocode is generated by using the current translation. This means that, for example, if Flowgorithm is in Spanish, it will produce Spanish pseudocode. This is the case for all other languages: Korean, Farsi, Arabic, Italian, etc...
 - *Added Smalltalk.* Created by Alan Kay, Smalltalk created the concept of Object Oriented Programming.
 - *Added Powershell.* This is scripting language, created by Microsoft, that combines concepts of Smalltalk, Perl, and .NET.
 - Incomplete shapes were improved. When a shape is incomplete, empty fields will now contain the field's name. It is read from the current translation. So, for example, the current translation is Spanish, then the incomplete fields will contain Spanish names.
 - The line numbers are now right aligned
 - VBA generated code was improved. *Special thanks to Nic Lightfoot.*
 - When source code is saved, the system will only use UTF-8 if a Unicode (non-ASCII) character is present. Otherwise, it will default to ASCII.
 - Greatly improved the generated code for: C#, Lua, Pascal, Perl, and Visual Basic .NET.
 - The last language used is now saved in the Registry.
- Bug fixes:
 - If a number is used with the Len() function, the system was incorrectly specifying it needed a "number" rather than "string". Thanks to Roberto Atzori for finding this.

Version 2.11.1 (January 25, 2018)

- The shape captions used in the Default (Descriptive) chart style are a bit more descriptive.
- Source Code Viewer:
 - Some of the language names were translated better. In particular, the word "pseudocode".
 - The mouse scroll wheel wasn't working on the Source Code Viewer.
 - QBasic code, in the Source Code Viewer, left off the '#' for generated functions - such as ToFixed.
 - Some of the languages, in the language dropdown menu, weren't being translated properly.

Version 2.11 (January 15, 2018)

- *The Slovenian translated was updated* - Thanks to Gregor Anželj
- Source Code Viewer:
 - *TypeScript was added.*
 - *Scala was added.*
 - The window GUI was revamped. Each programming language is now represented a unique graphical icon. This is used in the main toolbar as well as a new dropdown menu. The "Aligned/Hanging Brace" dropdown was also replaced with a graphical version. It will only be displayed when it is relevant (rather than just being disabled).
 - JavaScript code was improved. The generated code for Random last lacking the "Math." prefix and always used parenthesis.
 - The scrollbar behavior was improved. It is now disabled if not needed.
- The Change Chart Style toolbar icon was updated (yet again).
- The Clipboard popup menu now aligns to the left on right-to-left languages such as Arabic and Farsi.
- The Comment Shape's maximum width size (before word wrapping) was increased slightly.
- The "Pastel" color scheme was updated slightly - in particular, the shade of gray used for comments.
- Program Attributes window
 - It now shows the date that the file was last saved. This is not an editable, but useful, field.
 - Pressing F4 from the main window, will now open the Program Attributes window.
- Bug fixes:
 - Fixed a minor bug in the Color Scheme Editor. When an image file was opened, it remained locked (can't be moved, deleted, etc...) until the editor was closed.
 - Opening the Function Manager no longer marks the program as "changed".

Version 2.10 (November 22, 2017)

- The Function Manager was improved. The new version allows entire functions to be copied, cut, and pasted.
- Color Scheme Editor can now be opened with Control-E
- Fixed a bug in the Color Scheme Editor. The effects the "dropper" tool. If a transparent area of a shape is clicked, the color would be incorrect.
- Fixed some minor tab-order issues.

Version 2.9.2 (November 9, 2017)

- The Save-As feature, in both the Main Window and the Color Scheme Editor, works far better now. It will show the current name as a default.
- Updated the Retro Computer color scheme.
- Restored textboxes, that are used for expressions, to left-to-right for all cultures.

Version 2.9 (November 4, 2017)

- *Added support for Korean - special thanks to Inkee Jeong.*
- New error message. When a function is used in an expression, it must have a return value. The new error message properly explains, to the user, what happened. Beforehand, the error was being handled by Type Mismatch. This wasn't particularly helpful.
- Output Shapes were improved. Output can now include or exclude the newline character. This will give better control over output - and allow the "ASCII art" commonly used in beginning programming.

- Color schemes were improved. The colors of the Comment Shape can now be customized. Beforehand, it used the same colors as the page.
- A few built-in color schemes were modified.
- Bug fix: When an error occurs, the system will show the shape that cause the error. Unfortunately, the system failed to do this if the error occurred in a different function (than the one being displayed)
- Updated the About Window. The new version of the About Window was designed to maximum screen real estate. The old version was becoming far too large.
- Minor updates to the Error Dialog and Message Dialog windows.
- Minor GUI tweaks.

Version 2.8 (October 7, 2017)

- *Added support for Dutch - special thanks to Ralf Djojomoenawi & Sander Dol.*
- If a function is incomplete (missing its return variable), it will now display in gray.
- The comment shape was updated. The left-side connecting line will always use the color of the rest of the lines. The idea is to prevent it from "vanishing" if the current scheme's Selection border color is the same as the background color. In addition, the preview image no longer displays the left-side connecting line.
- Fixed a bug that occurred when a function, that doesn't return a value, is used in an expression. While the error message produced now isn't the best, it is far better than before.

Version 2.7.1 (September 16, 2017)

- Fixed a bug in Python generated code. Empty blocks were not including 'pass' statements. Special thanks to Clive Long for reporting the bug.
- The internal logic used to draw shapes was updated.

Version 2.7 (September 6, 2017)

- *Added support for Indonesian - special thanks to Abdul Kadir*
- The Polish Translation was updated - *special thanks to Robert Gajewski.*
- Updated the logic used to show the Add Shape and Clipboard pop-up windows. Hopefully, this change will work better for those using WINE to run Flowgorithm on the Macintosh.
- Fixed a bug where lines wouldn't "line up" on Comment Shapes.
- The Color Scheme Editor now has a "copy" dropper that will copy all the attributes (text, fill, and border) from another clicked shape.
- The colors used by Comment Shapes was redone. Now, both the text and border will use the page's text (label) color.

Version 2.6.1 (August 10, 2017)

- The Console Window now responds to pressing F5, F6, F7, and F8. These are used to Run, Step, Pause, and Stop respectively.
- The Italian Translation was updated - *special thanks to Roberto Atzori*
- In the Color Scheme Editor, the image scrollbars work far better than before.
- The graphic image, used the various shape edit windows, is better horizontally centered between the window's edge and the text.

Version 2.6 (August 7, 2017)

- *Added support for Farsi (Persian) - thanks to Vahid Baghi*
- *The Declare Shapes can now support multiple variables.* The user can list different variable names separated by commas.
- File changes:
 - To accomodate older versions of Flowgorithm, the application can now save programs in 1.0 and 2.0 format. The current format is 2.6. *The current version can still open any previous file version.*
 - One of the *redundant* legacy fields, "variables" in the Declare tag, was removed. This was designed for forward compatibility with older versions of Flowgorithm (pre 1.7). Since Flowgorithm can now save to older formats, it is unnecessary.
- The word-wrapping, used on shapes, was improved. Beforehand, the text could, sometimes, fail to be vertically centered. The new logic both resolves this problem and wraps based on syntactic elements (such as operators).
- Printing was improved:
 - The window now opens the Print Common Dialog window. This allows the user to find printers, set advance features, and much more.
 - An icon for portrait & landscape were added.
 - Printer settings are now saved to the registry. This includes the printer's name, landscape, and margins.
- Color Scheme Editor:
 - Added a Save-As button.
 - When a new color scheme is created, the hover color of the 'line' is now set to orange. Before, it was white.
- The system can catch, and report, Overflow Errors far better than it did before.
- The Console Window now has a limit on the number of items in the buffer. This prevents the system from running out of resources if (as an example) a loop prints from 1 to a million. No errors are caused. Early items are simply removed.
- The Turkish translation was updated.
- The Add Shape pop-up menu and Clipboard pop-up menu work better than before. They should appear quicker and properly return focus to the main window when closed.
- When the flowchat background is double-clicked, the Program Attributes window will open. Before, the user would *left* double-click. It has been changed to *right* double-click to avoid it being opened by accident.
- The message window was updated. In particular, the padding between graphical elements was increased.
- Bug fixes:
 - The thumbwheel on the mouse was moving the scrollbar, but not updating the window.
 - On the Gaddis Chart Style, the assignment caption had extra spaces.
 - The Console Window no longer aggressively sets its Z-Order. Before, anytime the window received output, it would force itself on top of all the other windows. This made it difficult to stop the program from the Main Window.

Version 2.5.4 (June 16, 2017)

- Fixed a bug in the Variable Watch Window. It was not translating the label "Main" into the current language.
- Double-clicking on the main window's background will now open the Program Properties window. This was an old feature that was removed before the shape selection object was updated. However, it wasn't re-enabled until now.

Version 2.5.3 (June 15, 2017)

- Updated an even better translation for Italian - *Thanks to Roberto Atzori*
- Improved the Color Scheme Editor. The internal logic, used to determine if a color scheme was changed, was improved.
- Fixed a bug in the Color Scheme Editor. The Open Image dialog was not being translated.
- The "If" Shape appearance was improved. Padding was added around the text - so the line and text should not overlap.
- Some miscellaneous changes to windows - minor and probably not noticable.

Version 2.5.2 (June 14, 2017)

- Updated the translation for Italian (fixing my Google Translate text) - *Thanks to Roberto Atzori*
- Fixed a bug that prevented the key combinations of Control+ and Control- from zooming the main window.

Version 2.5.1 (June 12, 2017)

- Updated the translation for Traditional Chinese - *Thanks to 黃昕暉 (Shin-Wei Hwang)*
- Updated the Color Scheme Editor. Now the editor, opens the current scheme. It also applies any changes to the main window.
- The icon used for Traditional Chinese (the Taiwanese flag), was updated with a better version.

Version 2.5 (June 8, 2017)

- Added support for Traditional Chinese - *Thanks to 黃昕暉 (Shin-Wei Hwang)*
- *Customized Color Schemes were added.*
 - Color schemes can be created and edited with the use of a new Color Scheme Editor.
 - The files are saved using the .fclr extension.
 - To make it easy to load schemes, "open" buttons were added to the Print, Export Bitmap, Export Scalable Graphics, and Change Chart Style windows.
- The Variable Watch Window now graphically displays the name of the current function.
- Chart preview images (used in the Print, Export Bitmap, Export Scalable Graphics, and Chart Chart style) were improved. They are larger, contain an additional shape, and show text on the shapes.
- Auto-scrolling, used when the window automatically scrolls when selecting shapes, works far better now. This should make it far easier to select shapes.
- Window activation works better now. Beforehand, if the user chose to show a window, but it was minimized, it stayed minimized. Now, the window will normalize.
- Source Code Viewer:
 - PHP was added. It contains both the "hanging" and "aligned" brace formatting styles.
 - Apple Swift 3 was added. Swift 2 was retained.
 - Fixed a bug in Python generated code. The code was using 'true' and 'false' rather than 'True' and 'False'. *Special thanks to Jon Rios for reporting this.*
- The "Traffic" color scheme was modified to have a blue background. The original color scheme can still be downloaded on the [Color Schemes](#) page.

Version 2.4.3 (March 18, 2017)

- Fixed a bug in C# generated code. The generated source code uses 'boolean' rather than 'Boolean'. *Special thanks to Jeffrey Burton.*

Version 2.4.2 (March 18, 2017)

- Fixed a bug in Python generated code. The bug affected For Loops. *Special thanks to Dave Braunschweig.*

Version 2.4.1 (March 17, 2017)

- Fixed a minor bug that caused the Function Dropdown to flicker when the language was set to Thai.

Version 2.4 (March 17, 2017)

- The modulo operator (mod, %) is now restricted to positive integers. Different programming languages treat negative integers differently - some inheriting the negative sign from the dividend and others the divisor. This can cause issues with code generation.
- The Console Window and the ToReal() intrinsic function will now accept commas as the decimal-point. Beforehand, the system would only accept the period for (for ToReal) and either a comma or period in the Console Window. To make the output consistent, the system will always use the international notation - the period.
- The main toolbar was updated. The icon for Clear Breakpoint was removed and replaced with an icon for Layout Windows.
- When a shape is edited, the system now clears the selection. This will allow the user to see the shape change from gray (incomplete) to color.
- Source Code Viewer:
 - Improved Java generated code. In particular, the Scanner library is only included if needed.
 - Fixed a **major** bug that caused the application to crash if a new Input Shape added.
 - Fixed For Loops in Python. For loops use the range() to create a series of values. However, this does not include the ending value. *Special thanks to Paul Gardner.*
 - Fixed the generated input functions for VBA. The name "input" conflicts with a reserved word. *Special thanks to Roberto Atzori.*
 - Cleaned up the IBO Pseudocode.
- Fixed a minor display bug that occurred when a new function was added. The screen was not updating the alignment - so the new function was being left-aligned rather than centered.
- Fixed some translation errors (that I made). *Special thanks to Roberto Atzori.*
- Fixed a formatting bug in the Console Window. When the information was displayed in simple text, it was not being aligned correctly.

Version 2.3.1 (March 9, 2017)

- Fixed a minor bug with the colors displayed in the Console Window.

Version 2.3 (March 8, 2017)

- Added support for Arabic - *Thanks to Amr Ibrahim Khudair*
- Added support for Slovenian - *Thanks to Gregor Anželj*
- The application can now support right-to-left languages. This required extensive testing and some major internal changes. *Special thanks to Amr Ibrahim Khudair for all his help.*
- Added IBO Pseudocode to the Source Code Viewer.
- Flowcharts are now centered in the window. Beforehand, the flowcharts would always start left-aligned.
- Resizing the main window works far better now. This was an oversight on my part.
- The scaling of the flowchart works better now.
- Fixed a bug with the ToInteger() and ToReal() intrinsic functions. They were not allowing the dash "-" used to denote negative numbers. *Thanks to Roberto Atzori for reporting this.*
- The icon on the "Enter" button in the Console Window was changed.

Version 2.2 (October 13, 2016)

- Added support for Turkish - *Thanks to Cafer Usta*
- Added support for Mongolian - *Thanks to Tuvshinbayar Amarsaikhan*
- The Italian translation was improved. As new features were added to Flowgorithm, I manually translated new fields. However, some were not accurate. *Thanks to Roberto Atzori for his help.*
- The Export to Vector tool was improved. In particular, Scalable Vector Graphics files will now work better on many websites and applications. Beforehand, text was aligned vertically using the "alignment-baseline" attribute. However, support for it is inconsistent across applications. So, text is now manually aligned.
- Fixed a bug that affected internal lookup tables. It affected Turkish-language computers.

Version 2.1.2 (October 3, 2016)

- The Print feature now uses generated EMF files. The resulting documents will print quicker and will be higher quality.
- In the last two updates, the size of the text on the Error Window was too small. I've restored the original appearance.

Version 2.1.1 (September 30, 2016)

- Enhanced the Most Recently Used (MRU) file list. If a file is not found, it will be removed from the list.
- The MRU will now only be saved to the user's local settings.
- Fixed a minor bug in EMF exported files. The character width of the ampersand (&) was not being measured correctly.

Version 2.1 (September 27, 2016)

- Added support for Polish - *Thanks to R. Robert Gajewski*
- Added support for Thai - *Thanks to Pongsathorn Sraouthai (พงศธรณ์ สระอุทัย)*
- Added a Most Recently Used (MRU) file list.
- The Print Window now displays the printing status on the main status bar (rather than a local one).
- Updated the EMF files created by the Export to Vector utility. They should print better than before.

Version 2.0.2 (August 16, 2016) - Bug fix

- Fixed a bug that occurred when a variable was assigned data that is incompatible with its data type (basic type mismatch). The system wasn't properly treating it as an error.

Version 2.0.1 (August 4, 2016) - Minor update

- Translations:
 - Updated the Hungarian translation.
 - Fixed a flaw in the British English translation.
- Fixed a bug in the Source Code Viewer for C++ generated code.
- The Console window now displays a message if a Breakpoint is reached.
- Made some configuration changes to improve installation.
- Some minor cosmetic changes.

Version 2.0 (August 2, 2016)

- Added a new shape: Breakpoint
 - The new shape is available from the Add Shape pop-up menu. It acts as the same as the breakpoint feature found in most IDEs.
 - Add a "Remove Breakpoints" icon to the main toolbar.
 - File format was updated. The "fileversion" now displays "2.0" rather than "1.7". This is based on that fact that newer programs can save the <breakpoint/> tag. This won't be a problem for older versions of Flowgorithm. They were designed to simply ignore unrecognized tags. So, if a program (that contains breakpoints) is opened in a older version, they will simply be absent.
- Easy right-clicking shapes. Single shapes can be simple clicked and cut/copy/pasted. This is a highly requested feature.
- Added buttons the Edit Function window that allows parameters to be reordered.
- New graphics library! The graphics engine was completely rewritten using a new design.
 - Shapes appear better - especially the curved edges used by the terminal shape.
 - Faster
 - Arrows now display a fine tip (the arrow heads are triangles). This will be apparent when the flowcharts are displayed in high resolution.
 - Mouse-overs highlight the entire line. Beforehand, it only highlighted the top or bottom half of the line.
- Miscellaneous improvements
 - Add Shape popup menu displays much, much better.
 - Auto scrolling works much better (and is much faster).
 - Layout Windows received some minor aesthetic improvements.
 - New message dialog windows. The new windows use custom icons (information, questions, etc...). Also, the buttons are now translated.
- Colors & Styles
 - New color scheme: Magic. This color scheme is a homage to Harry Potter. It uses the 4 house colors, the Sorting Hat, and snow/Hedwig/Buckbeak.
 - New color scheme: Traffic. This color scheme is based on colors used in traffic signs. It was inspired by the new breakpoint shape (which is an octagon).
 - Some of the other color schemes were tweaked slightly.
 - The Add Shape popup menu shows the "hover" effect using the a new color. Before, it was always using the menu label color.
 - Added a new chart style: Blocks. The style only uses the rectangle shape.
- New: Export to Scalable Vector Graphics. Flowcharts can now be exported to Scalable Vector Graphics (SVG) and Enhanced Metafiles (EMF).

Version 1.12.1 (May 8, 2016)

- Updated the language icons. I was displaying them in 8-bit rather than 32-bit. They should look much nicer now.
- Added the current language to the status bar of the main window. This was designed to make it easy for someone, who doesn't understand the current displayed language, change it. I didn't include the flag image. The flag is designed to help the user locate the correct icon, but seems a tad unnecessary on the main window.
- In the Export Image window, the margin (in pixels) inputbox was changed to a combobox. The user can enter any value they want, but it has the most common ones available on the drop down list.
- The visual styles manifest was removed from the installation script. This was originally added so the tile background would not appear red in Windows 8.1. However, the issue became moot with the release of Windows 10.

- Bug fixes:
 - The Print Windows wasn't showing its progress in its status bar.
 - The Export Image Window wasn't showing updates in its status bar.

Version 1.12 (March 30, 2016)

- Added support for Russian - *Thanks to Olga Cooke.*
- Updated the generated C++ code to use the vector template.

Version 1.11.3 (January 29, 2016)

- Added support for the Brazilian dialect of Portuguese - *Thanks to Theldo Cruz Franqueira.*

Version 1.11.2 (January 17, 2016)

- Added the Play, Step, Pause, and Stop buttons to the Console Window. These have the same behavior as the buttons on the main window.
- Made some internal changes to make the Console Window draw better.
- I forgot to fix a bug in Lua generated code. It was missing the -1 for descending loops.

Version 1.11.1 (January 15, 2016)

- Fixed a bug in JavaScript generated code. The For Loop was not being setup correctly for decreasing values.
- Fixed a bug in Python Code. A comma was missing for decreasing loops.
- Fixed a translation error in Galician. This was completely my fault.

Version 1.11 (January 11, 2016)

- Added support for Japanese - *Thanks to Saito Yutaka*
- Clipboard Popup Menu now follows the current color scheme.
- Rubén Castiñeiras Lorenzo released new versions of his Spanish (Castilian) and Galician translations.
- Added a new tab to the About Window. The new tab lists the names of everyone who has helped find bugs and give valuable feedback.
- Source Code Viewer:
 - To denote the start of the programs "main" code, the source code often contains a comment. Now, the comment will only be added if needed.
 - For Java, the generated class is now called "JavaApplication". This follows the defacto standard used by Java IDEs.
- Made some internal changes to the Add Shape Popup Menu.
- Bug fixes (thanks to Mathieu Kourouma):
 - Made some internal changes that prevent the Variable Watch Window from crashing on certain versions of Windows.
 - Fixed a various grammatical and spelling mistakes.

Version 1.10.2 (December 24, 2015)

- Added support for French - *Thanks to François Capon*
- Fixed a typo in the English translation (*thanks again to François Capon*)
- Updated the Change Translation and About windows. The languages/names are now displayed in a ListView control.

- Made some changes to the Add Shape menu so that it loads quicker and, hopefully, with less flicker.
- Fixed a small flaw in the Edit Comment window. It was displaying an icon and appearing in the Windows Task Bar. This wasn't consistent with the rest of the edit windows. I doubt anyone noticed.

Version 1.10.1 (December 18, 2015)

- Modified the Swift code created by the Source Code Viewer. The intrinsic function Random() now maps to the Swift random() function.
- Made some internal changes so the program loads faster.
- Some miscellaneous GUI tweaks.

Version 1.10 (December 17, 2015)

- Added the ability to control execution speed. The system now supports: Fast, Medium, and Slow. When set to medium or slow, the user can passively watch each shape execute. It essentially works like the user pressing the "Step" button.
- The behavior of the Play, Step, Pause, and Stop buttons were improved.
- Added **Apple's Swift** to the Source Code Viewer.
- Added a new color scheme: Vibrant.
- Made some cosmetic changes to the window headers. Now, each draws a line on the top and bottom. The goal was to prevent color clashes with the window's title bar.
- When the window is resized, it now makes sure it doesn't size it larger than the screen. Before, it was always defaulting to 800x600.
- Fixed an error in the Italian translation. It was my fault, not the original translator. When I changed the translation file format, I mistranslated "Assignment". *Thanks to Roberto Atzori for letting me know.*

Version 1.9.2 (November 17, 2015)

- Fixed an error that affected parameters displayed in the Edit Function window. The parameters weren't being translated correctly.

Version 1.9.1 (November 15, 2015)

- Fixed a typo.

Version 1.9 (November 14, 2015)

- Added Portuguese - *Thanks to Jorge Duque.*
- Chart styles:
 - When selecting a chart style and color scheme, the preview image now shows the proper colors and shapes. This preview is used in the Change Chart Style window, Print window, and Export Image window. The "Color Preview" caption was removed since it is no longer needed.
 - Added a new chart style: Default (Descriptive). It is identical to the "Default" chart style, but uses descriptive text on each shape.
 - Added a new color schemes: "Blueprints" and "Sacramento State". Eventually, the application will allow custom color schemes to be created.

- The current color scheme is now used in dialog windows. In previous versions, the window header background was always blue. With the change, the background color changes for each scheme.
- The Add Shape popup menu was modified to handle the new color scheme design. Changes include the hover effect as well as the border of the window.
- Most dialog windows were increased in size.
- Added a Copy button to the Console Window.
- Made some internal changes in preparation for a Macintosh/Linux port.
- Random number generation was improved. This is an internal change.
- Bug fixes:
 - Fixed a bug in For Loops. The system was not checking the range before executing the first time.
 - Fixed a bug in QBasic generated code. The dollar suffix was missing on the call to MID\$.

Additional Information

- [Older Revisions](#)