



Istituto Statale 'Biagio Pascal'
Istituto Tecnico Tecnologico – Liceo Scientifico
Via Brembio,97- 00188 - Via dei Robilant,2 - 00194 – Roma
Centralino: 06-12112-4205 via Brembio - 06-12112-4225 Via dei Robilant
Codice meccanografico RMTF330002 C.F. 97046890584 Web: www.itispascal.it
Email: RMTF330002@istruzione.it Pec: RMTF330002@pec.istruzione.it

Liceo Matematico

U.D. 11

AlgoBuild

Lezione "peer to peer"

presentazione organizzata da alcuni alunni della terza A sui programmi che costruiscono diagrammi a blocchi



L'obiettivo generale

Obiettivi formativi

La Peer education può essere usata in molti contesti, come approccio metodologico volto a rendere i ragazzi protagonisti del processo formativo. I giovani, in quanto promotori di apprendimento e di cambiamento di atteggiamenti nei confronti dei loro coetanei, vengono considerati "esperti" in determinati argomenti.

Si tratta quindi di un approccio educativo che assume l'attività fra pari come un metodo per diffondere informazioni e sviluppare strategie efficaci tramite un processo di condivisione di pensieri, assunzione di impegni reciproci e negoziazione di compromessi che, nel contempo, consente un atteggiamento di apertura verso nuove idee.

Nel dettaglio per Peer Education intendiamo un metodo educativo in base al quale alcuni membri di un gruppo, dopo essere stati opportunamente preparati, si reinseriscono nel gruppo per svolgere precise attività con i coetanei.

Si tratta di un sistema molto interessante, perché:

- rende più maturi i peer educator;
- insegna a tutti che il rapporto tra coetanei, pur sempre piacevole, può avere anche scopi più alti del semplice gioco - passatempo;
- facilita l'apprendimento, in quanto il peer educator è naturalmente in grado di utilizzare il linguaggio più consono e di adeguare il lavoro alle necessità del gruppo;

Obiettivi didattici



- Conoscere un ambiente visuale in cui disegnare dei flow-chart rispettando le regole della programmazione strutturata.
- Conoscere la sintassi di algobuild

Prerequisiti

- Uso di base del pc

Materiale

- Nel laboratorio di informatica uso del programma Algo Build
- Presentazione con Power Point

Esperienza

- Calcolo della somma tra due numeri
- Calcolo dell'area di un poligono
- Calcolo del perimetro
- Crivello di Eratostene

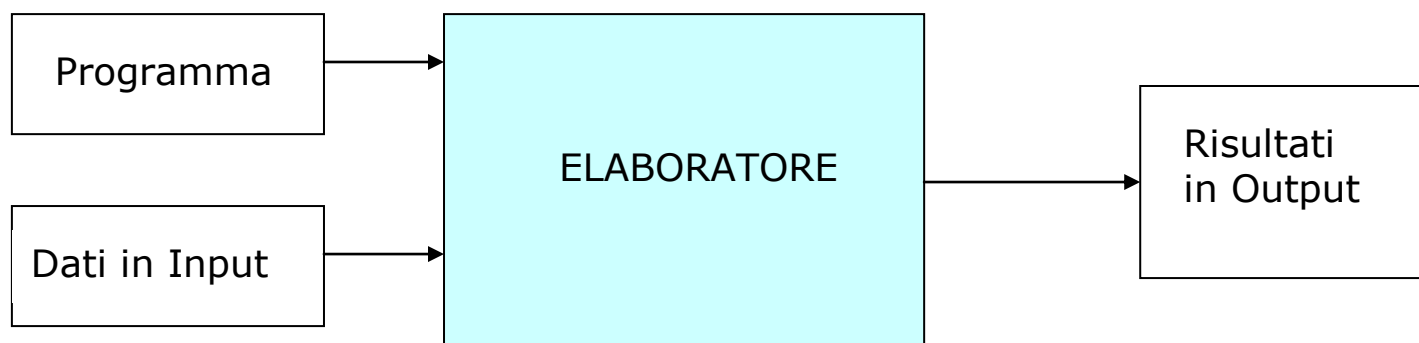
Verifica

- Saper creare un algoritmo con il programma Algo Build

Tempi: 6 ore

INTRODUZIONE ALLA PROGRAMMAZIONE

Un **programma** è un insieme finito di istruzioni che, eseguite in sequenza, permettono di elaborare i dati in ingresso per ottenere in uscita i risultati richiesti dall'elaborazione, in modo da risolvere un determinato problema.



Le **variabili** e le **costanti** sono gli oggetti utilizzati dalle istruzioni del programma. Esse risiedono nella memoria dell'elaboratore e corrispondono a contenitori dei valori che vengono man mano elaborati durante l'esecuzione del programma.

Alle variabili e alle costanti è associato un nome, detto **identificatore**, che le individua univocamente all'interno del programma. Alla variabile è associato un valore che può cambiare durante l'esecuzione del programma, mentre alla costante è associato un valore che non cambia per tutta l'esecuzione.

L'operazione tipica che viene eseguita su una variabile è detta **assegnazione** e corrisponde all'*inserimento di un valore nel contenitore associato alla variabile, sostituendo il valore precedentemente contenuto*.

Il simbolo utilizzato negli algoritmi per indicare un'operazione di assegnazione è una freccia orientata da destra verso sinistra, perchè tutto ciò che si trova a destra della freccia (un valore, una variabile, un'espressione, una costante) va assegnato al contenitore della variabile il cui identificatore si trova a sinistra della freccia, andando quindi a sostituire il valore precedentemente contenuto:

Identificatore variabile $\leftarrow$ Valore, Identificatore variabile, Identificatore costante, Espressione

Ad esempio, l'istruzione:

Calcola il discriminante dell'equazione usando la formula $\Delta = b^2 - 4ac$

può essere così scritta:

$\Delta \leftarrow B^2 - 4AC$

L'istruzione *Calcola la radice $x = -b/2a$* può essere così scritta:

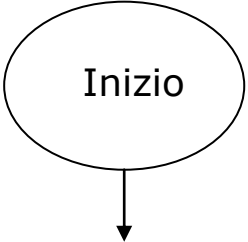
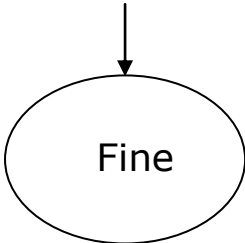
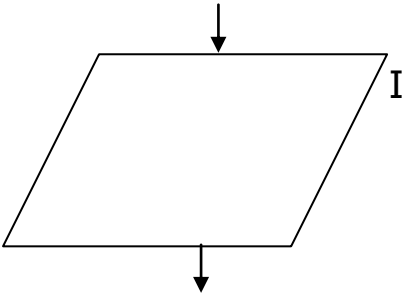
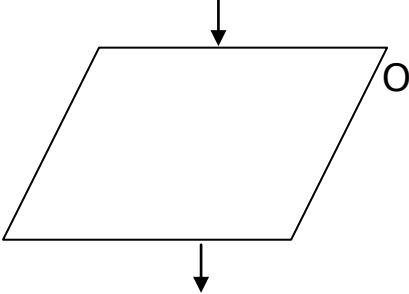
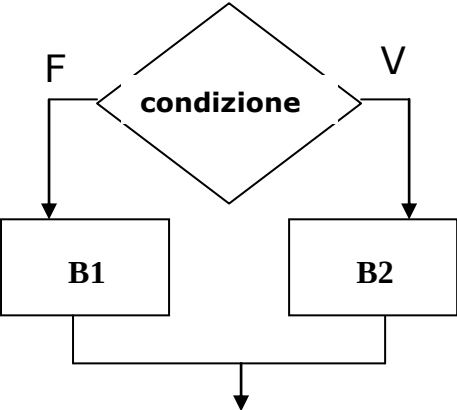
$X \leftarrow -B/(2A)$

ESEMPI DI ASSEGNAZIONE

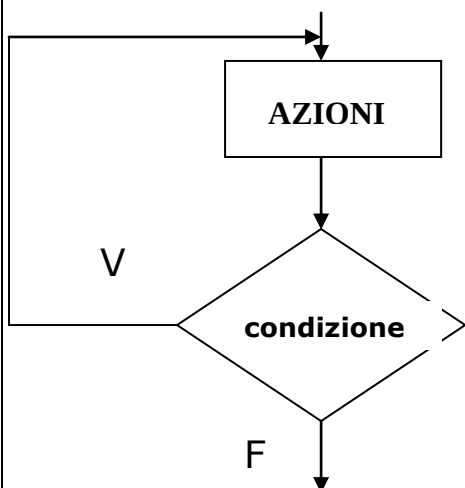
Assegnazione	Risultato	Commento
$A \leftarrow 9$	Prima: A ? Dopo: A 9	Nel contenitore associato alla variabile <i>A</i> viene inserito il valore 9
$B \leftarrow A$	Prima: A 9 B ? Dopo: A 9 B 9	Nel contenitore associato alla variabile <i>B</i> viene inserito il valore contenuto nel contenitore associato alla variabile <i>A</i>
$Area \leftarrow (B * H)/2$	Prima: Area ? B 5 H 3 Dopo: Area 7,5 B 5 H 3	Prima viene valutata l'espressione a destra dell'assegnazione quindi il risultato viene inserito nel contenitore associato alla variabile <i>Area</i>

TIPI DI ISTRUZIONI

Istruzione	Significato	Esempi
Inizio algoritmo	Si usa ad inizio algoritmo	Inizio
Fine algoritmo	Si usa a fine algoritmo	Fine
Azione	Istruzione che compie determinate azioni sui dati per produrre risultati intermedi o finali	Calcola $\Delta = b^2 - 4ac$ Dividi la somma per 5 Somma un numero ai precedenti
Comunicazione di Ingresso	Istruzione di ingresso: permette la comunicazione da utente a elaboratore	Leggi il 1° <i>valore</i> ricevuto in ingresso
Comunicazione di Uscita	Istruzione di uscita: permette la comunicazione da elaboratore a utente	Stampa a video x_1 ed x_2 Stampa a video la media Stampa a video il messaggio "Ciao"
Selezione	Istruzione che permette di prendere una decisione in base al risultato di una condizione, in modo da seguire una via oppure un'altra	Se $\Delta > 0$ allora altrimenti ... Se il 1° <i>valore</i> > 2° <i>valore</i> allora altrimenti ...
Ciclo o ripetizione	Istruzione che permette di ripetere una serie di azioni fin quando una condizione è vera	<i>Ripeti</i> <i>fino a quando</i> $A < 10$

Istruzione	Simbolo	Significato
Inizio algoritmo		<i>Blocco di Inizio algoritmo</i>
Fine algoritmo		<i>Blocco di Fine algoritmo</i>
Azione		<i>Blocco di Azione: eseguo l'azione descritta all'interno del rettangolo</i>
Comunicazione di Ingresso		<i>Blocco di Input dati: chiedo in ingresso all'utente un dato che verrà scritto in una variabile in memoria</i>
Comunicazione di Uscita		<i>Blocco di Output dati: fornisco in uscita all'utente un dato che verrà visualizzato a video</i>
Selezione		<i>Blocco di Selezione: verifico la condizione e se il risultato è vero passo ad eseguire le istruzioni sul ramo corrispondente a vero, altrimenti passo ad eseguire le istruzioni sul ramo corrispondente a falso</i>

Ciclo o ripetizione



Blocco di *Ciclo* o *Ripetizione*:
eseguo le AZIONI, verifico la condizione e ripeto le AZIONI fino a quando il risultato è vero, quando la condizione diventa falsa eseguo le istruzioni che seguono il blocco

GLI OPERATORI BOOLEANI

AND

C1	C2	C1 AND C2
Vera	Vera	Vera
Vera	Falsa	Falsa
Falsa	Vera	Falsa
Falsa	Falsa	Falsa

OR

C1	C2	C1 OR C2
Vera	Vera	Vera
Vera	Falsa	Vera
Falsa	Vera	Vera
Falsa	Falsa	Falsa

NOT

C	NOT C
Vera	Falsa
Falsa	Vera

PRINCIPALI TIPI DI DATI

- **Numerici**: sono i dati che contengono numeri, sui quali è possibile effettuare le operazioni aritmetiche (+, -, *, /) e quelle relazionali (=, <, ≤, >, ≥, ≠). Possono ancora essere ulteriormente suddivisi in:
 - **interi**: dati numerici che non prevedono cifre decimali;
 - **reali**: dati numerici che prevedono cifre decimali.
- **Alfanumerici**: sono i dati che contengono caratteri alfabetici minuscoli (dalla a fino alla z) e maiuscole (dalla A fino alla Z), cifre numeriche (da 0 a 9) e dei simboli speciali (', ", %, =, \$, ...). Gli operatori applicabili sono quelli relazionali usuali più altri operatori che dipendono dal linguaggio di programmazione che si sta utilizzando. Possono ancora essere ulteriormente suddivisi in:
 - **caratteri**: dati alfanumerici semplici;
 - **stringhe**: dati alfanumerici composti da più caratteri in sequenza.
- **Booleani**: i valori sono quelli appartenenti alle coppie (vero, falso), (true, false), (1, 0), (sì, no). Gli operatori applicabili sono quelli booleani.

I DATI IN C++

Tipo	Valore
<i>short int</i>	numero intero compreso tra -32768 e + 32767 (2 byte)
<i>long int</i>	numero intero compreso tra -2.147.483.648 e + 2.147.483.647 (4 byte)
<i>int</i>	A 2 byte come <i>short int</i> , a 4 byte come <i>long int</i> .
<i>float</i>	numero decimale in singola precisione
<i>double</i>	numero decimale in doppia precisione
<i>bool</i>	può assumere valore true (vero) o false (falso).
<i>char</i>	carattere ASCII
<i>char[n]</i>	Stringa formata da n caratteri

Nel linguaggio C++ ogni dato prima di essere usato deve essere **dichiarato**.

La dichiarazione di una **costante** si effettua usando la parola chiave *const* seguita dal tipo, dal nome e dal valore che la costante assume:

```
const char RISPOSTA = 's';
const int PER_IVA = 18;
const float PIGRECO = 3.14;
```

In C++ le **variabili** sono dichiarate facendo precedere il loro nome dal tipo.

```
short int i;
bool risposta;
char nome [10];
```

STRUTTURA DI UN PROGRAMMA IN C++

In C++ ogni istruzione termina con il separatore *"punto e virgola"* (;), le righe di commento devono iniziare con i caratteri //, mentre un commento che si protrae su più righe inizia con /* e termina con */.

Un blocco di istruzioni (due o più istruzioni) deve essere racchiuso tra *"parentesi graffe"*, la { si ottiene digitando 123 tenendo contemporaneamente premuto il tasto Alt 123 mentre la } si ottiene digitando 125 tenendo sempre contemporaneamente premuto il tasto Alt 125.

Operatori aritmetici

addizione	+
sottrazione	-
moltiplicazione	*
divisione	/

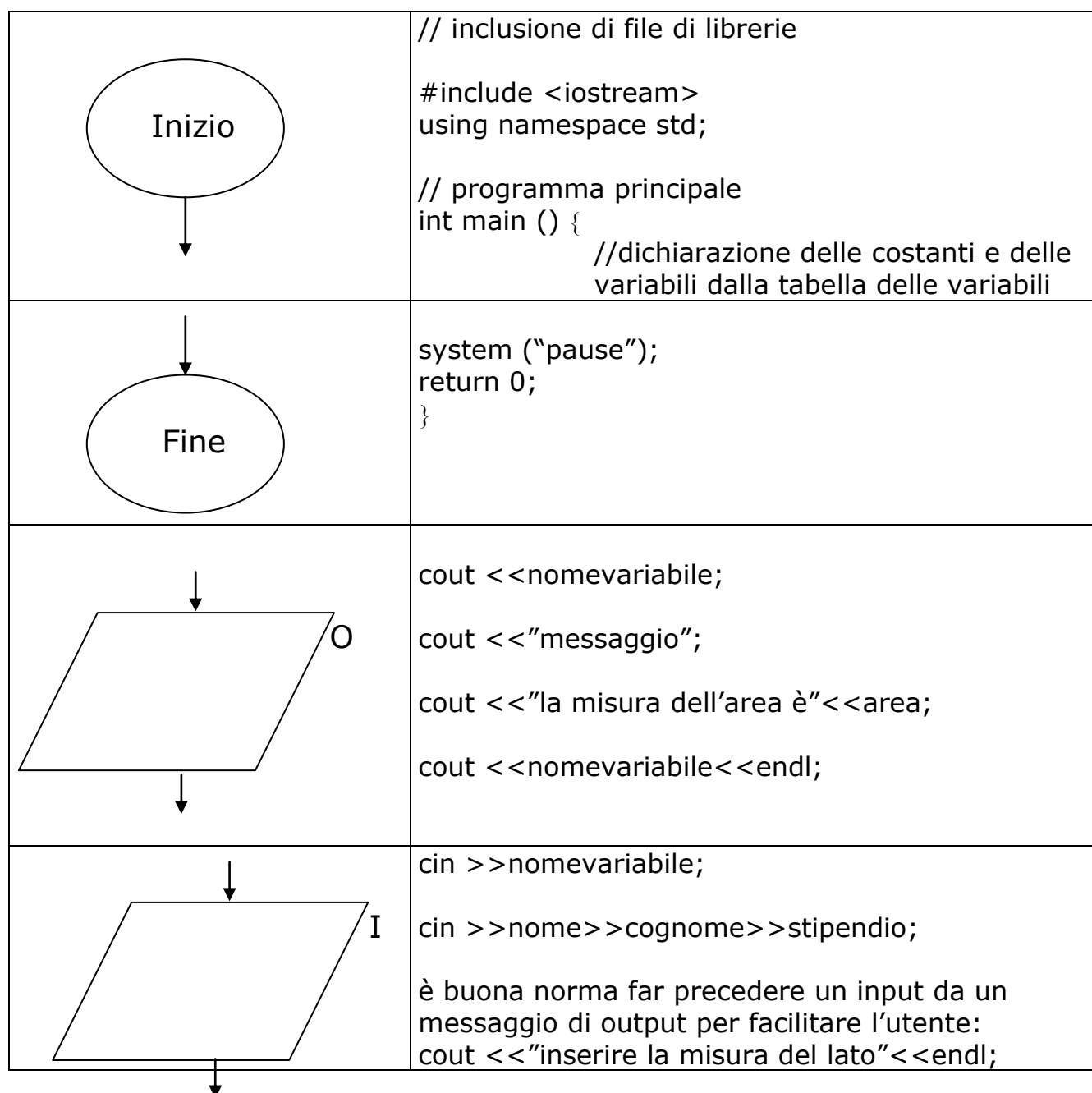
Operatori relazionali

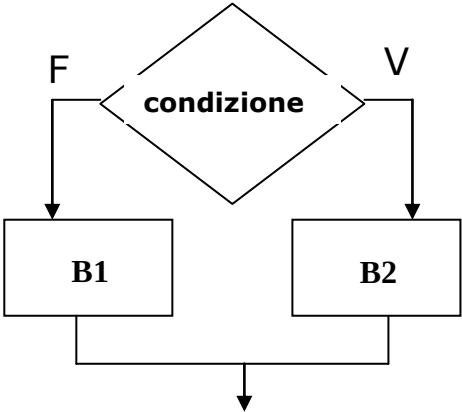
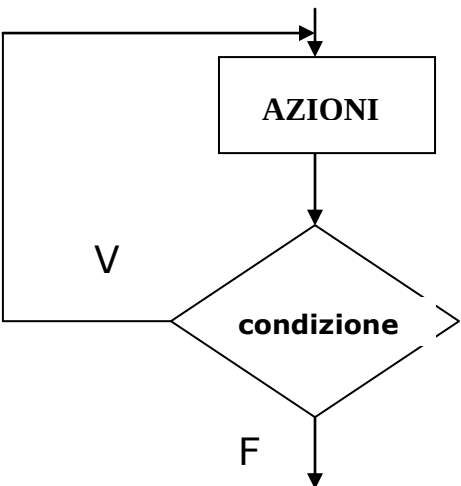
uguaglianza	==
minore	<
maggiore	>
diverso	!=
maggiore uguale	>=
minore uguale	<=

Operatori logici

and	&&
or	
not	!

L'istruzione di assegnazione (←) si traduce con il simbolo di uguale (=).



	cin >>lato;
	<pre>A=5; B=A; Area=Base*Altezza; Num=Num+1;</pre>
	<pre>if (condizione) { // blocco istruzioni B1(vero); } else { // blocco istruzioni B2 (falso); }</pre> <pre>if (A > B) { cout <<"il maggiore è "<<A; } else { cout <<"il maggiore è "<<B; }</pre> <i>In base alla condizione vengono eseguite in alternativa i due blocchi di istruzioni.</i>
	<pre>do { // blocco azioni; } while (condizione) ;</pre> <pre>do { cout<<"digita un numero<<endl; cin>>num; somma=somma + num; contatore = contatore + 1; } while (contatore < 10) ;</pre> <i>Le azioni vengono ripetute fin quando la condizione è vera, quando la condizione diventa falsa il programma prosegue in sequenza.</i>

Sintassi del c++

- << Vengono usati a stretto contatto con un cout per restituire dei dati all'utente (es cout<<x;).
- >> Vengono usati con il cin per inserire dei dati (es cin>>x;).
- “ ” Con una frase all' interno permettono di visualizzare un messaggio in uscita .(es cout<<"quanti sono i numeri?";).
- ; Ogni istruzione termina con questo separatore , fa capire al programma che deve passare all'istruzione successiva. (es x=a+b;).
- == Indica un uguaglianza
- = Viene utilizzato per assegnare dei valori alle variabili (c=a*b;).
- _ Viene utilizzato per unire nomi di variabili (area_triangolo).
- >= Maggiore o uguale
- <= Minore e uguale
- % E' chiamato operatore modulo restituisce semplicemente il resto di una divisione intera.
- != Indica una diversità
- { Si ottiene digitando alt 123 , è particolarmente importante perchè racchiude blocchi di istruzioni insieme alla graffa sottostante .
- } Si ottiene digitando alt 125.
- // Permette all' utente di apporre un commento su una riga (es //il programma fa questo//).
- /* Permette all' utente di apporre un commento su più righe , inizia con /* e finisce con */.

Connettivi logici

- And E' usato per due azioni entrambe verificate (&&)
- Not Viene usato quando l'istruzione è falsa(!)
- Or Viene usato per azioni dove almeno una è vera (||)

Variabili

Le variabili non sono altro che dei contenitori, identificati da un nome univoco (senza spazi), di un qualsiasi valore sia esso un numero o una stringa .

Tutte le variabili, prima di essere utilizzate, **devono essere dichiarate**, cioè deve essere detto al compilatore il tipo della variabile ed il suo nome, questo per permettergli di allocare la memoria necessaria alla variabile stessa; la dichiarazione generalmente viene fatta all'inizio del programma, ma in programmi di grandi dimensioni può trovarsi anche in altre posizioni (o altri file), ma bisogna ricordare che comunque la dichiarazione di una variabile può essere fatta **una ed una sola volta**.

Tipi di variabili:

- Intero variabile numerica con valore intero -int(6,250,22)
- Carattere Variabile formata da dati alfanumerici semplici -char (a,d,f)
- Stringa Variabile alfanumerica formata da n caratteri -string (ciao,marco)
- Reale variabile numerica formata da cifre decimali -float (3.5 , 7.20 , 300.24)
- Booleana Può assumere valore true o false

Le costanti sono variabili il cui valore non cambia all' interno del programma

AlgoBuild è un software didattico per progettare programmi.

Presenta un ambiente visuale in cui disegnare dei flow-chart rispettando le regole della programmazione strutturata.

Il programma prodotto viene visualizzato anche in forma di pseudo-codice.

Se le istruzioni vengono inserite rispettando una opportuna sintassi è possibile eseguire il programma passo-passo.

Lo stadio di sviluppo del software è, per ora, solo a livello di prototipo ed ha quindi possibilità ancora piuttosto limitate.

