

# PASCAL



# Per iniziare

Compariranno vari menu, ma quelli principali sono tre

- **File** in cui troviamo: **New** per creare un nuovo documento di lavoro, **Load** per aprire lavori già salvati, **Save** per registrare un lavoro.
- **Compile** l' unico comando che ci riguarda è **Compile**, che compila il programma.
- Una volta compilato il programma, però, dobbiamo eseguirlo, e perciò è necessario selezionare dal menu **Run** il comando **Run**.

# Struttura

- **sezione intestazione:** si inserisce il nome del programma,
- **sezione dichiarazioni:** vengono digitate le variabili e la loro tipologia
- **sezione esecutiva:** viene scritto il programma vero e proprio

Per intenderci possiamo fare un banalissimo esempio: possiamo considerare il TP come un teatro in cui si inscena un' opera: il titolo dell' opera è dato dalla sezione delle intestazioni, gli attori che prendono parte all' opera vengono presentati nella sezione dichiarativa, mentre lo svolgimento dello spettacolo è rappresentato dal corpo del programma.

# Regole generali

- Le parole riservate verranno scritte in maiuscolo o con l'iniziale maiuscola.
- E' da ricordare che alla fine di qualsiasi istruzione bisogna inserire il punto e virgola (;). Anche se non è obbligatorio andare a capo dopo di esso, è preferibile farlo per rendere più agevole la lettura e la comprensione del listato.
- Inoltre per inserire un commento che non interferisca con l' esecuzione del programma bisogna inserire il commento tra asterischi, e mettere il tutto tra parentesi tonde.

# Sezione dell'intestazione

- Inizia con la parola riservata **Program** seguita dal nome o titolo e finisce con un punto e virgola

# Sezione delle dichiarazioni

- Nel settore dichiarativo di un programma bisogna definire il tipo di costanti precedute dalla parola riservata **Const** e separate da un punto e virgola e le variabili che intendiamo utilizzare nel corso del programma stesso precedute dalla parola riservata **Var**. Nel nostro caso useremo i seguenti tipi di variabili:

**INTEGER** (numeri interi positivi e negativi) -32768 / +32767

**REAL** (numeri reali) -  $2.9 \times 10^{-36}$  /  $1.5 \times 10^{35}$

- Non è detto, però, che i tipi di dati corrispondano soltanto a cifre; potremmo avere un problema che necessita di dati alfanumerici, cioè di numeri e lettere. In questo caso dovremmo introdurre due nuovi tipi di variabile: 1) **STRING[x]**: Dichiarando così un tipo di dato assumeremo che quel dato è formato da un numero intero di x caratteri alfanumerici (compresi gli spazi). Usato senza parametri non definisce la lunghezza della stringa. Inoltre nel caso in cui si dichiara una lunghezza minore del dato inserito, l'elaboratore non considererà i caratteri che eccedono. Ad esempio, se dichiarassimo la variabile a di tipo **STRING[5]** e al momento di inserire i dati digitassimo la parola "folletto", essa risulterebbe troncata in "folle", poichè i caratteri in eccesso non verrebbero considerati nell'elaborazione. 2) **CHAR**: E' meno usata di **STRING**, ma qualche volta può risultare utile. Viene usata per definire variabili da un solo carattere (eccetto numeri).

# Sezione delle istruzioni, parole riservate

- La parte esecutiva è introdotta da **BEGIN** (= inizio).
- Poi troviamo una nuova parola riservata, **READLN** (abbreviazione Read Line = leggi riga). Essa serve a leggere i dati di input. La sua sintassi è molto semplice : **READLN** (nome variabile da leggere). L'istruzione **READLN** non seguita da alcun parametro, crea un ciclo di attesa che dura finchè non viene premuto Invio. E' d'obbligo alla fine di un listato, prima dell' **END**, poichè "congela" lo schermo quando dobbiamo leggere il risultato dell' elaborazione. Ad esempio, se eseguiamo un programma senza **READLN** finale, non potremmo esaminare i risultati della nostra operazione, perchè il programma si arresterebbe troppo in fretta.
- Un'altra fondamentale istruzione è **WRITELN** (abbreviazione Write Line = scrivi riga); essa serve a stampare i dati di output sullo schermo. La sua sintassi è **WRITELN** ('Stringa di testo') Il testo tra apici (') verrà stampato sullo schermo; nel caso in cui **WRITELN** venga usato senza parametri l'elaboratore lascerà una riga vuota nell'elaborazione.
- alla fine del programma troviamo **END** seguito da un punto. Esso indica la fine del flusso delle informazioni e il conseguente arresto del programma.

# Istruzioni

- Istruzione di assegnazione è l'istruzione mediante la quale viene attribuito un valore ad una variabile, è sempre preceduta da due punti e l'uguale  
Area:=B\*H;
- Istruzioni di input read o readln è la procedura di lettura di tutte le variabili dichiarate
- Istruzione di output write o writeln è la procedura di scrittura sul monitor di tutto ciò che ci interessa visualizzare



# Alcune osservazioni

- Se in una qualsiasi parte del programma dobbiamo identificare qualcosa con delle parole, queste inizieranno sempre con una lettera e non dovranno contenere spazi o simboli diversi
- Per separare parole o numeri useremo solo spazi
- Se le variabili sono di tipo REAL il computer le scrive in **notazione esponenziale in virgola mobile**  
7 diventa 7.00000E+00, che significa  $7,00000 * 10^0 = 7$   
221,13 diventa 2.21130E+02, che significa  $2,21130 * 10^2 = 221,13$

# Errori in Pascal

- 1) Out of memory : Il compilatore non ha abbastanza memoria per eseguire o compilare il programma. Provare a cambiare destinazione del codice eseguibile.(Vedi Capitolo1)
- 2) Unknown identifier : Identificatore sconosciuto. Non abbiamo dichiarato una variabile.
- 3) Duplicate identifier: Identificatore duplicato. Due variabili identiche.Cambiarne una.
- 4) Syntax error: E' sbagliata la sintassi di un istruzione.
- 5) Type mismatch: Si è dichiarata una variabile diversamente da come si è considerata. Ad esempio si è caricato il valore 1 in una variabile di tipo CHAR;
- 6) String constant exceeds lines: Si è dimenticato di mettere la stringa di testo fra apici.
- 7) Unespected end of file: Il numero dei BEGIN non coincide col numero degli END. Verificarli.

# Esempio

- PROGRAM triplo; ( \* Sezione dell' intestazione \* )
- VAR n:INTEGER; ( \* Sezione delle dichiarazioni \* )
- begin ( \* Corpo del programma \* )
- Writeln ( ' Questo programma calcola il triplo di un numero dato' );
- writeln;
- Writeln ( 'Introduci un numero intero' );
- Readln (n);
- n:=n\*3;
- writeln;
- Writeln ( ' Il triplo è ' ,n);
- writeln;
- Writeln ( ' Premi enter per finire' );
- readln;
- end.