

Il Pascal (continua)

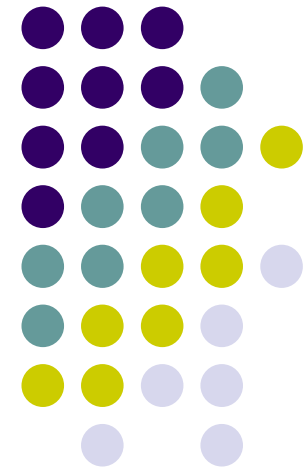
Università degli Studi di Salerno

Corso di Laurea in Scienze della Comunicazione

Informatica generale (matr. Dispari)

Docente: [Angela Peduto](#)

A.A. 2005/2006



Operatori



- I segni di operazione in Pascal , sono i seguenti:
 - **+** = ADDIZIONE
 - **-** = SOTTRAZIONE
 - ***** = MOLTIPLICAZIONE
 - **/** = DIVISIONE
 - **MOD** = RESTO DELLA DIVISIONE (MOD può essere usato soltanto se i numeri da dividere sono di tipo INTEGER)
- Inoltre per indicare il rapporto fra due cifre ci serviamo dei seguenti simboli matematici :
 - **<** = MINORE
 - **>** = MAGGIORE
 - **=** = UGUALE
 - **=>** = MAGGIORE O UGUALE
 - **<=** = MINORE O UGUALE
 - **<>** = DIVERSO DA ...



I tipi di dati

- Come già visto in precedenza, nel settore dichiarativo di un programma bisogna definire il tipo di variabili che intendiamo utilizzare nel corso del programma stesso.
- Esistono vari tipi di variabili, e sono raggruppate in questo breve specchietto.

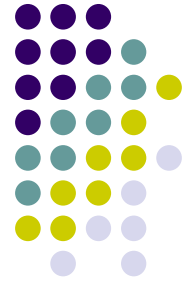
Nome tipo	Intervallo consentito	Commenti
INTEGER	-32768 / +32767	numeri interi
BYTE	0 / 255	numeri interi
WORD	0 / 65535	numeri interi
REAL	2.9×10^{-36} / 1.5×10^{35}	numeri reali

Stringhe



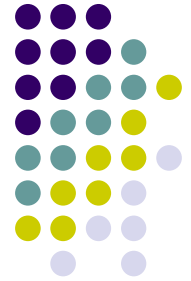
- potremmo trovarci di fronte a un problema che necessita di dati alfanumerici, cioè di numeri e lettere.
- In questo caso dovremmo introdurre due nuovi tipi di variabile
- **STRING[x]** : dichiarando così un tipo di dato assumeremo che quel dato è formato da un numero intero di x caratteri alfanumerici (compresi gli spazi).
 - usato senza parametri non definisce la lunghezza della stringa.

Stringhe (2)



- 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", poiché i caratteri in eccesso non verrebbero considerati nell'elaborazione

Caratteri



- CHAR : E' meno usata di STRING, ma qualche volta può risultare utile.
- Viene usata per definire variabili da un solo carattere (eccetto numeri).



Nota

- *write()* e *writeln()* che permettono, come già visto in precedenza, di stampare su schermo il contenuto delle parentesi.
- *read()* e *readln()* che permettono di leggere l'input dell'utente da tastiera inserendo fra le parentesi il nome della variabile in cui vogliamo salvare il dato.
- La differenza fra la versione con e senza *ln* è che la prima va a capo prima della scrittura o dell'immissione, la seconda versione no



L'assegnazione

- L'istruzione di assegnazione consente di inserire un valore in una variabile. Il formato è:
$$\textit{identificatore} := \textit{espressione} ;$$
- dove *espressione* è una qualsiasi espressione che produca un risultato che può essere inserito nella variabile il cui nome è *identificatore* (ovvero, un valore dello stesso tipo della variabile).
- Esempio:
$$\textit{area} := 3.14 * \textit{raggio} * \textit{raggio};$$
 - calcola il risultato dell'espressione a destra del simbolo ':=' e lo inserisce nella variabile il cui nome è 'area'.
- Si noti che l'espressione può anche contenere un riferimento alla stessa variabile in cui si inserisce il risultato:
$$\textit{i} := \textit{i} + 1;$$
 - significa: inserisci nella variabile 'i' il *vecchio* valore della variabile 'i' aumentato di 1.


```
PROGRAM text;                                {using text variables}
VAR
  name : STRING[10];
  age  : INTEGER;
  days : INTEGER;
BEGIN
  WRITE('enter your name : ');
  READLN(name);
  WRITE('enter your age  : ');
  READLN(age);
  days := 365 * age;
  WRITELN('hello ', name);
  WRITELN('you are ', age);
  WRITELN('you must be at least ', days, ' days old!');
  readln;
END.
```





Selezione binaria

- Se consideriamo un numero naturale qualsiasi e ci domandiamo se esso è pari, la risposta potrà avere solo due alternative: Sì oppure No.
- In Pascal c'è la possibilità di eseguire una scelta binaria tra due condizioni con il seguente comando:
- **IF** <proposizione1> **THEN** <istruzione1> **ELSE** <istruzione2>



- program Max;
- var x,y :integer;
- begin
- writeln ('Inserisci il primo numero: ');
- readln(x);
- writeln ('Inserisci il secondo numero: ');
- readln(y);
- writeln ('Valore maggiore: ');
- if (x>y) then writeln(x) else writeln(y);
- writeln;
- write ('Premi INVIO per terminare');
- readln;
- end.

Selezione binaria (2)



```
IF <proposizione1> THEN  
  begin  
    <istruzione1>  
    <istruzione2>  
    <istruzione...>  
  end;  
ELSE  
  begin  
    <istruzione1x>  
    <istruzione1x>  
    <istruzione..x>  
  end;
```

```
program area;
uses Crt;
var a,b,c: integer;
begin;
  CLRSCR;
  writeln ('scrive l'area di un rettangolo se maggiore di 20' );
  writeln ('inserisci base ');
  readln (b);
  writeln ('inserisci altezza ');
  readln (a);
  c:= a*b;
  if (c > 10) AND (c < 20) then
  begin
    writeln ('l'area del rettangolo e" ', c);
    readln;
  end
  else
  begin
    writeln ('l'area del rettangolo non e" compresa tra 10 e 20');
    readln;
  end;
end.
```



Librerie



- Una libreria non è altro che un file contenente molte funzioni e variabili che possono essere utilizzate una volta incluso il file all'interno del programma. Questo da un punto di vista teorico è molto utile perché permette di avere programmi molto brevi, ma da un punto di vista di sviluppo è piuttosto negativo, poiché una volta compilati anche programmi molto semplici possono essere estremamente “grandi” in dimensioni fisiche (ovvero in kByte).
- Il Pascal utilizza come libreria standard *Turbo.tpl*. Questa contiene tutte le funzioni di base. Non viene mai invocata, perché lo fa in automatico il compilatore alla compilazione. Ma come abbiamo già visto esistono molte altre librerie.
- Dall'ultimo programma osservato si nota come deve essere fatta la dichiarazione delle librerie:
 - *uses nomelibreria, nomelibreria, ...*; *uses* è una parola riservata, quindi non può essere usata come nome di variabile o funzione

CRT



- La libreria di gran lunga più usata è sicuramente *crt*, con tutte le funzioni relative all'estetica in ambito DOS. Ma non abbiamo ancora affrontato il problema dell'utilizzo di librerie all'interno dei nostri programmi.
- **funzioni di crt**
 - **clrscr**; Questa funzione permette di pulire lo schermo e di posizionare il cursore in alto a sinistra in posizione (1,1)



Istruzioni if in cascata

- Capita frequentemente all'interno di un programma, di dover verificare una serie di condizioni, a ciascuna delle quali è associata un'azione, fermandosi non appena viene trovata la prima condizione vera (dopo aver eseguito l'azione corrispondente). Sebbene questo caso si riconduca all'uso di istruzioni **if** annidate una dentro l'altra, risulta preferibile evitare la proliferazione eccessiva di **begin ... end** che pregiudicherebbe la leggibilità del programma



Istruzioni if in cascata (2)

- adottando invece il seguente formato:
 if *condizione1* **then**
 istruzione1
 else if *condizione2* **then**
 istruzione2

 else if *condizioneN* **then**
 istruzioneN
 else
 istruzione_di_default;
- dove l'istruzione dopo l'ultimo **else** viene eseguita nel caso di *default*, cioè se tutte le condizioni sono false. La parte **else** finale può essere omessa nel caso in cui non sia necessario prevedere un'istruzione di *default*.