

Algoritmi, Strutture Dati e Programmi

UD 2.b: Programmazione in Pascal

Prof. **Alberto Postiglione**

Dipartimento di **Scienze della Comunicazione**

AA 2007-2008

Università degli Studi di Salerno

Modificato il 22/11/2007	Algoritmi, Strutture Dati e Programmi: Programmazione in Pascal	
<h2>Il Linguaggio di Programmazione Pascal</h2>		
<ul style="list-style-type: none">■ Esistono molti linguaggi di programmazione. Per motivi didattici utilizzeremo una versione "ridotta" di un linguaggio di programmazione tra i più semplici, il Pascal.■ Questo linguaggio presenta tutte le caratteristiche dei linguaggi più "complessi", anche se non è molto utilizzato dai programmatori professionisti (se non nella versione, chiamata Delphi).		
UD 2.b	# 2	Prof Alberto Postiglione - Dipartimento di Scienze della Comunicazione - Università Salerno

Modificata il 22/11/2007

Algoritmi, Strutture Dati e Programmi: Programmazione in Pascal

II Linguaggio di Programmazione Pascal

- Un buon compilatore Pascal (Dev-Pascal 1.9.2 + Free Pascal compiler) lo si può scaricare, gratuitamente, all'indirizzo: <http://www.bloodshed.net/devpascal.html>.
- E' sicuramente utile un testo introduttivo sulla programmazione in Pascal. Un manuale di introduzione al linguaggio lo si può trovare in rete agli indirizzi
 - http://it.wikibooks.org/wiki/Programmare_in_Pascal (Wikipedia).
 - <http://www.marcocantu.it/epascal/default.htm> (Pascal nella versione Delphi)
 - <http://www.matematiche.org/> (seguendo il link "Dispense")

UD 2. b # 3 Prof Alberto Postiglione - Dipartimento di Scienze della Comunicazione - Università Salerno

Modificata il 22/11/2007

Algoritmi, Strutture Dati e Programmi: Programmazione in Pascal

II Linguaggio di Programmazione Pascal

- Un programma nel linguaggio di programmazione Pascal è suddiviso in due blocchi
 - **(DATI) Blocco delle dichiarazioni degli oggetti di lavoro** (principalmente variabili, costanti e tipi di dato)
 - **(ISTRUZIONI) Blocco delle istruzioni** racchiuse tra le parole chiave **begin End**
- **Ogni programma comincia con la parola "program"** seguita dal nome che noi attribuiamo al programma
- **Ogni istruzione termina con un punto e virgola ';'**

UD 2. b # 4 Prof Alberto Postiglione - Dipartimento di Scienze della Comunicazione - Università Salerno

Modificata il 22/11/2007

Algoritmi, Strutture Dati e Programmi: Programmazione in Pascal

Tipi di Dato "Predefiniti"

- **Tipi di dato elementari:**
 - **Integer** (numero intero)
 - **Real** (numero reale)
 - **Char** (singolo carattere)
 - **Boolean** (può valere VERO o FALSO)
 - ...
- **Tipi di dato "composti"**
 - **String** (sequenza di caratteri)
 - **Array** (Vettore o matrice)
 - **Record** (gruppo di dati eterogenei)
 - **Text** (File contenente testi)
 - ...

UD 2. b # 5 Prof Alberto Postiglione - Dipartimento di Scienze della Comunicazione - Università Salerno

Modificata il 22/11/2007

Algoritmi, Strutture Dati e Programmi: Programmazione in Pascal

Esempi di dichiarazioni

Const

```
Mio_Nome = "Luigi";  
Mia_Età = 35;
```

Var

```
Numero_Brano : integer;  
Titolo_Brano : string;  
Costo_CD : real;
```

UD 2. b # 6 Prof Alberto Postiglione - Dipartimento di Scienze della Comunicazione - Università Salerno

Modificata il 22/11/2007

Algoritmi, Strutture Dati e Programmi: Programmazione in Pascal

Istruzioni principali

- Principali istruzioni
 - Lettura/scrittura
 - Assegnazione (corrisponde alla Sequenza)
 - Decisione (corrisponde alla decisione binaria)
 - Ciclo
- Per facilitare la lettura del codice, è **opportuno che le istruzioni vengano scritte indentate**, in base alla profondità dell'istruzione

UD 2.b # 7 Prof Alberto Postiglione - Dipartimento di Scienze della Comunicazione - Università Salerno

Modificata il 22/11/2007

Algoritmi, Strutture Dati e Programmi: Programmazione in Pascal

Istruzioni di Lettura e scrittura

- Servono per leggere e scrivere il valore di una *variabile* dall'input o sull'output (ad es. lettura da tastiera e scrittura su video)
- **Write** (*Variabile*): corrisponde ad inviare il valore corrente della *Variabile* alla periferica di output (tipicamente il monitor);
- **Read** (*Variabile*): corrisponde a inserire nella *Variabile* il valore che viene immesso dalla periferica di input (tipicamente la tastiera)

UD 2.b # 8 Prof Alberto Postiglione - Dipartimento di Scienze della Comunicazione - Università Salerno

Modificata il 22/11/2007

Algoritmi, Strutture Dati e Programmi: Programmazione in Pascal

Istruzioni di I/O: Esempio

- **Programma 1: Legge un carattere da input e lo manda in output**

```
program Leggi;  
var car: char  
begin  
    read(car);  
    write(car);  
end.
```

UD 2. b # 9 Prof Alberto Postiglione - Dipartimento di Scienze della Comunicazione - Università Salerno

Modificata il 22/11/2007

Algoritmi, Strutture Dati e Programmi: Programmazione in Pascal

Istruzione di Assegnazione (Sequenza)

- **Assegnazione:** Istruzione che **assegna il valore *attuale* di un'espressione ad una variabile**
- **Sintassi:**
Nome della Variabile := Espressione;

UD 2. b # 10 Prof Alberto Postiglione - Dipartimento di Scienze della Comunicazione - Università Salerno

Istruzione di Assegnazione: Esempio 2

- Programma 2: Legge due numeri e manda in output la loro somma

```
program Somma;  
var totale: integer;  
begin  
  read(x);  
  read(y);  
  totale :=x+y;  
  write (totale);  
end.
```

Istruzione condizionale (Decisione Binaria)

- E' un'istruzione che valuta una condizione ed esegue, in alternativa, un insieme di istruzioni o un altro in base all'esito del confronto (VERO/FALSO).

if Condizione

then begin Istruzioni₁ end

else begin Istruzioni₂ end

- Se Istruzioni₁ o Istruzioni₂ sono composte da una sola istruzione, la coppia di begin ... end può essere omessa
- Se la condizione è soddisfatta si esegue il **then**, altrimenti si esegue l'**else**

Istruzione condizionale: Esempio

- Programma 3: Legge due numeri e manda in output il massimo

```
program Massimo;  
var x,y: integer;  
begin  
  read(x);  
  read(y);  
  if x > y  
  then write (x)  
  else write (y);  
end
```

NOTA: in queste posizioni **non va inserito il punto e virgola**, in quanto l'istruzione non è ancora terminata

Istruzione ciclica

- E' un'istruzione che valuta una condizione ed esegue un insieme di istruzioni se il confronto dà esito VERO e poi torna ad eseguire il test. In caso di esito FALSO si "esce dal ciclo", cioè viene eseguita l'istruzione immediatamente successiva l'end del ciclo

- Sintassi:

```
while Condizione do  
begin  
  Lista Istruzioni  
end
```

Modificata il 22/11/2007

Algoritmi, Strutture Dati e Programmi: Programmazione in Pascal

Istruzione ciclica: Esempio

- Programma 4: Calcola la somma di 10 numeri
- Algoritmo informale:
 - 1) inizializza Totale con 0
 - 2) Esegui per 10 volte le seguenti due istruzioni
 - 2.1) leggi (x)
 - 2.2) effettua la somma di x con Totale
 - 3) stampa Totale
- Ho bisogno di un meccanismo che mi permetta di eseguire per 10 volte le istruzioni 2.1 e 2.2 (qualcosa che conti il numero di volte che 2.1 e 2.2 vengono eseguite!)

UD 2.b # 15 Prof Alberto Postiglione - Dipartimento di Scienze della Comunicazione - Università Salerno

Modificata il 22/11/2007

Algoritmi, Strutture Dati e Programmi: Programmazione in Pascal

Istruzione ciclica: Esempio

```
Program Somma2;  
var x, K, totale: integer;  
begin  
  totale := 0;  
  K := 1;  
  while K <= 10 do  
    begin  
      read (x);  
      totale := totale + x;  
      K:=K+1;  
    end;  
  write (totale);  
end
```

UD 2.b # 16 Prof Alberto Postiglione - Dipartimento di Scienze della Comunicazione - Università Salerno

Istruzione ciclica: Esempio (alternativo)

```
Program Somma2bis;  
var x, K, totale: integer;  
begin  
  totale := 0;  
  for K :=1 to 10 do  
    begin  
      read (x);  
      totale := totale + x;  
    end;  
  write (totale);  
end
```

Dipartimento di Scienze della Comunicazione