

ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ PASCAL

ΓΕΝΙΚΗ ΔΟΜΗ ΠΡΟΓΡΑΜΜΑΤΟΣ

Program Όνομα_προγράμματος;

Begin

«πρόγραμμα»

End.

Πρόγραμμα 1

Program Lesson1_Program1;

Begin

Write('Hello World !!!'); {σχόλια}

Readln;

End.

Επεξήγηση Προγράμματος

► **Program** Lesson1_Program1;

Το όνομα του προγράμματος στην περίπτωση μας λέγεται Lesson1_Program1

► **Write**('Hello World !!!');

Παρουσιάζει στην οθόνη του μήνυμα που βρίσκεται μέσα στα ('....')

► **Readln**;

Το **Readln** εδώ χρησιμοποιείται για να σταματήσει το πρόγραμμα και να περιμένει τον χρήστη να πατήσει **Enter** για να τερματιστεί.

► **End.**

Τέλος του προγράμματος

Παρατηρήσεις

1. Όλες οι εντολές τελειώνουν με το ελληνικό ερωτηματικό « ; »
2. Ότι γράψουμε μέσα στις αγκύλες { } θεωρούνε σχόλια και δεν εκτελούνται από τον compiler.

Μεταβλητές

Μεταβλητή (variable) στις γλώσσες προγραμματισμούονομάζουμε ένα γλωσσικό αντικείμενο που μπορεί να λάβει διάφορες τιμές, μία κάθε φορά

Οι τιμές μιας μεταβλητής περιορίζονται συνήθως σε ένα τύπο δεδομένων.

Βασικοί τύποι δεδομένων της Pascal:

1. **integer** – ακραίοι, παίρνουν τιμές από 32768 μέχρι 32767
2. **real** – πραγματικοί, παίρνουν τιμές από 3.4×10^{-38} μέχρι 3.4×10^{38}
3. **char** – χαρακτήρες, όπως για παράδειγμα : 'a', 'B', '+'
4. **Boolean** – Παίρνει μόνο δυο τιμές True/False
5. **String** – Συμβολοσειρά (μέγιστου μεγέθους 255 χαρακτήρων) όπως για παράδειγμα : «Hello»

Σταθερές constant

Σταθερές είναι αντικείμενα τα οποία παίρνουν μόνο ΜΙΑ τιμή. Και τοποθετούνται πάνω από τις μεταβλητές.

Συντάσσονται έτσι:

Const

Όνομα_σταθεράς = τιμή;

ΔΗΛΩΣΗ ΜΕΤΑΒΛΗΤΩΝ

Χρησιμοποιούμε την VAR βάζουμε τις μεταβλητές χωρισμένες με , και στο τέλος βάζουμε τον τύπο και ;.

Πχ VAR

a,b :integer;

MO,S :REAL;

ΕΝΤΟΛΗ ΕΙΣΟΔΟΥ ReadIn

ΓΕΝΙΚΗ ΣΥΝΤΑΞΗ **ReadIn(METABΛ);**

Με την **ReadIn(όνομα_μεταβλητής)** το πρόγραμμα περιμένει μέχρι να δοθεί τιμή στην μεταβλητή

ΠΧ ReadIn(Num1);

ΕΝΤΟΛΕΣ ΕΞΟΔΟΥ Write ,WriteIn

Η εντολή **Write** τυπώνει στην οθόνη και ο κέρσοντας δεν αλλάζει γραμμή, ενώ με την εντολή **WriteIn** τυπώνει στην οθόνη και μετά ο κέρσοντας αλλάζει γραμμή.

ΠΧ WriteIn('Sum :',Sum);

Write('Input number 1:');

ReadIn(Num1);

ΕΝΤΟΛΗ ΕΚΧΩΡΗΣΗΣ

Χρησιμοποιούμε το σύμβολο **:=** και εκχωρούμε σε μια μεταβλητή την παράσταση που είναι δεξιά πχ **a:=6 , a:=b+c**

Program Lesson2_Program1;

Var Num1, Num2, Sum : Integer;

Begin

Write('Input number 1:');

ReadIn(Num1);

Write('Input number 2:');

ReadIn(Num2);

Sum := Num1 + Num2;

WriteIn('Sum :',Sum);

ReadIn;

End.

ΤΕΛΕΣΤΕΣ -ΣΥΝΑΡΤΗΣΕΙΣ

1. ΑΡΙΘΜΗΤΙΚΟΙ ΤΕΛΕΣΤΕΣ

ΤΕΛΕΣΤΗΣ	ΣΗΜΑΣΙΑ
+	ΠΡΟΣΘΕΣΗ
-	ΑΦΑΙΡΕΣΗ
*	ΠΟΛΛΑΠΛΑΣΙΑΣΜΟΣ
/	ΔΙΑΙΡΕΣΗ
^	ΔΥΝΑΜΗ
DIV	ΠΗΛΙΚΟ ΑΚΕΡΑΙΑΣ ΔΙΑΙΡΕΣΗΣ
MOD	ΥΠΟΛΟΙΠΟ ΑΚΕΡΑΙΑΣ ΔΙΑΙΡΕΣΗΣ

ΠΡΟΤΕΡΑΙΟΤΗΤΑ: ^,(*,/,DIV,MOD),(+,-)

ΠΧ $6 \text{ DIV } 2=3$, $6 \text{ MOD } 2=0$, $7 \text{ DIV } 2=3$, $7 \text{ MOD } 2=1$

2. ΣΥΓΡΙΤΙΚΟΙ ΤΕΛΕΣΤΕΣ :

(= ,> ,< , >= ,<= , <>)

3. ΛΟΓΙΚΟΙ ΤΕΛΕΣΤΕΣ

A. AND X AND Z το αποτέλεσμα ΑΛΗΘΕΣ αν είναι και οι δύο αληθείς .

B. OR X OR Z το αποτέλεσμα ΑΛΗΘΕΣ αν είναι μία τουλάχιστον αληθής

Γ. NOT NOT X το αποτέλεσμα ΑΛΗΘΕΣ αν η X είναι ΨΕΥΔΗΣ

ΣΥΝΑΡΤΗΣΕΙΣ

1. ΤΕΤΡΑΓΩΝΙΚΗ ΡΙΖΑ **SQRT** (X) ΠΧ $\text{SQRT}(25)=5$
2. ΑΠΟΛΥΤΗ ΤΙΜΗ **ABS**(X) ΠΧ $\text{ABS}(-3)=3$
3. ΤΕΤΡΑΓΩΝΟ ΑΡΙΘΜΟΥ **SQR**(X) ΠΧ $\text{SQR}(2)=4$

ΔΟΜΗ ΕΠΙΛΟΓΗΣ

1. Η Πρόταση Ελέγχου IF ... THEN

Συντάσσεται έτσι:

```
If «πρόταση» then  
Begin  
«εντολές»  
End;
```

Ο σκοπός της πρότασης ελέγχου IF...THEN είναι να ελέγχει μια έκφραση αν είναι αληθής ή ψευδής. Σε περίπτωση που είναι **αληθής** τότε εκτελούνται οι εντολές που ακολουθούν **αλλιώς** παραλείπονται άρα δεν εκτελούνται.

ΑΣΚΗΣΗ :Να γίνει πρόγραμμα που να διαβάζει ένα αριθμό και να εμφανίζει μήνυμα αν είναι θετικός-αρνητικός –μηδέν

```
Program Program1;  
Var a : Integer;  
Begin  
Write('Dose arithmo:');  
Readln(a);  
if (a>0) then  
  Begin  
    Write('Thetikos');  
  End;  
if (a<0) then  
  Begin  
    Write('Amitikos');  
  End;  
if (a=0) then  
  Begin  
    Write('Edoses 0');  
  End;  
Readln;  
End.
```

1. Η Πρόταση Ελέγχου IF ... THEN...ELSE

Συντάσσεται έτσι:

If «πρόταση» then

Begin

«εντολές»

End

else

Begin

«εντολές»

End;

Ο σκοπός που βάζουμε το ELSE (αλλιώςίκα) σε μια πρόταση ελέγχου IF...THEN είναι σε περίπτωση που δεν ισχύουν οι IF αλλά κάτι άλλο.

ΠΑΡΑΤΗΡΗΣΗ: Αν μέσα στο THEN ή ELSE υπάρχει ΜΙΑ ΜΟΝΟ ΕΝΤΟΛΗ τότε μπορούμε να γράψουμε τις εντολές ΧΩΡΙΣ ΤΟ BEGIN ...END;

Για παράδειγμα:

Αν η μεταβλητή b είναι μεγαλύτερη η ίση με 100 τότε τύπωσε «OK» αλλιώςίκα τύπωσε «Error».

```
program IF2;
Var b : Integer;
Begin
Write('Dose arithmo:');
Readln(b);
if (b>=100) then
  Begin
    Write('OK');
  End
else
  Begin
    Writeln('Error');
  End;
Readln;
End.
```

ΧΡΗΣΗ ΛΟΓΙΚΩΝ ΣΥΝΘΗΚΩΝ

ΑΡΤΙΟΣ-ΠΕΡΙΤΤΟΣ

```
Program LOGIKES;  
Var c : Integer;  
Begin  
Write('Dose arithmo:');  
Readln(c);  
if (c>=0) and (c<=100) then  
  Begin  
    Write('Eise mesa sta oria')  
  End  
else  
  Begin  
    Writeln('Eise ektos oriwn');  
  End;  
IF C MOD 2=0 THEN  
  BEGIN  
    WRITELN('ARTIOS');  
  END  
ELSE  
  BEGIN  
    WRITELN('PERITOS');  
  END;  
Readln;  
End.
```

ΔΟΜΗ ΕΠΑΝΑΛΗΨΗΣ

1. ΕΝΤΟΛΗ FOR...DO

ΓΕΝΙΚΗ ΣΥΝΤΑΞΗ :Για βήμα θετικό κατά 1

For μεταβλητή: = αρχή το τέλος **do**

Begin

«Κώδικας για επανάληψη»;

End;

Πρόγραμμα 1

Τυπώνει κάθετα τους αριθμούς από το 1 ως το 10.

```
Program Lesson4_Program1;
```

```
var count :integer;
```

```
begin
```

```
for count := 1 to 10 do
```

```
begin
```

```
writeln(count);
```

```
end;
```

```
Readln;
```

```
End.
```

Εδώ έχουμε την μεταβλητή `count` (τύπου `integer`) της οποίας δίνουμε αρχική τιμή 1 και τελική τιμή 10 (άρα ο βρόγχος μας θα εκτελεστεί 10 φορές), αυτό που γίνεται μέσα στον βρόγχο είναι ότι τυπώνετε η τιμή που παίρνει κάθε φορά η μεταβλητή `count` δηλαδή 1,2,3,4,5,6,7,8,9,10.

ΓΕΝΙΚΗ ΣΥΝΤΑΞΗ : Για βήμα αρνητικό κατά 1

For μεταβλητή: = αρχή downto τέλος do

Begin

«Κώδικας για επανάληψη»;

End;

Πρόγραμμα 2

Τυπώνει κάθετα τους αριθμούς από το 10 ως το 1.

```
Program Lesson4_Program2;
```

```
var count :integer;
```

```
begin
```

```
for count := 10 downto 1 do
```

```
begin
```

```
writeln(count);
```

```
end;
```

```
Readln;
```

```
End.
```

Κάνει το ίδιο με το πρόγραμμα 1 αλλά αυτή τηνφορά ο μετρητής count μετρά ανάποδα (downto). Άραη μεταβλητή count παίρνει τις τιμές:10,9,8,7,6,5,4,3,2,1

Παρατήρηση:

Η εντολή FOR χρησιμοποιείται όταν έχουμε συγκεκριμένο αριθμό επαναλήψεων.

2. ΕΝΤΟΛΗ WHILE ...DO

ΓΕΝΙΚΗ ΣΥΝΤΑΞΗ :

while «Συνθήκη» do

Begin

«Κώδικας για επανάληψη»;

End;

ΛΕΙΤΟΥΡΓΙΑ : Εκτελείται όσο η συνθήκη είναι αληθής .Χρησιμοποιείται όταν ο αριθμός των επαναλήψεων δεν είναι γνωστός.

Πρόγραμμα 3

Τυπώνει κάθετα τους αριθμούς από το 0 ως το 9.

```
Program Lesson4_Program3;
```

```
var a :integer;
```

```
begin
```

```
a := 0;
```

```
while a < 10 do
```

```
begin
```

```
writeln (a);
```

```
a := a + 1;
```

```
end;
```

```
Readln;
```

```
End.
```

Καταρχάς δίνουμε αρχική τιμή στη μεταβλητή $a:=0$. Όσο το a είναι μικρότερο του 10 τυπώνει το a και στη συνέχεια προσθέτει 1 ... Άρα το a θα παίρνει τις τιμές 0,1,2,3,4,5,6,7,8,9.

Στο $a<10$ (δηλαδή το 9) θα τερματιστεί ο βρόγχος για τον λόγο ότι ΔΕΝ θα ισχύει η συνθήκη και στην συνέχεια θα τερματιστεί το πρόγραμμα.