

Η Pascal με παραδείγματα



George Hadjikyriacou – 2007
ghadjikyriacou@yahoo.com

Περιεχόμενα

Μάθημα 1 - Εισαγωγή	3
Μάθημα 2 - Μεταβλητές	8
Μάθημα 3 - Προτάσεις Ελέγχου	13
Μάθημα 4 - Βρόγχοι	19
Μάθημα 5 - Πίνακες	25
Μάθημα 6 - Διαδικασίες - Συναρτήσεις	29
Μάθημα 7 - Σειριακά Αρχεία	34
Μάθημα 8 - Λίγο Χρώμα ...	39
Μάθημα 9 - Γραφικά (BGI)	44
Μάθημα 10 - Γραφικά (BGI) - Κίνηση	53
Μάθημα 11 - Ήχος	58
Παράρτημα	61
Λύσεις Ασκήσεων	65

Μάθημα 1

Εισαγωγή

Για να ξεκινήσουμε να προγραμματίζουμε στην Pascal το πρώτο πράγμα που χρειαζόμαστε να έχουμε έναν Compiler της Pascal. Για τα παρακάτω παραδείγματα χρησιμοποιήθηκε ο Compiler Bloodshed Dev-Pascal έκδοση 1.9.2.

Ας ξεκινήσουμε λοιπόν

Ένα πρόγραμμα στην Pascal έχει την εξής δομή.

```
Program Ονομα_προγραμματος ;
```

```
Begin
```

```
«πρόγραμμα»
```

```
End.
```

Πρόγραμμα 1

Ο σκοπός του πρώτου μας προγράμματος είναι να παρουσιάσει στην οθόνη μας το μήνυμα «Hello World !!!». Παρακάτω δίνετε η λίστα:

```
Program Lesson1_Program1 ;
```

```
Uses Crt ;
```

```
Begin
```

```
Clrscr ;
```

```
Write('Hello World !!!'); {σχόλια}
```

```
Readln ;
```

```
End.
```

Επεξήγηση Προγράμματος

```
Program Lesson1_Program1 ;
```

Το όνομα του προγράμματος στην περίπτωση μας λέγεται

```
Lesson1_Program1
```

Begin

Αρχή του προγράμματος

Clrscr;

Καθαρισμός της Οθόνης.

Write('Hello World !!!');

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

Readln;

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

End.

Τέλος του προγράμματος

Παρατηρήσεις

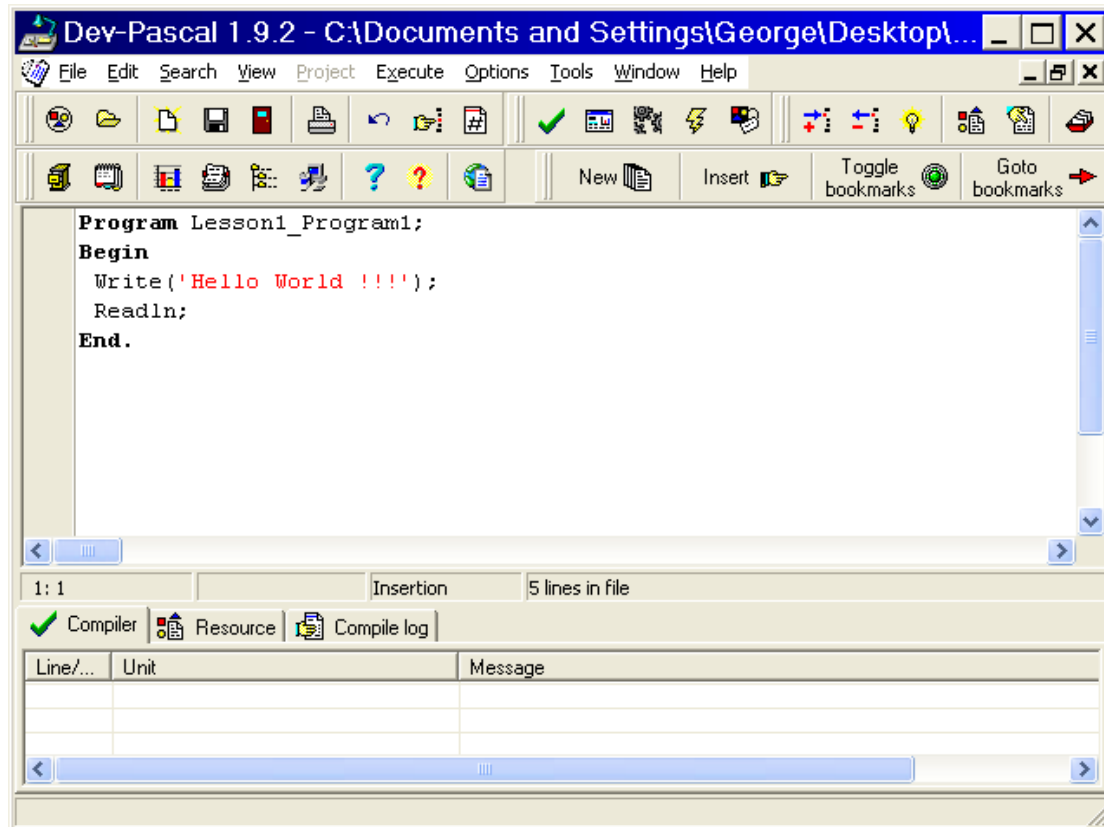
1. Όλες οι εντολές τελειώνουν με το ελληνικό ερωτηματικό « ; »
2. Ότι γράψουμε μέσα στις αγκύλες {} θεωρούνε σχόλια και δεν εκτελούνται από τον compiler.

Βήματα Υλοποίησης

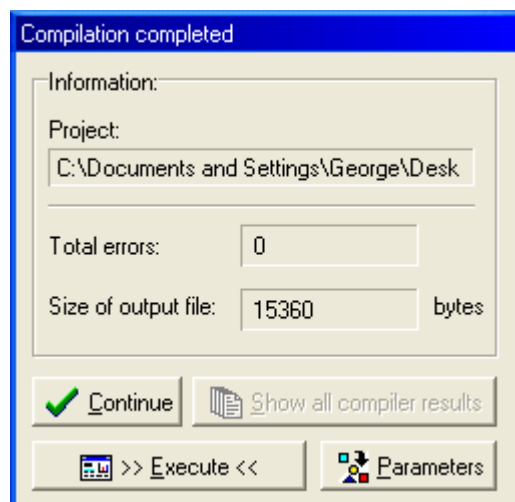
Βήμα 1. Ανοίγουμε το πρόγραμμα Dev-Pascal

Βήμα 2. Από το μενού File επιλεγούμε New Source File

Βήμα 3. Πληκτρολογούμε το κώδικα



Βήμα 4. Κάνουμε compile το πρόγραμμα που μόλις γράψαμε (αφού γίνει έλεγχος για τυχόν λάθη) από το μενού Execute – Compile



Βήμα 5. Αφού πάνε όλα καλά (Total errors: 0) τρέχουμε το πρόγραμμα μας με Execute – Run. Στην οθόνη παρουσιάζετε:

A screenshot of a Windows command prompt window. The title bar reads 'C:\ Documents and Settings...'. The main area of the window is black with white text that says 'Hello World !!!'. There is a cursor at the end of the line. The window has standard Windows window controls (minimize, maximize, close) in the top right corner and a scroll bar on the right side.

Τέλος Μαθήματος

Ορισμοί

γλώσσα προγραμματισμού. Τεχνική γλώσσα σχεδιασμένη για να δημιουργεί ή να εκφράζει προγράμματα.

Compiler. Πρόγραμμα υπολογιστή που χρησιμοποιείται για να μεταγλωττίζει (από γλώσσα προγραμματισμού σε γλώσσα που μπορεί να καταλάβει ο ηλεκτρονικός υπολογιστής)

Μάθημα 2

Μεταβλητές

Μεταβλητή (variable) στις γλώσσες προγραμματισμού ονομάζουμε 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. LongInt – ακραίοι, παίρνουν τιμές από -2^{31} μέχρι $2^{31} - 1$
3. real – πραγματικοί, παίρνουν τιμές από 3.4×10^{-38} μέχρι 3.4×10^{38}
4. char – χαρακτήρες, όπως για παράδειγμα : 'a', 'B', '+'
5. Boolean – Παίρνει μόνο δυο τιμές True/False
6. 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;
```

Πρόγραμμα 1

Ο σκοπός του προγράμματος είναι να διαβάσει (είσοδος) δυο ακεραίους από το πληκτρολόγιο και να τυπώσει (έξοδος) το αποτέλεσμα. Παρακάτω δίνετε η λίστα:

```
Program Lesson2_Program1;
```

```
Var Num1, Num2, Sum : Integer;
```

```
Begin
```

```
  Write('Input number 1:');
```

```
  Readln(Num1);
```

```
  Write('Input number 2:');
```

```
  Readln(Num2);
```

```
  Sum := Num1 + Num2;
```

```
  Writeln('Sum :', Sum);
```

```
  Readln;
```

```
End.
```

Ακολουθήστε τα βήματα υλοποίησης του Μαθήματος 1 (σελίδα 3) για να τρέξετε το πρόγραμμα.

Επεξήγηση Προγράμματος

```
Var Num1, Num2, Sum : Integer;
```

Δήλωση μεταβλητών αυτό γίνεται ξεκινώντας με το Var (από το variable), δηλώνουμε τα ονόματα των μεταβλητών που θα χρησιμοποιήσουμε (Num1, Num2, Sum) και στο τέλος δηλώνουμε τι τύπου είναι αυτές οι μεταβλητές (: Integer).

```
Readln(Num1);
```

Με την Readln(όνομα_μεταβλητής) το πρόγραμμα περιμένει μέχρι να δοθεί τιμή στην μεταβλητή.

```
Sum := Num1 + Num2;
```

Εδώ γίνεται εκχώρηση τιμής στην μεταβλητή Sum αφού κάνει την πράξη μεταξύ της Num1 και Num2 (πρόσθεση).

```
Writeln('Sum :', Sum);
```

Τυπώνει το αποτέλεσμα της πρόσθεσης, δηλαδή την τιμή της μεταβλητής Sum.

Παρατηρήσεις

3. Η εντολή Write τυπώνει στην οθόνη και ο κέρσορας δεν αλλάζει γραμμή, ενώ με την εντολή Writeln τυπώνει στην οθόνη και μετά ο κέρσορας αλλάζει γραμμή.
4. Με το := κάνουμε εκχώρηση τιμής σε μεταβλητή μέσα στο πρόγραμμα.
5. Ισχύουν οι πράξεις: +(πρόσθεση), -(αφαίρεση), *(πολλαπλασιασμός), /(διαίρεση), (...)(παρενθέσεις)

Πρόγραμμα 2

Συμβολοσειρές ...

```
Program Lesson2_Program2;  
Var name : String;  
  
Begin  
  Write('What is your Name ?:');  
  Readln(name);  
  Writeln('Hello, ', name, ' are you OK ?');  
  Readln;  
End.
```

Άσκηση

Να γράψετε ένα πρόγραμμα σε Pascal που θα διαβάζει τρεις ακεραίους αριθμούς από το πληκτρολόγιο και να τυπώνει τον μέσο όρο τους.

$$\text{Μέσος όρος} = \frac{\text{Αριθμός1} + \text{Αριθμός2} + \text{Αριθμός3}}{3}$$

Τέλος Μαθήματος

Μάθημα 3

Προτάσεις Ελέγχου

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. Για παράδειγμα:

Αν η μεταβλητή *a* είναι μεγαλύτερη από 0 **τότε** τύπωσε στην οθόνη «Θετικός» **αν** είναι μικρότερη από 0 **τότε** τύπωσε «Αρνητικός» και **αν** είναι ίση με 0 **τότε** τύπωσε «Έδωσες 0».

Για να δούμε τώρα πως μπορούμε να το πούμε αυτό στην Pascal

Πρόγραμμα 3

```
Program Lesson3_Program1;  
Var a : Integer;  
Begin  
    Write('Dose arithmo:');  
    Readln(a);  
    if (a>0) then  
        Begin  
            Write('Thetikos');  
        End;  
    if (a<0) then  
        Begin
```

```

    Write('Arnitikos');
    End;
if (a=0) then
    Begin
        Write('Edoses 0');
        End;
    Readln;
End.

```

Η Πρόταση Ελέγχου IF...THEN...ELSE

Συντάσσεται έτσι:

```

If «πρόταση» then
    Begin
        «ΕΝΤΟΛΕΣ»
    End
else
    Begin
        «ΕΝΤΟΛΕΣ»
    End;

```

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

Για παράδειγμα:

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

Πρόγραμμα 2

```

Program Lesson3_Program2;
Var b : Integer;
Begin
    Write('Dose arithmo:');
    Readln(b);
    if (b>=100) then

```

```

Begin
Write('OK');
End
else
Begin
Writeln('Error');
End;
Readln;
End.

```

Σε περίπτωση τώρα που έχουμε πάνω από μια συνθήκες να ελέγξουμε χρησιμοποιούμε τους λογικούς τελεστές AND και OR. Το AND το χρησιμοποιούμε άμα ισχύουν ΟΛΕΣ οι προτάσεις ενώ το OR χρησιμοποιείτε αν ισχύει ΜΙΑ από τις προτάσεις.

Για παράδειγμα:

Αν η μεταβλητή c είναι **μικρότερη** από 100 **ΚΑΙ** **μεγαλύτερη** από 0 τότε τύπωσε «Είσαι μέσα στα όρια» **αλλιώς** τύπωσε «Είσαι εκτός ορίων».

Πρόγραμμα 3

```

Program Lesson3_Program3;
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
Write('Eise ektos oriwn');
End;
Readln;
End.

```


Η Πρόταση Ελέγχου CASE

Παίζει τον ρόλο των πολλών IF, από τις οποίες θα εκτελεστεί μόνο η μία.

Πρόγραμμα 4

```
Program Lesson3_Program4;  
Var c : char;  
Begin  
  Write('Dose a,b or c:');  
  Readln(c);  
case c of  
  'a':  
    Begin  
      writeln('alpha');  
    End;  
  'b':  
    Begin  
      writeln('beta');  
    End;  
  'c':  
    Begin  
      writeln('gamma');  
    End;  
End;  
  Readln;  
End.
```

Παρατηρήσεις

Προσέξτε σε ποια End μπαίνει « ; » σε ποιο End μπαίνει « . » και σε ποιο End δεν μπαίνει τίποτα.

Ασκήσεις

1. Να γραφεί πρόγραμμα στη Pascal που θα ζητά δυο ακεραίους αριθμούς και να τυπώνει τον μεγαλύτερο.
2. Να γραφεί πρόγραμμα στη Pascal που θα μετατρέπει ποσά από Κυπριακή λύρα σε Ευρώ καθώς και από Ευρώ σε Κυπριακή λύρα. Στην αρχή του προγράμματος να υπάρχει μενού όπου θα επιλέγουμε τι μετατροπή θέλουμε να κάνουμε.
Να υλοποιηθεί με Case . Το μενού να είναι της μορφής:

```
1. EUR - CYP
2. CYP - EUR
Select>_
```

Τέλος Μαθήματος

Μάθημα 4

Βρόγχοι

Ο λόγος που χρησιμοποιούμε βρόγχους είναι όταν θέλουμε να επαναλάβουμε ένα κομμάτι κώδικα πολλές φορές. Οι βρόγχοι που θα εξετάσουμε είναι οι : **For, While....do** και ο βρόγχος **Repeat....Until**.

Ο Βρόγχος FOR

Συντάσσεται έτσι:

```
For μεταβλητή: = αρχή to 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 γίνετε 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.

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 την φορά ο μετρητής count μετρά ανάποδα (downto). Άρα η μεταβλητή count παίρνει τις τιμές:
10,9,8,7,6,5,4,3,2,1

Ο Βρόγχος While...do

Συντάσσεται έτσι:

```

while «Συνθήκη» do
Begin
«Κώδικας για επανάληψη»;
End;

```

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.

Ο Βρόγχος Repeat...Until

Συντάσσεται έτσι:

```
Repeat  
«Κώδικας για επανάληψη»;  
Until «Συνθήκη»;
```

Πρόγραμμα 4

```
Program Lesson4_Program4;  
var a :integer;  
begin  
a := 0;  
  
Repeat  
writeln(a);  
a := a + 1;  
Until (a > 9);  
  
Readln;  
End.
```

Όπως και στο Πρόγραμμα 3 δίνουμε αρχική τιμή στη μεταβλητή `a :=0;`.

Θα προσθέτει 1 στο `a` μέχρι το `a` να πάρει τιμή μεγαλύτερη του 9

Άρα το `a` θα παίρνει τις τιμές

`0,1,2,3,4,5,6,7,8,9.`

Μόλις το `a` πάρει την τιμή 10 (`a >9`) θα τερματιστεί ο βρόγχος για τον λόγο ότι ΔΕΝ θα ισχύει η συνθήκη και στην συνέχεια θα τερματιστεί το πρόγραμμα.

Προσοχή! Οι βρόγχοι με συνθήκη πρέπει να περιλαμβάνουν κάτι που να αλλάζει την τιμή της έκφρασης, ώστε αυτή να γίνει ψευδής και ο βρόγχος να λήξει, αλλιώς δεν θα τελειώσει ΠΟΤΕ.

Άσκηση

Να γραφεί πρόγραμμα στη Pascal (χρησιμοποιώντας βρόγχο) που θα διαβάζει 10 ακεραίους από το πληκτρολόγιο και να τυπώνει το άθροισμα τους.

Τέλος Μαθήματος

Μάθημα 5

Πίνακες

Πίνακες – Arrays

Πίνακα ονομάζουμε διάταξη δεδομένων μιας ή περισσοτέρων διαστάσεων η οποία είναι συγκεκριμένου τύπου δεδομένων.

Για παράδειγμα έχουμε ένα πίνακα ακραίων 5 θέσεων του οποίου ονομάζουμε `myarray` και τον δηλώνουμε έτσι στις μεταβλητές μας:

```
...  
myarray :array[1..5] of integer;  
....
```

<code>myarray[1]</code>	<code>myarray[2]</code>	<code>myarray[3]</code>	<code>myarray[4]</code>	<code>myarray[5]</code>
4	7	1	3	6

Πρώτο στοιχείο του πίνακα `myarray` είναι ο `myarray[1]` που έχει τιμή τον ακέραιο 4 και το τελευταίο στοιχείο είναι ο `myarray[5]` που έχει τιμή τον ακέραιο 6.

Για να δώσουμε τιμή σε ένα στοιχείο του πίνακα `myarray` γράφουμε :

Π.χ

Κατευθείαν ...

```
myarray[2] := 7;
```

ή μέσω πληκτρολογίου...

```
readln(myarray[2]);
```

Για να παρουσιάσουμε μια τιμή από τον πίνακα `myarray` γράφουμε :

Π.χ

```
write(myarray[3]);
```

Άμα είναι μεγάλος ο πίνακας καλό είναι να χρησιμοποιούμε βρόγχους για την προσπέλαση του.

Πρόγραμμα 1

Γέμισμα και Παρουσίαση στοιχείων του πίνακα στην οθόνη.

```
Program Lesson5_Program1;  
  
var  
myarray :array[1..5] of integer;  
count1,count2 :integer;  
  
begin  
  
for count1 := 1 to 5 do  
begin  
readln(myarray[count1]);  
end;  
  
for count2 := 1 to 5 do  
begin  
write(myarray[count2]);  
end;  
  
readln;  
end.
```

Άσκηση

1. Να γραφεί πρόγραμμα στη Pascal που θα γεμίζει έναν πίνακα ακεραίων τον pin1 ο ποιος είναι 10 θέσεων, στην συνέχεια θα αθροίζει τα στοιχεία και θα τυπώνει το αποτέλεσμα.
2. Να γραφεί πρόγραμμα στη Pascal που θα γεμίζει έναν πίνακα ακεραίων τον pin2 ο ποιος είναι 5 θέσεων, και

στην συνέχεια θα τυπώνει τους αρνητικούς αριθμούς καθώς και το πλήθος τους (πόσοι είναι).

3. Να γραφεί πρόγραμμα στη Pascal που θα γεμίζει έναν πίνακα ακεραίων τον `pin3` ο οποίος είναι 5 θέσεων, στην συνέχεια θα βρίσκει το μικρότερο στοιχείο και θα το τυπώνει στην οθόνη.

Οι ασκήσεις να υλοποιηθούν με βρόγχους FOR.

Τέλος Μαθήματος

Μάθημα 6

Διαδικασίες - Συναρτήσεις

Διαδικασίες – procedures

Διαδικασία ονομάζετε μια ομάδα εντολών με ή χωρίς παραμέτρους, η εκτέλεση της οποίας προκαλείται μέσω μιας κλήσης διαδικασίας.

Μαζί με τις συναρτήσεις (functions) μας γλυτώνουν από τον **επαναληπτικό προγραμματισμό**.

Πρόγραμμα 1

```
Program Lesson6_Program1;  
  
procedure dia(aa, bb, cc, dd : integer);  
begin  
  writeln ('aa :',aa,' bb :', bb,' cc :', cc,' dd :', dd);  
end;  
  
begin  
  dia(10,22,37,49);  
  readln;  
end.
```

Βήματα εκτέλεσης:

4. Γίνετε η κλήση της διαδικασίας dia(10, 22, 37, 49); Με ορισματα τους ακεραίους 10, 22, 37, 49.
5. Οι ακέραιοι 10, 22, 37, 49 περνούν σαν ορίσματα στην διαδικασία dia(aa, bb, cc, dd : integer); στις μεταβλητές aa, bb, cc, dd
6. Γίνετε η παρουσίαση των μεταβλητών στην οθόνη και τερματίζετε η διαδικασία.
7. Πατάμε [ENTER] για να τερματιστεί το πρόγραμμα.

Ένα άλλο παράδειγμα:

Πρόγραμμα 2

```
Program Lesson6_Program2;
```

```

var a,b:integer;

procedure prosthesi(aa, bb: integer);
begin
  writeln ('prosthesi :',aa+bb);
end;

procedure aferesi(aa, bb: integer);
begin
  writeln ('aferesi :',aa-bb);
end;

begin
Write('Dose ton 1o akereo :');
readln(a);
Write('Dose ton 2o akereo :');
readln(b);
prosthesi(a,b);
aferesi(a,b);
readln;
end.

```

Βήματα εκτέλεσης:

1. Το πρόγραμμα μας ζητά δυο αριθμούς που αποθηκεύονται στις μεταβλητές a και b.
2. Στη συνέχεια καλείτε η διαδικασία `prosthesi(a,b)`; όπου εκτυπώνει την πρόσθεση των δυο μεταβλητών και τερματίζετε η διαδικασία.
3. Μετά καλείτε η διαδικασία `aferesi(a,b)`; όπου εκτυπώνει την αφαίρεση των δυο μεταβλητών και τερματίζετε η διαδικασία.
4. Πατάμε [ENTER] για να τερματιστεί το πρόγραμμα.

Συναρτήσεις- functions

Δουλεύουν όπως τις διαδικασίες αλλά οι συναρτήσεις επιστρέφουν μια τιμή στο κυρίως πρόγραμμα μέσω του ονόματος της.

Πρόγραμμα 3

```
Program Lesson6_Program3;
var a,b,pro,afe:integer;

function prosthesi(aa, bb: integer):integer;
begin
    prosthesi := aa+bb;
end;

function aferesi(aa, bb: integer):integer;
begin
    aferesi := aa-bb;
end;

begin
Write('Dose ton 1o akereo :');
readln(a);
Write('Dose ton 2o akereo :');
readln(b);
pro:= prosthesi(a,b);
afe:= aferesi(a,b);
writeln('Prosthesh :',pro,' Aferesh :',afe);
readln;
end.
```

Βήματα εκτέλεσης:

1. Το πρόγραμμα μας ζητά δυο αριθμούς που αποθηκεύονται στις μεταβλητές a και b.
2. Στη συνέχεια καλείτε η συνάρτηση `prosthesi(a,b)`; όπου υπολογίζει την πρόσθεση των δυο μεταβλητών το αποτέλεσμα επιστρέφει και γίνεται εκχώρηση τιμής στη μεταβλητή `pro` του κυρίως προγράμματος.

3. Μετά καλείτε η συνάρτηση $aferesi(a,b)$; όπου υπολογίζει την αφαίρεση των δυο μεταβλητών το αποτέλεσμα επιστρέφει και γίνεται εκχώρηση τιμής στη μεταβλητή afe του κυρίως προγράμματος.
4. Γίνετε παρουσίαση των αποτελεσμάτων στην οθόνη.
5. Πατάμε [ENTER] για να τερματιστεί το πρόγραμμα.

Ασκήσεις

1. Τροποποιείτε το Πρόγραμμα 1 ούτως ώστε η διαδικασία $dia(10,22,37,49)$; να εκτελεστεί 10 φορές (να κάνετε χρήση βρόγχου).
2. Να γραφεί μια συνάρτηση (function) στη Pascal που θα δέχεται σαν ορίσματα δυο ακεραίους και να επιστρέφει στο κυρίως πρόγραμμα τον μικρότερο. Το διάβασμα των δυο ακεραίων καθώς και η κλήση της συνάρτησης να γίνεται στο κυρίως πρόγραμμα.

Τέλος Μαθήματος

Μάθημα 7

Σειριακά Αρχεία

Αρχεία κειμένου

Η δυνατότητα να αποθηκεύσεις σε ένα αρχείο δεδομένα ή να ανοίξεις ένα αρχείο κειμένου και να διαβάσεις τα περιεχόμενα του μπορεί να μας βοηθήσει πάρα πολύ στην ανάπτυξη προγραμμάτων.

Εδώ θα ασχοληθούμε με την εγγραφή και την ανάγνωση αρχείων κειμένου ή σειριακών αρχείων (sequential files).

Εγγραφή σε αρχείο κειμένου

Πρόγραμμα 1

```
Program Lesson7_Program1;  
Var FName, Txt : String;  
    UserFile    : Text;  
  
Begin  
Write('Dose onoma arxeiou (px.C:\test.txt) :');  
  readln(FName);  
  Assign(UserFile, FName);  
  Rewrite(UserFile);  
  Writeln('Grapse to kimeno : ');  
  Readln(Txt);  
  Writeln(UserFile, Txt);  
  Close(UserFile);  
End.
```

Επεξήγηση Προγράμματος

```
UserFile    : Text
```

Δηλώνουμε ότι το αρχείο θα περιέχει κείμενο(text).

```
Assign(UserFile, FName);
```

Αντιστοίχιση του αρχείου. Περνούν τα ορίσματα τύπος αρχείου και όνομα αρχείου(πλήρης διαδρομή). Δηλαδή το αρχείο FName θα είναι τύπου UserFile δηλαδή κείμενο.

```
Rewrite(UserFile);
```

Άνοιγμα του αρχείου FName για εγγραφή (αν έχει ήδη κείμενο το αρχείο σβήνετε και γράφει το νέο).

Κάνοντας χρήση του Append(UserFile); προσθέτουμε το κείμενο στο τέλος του αρχείου (**εφόσον υπάρχει το αρχείο !!!**) κρατώντας το προηγούμενο κείμενο.

```
Writeln(UserFile, Txt);
```

Εγγραφή κειμένου στο αρχείο (όχι στην οθόνη).

```
Close(UserFile);
```

Κλείσιμο του αρχείου FName (απαιτείτε).

Ανάγνωση από αρχείο κειμένου

Πρόγραμμα 2

```
Program Lesson7_Program2;
```

```
Var UserFile : Text;
```

```
    FileName, TFile : String;
```

```
Begin
```

```
    Write('Dose onoma arxeiou (px.C:\test.txt) :');
```

```
    readln(FileName);
```

```
    Assign(UserFile, FileName);
```

```
    Reset(UserFile);
```

```
    Repeat
```

```
        Readln(UserFile, TFile);
```

```
        Writeln(TFile);
```

```
    Until Eof(UserFile);
```

```
    Close(UserFile);
```

```
    Readln;
```

```
End.
```

Επεξήγηση Προγράμματος

```
Reset(UserFile);
```

Άνοιγμα του αρχείου FileName για ανάγνωση.

...

```
Repeat
```

```
  Readln(UserFile, TFile);
```

```
  Writeln(TFile);
```

```
Until Eof(UserFile);
```

...

Διάβαζε από το αρχείο `Readln(UserFile, TFile);` Και να παρουσιάζεις στην οθόνη `Writeln(TFile);` μέχρι να φτάσεις στο τέλος του αρχείου `Until Eof(UserFile);`.
(Eof – End of file)

```
Close(UserFile);
```

Κλείσιμο του αρχείου FileName (απαιτείτε).

Παρατηρήσεις

Η ανάγνωση του αρχείου γίνεται γραμμή-γραμμή μέχρι το τέλος του EOF.

Διαγραφή Αρχείου (Προσοχή - Κίνδυνος !!!)

Πρόγραμμα 3

```
Program Lesson7_Program3;
```

```
Var UFile : Text;
```

```
FileName : String;
```

```
Begin
```

```
Write('Dose onoma arxeiou (px.C:\test.txt) :');
```

```
readln(FileName);
```

```
Assign(UFile, FileName);
```

```
Erase(UFile);  
End.
```

Επεξήγηση Προγράμματος

```
Erase(UFile);  
Άνοιγμα του αρχείου FileName για διαγραφή.
```

Άσκηση

Να γράψετε μια **διαδικασία** (procedure) στην pascal που θα ζητά τις πληροφορίες: όνομα, επίθετο, e-mail, τηλέφωνο και να τις αποθηκεύει σε αρχείο κειμένου(Append). Στη συνέχεια θα γίνεται ερώτηση αν θέλουμε να συνεχίσουμε να προσθέτουμε δεδομένα στο αρχείο. Αν η απάντηση είναι θετική θα ξανακαλείται η διαδικασία, ενώ αν είναι αρνητική θα τερματίζεται το πρόγραμμα.

Τα δεδομένα να αποθηκεύονται στο αρχείο στην εξής μορφή:

```
Onoma : Γιώργος  
Epitheto : Χ.  
e-mail : ghadjikyriacou@yahoo.com  
Tilefono : 99123456
```

Τέλος Μαθήματος

Μάθημα 8

Λίγο Χρώμα...

Ας δώσουμε λίγο χρώμα ...

Εδώ πρέπει να χρησιμοποιήσουμε ένα Unit στη Pascal που ενεργοποιεί ορισμένες λειτουργίες που είναι υπεύθυνες για την έξοδο προς την οθόνη, το crt.

Μπαίνει στην αρχή του προγράμματος μας κάτω από το Program ...

Συντάσσετε έτσι:

```
Uses crt;
```

Η εντολή Textcolor (int);

Η εντολή `textcolor(int)`; μας δίνει την δυνατότητα να αλλάξουμε το χρώμα των γραμμάτων του προγράμματος μας.

Συντάσσετε έτσι:

```
textcolor(int);
```

Όπου `int` δηλώνουμε έναν ακέραιο από το 0 ως το 15 που αντιστοιχεί το κάθε χρώμα.

Τα χρήματα:

- 0 - Black
- 1 - Dark Blue
- 2 - Dark Green
- 3 - Dark Cyan
- 4 - Dark Red
- 5 - Purple
- 6 - Brown
- 7 - Light Grey
- 8 - Dark Grey
- 9 - Light Blue
- 10 - Light Green
- 11 - Cyan
- 12 - Light Red
- 13 - Pink

14 - Yellow

15 - White

Παράδειγμα η εντολή `textcolor(14);` Κάνει τα γράμματα του προγράμματος κίτρινα!

Πρόγραμμα 1

```
program Lesson8_Program1;  
  
Uses crt;  
  
var  
i : integer;  
begin  
for i:=0 to 15 do  
begin  
textcolor(i);  
delay(200);  
clrscr;  
writeln('Hello World !!!');  
end;  
readln;  
end.
```

Επεξήγηση Προγράμματος

```
delay(200);
```

Δίνει χρονική καθυστέρηση 200 ms (1000ms=1sec)

```
clrscr;
```

Καθαρισμός της οθόνης

Η εντολή textBackground (int)

Η εντολή `textcolor(int);` μας δίνει την δυνατότητα να αλλάξουμε το χρώμα του `background` των γραμμάτων του προγράμματος μας.

Συντάσσετε έτσι:

```
Background(int);
```

Όπου int δηλώνουμε έναν ακέραιο από 0 ως 7 που αντιστοιχεί το κάθε χρώμα.

Τα χρώματα:

0 - Black
1 - Dark Blue
2 - Dark Green
3 - Dark Cyan
4 - Dark Red
5 - Purple
6 - Brown
7 - Light Grey

Πρόγραμμα 2

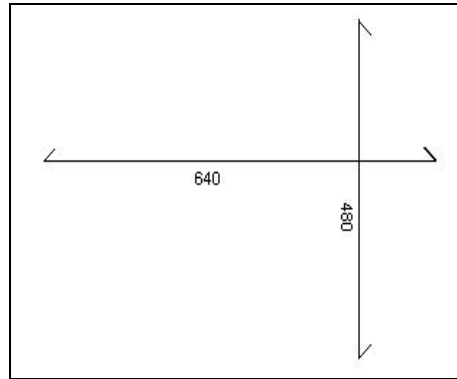
```
program Lesson8_Program2;  
  
Uses crt;  
  
var  
i : integer;  
begin  
for i:=0 to 7 do  
begin  
textBackground(i);  
delay(100);  
writeln('Hello World !!!');  
end;  
readln;  
end.
```

Τέλος Μαθήματος

Μάθημα 9

Γραφικά (BGI)

Για τα παρακάτω παραδείγματα χρησιμοποιήθηκε ο compiler της εταιρίας Borland Turbo Pascal 7 (για DOS). Τα αρχικά BGI σημαίνουν Borland Graphics Interface. Τα BGI γραφικά χρησιμοποιούν ανάλυση 640x480 (pixels)...



Αρχικοποίηση των γραφικών

```
USES CRT, GRAPH;  
  
VAR  
  
GD, GM : INTEGER;  
  
BEGIN  
    InitGraph(GD, GM, '..\BGI');  
  
    {...Πρόγραμμα...}  
  
    CloseGraph;  
END.
```

Εντολές Σχεδιασμού στην Pascal

Γραμμές – `Line(x, y, xx, yy);`

Κύκλοι – `Circle(x, y, size);`

Τετράγωνα – `Rectangle(x, y, xx, yy);`

Χρώματα –
 `SetColor(int);`

```
SetBkColor(int);
```

Γεμίματα –

```
SetFillStyle(style, color);  
FloodFill(x, y, stop_color);
```

Κείμενο –

```
Outtextxy(x, y, 'Text Here');
```

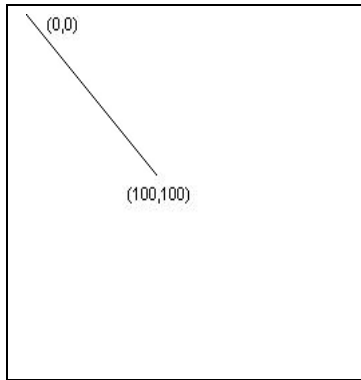
Γραμμές

Για να κάνουμε μια γραμμή χρησιμοποιούμε την εντολή `Line(x, y, xx, yy);`. Όπου `x,y` είναι οι συντεταγμένες (σε pixel), της αρχής της γραμμής ενώ τα `xx,yy` είναι οι συντεταγμένες του τέλους της γραμμής.

Πρόγραμμα 1

```
USES CRT, GRAPH;  
  
VAR  
  
GD, GM : INTEGER;  
  
BEGIN  
    InitGraph(GD,GM, '..\BGI');  
    line(0,0,100,100);  
    readln;  
    CloseGraph;  
END.
```

Έξοδος:



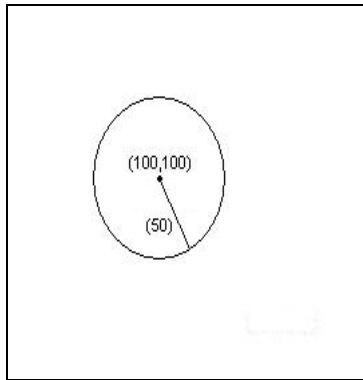
Κύκλοι

Για να κάνουμε μια γραμμή χρησιμοποιούμε την εντολή `Circle(x, y, size);` .Όπου x, y είναι οι συντεταγμένες (σε pixel), του κέντρου του κύκλου ενώ το `size` είναι η ακτίνα του κύκλου.

Πρόγραμμα 2

```
USES CRT, GRAPH;  
  
VAR  
  
GD, GM : INTEGER;  
  
BEGIN  
    InitGraph(GD, GM, '..\BGI');  
    circle(100, 100, 50);  
    readln;  
    CloseGraph;  
END.
```

Έξοδος:



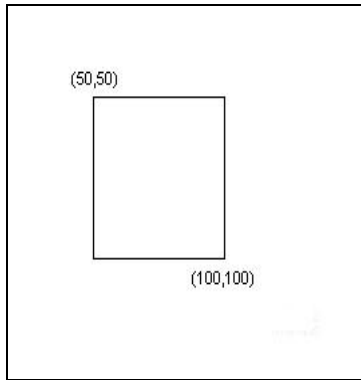
Τετράγωνα

Για να κάνουμε ένα τεράγωνο χρησιμοποιούμε την εντολή `Rectangle(x, y, xx, yy);` όπου x, y είναι οι συντεταγμένες (σε pixel), της αρχής του τετραγώνου (πάνω αριστερά) ενώ τα xx, yy είναι οι συντεταγμένες του τέλους του τετραγώνου (κάτω δεξιά).

Πρόγραμμα 3

```
USES CRT, GRAPH;  
  
VAR  
  
GD, GM : INTEGER;  
  
BEGIN  
    InitGraph(GD, GM, '..\BGI');  
    rectangle(50, 50, 100, 100);  
    readln;  
    CloseGraph;  
END.
```

Έξοδος:



Χρώματα

Για να δώσουμε χρώμα σε ένα σχήμα τότε χρησιμοποιούμε την εντολή `SetColor(int)`; Ενώ για να δώσουμε χρώμα στο φόντο χρησιμοποιούμε την εντολή `SetBkColor(int)`; . Όπου `int` ο αριθμός του χρώματος (0 - 15) που αναφέρουμε στο Μάθημα 8.

Πρόγραμμα 4

```
USES CRT, GRAPH;
```

```
VAR
```

```
GD, GM : INTEGER;
```

```
BEGIN
```

```
  InitGraph(GD,GM, '..\BGI');
```

```
  SetColor(14); {xroma sxhmatos kitrino}
```

```
  SetBkColor(1); {xroma fontou mple}
```

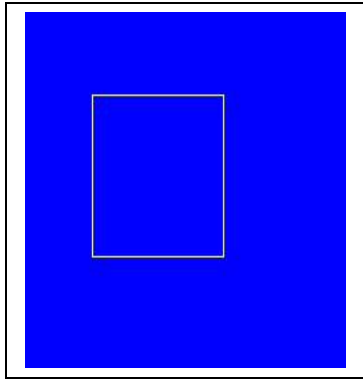
```
  rectangle(50,50,100,100);
```

```
  readln;
```

```
  CloseGraph;
```

```
END.
```

Έξοδος:



Γεμίματα

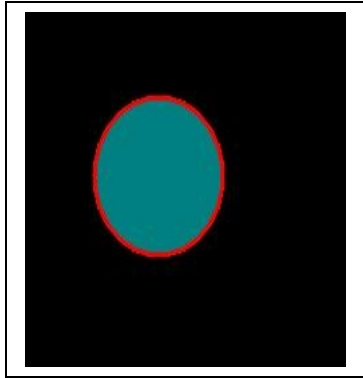
Για να γεμίσουμε ένα σχήμα με χρώμα χρησιμοποιούμε 2 εντολές. Με την εντολή `SetFillStyle(style, color);` Δηλώνουμε `style` γεμίματος (1-10) και χρώμα γεμίματος (0-15 βλέπε μάθημα 8), ενώ με την εντολή `FloodFill(x, y, stop_color);` δίνουμε την εντολή να γεμίζει από το σημείο `x,y` μέχρι να βρει το χρώμα `stop_color` (0-15 βλέπε μάθημα 8).

Πρόγραμμα 5

```
USES CRT, GRAPH;  
  
VAR  
  
GD, GM : INTEGER;  
  
BEGIN  
    InitGraph(GD,GM, '..\BGI');  
    cleardevice;  
  
    SetColor(4); {kokkino}  
    Circle(100,100,50);  
  
    SetFillStyle(1,3);  
    FloodFill(101,101,4);  
  
    readln;
```

```
    CloseGraph;  
END.
```

Έξοδος:



Κάπως έτσι ... :-)

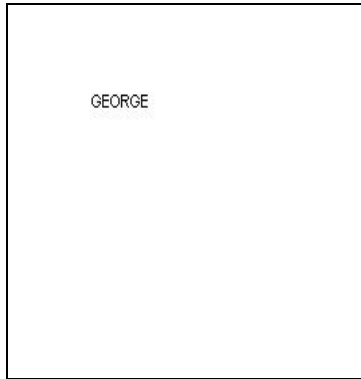
Κείμενο

Για να γράψουμε κείμενο μπορούμε να χρησιμοποιήσουμε την εντολή `Outtextxy(x, y, 'Text Here');` όπου `x,y` είναι οι συντεταγμένες που θα εμφανιστεί το κείμενο.

Πρόγραμμα 6

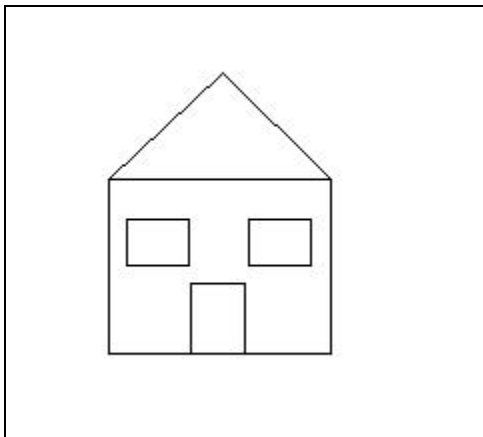
```
USES CRT, GRAPH;  
  
VAR  
  
GD, GM : INTEGER;  
  
BEGIN  
    InitGraph(GD,GM, '..\BGI');  
    outtextxy(50,50,'GEORGE');  
    readln;  
    CloseGraph;  
END.
```

Έξοδος:



Άσκηση

Να γράψετε ένα πρόγραμμα σε Pascal που να σχεδιάζει ένα σπίτι (με πόρτα και παράθυρα). Όπως το σχήμα (περίπου):



Για δική σας ευκολία καλό είναι να κάνετε χρήση τετραγωνισμένου χαρτιού για τον υπολογισμό των σημείων.

Τέλος Μαθήματος

Μάθημα 10

Γραφικά (BGI) - Κίνηση

Ας δώσουμε λίγη κίνηση τώρα στα γραφικά μας.

Πρόγραμμα 1

Μπροστά ...

```
PROGRAM Lesson9_Program1;  
  
USES CRT, GRAPH;  
  
VAR  
  
GD, GM, i : INTEGER;  
  
BEGIN  
write('Press [Enter] to Start...');  
readln;  
  
InitGraph(GD, GM, '..\BGI');  
  
for i:=10 to 350 do  
begin  
cleardevice;  
outtextxy(i, 240, '-= TEST');  
delay(1000);  
end;  
outtextxy(0, 0, 'Press [ENTER] to Exit ...');  
readln;  
CloseGraph;  
  
END.
```

Επεξήγηση του προγράμματος

```
cleardevice;
```

Κάνει καθαρισμό της οθόνης από τα γραφικά.

Πρόγραμμα 2

Και πίσω...

```
PROGRAM Lesson9_Program2;

USES CRT, GRAPH;

VAR

GD, GM,i : INTEGER;

BEGIN
write('Press [Enter] to Start...');
readln;

InitGraph(GD,GM, '..\BGI');

    for i:=10 to 350 do
    begin
cleardevice;
outtextxy(i,240,'-= TEST');
delay(1000);
end;

    for i:=350 downto 10 do
    begin
cleardevice;
outtextxy(i,240,'TEST =-');
delay(1000);
end;

    outtextxy(0,0,'Press [ENTER] to Exit ...');
    readln;
CloseGraph;

END.
```

Πρόγραμμα 3

Ένα Screensaver ...

```
PROGRAM Lesson9_Program3;

USES CRT, GRAPH;

VAR

GD, GM,i : INTEGER;

BEGIN
Randomize;
write('Press [Enter] to Start...');
readln;

InitGraph(GD,GM, '..\BGI');

for i:=0 to 500 do
begin
setcolor(random(15));
line(random(640),random(480),random(640),random(480));
delay(5000);
end;
cleardevice;
setcolor(15);
outtextxy(0,0,'Press [ENTER] to Exit ...');
readln;
CloseGraph;

END.
```

Επεξήγηση του προγράμματος

```
Randomize;
```


Αρχικοποίηση της συνάρτησης παραγωγής τυχαίων αριθμών.

```
random(int);
```

Συνάρτηση που παράγει τυχαίους αριθμούς από 1 ως int.

Άσκηση

Να γράψετε ένα πρόγραμμα στην Pascal (τύπου screensaver) που να εμφανίζει σε τυχαία σημεία, τυχαίου μεγέθους κύκλους.

Τέλος Μαθήματος

Μάθημα 11

Ήχος

Για τα παρακάτω παραδείγματα χρησιμοποιήθηκε ο compiler της εταιρίας Borland Turbo Pascal 7 (για DOS).

Για να βάλουμε ήχο σε ένα πρόγραμμα της Pascal χρησιμοποιούμε την συνάρτηση `Sound(int)`, όπου `int` βάζουμε την συχνότητα του ήχου σε Hz (Hertz).

Πρόγραμμα 1

Οι Νότες ...

```
program Lesson11_Program;

USES CRT;

{dilonoume tis sixnotites se statheres}
const
C=262;
D=294;
E=330;
F=349;
G=392;
A=440;
B=494;
H=524;
{dilonoume stathera tin diarkia se millisecond}
MS=20000;

begin
    Sound(C);
    Delay(MS);
    NoSound;
    Sound(D);
    Delay(MS);
    NoSound;
    Sound(E);
    Delay(MS);
    NoSound;
```

```
Sound(F);  
Delay(MS);  
NoSound;  
Sound(G);  
Delay(MS);  
NoSound;  
Sound(A);  
Delay(MS);  
NoSound;  
Sound(B);  
Delay(MS);  
NoSound;  
Sound(H);  
Delay(MS);  
NoSound;  
end.
```

Τέλος Μαθήματος

Παράρτημα

Βασικές Μαθηματικές Πράξεις

Πρόσθεση	$x := y + z;$
Αφαίρεση	$x := y - z;$
Πολλαπλασιασμός	$x := y * z;$
Διαίρεση	$x := y / z;$
Ακέραια διαίρεση	$x := y \text{ div } z;$
Υπόλοιπο διαίρεσης	$x := y \text{ mod } z;$

Μαθηματικές Συναρτήσεις

Τετράγωνο	SQR(Real Variable)
Τετραγωνική Ρίζα	SQRT(Real Variable)
Ημίτονο	SIN(Real variable)
Συνημίτονο	COS(Real variable)
Συνεφαπτομένη	ARCTAN(Real variable)

Πίνακας ASCII

Ctrl	Dec	Hex	Char	Code	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
␣	0	00		NUL	32	20	sp	64	40	␣	96	60	`
␣	1	01	␣	SOH	33	21	!	65	41	A	97	61	a
␣	2	02	␣	SIX	34	22	"	66	42	B	98	62	b
␣	3	03	♠	EIX	35	23	#	67	43	C	99	63	c
␣	4	04	♣	EOI	36	24	\$	68	44	D	100	64	d
␣	5	05	♠	ENQ	37	25	%	69	45	E	101	65	e
␣	6	06	♣	ACK	38	26	&	70	46	F	102	66	f
␣	7	07	♠	BEL	39	27	'	71	47	G	103	67	g
␣	8	08	␣	BS	40	28	(72	48	H	104	68	h
␣	9	09	␣	HI	41	29)	73	49	I	105	69	i
␣	10	0A	␣	LF	42	2A	*	74	4A	J	106	6A	j
␣	11	0B	♠	VI	43	2B	+	75	4B	K	107	6B	k
␣	12	0C	♣	FF	44	2C	,	76	4C	L	108	6C	l
␣	13	0D	♠	CR	45	2D	-	77	4D	M	109	6D	m
␣	14	0E	♣	SO	46	2E	.	78	4E	N	110	6E	n
␣	15	0F	♠	SI	47	2F	/	79	4F	O	111	6F	o
␣	16	10	␣	SLE	48	30	0	80	50	P	112	70	p
␣	17	11	␣	CS1	49	31	1	81	51	Q	113	71	q
␣	18	12	␣	DC2	50	32	2	82	52	R	114	72	r
␣	19	13	!!	DC3	51	33	3	83	53	S	115	73	s
␣	20	14	␣	DC4	52	34	4	84	54	T	116	74	t
␣	21	15	␣	NAK	53	35	5	85	55	U	117	75	u
␣	22	16	␣	SYN	54	36	6	86	56	V	118	76	v
␣	23	17	␣	EIB	55	37	7	87	57	W	119	77	w
␣	24	18	␣	CAN	56	38	8	88	58	X	120	78	x
␣	25	19	␣	EM	57	39	9	89	59	Y	121	79	y
␣	26	1A	␣	SIB	58	3A	:	90	5A	Z	122	7A	z
␣	27	1B	␣	ESC	59	3B	;	91	5B	[123	7B	{
␣	28	1C	␣	FS	60	3C	<	92	5C	\	124	7C	
␣	29	1D	␣	GS	61	3D	=	93	5D]	125	7D	}
␣	30	1E	␣	RS	62	3E	>	94	5E	^	126	7E	~
␣	31	1F	␣	US	63	3F	?	95	5F	_	127	7F	Δ [†]

DOS	WIN	Dec	Hex	DOS	WIN	Dec	Hex	DOS	WIN	Dec	Hex	DOS	WIN	Dec	Hex
Ç	€	128	80	á		160	A0	À		192	C0	α	à	224	E0
ü		129	81	í	ı	161	A1	Á		193	C1	β	á	225	E1
é	,	130	82	ó	¢	162	A2	Â		194	C2	Γ	â	226	E2
â	f	131	83	ú	£	163	A3	Ã		195	C3	π	ã	227	E3
ä	„	132	84	ñ	¤	164	A4	Ä		196	C4	Σ	ä	228	E4
à	...	133	85	Ñ	¥	165	A5	Å		197	C5	σ	å	229	E5
â	†	134	86	ª	¦	166	A6	Æ		198	C6	μ	æ	230	E6
ç	‡	135	87	º	§	167	A7	Ç		199	C7	τ	ç	231	E7
ê	^	136	88	¿	¨	168	A8	È		200	C8	Φ	è	232	E8
ë	‰	137	89	¬	©	169	A9	É		201	C9	Θ	é	233	E9
è	Š	138	8A	¬	ª	170	AA	Ê		202	CA	Ω	ê	234	EA
ı	‹	139	8B	½	«	171	AB	Ë		203	CB	δ	ë	235	EB
î	Œ	140	8C	¼	¬	172	AC	Ì		204	CC	∞	ı	236	EC
i		141	8D	ı	-	173	AD	Í		205	CD	ø	í	237	ED
Ä	Ž	142	8E	«	®	174	AE	Î		206	CE	ε	î	238	EE
Å		143	8F	»	¯	175	AF	Ï		207	CF	∩	ï	239	EF
É		144	90	◌◌◌	°	176	B0	Ð		208	D0	≡	ð	240	F0
æ	‘	145	91	◌◌◌◌	±	177	B1	Ñ		209	D1	±	ñ	241	F1
Æ	’	146	92	◌◌◌◌◌	²	178	B2	Ò		210	D2	≥	ò	242	F2
ô	“	147	93		³	179	B3	Ó		211	D3	≤	ó	243	F3
ö	”	148	94		´	180	B4	Ô		212	D4		ô	244	F4
ò	•	149	95		µ	181	B5	Õ		213	D5		ö	245	F5
û	—	150	96		¶	182	B6	Ö		214	D6		û	246	F6
ù	—	151	97		·	183	B7	×		215	D7	≈	÷	247	F7
ÿ	~	152	98		,	184	B8	∅		216	D8	°	ø	248	F8
Ö	™	153	99		ı	185	B9	Û		217	D9	•	ù	249	F9
Û	š	154	9A		º	186	BA	Ú		218	DA	·	ú	250	FA
¢	›	155	9B		»	187	BB	Û		219	DB	√	û	251	FB
£	œ	156	9C		¼	188	BC	Ü		220	DC	ª	ü	252	FC
¥		157	9D		½	189	BD	Ý		221	DD	²	ý	253	FD
Pls	ž	158	9E		¾	190	BE	Þ		222	DE	■	þ	254	FE
f	ÿ	159	9F		¿	191	BF	ß		223	DF		ÿ	255	FF

Λύσεις Ασκήσεων

Μάθημα 2

```
Program Mathima2_Askisi1;
Var num1,num2,num3,sum : integer;
Var mesos : real;
Begin
write ('input number1 :');
readln(num1);
write ('input number2 :');
readln(num2);
write ('input number3 :');
readln(num3);
mesos := (num1+num2+num3)/3;
writeln('mesos oros:',mesos);
readln;
End.
```

Μάθημα 3

```
Program Mathima3_Askisi1;
var a,b: integer;
Begin
write ('dose 1 ari8mo:');
readln(a);
write ('dose 2 ari8mo:');
readln(b);
If (a>b) then
begin
write (a);
end;
if (a=b) then
begin
write ('isoi !');
end
else
begin
write(b);
end;
readln;
end.
```

Program Mathima3_Askisi2

```
Var sel : integer;
Var a, met :real;
Begin
```

```

Writeln('1. EUR - CYP');
Writeln('2. CYP - EUR');
Write('Select> ');
Readln(sel);
case sel of
  1:
    Begin
      write('dose EUR: ');
      readln(a);
      met:=a * 0.58;
      writeln('CYP : ',met)
    End;
  2:
    Begin
      writeln('dose CYP: ');
      readln(a);
      met:=a * 1.74;
      writeln('EUR : ',met);
    End;
End;
Readln;
End.

```

Μάθημα 4

```

Program Mathima4_Askisi1;
var count: integer ;
var a,sum : integer ;
Begin
count:= 1;
sum:=0;
while count <=10 do
begin
readln(a);
sum := sum + a;
count := count + 1;
end;
writeln ('sum:',sum);
readln;
end.

```

Μάθημα 5

```

Program Mathima5_Askisi1;
var pin1 :array[1..10] of integer;

```

```

var count1,count2,sum :integer;
begin

for count1 := 1 to 10 do
begin
readln(pin1[count1]);
end;

for count2 := 1 to 10 do
begin
sum:= sum + pin1[count2];
end;

writeln ('sum:',sum);
readln;
end.

```

Program Mathima5_Askisi2;

```

var pin2:array[1..5] of integer;
var count1,count2,count3:integer;
begin

```

```

for count1 := 1 to 5 do
begin
readln(pin2[count1]);
end;

```

```

for count2 := 1 to 5 do
begin
if pin2[count2]<0 then
Begin
Writeln(pin2[count2]);
count3:=count3 + 1;
End;

```

```

end;
writeln ('Oi arnitikoi einai:',count3);
readln;
end.

```

Program Mathima5_Askisi3;

```

var pin3:array[1..5] of integer;
var count1,count2,min :integer;

```

```

begin
for count1 := 1 to 5 do

```

```

begin
readln(pin3[count1]);
end;

min:= pin3[1];

for count2 := 2 to 5 do
begin
if pin3[count2]<min then
    Begin
    min := pin3[count2];
    End;
end;

writeln ('Ο Mikroteros einai:',min);
readln;
end.

```

Μάθημα 6

Program Lesson6_Askisi1

```

var count:integer;

procedure dia(aa, bb, cc, dd : integer);
begin
writeln ('aa :',aa,' bb :', bb,' cc :', cc,' dd :', dd);
end;

begin
For count:=1 to 10 do
Begin
dia(10,22,37,49);
end;
readln;
end.

```

Program Lesson6_Askisi2;

```

var a,b,mik:integer;

function mikroteros(aa, bb: integer):integer;
begin
if(aa<bb)then
    begin
mikroteros := aa;
end;
if(aa>bb) then

```

```

begin
mikroteros := bb;
end;
if(aa=bb) then
begin
mikroteros := bb;
end;
end;

begin
Write('Dose ton 1o akereo :');
readln(a);
Write('Dose ton 2o akereo :');
readln(b);
mik:= mikroteros(a,b);
writeln('mikroteros :',mik);
readln;
end.

```

Μάθημα 7

Program Lesson7_Askisi;

Uses Crt;

procedure insert();

Var

```

UserFile : Text;
name, surname, mail, phone : String;
yn : char;

```

begin

```

Assign(UserFile, 'C:\Phonebook.txt');
{To arxeio C:\Phonebook.txt prepei na yparxei
gia na mporesei na ginei Append !!!}
Append(UserFile);

```

```

Write('Onoma: ');
Readln(name);

```

```

Write('Epitheto: ');
Readln(surname);

```

```

Write('E-Mail: ');

```

```

Readln(mail);

Write('Τηλεφωνο: ');
Readln(phone);

Writeln(UserFile, 'Όνομα: ' + name);
Writeln(UserFile, 'Επιθετο: ' + surname);
Writeln(UserFile, 'E-Mail: ' + mail);
Writeln(UserFile, 'Τηλεφωνο: ' + phone);
Writeln(UserFile, '-----');
Close(UserFile);

Write('Θελετε να sinexisete (y/n) :');
Readln(yn);

if(yn='y') then
begin
clrscr; {Katharismos tis othonis}
insert();
end
else
begin
Write('Bye Bye ...');
exit; {exodos apo to programma}
end;

end;

begin
insert();
readln;
end.

```

Μάθημα 9

```

PROGRAM BGI_HOUSE;

USES CRT, GRAPH;

VAR

GD, GM : INTEGER;

BEGIN

```

```

    InitGraph(GD,GM, '..\BGI');
    setcolor(9);
    line(300,0,200,200);
    line(300,0,400,200);
    rectangle(200,200,400,400);
    setcolor(3);
    rectangle(220,220,270,270);
    rectangle(330,220,380,270);
    setcolor(3);
    rectangle(270,320,330,400);
    setcolor(14);
    outtextxy(170,420,'BGI HOUSE BY GEORGE
HADJIKYRIACOU');
    setcolor(7);
    outtextxy(220,440,'PRESS [ENTER] TO EXIT');

    readln;
    CloseGraph;
END.

```

Μάθημα 10

```

PROGRAM Lessor10_Program;

USES CRT, GRAPH;

VAR

GD, GM,i : INTEGER;

BEGIN
write('Press [Enter] to Start...');
readln;

InitGraph(GD,GM, '..\BGI');

for i:=50 to 500 do
    begin
        circle(random(640),random(480),random(100));
        delay(5000);
    end;
    cleardevice;
    setcolor(15);
    outtextxy(0,0,'Press [ENTER] to Exit ...');
    readln;
CloseGraph;

```


END.