

Εισαγωγή στην πληροφορική και τις εφαρμογές της

ΠΑΝΑΓΙΩΤΗΣ ΠΑΠΑΖΟΓΛΟΥ

Website: microdev.gr



Επιστημονικές Εκδόσεις
ΤΖΙΟΛΑ

ΚΕΦΑΛΑΙΟ 14

Εισαγωγή στον Προγραμματισμό



Γενικά για τον προγραμματισμό

- **Ανάπτυξη κώδικα, η οποία αποτελεί στην ουσία την υλοποίηση του αντίστοιχου αλγόριθμου**
- **Προγραμματισμός \neq Αλγόριθμος**
- **Προγραμματισμός = σύνθεση του κώδικα**
- **Δεν αρκεί απλά να γνωρίζω τις εντολές**
- **Γλώσσα προγραμματισμού = επιλογή του προγραμματιστή**
- **Πολλαπλά κριτήρια για την επιλογή γλώσσας προγραμματισμού**



Το πρώτο μας πρόγραμμα (1)

Κώδικας C

```
//My first program
#include <stdio.h>
int main()
{
    printf("Hello, world");
    return 0;
}
```

Κώδικας Java

```
// My first program
class hello
{
    public static void main(String args[])
    {
        System.out.println("Hello, World");
    }
}
```

Κώδικας Python

```
# My first program
print ("Hello, world")
```



Το πρώτο μας πρόγραμμα (2)

Κώδικας C

```
//My first program
#include <stdio.h>
int main()
{
    printf("Hello, world");
    return 0;
}
```

#include <stdio.h>

Ενσωμάτωση συναρτήσεων από το αρχείο `stdio.h`, το οποίο υποστηρίζει την είσοδο και την έξοδο δεδομένων

int main()

Κυρίως πρόγραμμα (συνάρτηση). Αυτή η ενότητα προγράμματος τρέχει πρώτη μετά την ενεργοποίηση του προγράμματος.

```
{
```

printf("Hello, world");

Εμφάνιση μηνύματος στην οθόνη

return 0;

Επιστροφή του αριθμού 0 κατά την έξοδο από το πρόγραμμα (ο τύπος επιστροφής ταιριάζει με το `int` που σημαίνει ακέραιος)

```
}
```



Το πρώτο μας πρόγραμμα (3)

Κώδικας Java

```
// My first program
class hello
{
    public static void main(String args[])
    {
        System.out.println("Hello, World");
    }
}
```

// My first program

Αυτή η γραμμή αποτελεί σχόλιο

class hello

Κάθε πρόγραμμα Java ξεκινά με τον ορισμό μιας κλάσης. Το όνομα της κύριας κλάσης ταυτίζεται και με το όνομα του αρχείου που περιέχει τον πηγαίο κώδικα (hello.java)

```
{
    public static void main(String args[])
    {
```

Κυρίως πρόγραμμα που δεν επιστρέφει δεδομένα στο ΛΣ (τύπος void). Ο πίνακας args[] φιλοξενεί τυχόν ορίσματα που δίνονται από τη γραμμή εντολής κατά την ενεργοποίηση του προγράμματος.

```
        System.out.println("Hello, World");
```

Εμφάνιση μηνύματος στην οθόνη

```
    }
}
```



Το πρώτο μας πρόγραμμα (4)

Κώδικας Python

```
# My first program  
print ("Hello, world")
```

My first program

Αυτή η γραμμή αποτελεί σχόλιο

print ("Hello, world")



Είσοδος και έξοδος δεδομένων (1)

```
#include <stdio.h>
int main()
{
    int varA,varB;
    char name[6];
    float varC=4.53,varD;
    varA=32;
    printf("integer B=");
    scanf("%d",&varB);
    printf("real D="); scanf("%f",&varD);
    printf("My name:"); scanf("%s",name);
    float AD=varA*varD;
    int AB=varA+varB;
    printf("start\n");
    printf("A=%d, B=%d, C=%.3f,
D=%f\n",varA,varB,varC,varD);
    printf("%d+%d=%d\n",varA,varB,AB);
    printf("%d*%.2f=%.2f\n",varA,varD,AD);
    printf("My name=%s",name);
    return 0;
}
```

Είσοδος από το πληκτρολόγιο

```
integer B=2
real D=1.22
My name:panos
```

Έξοδος στην οθόνη

```
start
A=32, B=2, C=4.530, D=1.220000
32+2=34
32*1.22=39.04
My name=panos
```



Είσοδος και έξοδος δεδομένων (2)

Μια ματιά στους τύπους δεδομένων

Τύπος	Byte (τυπικό)	Εύρος τιμών
char	1	-128..127
short int	2	-32,768..32,767
int	4	-2,147,483,648.. 2,147,483,647
long int	4	-2,147,483,648.. 2,147,483,647
float	4	1.2E-38 to 3.4E+38
double	8	2.3E-308 to 1.7E+308
unsigned char	1	0..255
unsigned short int	2	0..65535
unsigned int	4	0..4,294,967,295
unsigned long int	4	0..4,294,967,295

Είσοδος και έξοδος δεδομένων (3)

Μορφοποίηση εξόδου

```
printf("A=%d, B=%d, C=%.3f, D=%f\n",varA,varB,varC,varD);
```

```
printf("A=%d, B=%d, C=%.3f, D=%f\n",varA,varB,varC,varD);
```

Χαρακτήρας μορφοποίησης	Περιγραφή
%d	Ακέραιος με πρόσημο
%c	Απλός χαρακτήρας
%f	Αριθμός κινητής υποδιαστολής
%s	Αλφαριθμητικό
%u	Μη προσημασμένος ακέραιος

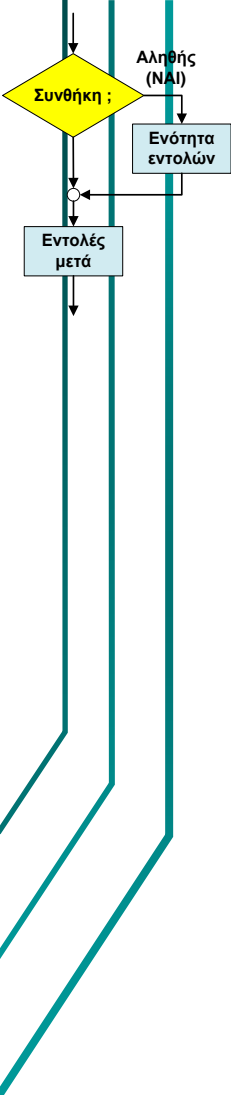
Χαρακτήρας εκτύπωσης	Αποτέλεσμα εκτύπωσης
\n	Αλλαγή γραμμής
\b	Backspace
\t	Tab
\"	Διπλά εισαγωγικά





Δείτε στο βιβλίο την αντίστοιχη υλοποίηση σε Python για κάθε πρόγραμμα σε γλώσσα C

Δείτε στο βιβλίο την υλοποίηση με κώδικα για κάθε αλγόριθμο και ψευδοκώδικα που παρουσιάζεται



Εφαρμογή του δομημένου προγραμματισμού (1)

Ακολουθία εντολών

Στην προηγούμενη ενότητα υλοποιήθηκε η ακολουθία εντολών

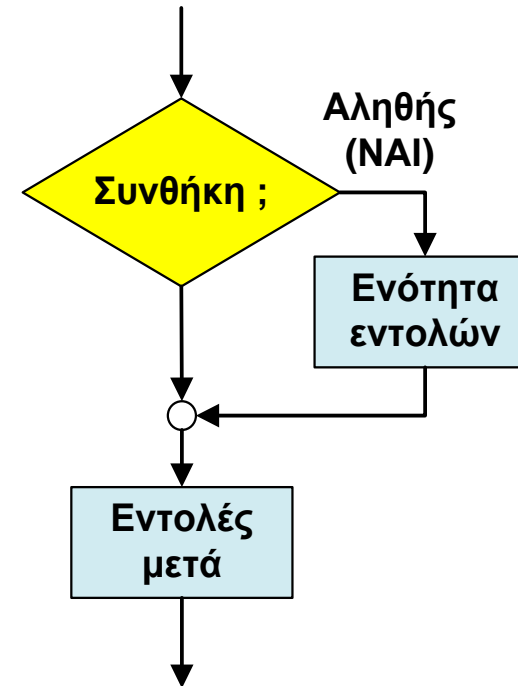
Δομή ελέγχου

Ψευδοκώδικας

`if (συνθήκη)`

`Ενότητα εντολών`

`Εντολές μετά`



Απλή εντολή if



Εφαρμογή του δομημένου προγραμματισμού (2)

Δομή ελέγχου

Ψευδοκώδικας

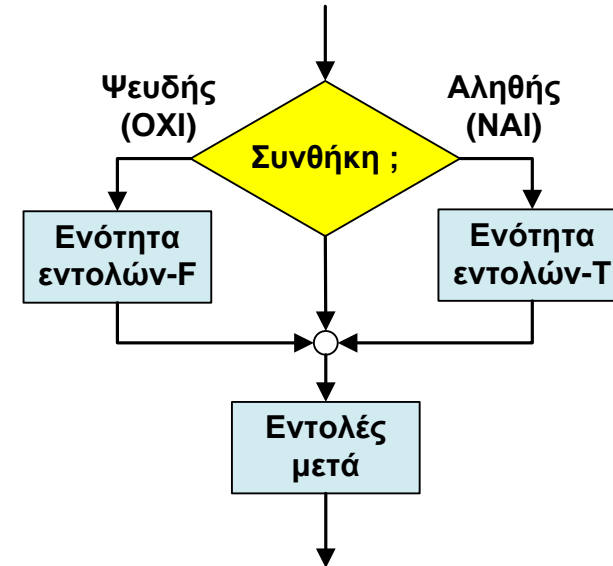
if (συνθήκη)

 Ενότητα εντολών-T

Else

 Ενότητα εντολών-F

Εντολές μετά



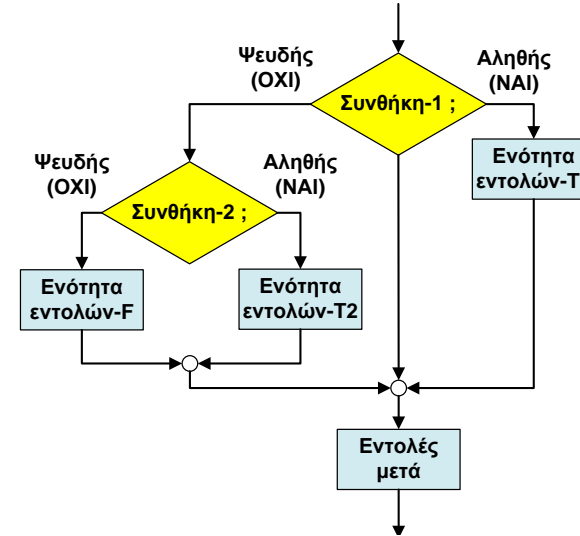
Δομή if-else

Εφαρμογή του δομημένου προγραμματισμού (3)

Δομή ελέγχου

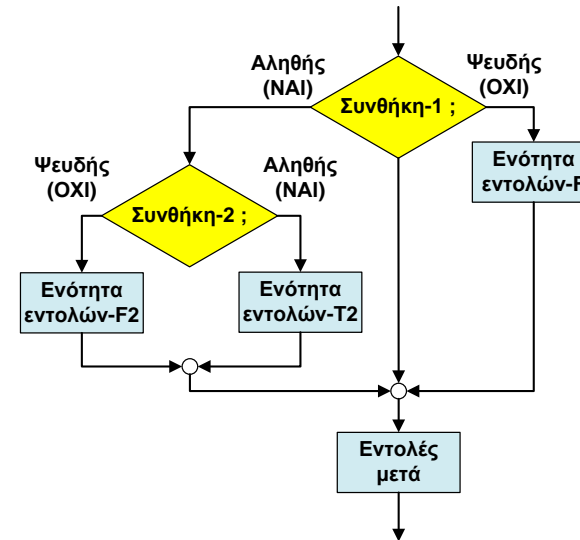
Ψευδοκώδικας

```
if (συνθήκη-1)  
    Ενότητα εντολών-T1  
Else  
    if (συνθήκη-2)  
        Ενότητα εντολών-T2  
    Else  
        Ενότητα εντολών-F  
Εντολές μετά
```



Ψευδοκώδικας

```
if (συνθήκη-1)  
    if (συνθήκη-2)  
        Ενότητα εντολών-T2  
    Else  
        Ενότητα εντολών-F2  
Else  
    Ενότητα εντολών-F  
Εντολές μετά
```



Δομή if-else-if και σύνθετες συνθήκες

Υπολογισμός συνάρτησης

$$f(x) = \begin{cases} 2x + x^2 & \text{αν } x \geq 10 \\ \frac{3x + 2}{x^2 + 1} & \text{αν } x < 10 \end{cases}$$

Ψευδοκώδικας

ΑΡΧΗ

Διάβασμα x

Αν $x \geq 10$ ΤΟΤΕ

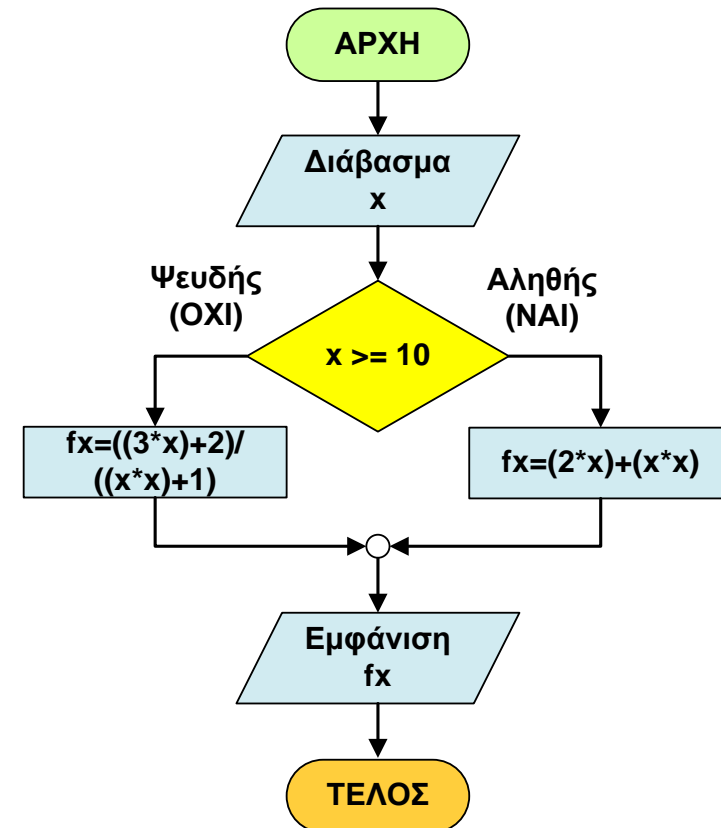
$$fx = (2 * x) + (x * x)$$

Διαφορετικά

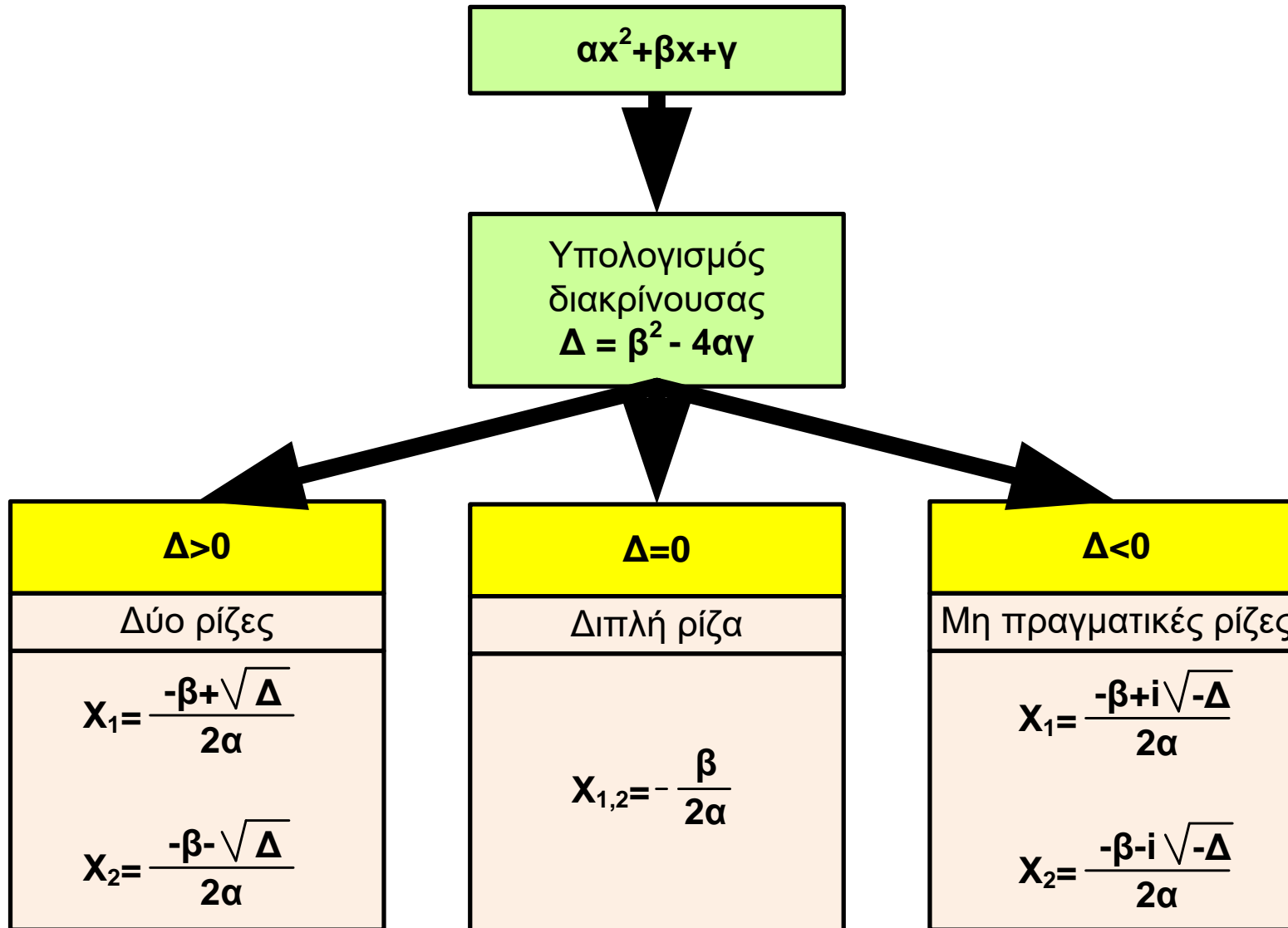
$$fx = ((3 * x) + 2) / ((x * x) + 1)$$

Εμφάνισε fx

ΤΕΛΟΣ



Επίλυση τριωνύμου (1)



Επίλυση τριωνύμου (2)

Ψευδοκώδικας

ΑΡΧΗ

Διάβασμα α, β, γ

$$\Delta = (\beta * \beta) - (4 * \alpha * \gamma)$$

Αν $\Delta < 0$ ΤΟΤΕ

Εμφάνιση «Μιγαδικές ρίζες»

Διαφορετικά

Αν $\Delta = 0$ ΤΟΤΕ

$$X12 = -\beta / (2 * \alpha)$$

Εμφάνιση «Διπλή ρίζα»

Εμφάνιση $X12$

Διαφορετικά

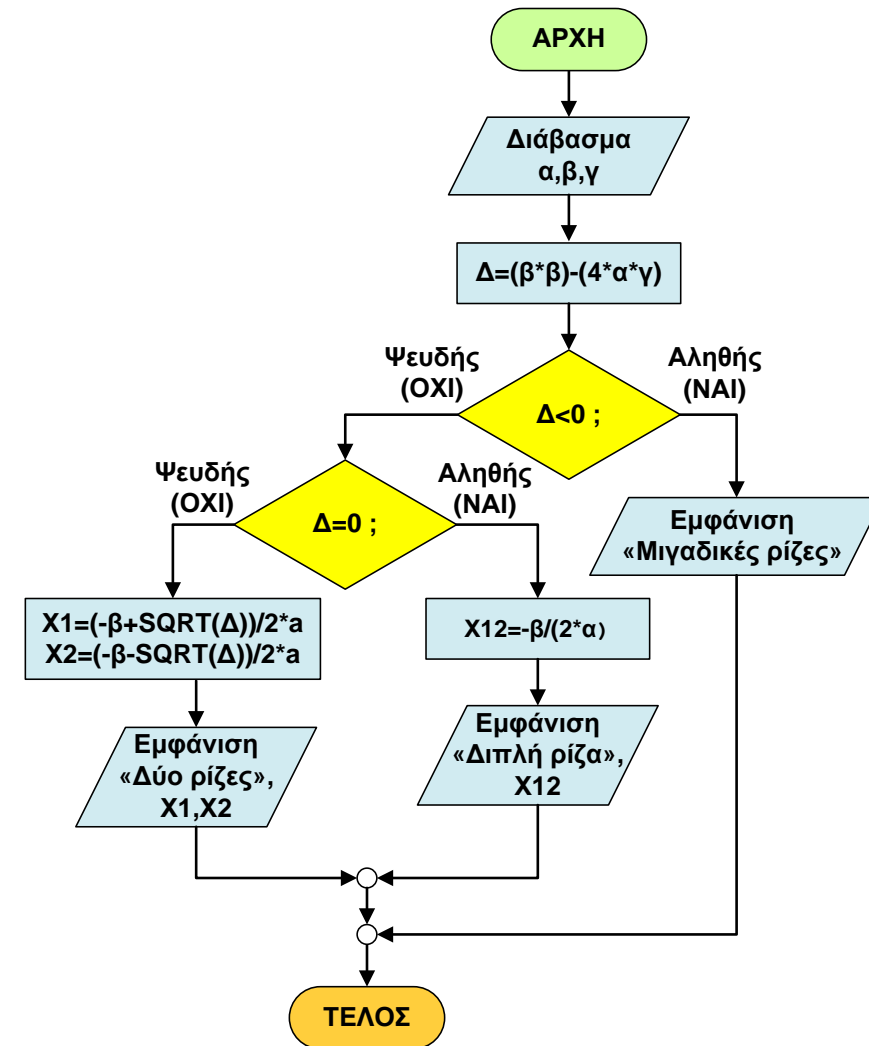
$$X1 = (-\beta + \text{SQRT}(\Delta)) / 2 * \alpha$$

$$X2 = (-\beta - \text{SQRT}(\Delta)) / 2 * \alpha$$

Εμφάνιση «Δύο ρίζες»

Εμφάνιση $X1, X2$

ΤΕΛΟΣ



Επίλυση τριωνύμου (3)

Κώδικας γλώσσας C

```
#include <stdio.h>
int main()
{
    float a,b,c,x12,x1,x2,D;
    printf("a="); scanf("%f",&a);
    printf("b="); scanf("%f",&b);
    printf("c="); scanf("%f",&c);
    D=(b*b)-4*a*c;
    if (D<0)
        printf("Migadikes rizes");
    else
        if (D==0)
            {
                x12=-b/(2*a);
                printf("dipli riza,x12=%.2f" ,x12);
            }
        else
            {
                x1=(-b+sqrt(D))/2*a;
                x2=(-b-sqrt(D))/2*a;
                printf("dyo rizes, x1=%.2f,
                x2=%.2f",x1,x2);
            }
    return 0;
}
```

Κώδικας γλώσσας Python

```
import math
a=float(input("a="))
b=float(input("b="))
c=float(input("c="))
D=(b*b)-4*a*c
if (D<0):
    print("Migadikes rizes")
else:
    if (D==0):
        x12=-b/(2*a)
        printf("dipli riza, x12=%.2f" %x12)
    else:
        x1=(-b+math.sqrt(D))/2*a
        x2=(-b-math.sqrt(D))/2*a
        print("dyo rizes, x1=%.2f, x2=%.2f" % (x1,x2))
```

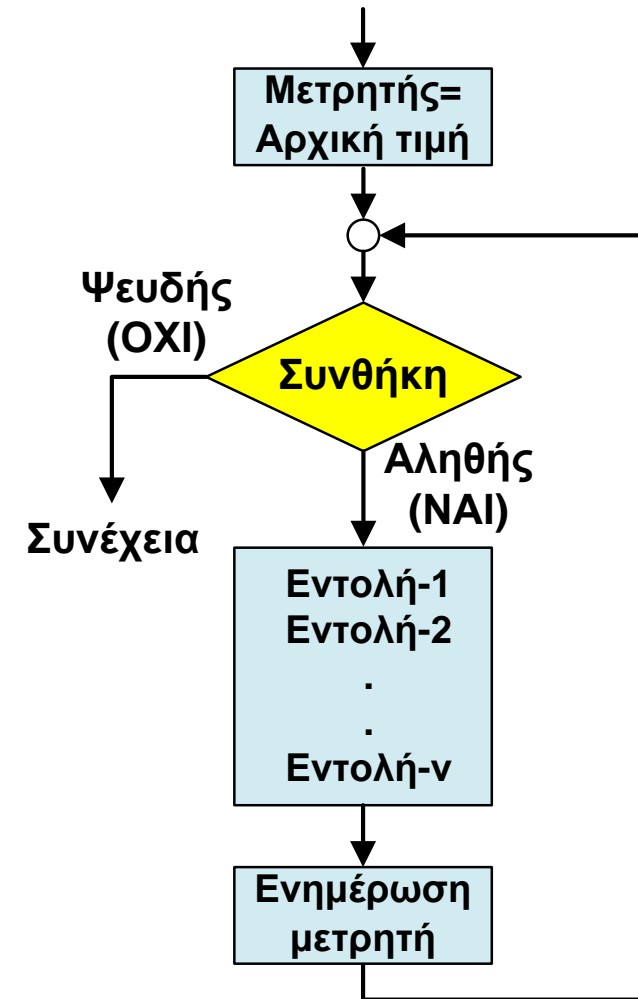


Εφαρμογή του δομημένου προγραμματισμού (4)

Δομή επανάληψης (1)

Ψευδοκώδικας

Μετρητής=αρχική τιμή
ΟΣΟ (συνθήκη) αληθής ΕΠΑΝΕΛΑΒΕ
 Εντολή-1
 Εντολή-2
 .
 .
 Εντολή-ν
 Ενημέρωση μετρητή
ΤΕΛΟΣ-ΟΣΟ



Εφαρμογή του δομημένου προγραμματισμού (5)

Δομή επανάληψης (2)

Ψευδοκώδικας

Μετρητής=αρχική τιμή
ΕΠΑΝΕΛΑΒΕ

Εντολή-1

Εντολή-2

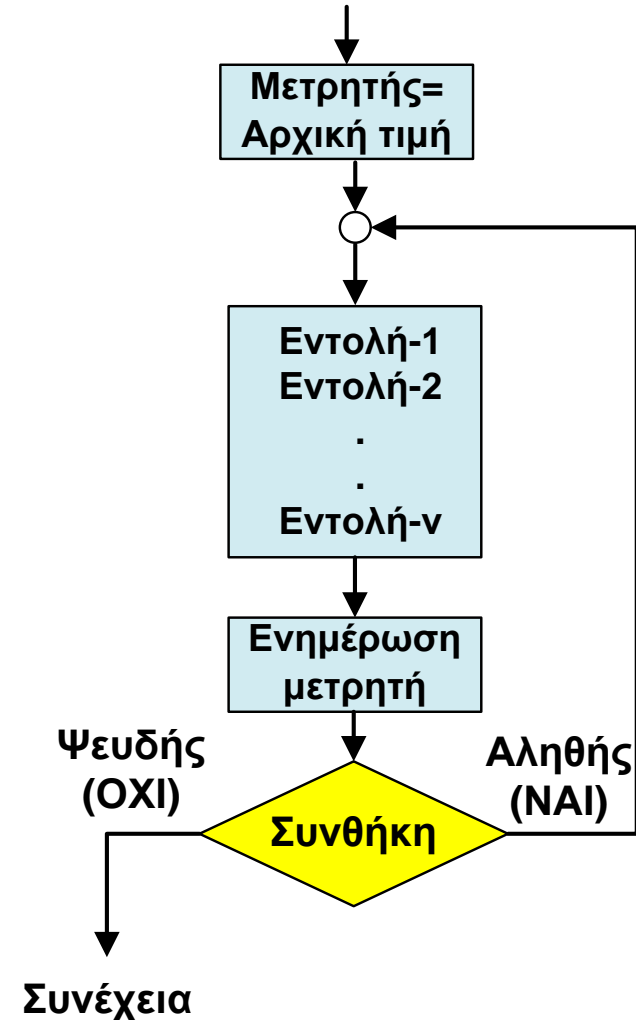
·

·

Εντολή-ν

Ενημέρωση μετρητή

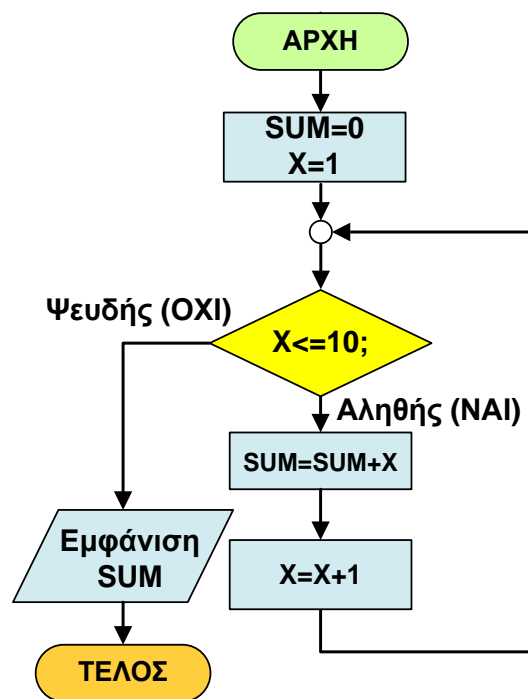
ΟΣΟ (συνθήκη) αληθής



Υπολογισμός αθροίσματος (1)

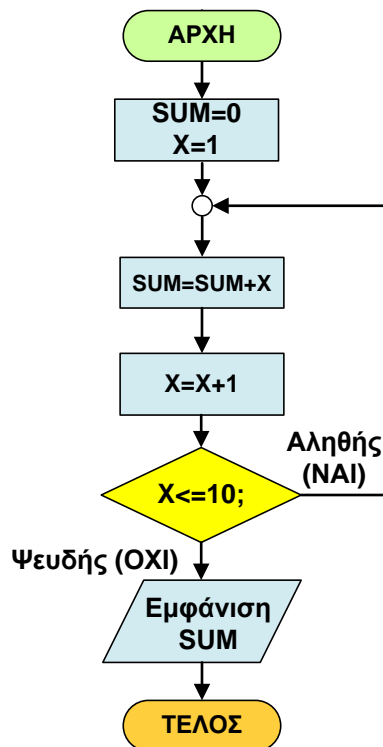
Ψευδοκώδικας While-do

```
X=1  
SUM=0  
ΟΣΟ (X<=10) ΕΠΑΝΕΛΑΒΕ  
    SUM=SUM+X  
    X=X+1  
ΤΕΛΟΣ-ΟΣΟ  
Εμφάνιση SUM
```



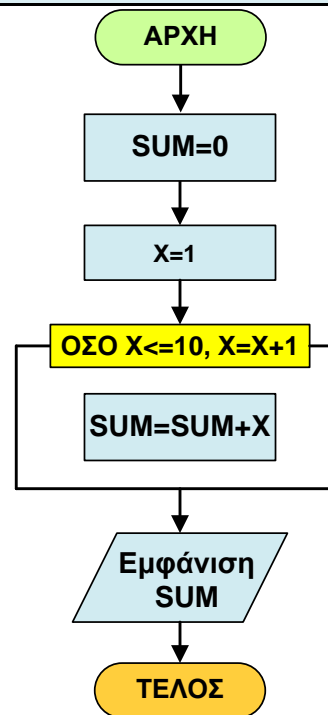
Ψευδοκώδικας do-While

```
X=1  
SUM=0  
ΕΠΑΝΕΛΑΒΕ  
    SUM=SUM+X  
    X=X+1  
ΟΣΟ (X<=10)  
Εμφάνιση SUM
```



Ψευδοκώδικας For

```
SUM=0  
Για X από 1 έως 10,  
βήμα 1  
    SUM=SUM+X  
Τέλος-Για  
Εμφάνιση SUM
```



Υπολογισμός αθροίσματος (2)

Κώδικας C

```
#include <stdio.h>
int main()
{
    int X,SUM;
    SUM=0;X=1;
    while (X<=10)
    {
        SUM=SUM+X;
        X=X+1;
    }
    printf("SUM=%d",SUM);
    return 0;
}
```

Κώδικας C

```
#include <stdio.h>
int main()
{
    int X,SUM;
    SUM=0;X=1;
    do
    {
        SUM=SUM+X;
        X=X+1;
    }
    while (X<=10);
    printf("SUM=%d",SUM);
    return 0;
}
```

Κώδικας C

```
#include <stdio.h>
int main()
{
    int X,SUM;
    SUM=0;
    for (X=1;X<=10;X++)
        SUM=SUM+X;
    printf("SUM=%d",SUM);
    return 0;
}
```

Στοιχειώδης αριθμομηχανή

Κώδικας C

```
#include <stdio.h>
int main()
{
    float a,b,ap=0;
    short int error;
    unsigned char p="";
    do
    {
        error=0;
        printf("**[ calculator ]** (A=999 for exit)\n");
        printf("A=");scanf("%f",&a);
        if (a==999) return 0;
        printf("B=");scanf("%f",&b);
        printf("Enter + - * or / :");
        scanf("%s",&p);
        if (p=='+') ap=a+b;
        if (p=='-') ap=a-b;
        if (p=='*') ap=a*b;
        if (p=='/')
            if (b==0)
            {
                printf("Error division by zero!!\n");
                error=1;
            }
        else ap=a/b;
        if (!error) printf("%.2f %c %.2f %.2f\n\n", a,p,b,ap);
    }
    while (1);
    return 0;
}
```



Εισπράξεις διοδίων (1)

Ψευδοκώδικας

ΑΡΧΗ

TIMH_ΦΟΡΤΗΓΟΥ=12.35;

TIMH_ΕΠΙΒΑΤΙΚΟΥ=3.85;

TIMH_ΔΙΚΥΚΛΟ=1.05;

ΣΥΝΟΛΟ_ΦΟΡΤΗΓΩΝ=0

ΣΥΝΟΛΟ_ΕΠΙΒΑΤΙΚΩΝ=0

ΣΥΝΟΛΟ_ΔΙΚΥΚΛΩΝ=0

ΣΥΝΟΛΟ=0

ΕΠΑΝΕΛΑΒΕ

Διάβασμα επιλογής (ΕΠ)

Αν ΕΠ=1 ΤΟΤΕ

Εμφάνιση τιμής TIMH_ΦΟΡΤΗΓΟΥ

ΣΥΝΟΛΟ_ΦΟΡΤΗΓΩΝ= ΣΥΝΟΛΟ_ΦΟΡΤΗΓΩΝ+TIMH_ΦΟΡΤΗΓΟΥ

Τέλος-Αν

Αν ΕΠ=2 ΤΟΤΕ

Εμφάνιση TIMH_ ΕΠΙΒΑΤΙΚΟΥ

ΣΥΝΟΛΟ_ ΕΠΙΒΑΤΙΚΩΝ = ΣΥΝΟΛΟ_ ΕΠΙΒΑΤΙΚΩΝ +TIMH_ ΕΠΙΒΑΤΙΚΟΥ

Τέλος-Αν

Αν ΕΠ=3 ΤΟΤΕ

Εμφάνιση TIMH_ ΔΙΚΥΚΛΟΥ

ΣΥΝΟΛΟ_ ΔΙΚΥΚΛΩΝ = ΣΥΝΟΛΟ_ ΔΙΚΥΚΛΩΝ +TIMH_ ΔΙΚΥΚΛΟΥ

Τέλος-Αν

ΟΣΟ ΕΠ<>0

ΣΥΝΟΛΟ= ΣΥΝΟΛΟ_ΦΟΡΤΗΓΩΝ+ ΣΥΝΟΛΟ_ ΕΠΙΒΑΤΙΚΩΝ+ ΣΥΝΟΛΟ_ ΔΙΚΥΚΛΩΝ

Εμφάνιση ΣΥΝΟΛΟ

Εμφάνιση ΣΥΝΟΛΟ_ΦΟΡΤΗΓΩΝ

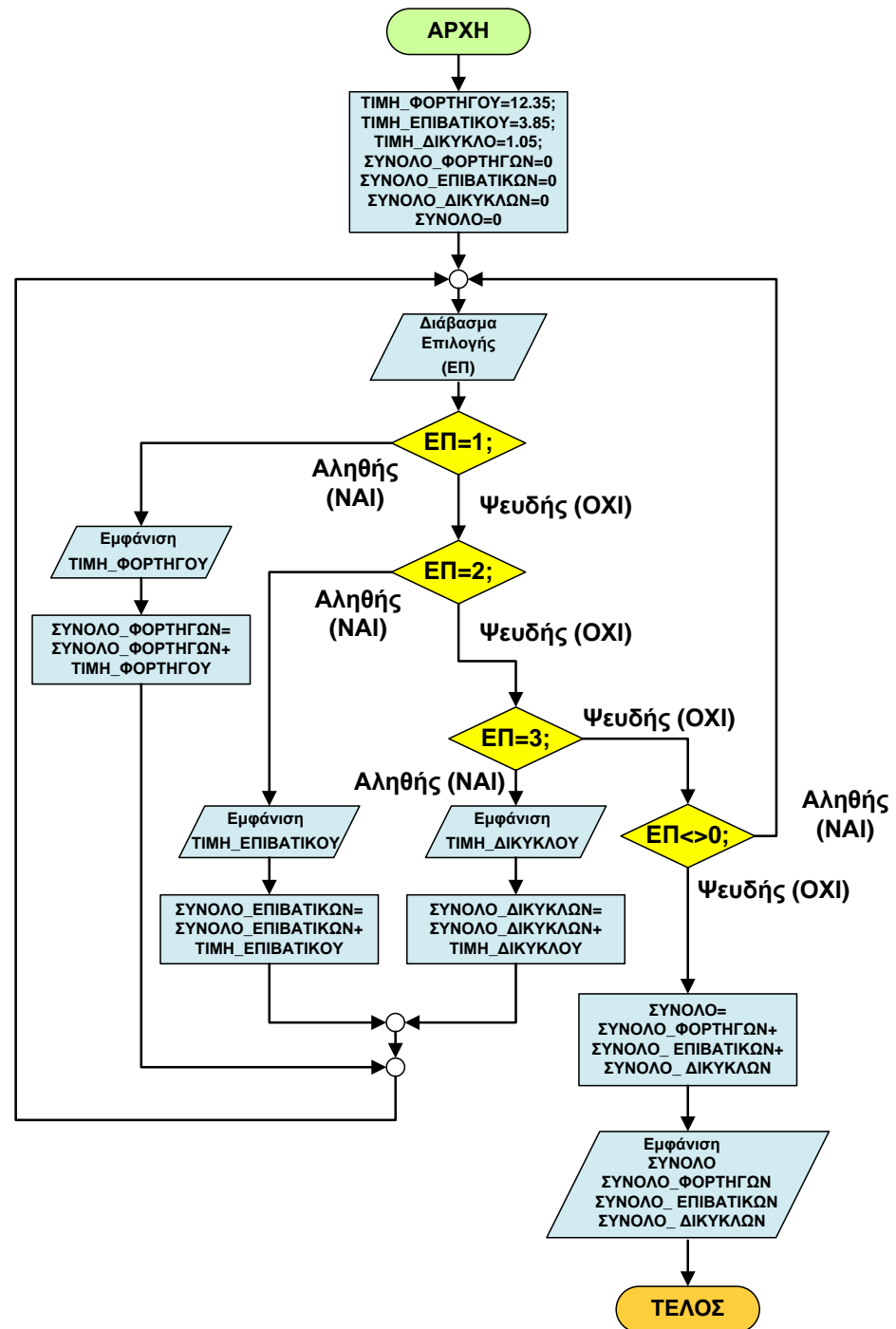
Εμφάνιση ΣΥΝΟΛΟ_ ΕΠΙΒΑΤΙΚΩΝ

Εμφάνιση ΣΥΝΟΛΟ_ ΔΙΚΥΚΛΩΝ

ΤΕΛΟΣ



Εισπράξεις διοδίων (2)



Κώδικας C

```
#include <stdio.h>
int main()
{
    const float fortigo=12.35;
    const float epivatiko=3.85;
    const float dikyklo=1.05;
    float S_fortiga=0, S_epivatika=0, S_dikykla=0, SUM=0;
    int ep;
    do
    {
        printf("*****\n");
        printf("(1) FORTHGO, (2) EPIVATIKO, (3) DIKYKLO, (0)
EXODOS :");
        scanf("%d",&ep);
        switch (ep)
        {
            case 1:
                {
                    printf("\n*** FORTHGO=%.2f Euros ***\n",fortigo);
                    S_fortiga+=fortigo;
                    break;
                }
            case 2:
                {
                    printf("\n*** EPIVATIKO=%.2f Euros ***\n",epivatiko);
                    S_epivatika+=epivatiko;
                    break;
                }
            case 3:
                {
                    printf("\n*** DIKYKLO=%.2f Euros ***\n",dikyklo);
                    S_dikykla+=dikyklo;
                    break;
                }
        }
    }
    while (ep!=0);
```

1

```
SUM=S_fortiga+S_epivatika+S_dikykla;
printf("-----\n");
printf("EISPRAXEIS=%.2f Euros \n",SUM);
printf("-----\n");
printf("FORTHGA=%.2f Euros \n",S_fortiga);
printf("EPIVATIKA=%.2f Euros \n",S_epivatika);
printf("DIKYKLA=%.2f Euros \n",S_dikykla);
printf("-----\n");

return 0;
}
```

2

Οργάνωση με Συναρτήσεις

Συναρτήσεις ως απλά υποπρογράμματα (1)

Έστω ότι θα αναπτύξουμε πρόγραμμα για τον ακόλουθο υπολογισμό:

$$f(x) = \frac{3x + 2}{x^3 - 1}$$

Ψευδοκώδικας

ΑΡΧΗ

Fx=0

Διάβασμα X

P=X³-1

Αν P≠0 ΤΟΤΕ

Fx=((3*X)+2)/P

Διαφορετικά

Εμφάνιση «Σφάλμα, διαίρεση με μηδέν, αγνοήστε την τιμή της Fx»

Εμφάνιση Fx

ΤΕΛΟΣ



Οργάνωση με Συναρτήσεις

Συναρτήσεις ως απλά υποπρογράμματα (2)

Χωρίς υποπρογράμματα

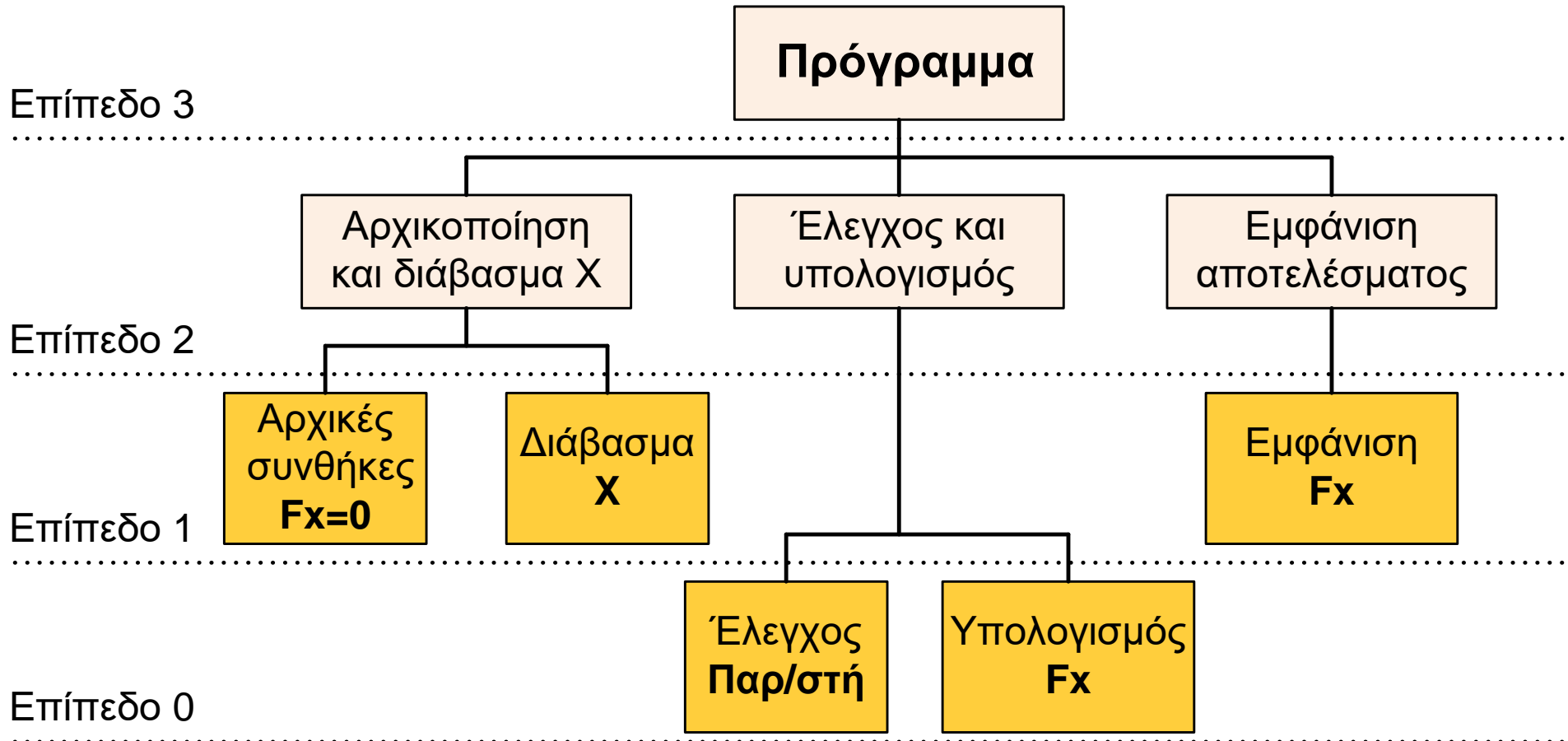
```
#include <stdio.h>
int main()
{
    float x,P,Fx=0;
    printf("X="); scanf("%f",&x);
    P=(x*x*x)-1;
    if (P!=0)
        Fx=((3*x)+2)/P;
    else
        printf("Error division by zero, Ignore f(x) result\n");
    printf("f(x)=%.2f\n",Fx);
    return 0;
}
```



Οργάνωση με Συναρτήσεις

Συναρτήσεις ως απλά υποπρογράμματα (3)

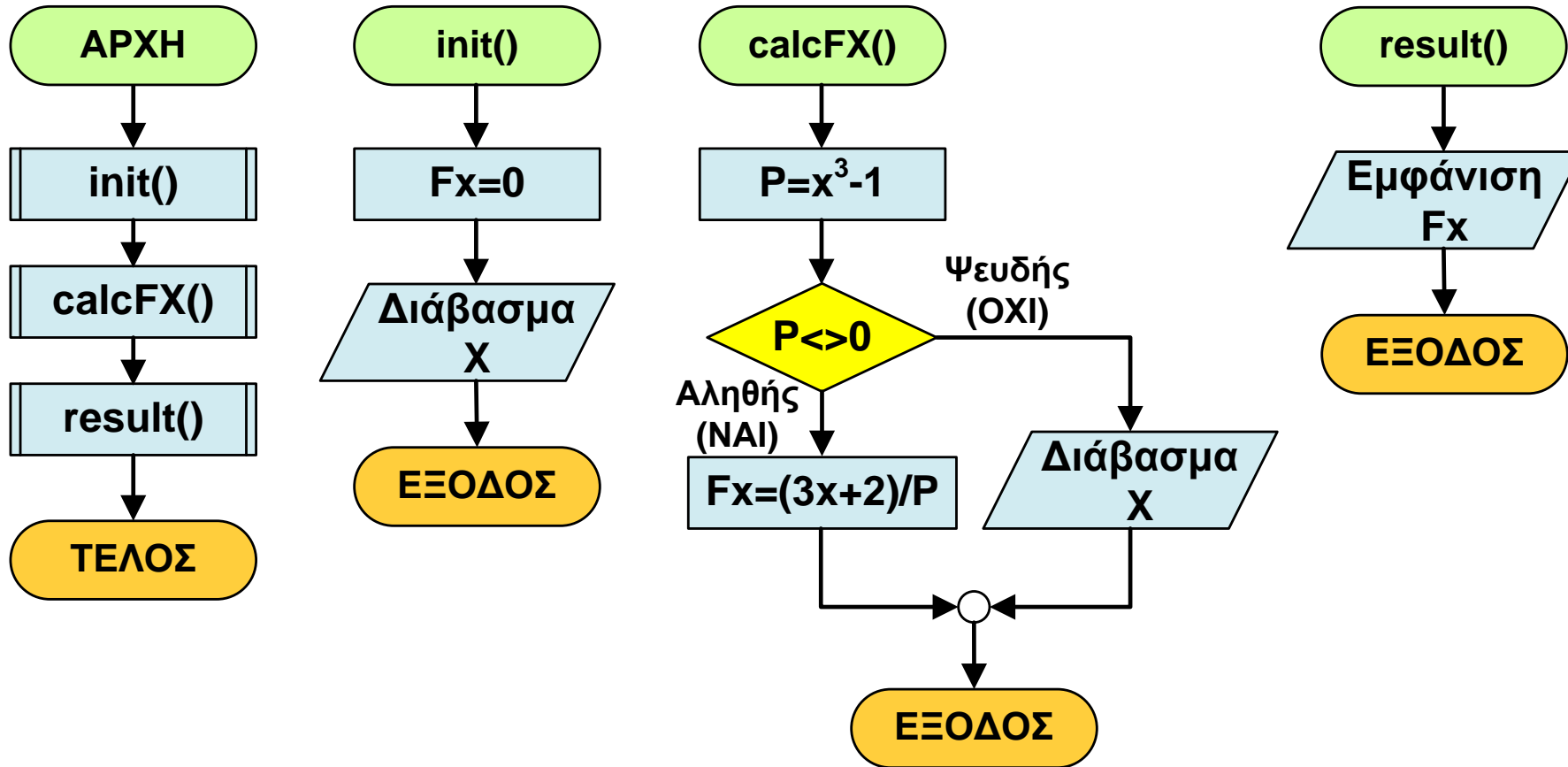
Διάσπαση του προβλήματος σε απλούστερα



Οργάνωση με Συναρτήσεις

Συναρτήσεις ως απλά υποπρογράμματα (4)

Υλοποίηση με συναρτήσεις-υποπρογράμματα



Οργάνωση με Συναρτήσεις

Συναρτήσεις ως απλά υποπρογράμματα (5)

Υλοποίηση με συναρτήσεις-υποπρογράμματα

Κώδικας C

```
#include <stdio.h>
void init();
void calcFX();
void result();
float x,P,Fx;
int main()
{
    init();
    calcFX();
    result();
    return 0;
}
void init()
{
    Fx=0;
    printf("X="); scanf("%f",&x);
}
void calcFX()
{
    P=(x*x*x)-1;
    if (P!=0)
        Fx=((3*x)+2)/P;
    else
        printf("Error division by zero, Ignore f(x) result\n");
}
void result()
{
    printf("f(x)=%.2f\n",Fx);
}
```

Οργάνωση με Συναρτήσεις

Συναρτήσεις ως απλά υποπρογράμματα (6)

Υλοποίηση με συναρτήσεις- υποπρογράμματα

```
#include <stdio.h>
```

```
void init();  
void calcFX();  
void result();
```

```
float x,P,Fx;
```

```
int main()  
{  
    init();  
    calcFX();  
    result();  
    return 0;  
}
```

```
void init()  
{  
    Fx=0;  
    printf("X="); scanf("%f",&x);  
}
```

```
void calcFX()  
{  
    P=(x*x*x)-1;  
    if (P!=0)  
        Fx=((3*x)+2)/P;  
    else  
        printf("Error division by zero, Ignore f(x)  
result\n");  
}
```

```
void result()  
{  
    printf("f(x)=%.2f\n",Fx);  
}
```


Οργάνωση με Συναρτήσεις

Συναρτήσεις ως απλά υποπρογράμματα (7)

Παραμετροποίηση

Συνάρτηση **ΠΡΟΣΘΕΣΗ()**

$X=1+3$

Επίστρεψε X ή εμφάνιση X

Τέλος-Συνάρτηση

Επιστρέφει πάντα το ίδιο αποτέλεσμα



Συνάρτηση **ΠΡΟΣΘΕΣΗ (A,B)**

$X=A+B$

Εμφάνιση X

Τέλος-Συνάρτηση

Παραδείγματα κλήσης

(α) $Z=\text{ΠΡΟΣΘΕΣΗ}(1,2)$ ή $Z=\text{ΠΡΟΣΘΕΣΗ}(x,y)$

(β) $Z=(4*5)+\text{ΠΡΟΣΘΕΣΗ}(1,2)$ ή $Z=(4*5)+\text{ΠΡΟΣΘΕΣΗ}(x,y)$

Οργάνωση με Συναρτήσεις

Συναρτήσεις ως απλά υποπρογράμματα (8)

Παραδείγματα υλοποίησης

```
int ADD(int a, int b)
{
    int x;
    x=a+b;
    return x;
}
```

```
int ADD(int a, int b)
{
    return a+b;
}
```

Κλήση
συνάρτησης

```
x = ADD(1,2);
```

Συνάρτηση

```
int ADD(int a, int b)
{
    return a+b;
}
```

Ολοκλήρωση κεφαλαίου
Δείτε τις ασκήσεις από το βιβλίο

