

Απλό πρόγραμμα

```
class hello
{
    public static void main (String args[])
    { System.out.println("Hello World!!!"); }
}
```

Σώζουμε ως *hello.java* . Σε *dos* πηγαίνουμε στο φάκελο και γράφουμε *javac hello.java*
Δημιουργείται το *hello.class*. Γράφουμε *java hello* και εκτελείται το πρόγραμμα.

Εισαγωγή βιβλιοθηκών

```
import java.applet.Applet;
import java.awt.Graphics;
ή import java.*.*
```

Τύποι Δεδομένων

Byte 8 bit
Short 16 bit
Int 32 bit
Long 64 bit
Float 4 byte *Ακρίβεια: 7 ψηφία*
Double 8 byte *Ακρίβεια: 18 ψηφία*

x = 10	x = 10	x = x+y
y = x++	y = ++x	x += y, x /= y
y = 10, x = 11	y = 11, x = 11	x -= y, x *= y

Λογικές Τιμές

```
boolean state;
state = true;
ή boolean state = true;
```

Τελεστές σύγκρισης

= *Ισον*
!= *Διάφορο*
< *Μικρότερο* **<=** *Μικρότερο ή ίσο*
> *Μεγαλύτερο* **>=** *Μεγαλύτερο ή ίσο*

Λογικοί Τελεστές

AND &&
OR || (τουλ.μια αληθής)
NOT !
XOR ^ (μια και μόνο μια αληθής)

Τελεστές bitwise

bitwise AND &
bitwise OR ^
συμπλήρωμα ~
bitwise XOR |

Ακολουθίες διαφυγής

\b backspace	\r carriage return
\t Tab	\ " "
\n line feed	\ ' ', \\ \
\f form feed	\uxxxx Unicode character

Μετατροπή τύπου δεδομένων

```
double x=3,31
y = (int)x; → y = 3
```

Συναρτήσεις

abs(var) *Absolute Value*

ceil(var) *Στρογ. στο μεγαλύτερο ακέραιο*

cos(var) *cosine*

exp(var) e^x

floor(var) *Στρογ. στο μικρότερο ακέραιο*

log(var) *log (base e)*

max(var1, var2) *Μέγιστο*

min(var1, var2) *Ελάχιστο*

pow(var1, var2) $var1^{var2}$

round(var) *Στρογ. στον κοντινότερο ακέραιο*

tan(var) *tangent*

random() *random $0 \leq x < 1$*

sin(var) *sinus*

sqrt(var) *square root*

Τις χρησιμοποιούμε γράφοντας : Math.abs(var) π.χ. math.pi

Μετατροπή αλφαριθμητικού σε αριθμό

int i = Integer.valueOf("str1").intValue();

long l = Long.valueOf("str1").longValue();

Εισαγωγή δεδομένων από πληκτρολόγιο

int x;

x = Integer.valueOf(args[0]).intValue();

Η δομή if

if (condition)

{

...;

...;

}

Ο όρος else

if (condition)

{ ...; }

else

{ ...; }

Τριαδικός τελεστής ?:

if (a>b)

max=a; $\rightarrow \rightarrow$ max = a>b ? a:b;

else $\rightarrow \rightarrow$

max = b;

Δομή Switch

Switch (var)

{

case 1:

...;

break;

...

case n:

...;

break;

...

default:

...;

break;

}

Ο βρόχος for

for (c=0; c<10; c++)

{ ...; }

Ο βρόχος while

while(condition)

{ ...; ...; }

...

Ο βρόχος do-while

do

{ ...; }

while (condition);

Η εντολή continue

if (condition)

continue;

τότε δεν εκτελείται το περιεχόμενο του βρόχου

Το break το χρησιμοποιούμε για να βγούμε από ένα βρόχο

Μέθοδος

Επιστρέφει αποτέλεσμα ή στο κυρίως πρόγραμμα με την εντολή return ...;

Δήλωση ενός πίνακα

float []array1;

int array1[];

Καταχώρηση πίνακα

```
Array1 = new int[5]
```

Απόδοση τιμών

```
C[0]= 3;
```

```
...;
```

```
system.out.println (C[2])
```

```
int num[]={1,2,3,4,5,6,7};
```

```
system.out.println (num.length)
```

```
float []numbers1=numbers;
```

```
M = new int [4][5];
```

Ταξινόμηση

Ταξινόμηση με επιλογή

Μέθοδος της φουσαλίδας

Αλφαριθμητικές μεταβλητές

```
str1 + str2
```

Ισότητα strings

```
str1.equals(str2)
```

```
str1.equalsIgnoreCase(str2)
```

```
str2=str1.toLowerCase()
```

```
str2=str1.toUpperCase()
```

```
String substring(int start, int end);
```

```
str1.length Μήκος
```

```
str1.charAt(4);
```

```
str2=str1.replace('a','b')
```

Εντοπισμός χαρακτήρων

```
int index=0;
```

```
index=str1.indexOf('a');
```

```
index=str1.lastIndexOf('a');
```

```
int index=0;
```

```
index=str1.indexOf('a', 4);
```

```
int index=0;
```

```
index=str1.lastIndexOf('a', 4);
```

```
int x=12;
```

```
String str1="123"
```

```
string str1=String.valueOf(x); int x=integer.parseInt(str1);
```

```
StringBuffer sbf=new StringBuffer(40);
```

```
sbf.length(); sbf.capacity();
```

```
sbf.append("Nikolas"); sbf.insert(8,'w');
```

```
sbf.setCharAt(9,'a'); sbf.reverse();
```

```
StringBuffer sbf=new StringBuffer("Nikolas")
```

```
String str1=sbf.toString();
```

Ορισμός μιας κλάσης

```
Class MyClass
```

```
{  
}  
}
```

Για να ορίσουμε μια μεταβλητή κλάσης που δημιουργείται ακόμα και αν δεν έχει δημιουργηθεί κανένα αντικείμενο ακόμα, γράφουμε `static int count = 0;`

Για να ορίσουμε μια μεταβλητή στιγμιότυπου που δημιουργείται με κάθε αντικείμενο γράφουμε `int x;`
Αντίστοιχα και για τις μεθόδους κλάσης και στιγμιότυπου.

Αν μια μέθοδος δεν επιστρέφει τιμή τότε γράφουμε **void** αλλιώς τον τύπο της τιμής που επιστρέφει και βάζουμε **return**.

Η μεταβλητή **this** αναφέρεται στο τρέχον αντικείμενο π.χ. `this.radius`

Αναφορά : `Circle troxos = new Circle (1,2,3);`

Πακέτα : `import java.util.Date; Date thisDate = new Date();`
ή `java.util.Date thisDate = new java.util.Date();`

Τυπικά Πακέτα Java

java.lang	java.util	java.sql
java.io	java.awt.peer	java.beans
java.awt (GUI)	java.security	java.rmi Απομακρυσμένες Μέθοδοι
java.awt.event	java.awt.image	javax.swing Νέο GUI
java.applet	java.net	

Δήλωση κλάσης σε πακέτο `package Geometry; public class Circle`
`Import geometry.Circle;`

Δυνατότητα Προσπέλασης Από την ίδια κλάση	public NAI	protected NAI	private NAI	Χωρίς προσδιοριστή NAI
Από τις κλάσεις του ίδιου πακέτου	NAI	NAI	OXI	NAI
Από οποιαδήποτε κλάση εκτός του πακέτου όπου ανήκει η κλάση	NAI	OXI	OXI	OXI
Από μια υποκλάση του ίδιου πακέτου	NAI	NAI	OXI	NAI
Από μια υποκλάση έξω από το πακέτο όπου ανήκει η κλάση	NAI	NAI	OXI	OXI

Για να δηλώσουμε ότι η κλάση `Car` είναι υποκλάση της `motorVehicle` γράφουμε :

```
Class Car extends motorVehicle
```

Η λέξη `super` καλεί μέλη της υπερκλάσης. Η `super()` καλεί τον κατα/στή και η `super.counter()` την `counter`.

Μια τελική κλάση δεν μπορεί να είναι υπερκλάση και τη δηλώνουμε : `public final class MyClass`

Ομοίως και για μια μέθοδο που τότε δεν αλλάζει και για μια μεταβλητή που καθίσταται σταθερά.

Μια αφηρημένη κλάση τη δηλώνουμε με `public abstract getname();`

```
Public interface LengthConversion {} Ορίζουμε μια διασύνδεση.
```

```
Public class LengthMethods implements LengthConversion {} Καλούμε μια διασύνδεση.
```

*Αν μια κλάση που υλοποιεί τις μεθόδους μιας διασύνδεσης δεν τις υλοποιεί όλες, πρέπει να δηλωθεί ως αφηρημένη.

Applets

```
Import java.applet.Applet;  
Import java.awt.Graphics ;
```

```
Public class Hello extends Applet  
{  
String name1  
Public void init()  
{  
name = getParameter("message");  
name= "Hello " + name;  
}  
public void paint (Graphics g)  
{g.drawString (name, 30, 40);}  
}
```

Κώδικας HTML

```
<HTML>  
<HEAD>  
<TITLE> mplamplampla </TITLE>  
</HEAD>  
<BODY>  
<applet code = "Hello.class" width = 300 height = 200>  
<param name = "message" value = "Someone">  
</applet>  
</BODY>  
</HTML>
```

Άλλες εντολές στα applet : start(), stop(), destroy(), paint(Graphics g), getAppletInfo(),String [][]
getParameterInfo()

Exceptions

Η εξαίρεση είναι ένα αντικείμενο της υποκλάσης java.lang.Throwable και δημιουργείται όταν εμφανίζεται στο πρόγραμμα μια μη κανονική κατάσταση.

Για να δημιουργήσουμε μια δική μας εξαίρεση :

```
class MyException extends Exception  
{  
MyException (String message)  
{super(message);}  
}
```

Δήλωση ότι η μέθοδος μπορεί να δημιουργήσει εξαίρεση :

```
double myMethod() throws EOFException, IOException  
{//κώδικας}
```

Για να συλλάβουμε μια εξαίρεση, βάζουμε τον επικίνδυνο κώδικα σε ένα try block try {} και ακολουθεί ένα catch block : catch(τύπος_εξαίρεσης e) {}

Ακόμα μπορούμε να βάλουμε ένα finally block στο τέλος του κώδικα που "συμμαζεύει" τον κώδικα και εκτελείται πάντα εκτός και αν τερματιστεί το πρόγραμμα μέσα στο try ή το catch.

Αν θέλουμε, μπορούμε να δημιουργήσουμε μια δική μας εξαίρεση :

π.χ.

```
catch (MyZeroDivideException zde)
{System.err.println("Exception1 : " + zde.getMessage());}
```

```
public class MyZeroDivideException extends ArithmeticException
{ public MyZeroDivideException()
  {super ("Diaresi me to 0");}}
```

Input / Output

Για να διαβάσουμε ή να γράψουμε πληροφορίες, χρησιμοποιούμε ένα Stream.

Υπάρχουν 2 είδη. Τα **character Streams** που είναι 16-bit και είναι τα **Reader** και **Writer** και τα **Byte Streams** που είναι 8-bit και είναι τα **Input Stream** και **Output Stream**.

Οι εντολές είναι :

int read()	int write(int c)
int read(char cbuf[])	int write(char cbuf[])
int read(char cbuf,int offset,intlength)	int write(char cbuf,int offset,intlength)

και *αντίστοιχα* έχουμε αντί για char με byte.

```
import java.io.*;
public class Copy {
    public static void main(String[] args) throws IOException {
        File inputFile = new File("farrago.txt");
        File outputFile = new File("outagain.txt");
        FileReader in = new FileReader(inputFile);
        FileWriter out = new FileWriter(outputFile);
        int c;
        while ((c = in.read()) != -1)
            out.write(c);
        in.close();
        out.close();    }}
```

Αντί για FileReader μπορούμε να έχουμε και FileInputStream κλπ.

Για να κάνουμε wrap ένα Stream

```
FileOutputStream fout = new FileOutputStream ("C:\...\test.txt");
BufferedOutputStream bos = new BufferedOutputStream (fout);
DataOutputStream d = new DataOutputStream (bos);
```