

2. ΕΙΣΑΓΩΓΗ ΣΤΗ JAVA

Η γλώσσα προγραμματισμού Java

Βασικό συντακτικό, ορισμός μεταβλητών,
έλεγχος ροής

Η εξέλιξη των γλωσσών προγραμματισμού

- Η εξέλιξη των γλωσσών προγραμματισμού είναι μια διαδικασία **αφαίρεσης**
 - Στην αρχή ένα πρόγραμμα ήταν μια σειρά από εντολές σε γλώσσα μηχανής.
 - Με τον **Διαδικασιακό Προγραμματισμό (procedural programming)**, ένα πρόγραμμα έγινε μια συλλογή από **διαδικασίες** που η μία καλεί την άλλη.
 - Στον **Συναρτησιακό Προγραμματισμό (functional programming)** ένα πρόγραμμα είναι μια συλλογή από **συναρτήσεις** όπου η μία εφαρμόζεται πάνω στην άλλη.
 - Στον **Λογικό Προγραμματισμό (logic programming)** ένα πρόγραμμα είναι μια συλλογή από **κανόνες** και **γεγονότα**.
 - Στον **Αντικειμενοστραφή Προγραμματισμό (object oriented programming)** ένα πρόγραμμα είναι μια συλλογή από **κλάσεις** και **αντικείμενα** όπου το ένα μιλάει με το άλλο

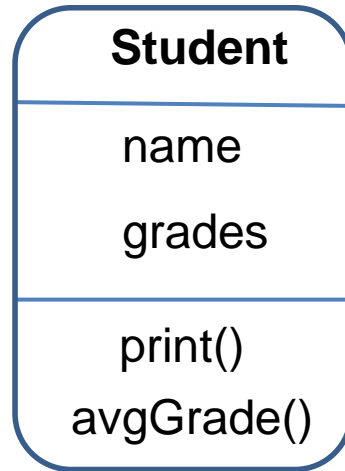
Αντικειμενοστραφής Προγραμματισμός

- Οι πέντε αρχές του Allan Kay:
 - Τα πάντα είναι **αντικείμενα**.
 - Ένα πρόγραμμα είναι μια **συλλογή** από **αντικείμενα** όπου το ένα λέει στο άλλο τι να κάνει.
 - Κάθε αντικείμενο έχει δικιά του **μνήμη** και αποτελείται από **άλλα αντικείμενα**.
 - Κάθε αντικείμενο έχει ένα συγκεκριμένο **τύπο**.
 - Τύπος = **Κλάση**
 - Αντικείμενα του **ίδιου τύπου** μπορούν να δεχτούν **τα ίδια μηνύματα**
 - Δηλαδή έχουν τις **ίδιες λειτουργίες**

Κλάσεις και Αντικείμενα

Κλάση

Μια αφηρημένη περιγραφή μιας οντότητας



Όνομα κλάσης

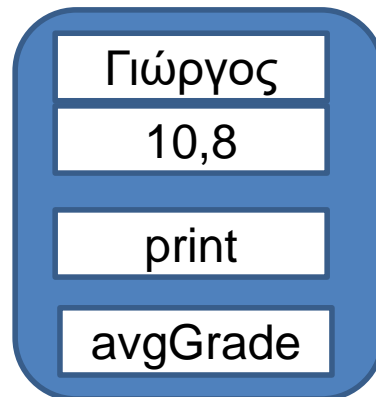
Πεδία κλάσης: Ιδιότητες/Χαρακτηριστικά

Μέθοδοι κλάσης: λειτουργίες

Αντικείμενα

Ένα συγκεκριμένο στιγμιότυπο της αφηρημένης κλάσης

Student **studentGeorge**:



Πρόσβαση στο αντικείμενο μόνο μέσω κλήσεων των μεθόδων:

`studentGeorge.print()`: Τυπώνει τις πληροφορίες για το αντικείμενο.

`studentGeorge.avgGrade()`: Υπολογίζει και επιστέφει τον μέσο όρο του φοιτητή

Σύντομη ιστορία του Αντικειμενοστραφούς Προγραμματισμού

- Η πρώτη γλώσσα που χρησιμοποίησε τις έννοιες της κλάσης και του αντικειμένου θεωρείται η **SIMULA** (1960s)
 - Γλώσσα για προσομοιώσεις συστημάτων
- Εμπνευσμένος από την SIMULA ο **Allan Kay** δημιούργησε στην HP την γλώσσα **SmallTalk** με στόχο μια γλώσσα που να υποστηρίζει γραφικά (1970s)
 - Ήταν αυτός που εισήγαγε την έννοια «**Αντικειμενοστραφής Προγραμματισμός**» (**Object Oriented Programming**)
 - Το 2003 βραβεύτηκε με το Turing Award
- Οι ιδέες του αντικειμενοστραφούς προγραμματισμού άρχισαν να εισάγονται σε πολλές υπάρχουσες ή νέες γλώσσες. Ο **Bjorn Stroustrup** δημιούργησε την **C++** (1980s)
- Η Sun δημιούργησε την γλώσσα **Java** η οποία βρίσκει εφαρμογή σε ανάπτυξη εφαρμογών στο διαδίκτυο (1990s)
 - Ακολούθησε η Microsoft με την .NET πλατφόρμα και τις γλώσσες **Visual Basic** και **C#**

Σύντομη ιστορία της Java

- Ο **Patrick Naughton** απειλεί την Sun ότι θα φύγει.
- Τον βάζουν σε μία ομάδα αποτελούμενη από τους **James Gosling** και **Mike Sheridan** για να σχεδιάσουν τον προγραμματισμό των έξυπνων συσκευών της επόμενης γενιάς.
 - The **Green project**.
- Ο Gosling συνειδητοποιεί ότι η C++ δεν είναι αρκετά αξιόπιστη για να δουλεύει σε συσκευές περιορισμένων δυνατοτήτων και με διάφορες αρχιτεκτονικές.
 - Δημιουργεί τη γλώσσα **Oak**
- Το 1992 η ομάδα κάνει ένα demo μιας συσκευής **PDA, *7 (star 7)**
 - Δημιουργείται η θυγατρική εταιρία **FirstPerson Inc**
- Η δημιουργία των έξυπνων συσκευών αποτυγχάνει και η ομάδα (μαζί με τον **Eric Schmidt**) επικεντρώνεται στην εφαρμογή της πλατφόρμας στο **Internet**.
 - Ο Naughton φτιάχνει τον **WebRunner browser** (μετα **HotJava**)
 - Η γλώσσα μετονομάζεται σε **Java** και το ενδιαφέρον επικεντρώνεται σε εφαρμογές που τρέχουν μέσα στον browser.
- Ο **Marc Andersen** ανακοινώνει ότι ο **Netscape browser** θα υποστηρίζει Java μικροεφαρμογές (applets)

Java

- Η Java είχε τους εξής στόχους:
 - "simple, object-oriented and familiar"
 - "robust and secure"
 - "architecture-neutral and portable"
 - "high performance"
 - "interpreted, threaded, and dynamic"

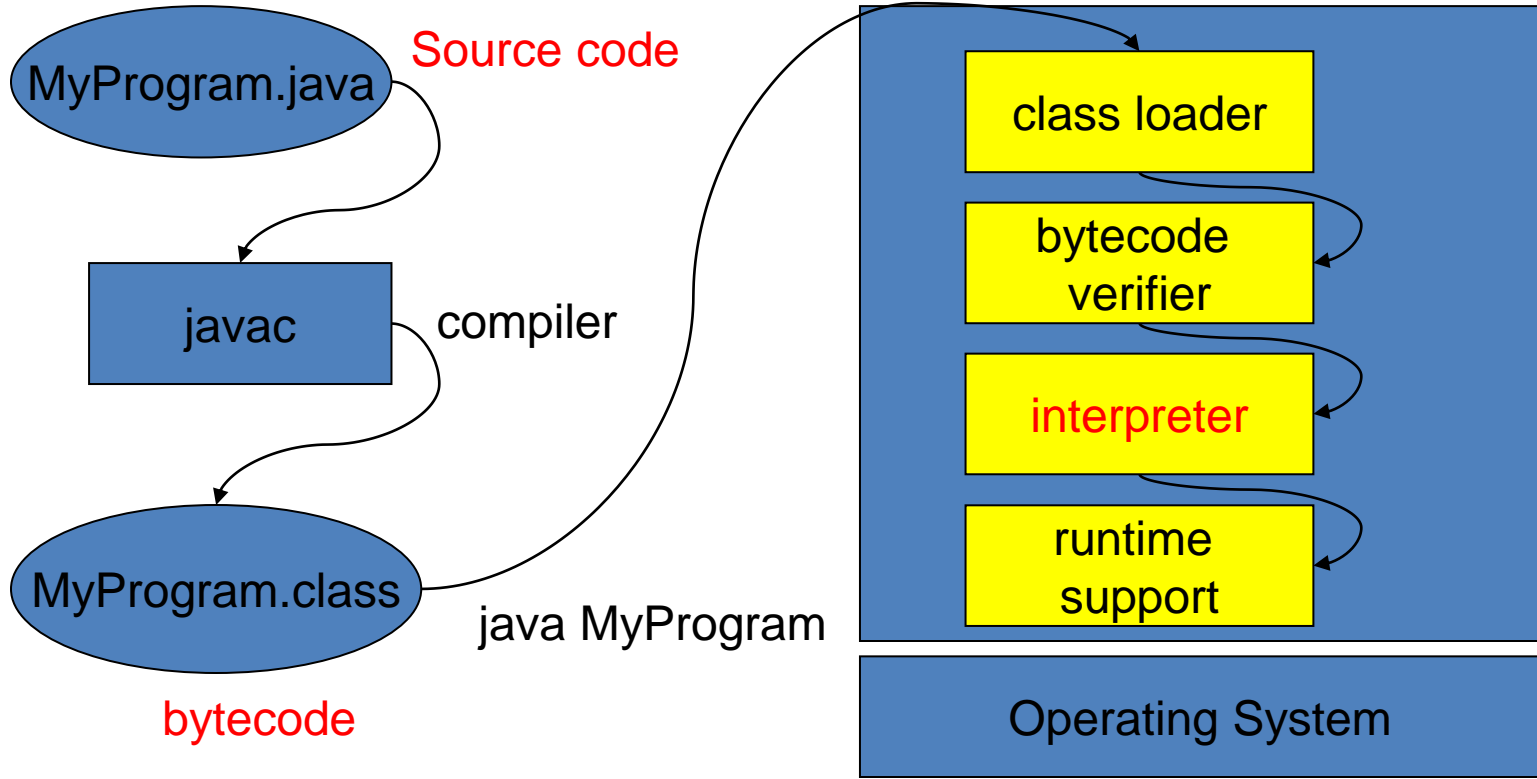
Java

- Η Java είχε τους εξής στόχους:
 - "simple, object-oriented and familiar"
 - "robust and secure"
 - "architecture-neutral and portable"
 - "high performance"
 - "interpreted, threaded, and dynamic"

“architecture-neutral and portable”

- Το μεγαλύτερο πλεονέκτημα της Java είναι η **μεταφερισιμότητα (portability)**: ο κώδικας μπορεί να τρέξει πάνω σε οποιαδήποτε πλατφόρμα.
 - **Write-Once-Run-Anywhere** μοντέλο, σε αντίθεση με το σύνηθες **Write-Once-Compile-Anywhere** μοντέλο.
- Αυτό επιτυγχάνεται δημιουργώντας ένα **ενδιάμεσο κώδικα (bytecode)** ο οποίος μετά τρέχει πάνω σε μια **εικονική μηχανή (Java Virtual Machine)** η οποία το μεταφράζει σε **γλώσσα μηχανής**.
 - Οι προγραμματιστές πλέον γράφουν κώδικα για την εικονική μηχανή, η οποία δημιουργείται **για οποιαδήποτε πλατφόρμα**.

Java Virtual Machine (JVM)

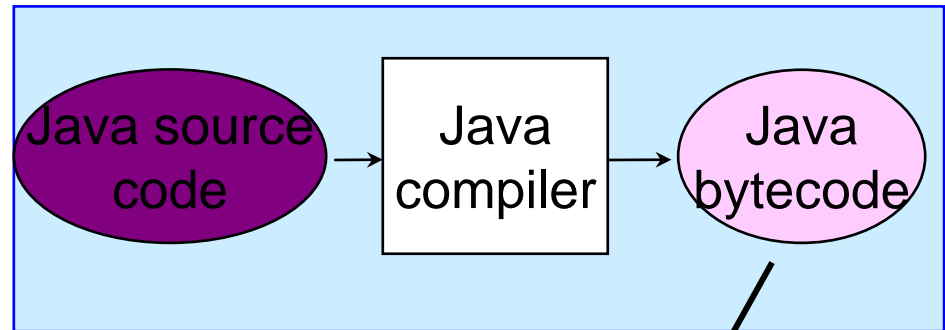


Java και το Internet

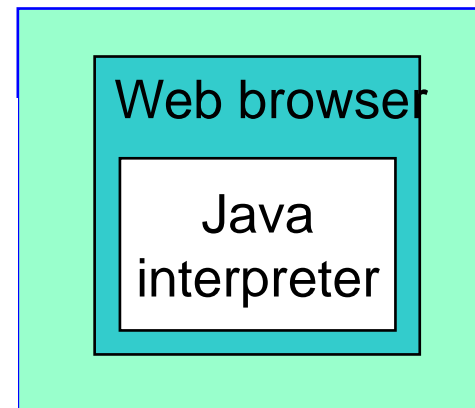
- Η προσέγγιση της Java είχε μεγάλη επιτυχία για **Web εφαρμογές**, όπου έχουμε ένα τεράστιο κατακευματισμένο **client-server** μοντέλο με πολλές διαφορετικές αρχιτεκτονικές
 - **Client-side programming**: Αντί να κάνει όλη τη δουλειά ο server για την δημιουργία της σελίδας κάποια από την επεξεργασία των δεδομένων γίνεται στη μηχανή του client.
 - **Web Applets**: κώδικας ο οποίος κατεβαίνει μαζί με τη Web σελίδα και τρέχει στη μηχανή του client. Είναι πολύ σημαντικό στην περίπτωση αυτή ο κώδικας να είναι portable.
 - **Server-side programming**: μία web σελίδα μπορεί να είναι το αποτέλεσμα ενός προγράμματος που συνδυάζει δυναμικά δεδομένα και είσοδο του χρήστη.
 - **Java Service Pages (JSPs)**: Η λύση της Java. Γίνεται compiled σε **servlets** και τρέχει στη μεριά του server.

Java Applets

- Το Web Browser software περιλαμβάνει ένα **JVM**
 - ◆ **Φορτώνει** τον java byte code από τον remote υπολογιστή
 - ◆ **Τρέχει** τοπικά το Java πρόγραμμα μέσα στο παράθυρο του Browser



Remote computer



Local computer



"simple, object-oriented and familiar"

- **Familiar:** Η Java είχε ως έμπνευση της την C++, και δανείζεται αρκετά από τα χαρακτηριστικά της.
- **Object-oriented:** Η Java είναι «**πιο αντικειμενοστραφής**» από την C++ η οποία προσπαθεί να μείνει συμβατή με την C
 - Στην Java **τα πάντα** είναι **αντικείμενα**
- **Simple:** Η Java δίνει λιγότερο έλεγχο στο χρήστη, αλλά κάνει τη ζωή του πιο εύκολη. Η **διαχείριση της μνήμης** γίνεται **αυτόματα**.
 - Η γλώσσα φροντίζει να κάνει πιο εύκολο και πιο σταθερό (robust) τον προγραμματισμό παρότι αυτό μπορεί να έχει αποτέλεσμα τα προγράμματα να γίνονται **πιο αργά**.

ΕΓΚΑΤΑΣΤΑΣΗ JAVA – ΧΡΗΣΗ ΤΕΡΜΑΤΙΚΟΥ – EDITOR

Java Installation

- Για να μπορείτε να μεταγλωττίσετε και να τρέξετε Java προγράμματα στον υπολογιστή σας θα πρέπει να **εγκαταστήσετε** το Java Development Kit (JDK).
 - Θα κάνετε download and install από τη σελίδα της Oracle.
 - Ψάξετε “**download java jdk**” στο Google
 - <https://www.oracle.com/java/technologies/javase-downloads.html>
- Υπάρχει περίπτωση μετά την εγκατάσταση να πρέπει να προσθέσετε το **path στο directory** στο οποίο εγκαταστάθηκε η Java στο **Path environmental variable**
 - Συνήθως αυτό γίνεται αυτόματα.



download java jdk



All Videos Images Books News More Settings Tools

About 19,600,000 results (0.59 seconds)

www.oracle.com > java > technologies > javase-downlo...

Java SE - Downloads | Oracle Technology Network | Oracle

Oracle **JDK** - Oracle Customers and ISVs targeting Oracle LTS releases: Oracle **JDK** is Oracle's supported **Java** SE version for customers and for developing, ...

Java SE Development Kit 8

Java SE Development Kit 8 - JDK 8 - Java JDK 8 - Checksum - ...

Java SE Development Kit 15

Thank you for downloading this release of the Java™ Platform ...

Java SE Development Kit 11

Java SE Development Kit 11 Downloads - Important Oracle ...

Java SE - Downloads

Oracle Customers and ISVs targeting Oracle LTS releases ...

Downloads

Java SE Development Kit 8 - JDK Download - JRE Download - ...

Java Runtime Environment

If you want to run Java programs, but not develop them ...

More results from oracle.com »

www.oracle.com > java > technologies > javase-jdk15-d...

Java SE Development Kit 15 - Downloads | Oracle India

Thank you for **downloading** this release of the **Java™** Platform, Standard Edition **Development Kit (JDK™)**. The **JDK** is a development environment for building ...

Developer: Oracle Corporation

www.oracle.com > java > technologies > jdk12-archive-...

Java Archive Downloads - Java SE 12 - Oracle

The **JDK** includes tools useful for developing and testing programs written in the **Java** programming language and running on the **Java™** platform. Important ...

People also ask

How do I download Java JDK?

Java SE Downloads

Java Platform, Standard Edition

Java SE 15

Java SE 15.0.2 is the latest release for the Java SE Platform

- [Documentation](#)
- [Installation Instructions](#)
- [Release Notes](#)
- [Oracle License](#)
 - [Binary License](#)
 - [Documentation License](#)
- [Java SE Licensing Information User Manual](#)
 - [Includes Third Party Licenses](#)
- [Certified System Configurations](#)
- [Readme](#)

Oracle JDK

- ↓ [JDK Download](#)
- ↓ [Documentation Download](#)










Looking for Oracle OpenJDK builds?

- **Oracle Customers and ISVs targeting Oracle LTS releases:** Oracle JDK is Oracle's supported Java SE version for customers and for developing, testing, prototyping or demonstrating your Java applications.
- **End users and developers looking for free JDK versions:** [Oracle OpenJDK](#) offers the same features and performance as Oracle JDK under the [GPL license](#).

To Learn more about these options visit [Oracle JDK Releases for Java 11 and Later](#)

Java SE Development Kit 15.0.2

This software is licensed under the [Oracle Technology Network License Agreement for Oracle Java SE](#)

Product / File Description	File Size	Download
Linux ARM 64 RPM Package	141.82 MB	 jdk-15.0.2_linux-aarch64_bin.rpm
Linux ARM 64 Compressed Archive	157 MB	 jdk-15.0.2_linux-aarch64_bin.tar.gz
Linux x64 Debian Package	154.81 MB	 jdk-15.0.2_linux-x64_bin.deb
Linux x64 RPM Package	162.03 MB	 jdk-15.0.2_linux-x64_bin.rpm
Linux x64 Compressed Archive	179.35 MB	 jdk-15.0.2_linux-x64_bin.tar.gz
macOS Installer	175.93 MB	 jdk-15.0.2_osx-x64_bin.dmg
macOS Compressed Archive	176.51 MB	 jdk-15.0.2_osx-x64_bin.tar.gz
Windows x64 Installer	159.71 MB	 jdk-15.0.2_windows-x64_bin.exe
Windows x64 Compressed Archive	179.28 MB	 jdk-15.0.2_windows-x64_bin.zip

Χρήση του τερματικού

- Για τα πρώτα προγράμματα σας θα χρησιμοποιούμε το **τερματικό** για να μεταγλωττίζουμε (compile) και να τρέχουμε τα προγράμματα μας

The image shows a Windows Start menu search interface. The search bar at the top contains the text "cmd". Below the search bar, the results are categorized into "Best match", "Search the web", and "Settings (1)". Under "Best match", the "Command Prompt" app is listed with a small icon and the label "App". To the right of this result, a larger preview card for "Command Prompt" is shown, also labeled "App". This card contains a list of actions: "Open", "Run as administrator", "Open file location", "Pin to Start", and "Pin to taskbar". The "Open" option is circled in red. At the bottom of the search window, there is a search bar with "cmd" entered. The background of the entire image is the Windows 10 desktop with the blue "Windows" logo wallpaper.

All Apps Documents Web More

Best match

Command Prompt
App

Search the web

cmd - See web results

Settings (1)

Open

Run as administrator

Open file location

Pin to Start

Pin to taskbar

cmd

```
Command Prompt
Microsoft Windows [Version 10.0.17763.107]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\tsap>
```



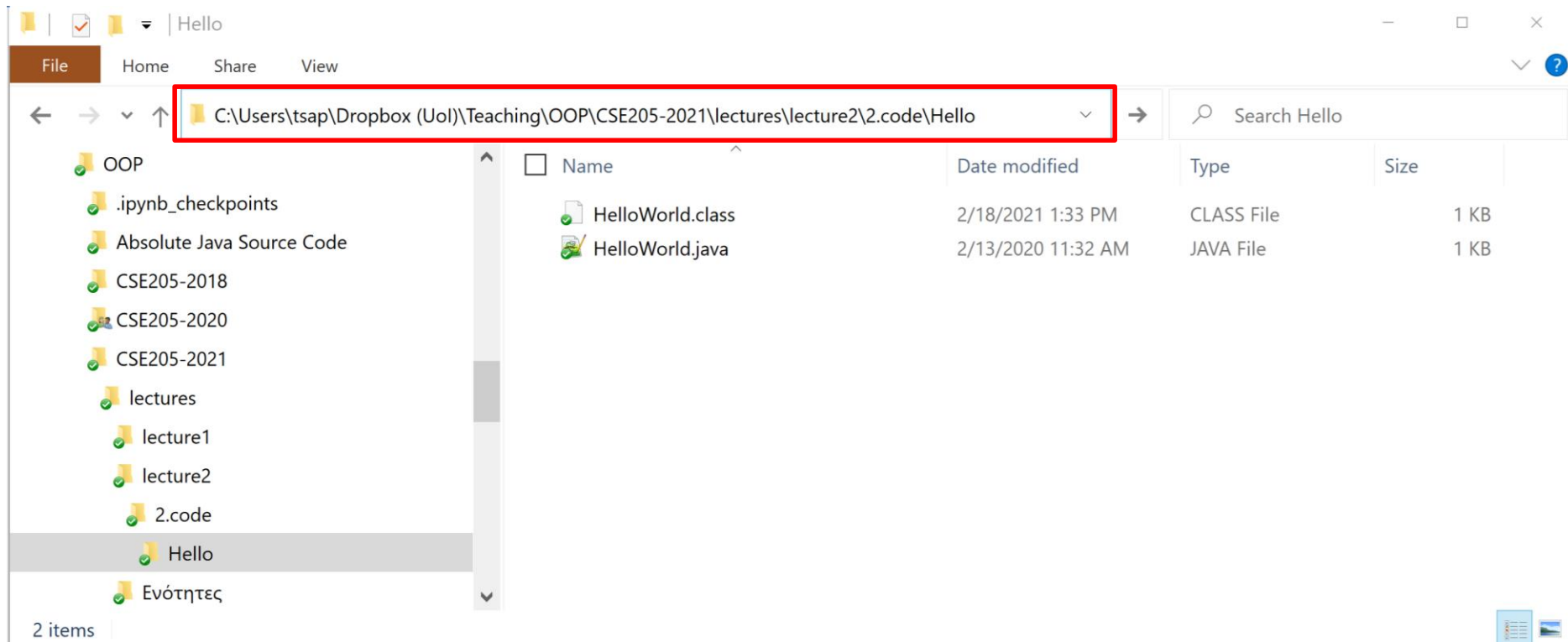
Command Prompt

Microsoft Windows [Version 10.0.17763.107]
(c) 2018 Microsoft Corporation. All rights reserved.

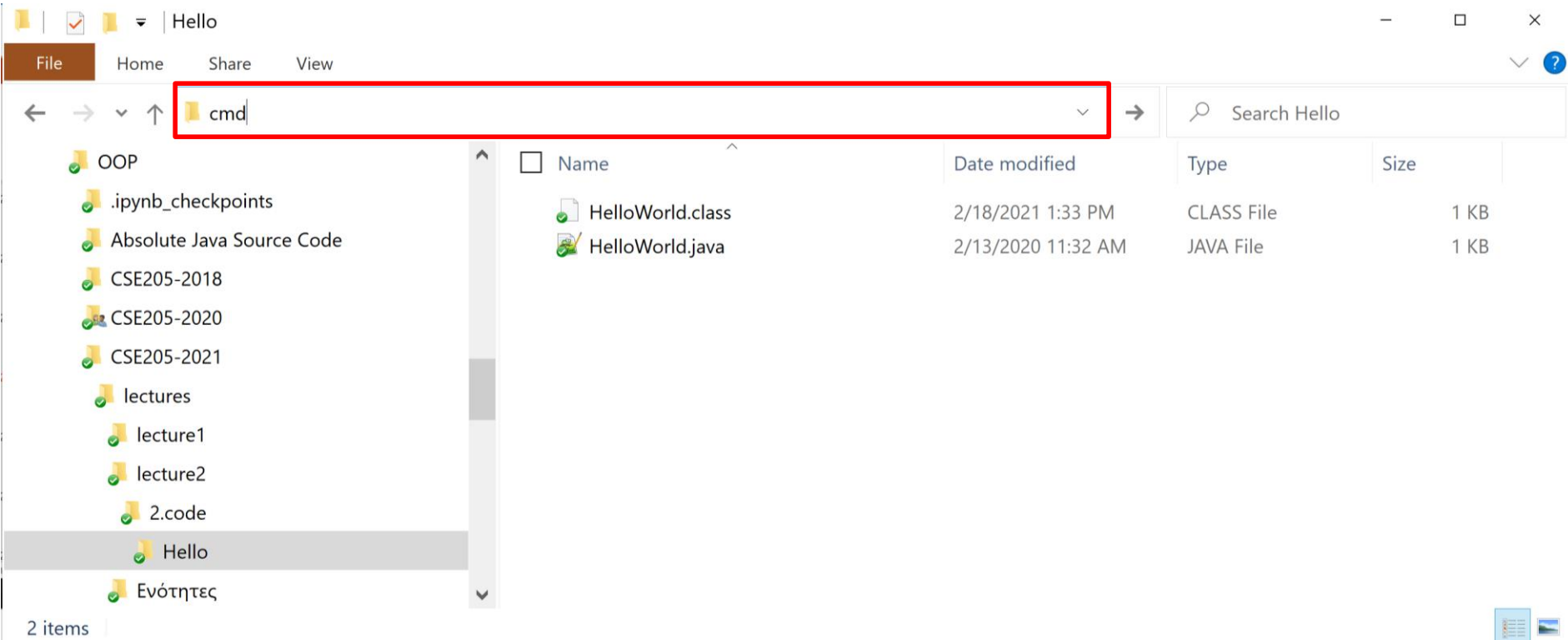
C:\Users\tsap>cd "C:\Users\tsap\Dropbox (UoI)\Teaching\OOP\CSE205-2021\lectures\lecture2\2.code\Hello"

C:\Users\tsap\Dropbox (UoI)\Teaching\OOP\CSE205-2021\lectures\lecture2\2.code\Hello>_

Μπορείτε επίσης να ανοίξετε ένα τερματικό σε συγκεκριμένο φάκελο γράφοντας την εντολή cmd στο παράθυρο που είναι το path του directory



Μπορείτε επίσης να ανοίξετε ένα τερματικό σε συγκεκριμένο φάκελο γράφοντας την εντολή `cmd` στο παράθυρο που είναι το path του directory

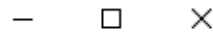


The screenshot shows a Windows File Explorer window titled "Hello". The address bar contains the text "cmd", which is highlighted with a red rectangle. The left sidebar shows a tree view of folders, with "Hello" selected. The main pane displays a table of files:

Name	Date modified	Type	Size
HelloWorld.class	2/18/2021 1:33 PM	CLASS File	1 KB
HelloWorld.java	2/13/2020 11:32 AM	JAVA File	1 KB

At the bottom left, it says "2 items".

C:\Windows\System32\cmd.exe



Microsoft Windows [Version 10.0.17763.1757]

(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\tsap\Dropbox (UoI)\Teaching\OOP\CSE205-2021\lectures\lecture2\2.code\Hello>

Editor

- Για την συγγραφή των πρώτων προγραμμάτων μας θα χρησιμοποιούμε ένα απλό **text editor**, το **Notepad++** . Στα Linux μηχανήματα μπορείτε να χρησιμοποιείτε το **gedit**. Μπορείτε να χρησιμοποιήσετε όποιον editor θέλετε.
- Μπορείτε να χρησιμοποιήσετε και κάποιο IDE (**Eclipse**, **NetBeans**, **IntelliJ**)
- Είναι χρήσιμο ο editor που χρησιμοποιείτε να κάνει **highlighting** δηλαδή να βάζει χρώματα σε λέξεις κλειδιά στον κώδικα.

HELLO WORLD

Το πρώτο μας πρόγραμμα σε Java

Δομή ενός απλού Java προγράμματος

- Το **όνομα** του αρχείου που κρατάει το πρόγραμμα έχει το επίθεμα **.java**
 - Στο παράδειγμα μας ονομάζουμε το πρόγραμμα μας: **HelloWorld.java**
- Αν έχουμε το αρχείο **X.java**, συνήθως μέσα ορίζουμε μια **κλάση** με το όνομα **X**.
 - **class HelloWorld** στο παράδειγμα μας
- Για να τρέξουμε το πρόγραμμα, η κλάση θα πρέπει να περιέχει μια **μέθοδο main** η οποία είναι το **σημείο εκκίνησης** του προγράμματος μας
 - **public static void main(String[] args)**

File HelloWorld.java

```
class HelloWorld
{
    public static void main(String args[])
    {
        // print message
        System.out.println("Hello world!");
    }
}
```

Το όνομα του `.java` αρχείου και το όνομα της κλάσης (που περιέχει την μέθοδο `main`) είναι καλό να είναι τα **ίδια**

- Αν η κλάση έχει οριστεί `public` τότε υποχρεωτικά πρέπει να είναι τα ίδια αλλιώς θα πάρετε `compile error`

Μεταγλώττιση – Compiling

Η μεταγλώττιση γίνεται με την εντολή **javac** και όρισμα το αρχείο **X.java**

- `javac X.java`

Π.χ.

```
➤ javac HelloWorld.java
```

Command Prompt

```
Microsoft Windows [Version 10.0.17763.107]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\tsap>cd "C:\Users\tsap\Dropbox (UoI)\Teaching\OOP\CSE205-2021\lectures\lecture2\2.code\Hello"
C:\Users\tsap\Dropbox (UoI)\Teaching\OOP\CSE205-2021\lectures\lecture2\2.code\Hello>javac HelloWorld.java
C:\Users\tsap\Dropbox (UoI)\Teaching\OOP\CSE205-2021\lectures\lecture2\2.code\Hello>_
```

Μεταγλώττιση – Compiling

Το αποτέλεσμα είναι η δημιουργία του αρχείου **X.class** που περιέχει τον ενδιάμεσο κώδικα (bytecode)

Το αρχείο **HelloWorld.class** στο παράδειγμα μας

```
Command Prompt
Microsoft Windows [Version 10.0.17763.107]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\tsap>cd "C:\Users\tsap\Dropbox (UoI)\Teaching\OOP\CSE205-2021\lectures\lecture2\2.code\Hello"
C:\Users\tsap\Dropbox (UoI)\Teaching\OOP\CSE205-2021\lectures\lecture2\2.code\Hello>javac HelloWorld.java
C:\Users\tsap\Dropbox (UoI)\Teaching\OOP\CSE205-2021\lectures\lecture2\2.code\Hello>dir
Volume in drive C has no label.
Volume Serial Number is EE33-D3D0

Directory of C:\Users\tsap\Dropbox (UoI)\Teaching\OOP\CSE205-2021\lectures\lecture2\2.code\Hello

02/18/2021  11:44 AM    <DIR>          .
02/18/2021  11:44 AM    <DIR>          ..
02/18/2021  11:44 AM                426 HelloWorld.class
02/13/2020  11:32 AM                110 HelloWorld.java
                2 File(s)      536 bytes
                2 Dir(s)  183,254,695,936 bytes free

C:\Users\tsap\Dropbox (UoI)\Teaching\OOP\CSE205-2021\lectures\lecture2\2.code\Hello>
```

Εκτέλεση - Running

- Η εκτέλεση του κώδικα γίνεται με την εντολή **java** με όρισμα **το όνομα του .class** αρχείου **χωρίς κανένα επίθεμα**
 - java X

➤ **java HelloWorld**

Χωρίς **κανένα** επίθεμα!

Command Prompt

```
C:\Users\tsap\Dropbox (UoI)\Teaching\OOP\CSE205-2021\lectures\lecture2\2.code\Hello>java HelloWorld  
Hello World!
```

```
C:\Users\tsap\Dropbox (UoI)\Teaching\OOP\CSE205-2021\lectures\lecture2\2.code\Hello>_
```



```
class HelloWorld
{
    public static void main(String[] args)
    {
        // print message
        System.out.println("Hello world!");
    }
}
```

Λέξεις σε κόκκινο: δεσμευμένες λέξεις

Ορίζει την
κλάση

Όνομα της κλάσης

```
class HelloWorld
{
    public static void main(String[] args)
    {
        // print message
        System.out.println("Hello world!");
    }
}
```

```
class HelloWorld
```

```
{
```

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

```
    {
```

```
        // print message
```

```
        System.out.println("Hello world!");
```

```
    }
```

```
}
```

Τα άγκιστρα { ... } ορίζουν ένα **λογικό block** του κώδικα

- Αυτό μπορεί να είναι **μία κλάση**, **μία συνάρτηση**, **ένα if statement**
- Αντίστοιχο των tabs στην Python, εδώ δεν χρειάζονται αλλά τα βάζουμε πάντα για να διαβάζεται ο κώδικας πιο εύκολα.
- Ένα λογικό block μπορεί να έχει τις δικές του **τοπικές μεταβλητές** που έχουν **εμβέλεια** μόνο μέσα στο block

Ορισμός της συνάρτησης main

```
class HelloWorld
{
    public static void main(String[] args)
    {
        // print message
        System.out.println("Hello world!");
    }
}
```

Ορισμός της συνάρτησης main

```
class HelloWorld
{
    public static void main(String[] args)
    {
        // print message
        System.out.println("Hello world!");
    }
}
```

public, static: θα τα εξηγήσουμε αργότερα

Ορισμός της συνάρτησης main

```
class HelloWorld
{
    public static void main (String[] args)
    {
        // print message
        System.out.println("Hello world!");
    }
}
```

Πριν από το όνομα της μεθόδου έχουμε τον **τύπο της επιστρεφόμενης τιμής**, δηλαδή, τι επιστρέφει η μέθοδος

void: Η μέθοδος δεν επιστρέφει **τίποτα**.

Ορισμός της συνάρτησης main

```
class HelloWorld
{
    public static void main (String[] args)
    {
        // print message
        System.out.println("Hello world!");
    }
}
```

Το όνομα της μεθόδου

- **main**: ειδική περίπτωση που σηματοδοτεί το σημείο εκκίνησης του προγράμματος.

Ορισμός της συνάρτησης main

```
class HelloWorld
{
    public static void main (String[] args)
    {
        // print message
        System.out.println("Hello world!");
    }
}
```

Ορίσματα της μεθόδου

- Ένας **πίνακας** από **Strings** που αντιστοιχούν στις **παραμέτρους** με τις οποίες τρέχουμε το πρόγραμμα.

Η κλάση String

```
class HelloWorld
{
    public static void main (String[] args)
    {
        // print message
        System.out.println("Hello world!");
    }
}
```

- **String**: κλάση που χειρίζεται τα **αλφαριθμητικά**.
- Στη Java χρειάζεται να ορίσουμε τον **τύπο** της κάθε μεταβλητής
- **Strongly typed language**

Σχόλια!

```
/**  
 * A class that prints a message "hello world"  
 **/
```

```
class HelloWorld
```

```
{
```

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

```
    {
```

```
        // print message
```

```
        System.out.println("Hello world!");
```

```
    }
```

```
}
```

```
class HelloWorld
{
    public static void main(String[] args)
    {
        // print message
        System.out.println("Hello world!");
    }
}
```

Κάθε εντολή στη Java πρέπει να τερματίζει με το ;

Έξοδος: Χρησιμοποιούμε την μέθοδο `System.out.println()`
Η μέθοδος παίρνει σαν όρισμα ένα String, και το εκτυπώνει στην οθόνη και αλλάζει γραμμή.

```
class HelloWorld
{
    public static void main(String[] args)
    {
        // print message
        System.out.println("Hello world!");
    }
}
```

Αντικείμενο `System.out`
Ορίζει το ρεύμα εξόδου

Μέθοδος `println`:
Τυπώνει το String αντικείμενο που δίνεται ως όρισμα και αλλάζει γραμμή

```
class HelloWorld
{
    public static void main(String[] args)
    {
        // print message
        System.out.println("Hello world!");
    }
}
```

Το "Hello World" είναι μία **String** σταθερά.
Όταν βάζουμε κείμενο μέσα σε "" ορίζουμε ένα String.
Πιο ακριβώς, ένα αντικείμενο της κλάσης String

Programming Style

Το όνομα της κλάσης ξεκινάει με κεφαλαίο και χρησιμοποιούμε την **CamelCase** σύμβαση

```
class HelloWorld
{
    public static void main(String[] args)
    {
        // print message
        System.out.println("Hello world!");
    }
}
```

Στοίχιση του κώδικα:

- Οι εντολές μέσα σε ένα block του κώδικα ξεκινάνε ένα tab πιο μπροστά από το προηγούμενο.
- Όλες οι εντολές σε ένα block είναι στοιχισμένες
- Τα άγκιστρα είναι στοιχισμένα με την εντολή που ορίζει το block

Programming Style: Ονόματα

- Τα ονόματα των **κλάσεων** ξεκινάνε με κεφαλαίο, τα ονόματα των **πεδίων**, **μεθόδων** και **αντικειμένων** με μικρό.
 - Π.χ., **HelloWorld**, **position**, **print**
- Κάποιες συμβάσεις ξεκινούν τα ονόματα πεδίων με **'_'** για να ξεχωρίζουν από τις μεθόδους
 - Π.χ., **_position**
- Χρησιμοποιούμε **ολόκληρες λέξεις** (και συνδυασμούς τους) για τα ονόματα
 - Δεν πειράζει αν βγαίνουν μεγάλα ονόματα
- Χρησιμοποιούμε το **CamelCase Style**
 - Όταν για ένα όνομα έχουμε πάνω από μία λέξη, τις συνενώνουμε και στο σημείο συνένωσης κάνουμε το πρώτο γράμμα της λέξης κεφαλαίο
 - **printName** όχι **print_name**
- Χρησιμοποιούμε **κεφαλαία** και **'_'** για τις **σταθερές**.
 - Π.χ., **PI_NUMBER**

