

# Αντικειμενοστραφής Προγραμματισμός

Β' εξάμηνο

**Δημήτρης Ρίγγας**

Μηχανικός Η/Υ & Πληροφορικής, MSc, PhD

Ε.Δι.Π. Τμήματος Πληροφορικής – Ιόνιο Παν/μιο

[riggas@ionio.gr](mailto:riggas@ionio.gr)



● επιλογή  java ή c++

 εδώ `java`



## Η σελίδα του μαθήματος

<https://opencourses.ionio.gr/courses/DDI302/>

<https://github.com/riggas-ionio/java>



# Επικοινωνία (για το Java branch!)

Δημήτριος Ρίγγας - [riggas@ionio.gr](mailto:riggas@ionio.gr)

Προφανώς..

- Στο μάθημα (ή το εργαστήριο)

Αν, και μόνο αν, είναι κάτι επείγον..

- Μηνύματα μέσα από τη σελίδα του μαθήματος
- Συζητήσεις (εργαλείο του [opencourses](https://open.courses))
- email που δεν αναφέρουν με σαφήνεια:
  - το μάθημα
  - τα στοιχεία σας (ονοματεπώνυμο και ΑΜ)
  - το ερώτημά σας
  - ΔΕΝ θα απαντώνται...

# Διαλέξεις

- Τρίτη 11:00 – 13:00 – Αίθουσα TABM 2.45
  - Θεωρητικές Διαλέξεις
- Παρασκευή 11:00 - 15:00 - Εργ. Γαληνός
  - Εργαστηριακές Διαλέξεις
  - Ενδιάμεσες Αξιολογήσεις

## Κατανομή στα τμήματα (υποχρεωτικά δια ζώσης μόνο για γ' εξάμηνο)

→ για την πρώτη εβδομάδα ←

- φοιτητές με Α.Μ. που λήγει σε 00-33: 11:00 πμ
- φοιτητές με Α.Μ. που λήγει σε 34-66: 12:00 μμ
- φοιτητές με Α.Μ. που λήγει σε 67-99: 13:00 μμ

→ για την συνέχεια θα χωριστούμε αναλόγως επιλογής γλώσσας ←

## Περιγραφή μαθήματος



# Σκοπός του μαθήματος

Το μάθημα επιδιώκει την εμπέδωση των αρχών του αντικειμενοστραφούς προγραμματισμού μέσω της γλώσσας `Java`, ώστε οι φοιτητές/φοιτήτριες να αποκτήσουν τις απαραίτητες γνώσεις και δεξιότητες για τη σχεδίαση, ανάπτυξη και αξιολόγηση αποδοτικών, επεκτάσιμων και συντηρήσιμων εφαρμογών λογισμικού.

# Στόχοι του μαθήματος - Γνώσεις

- Κατανοούν τις βασικές αρχές του αντικειμενοστραφούς προγραμματισμού (OOP) και να τις διακρίνουν από τον διαδικαστικό προγραμματισμό.
- Γνωρίζουν τη σύνταξη (syntax) και τη σημασιολογία (semantics) γλωσσών αντικειμενοστραφούς προγραμματισμού, με έμφαση στη γλώσσα Java.
- Περιγράφουν και διακρίνουν τα δομικά στοιχεία ενός προγράμματος στη γλώσσα Java και τον τρόπο οργάνωσής του σε αρχεία πηγαίου κώδικα.
- Αναγνωρίζουν βασικές δομές δεδομένων (πίνακες, λίστες, στοίβες, ουρές) και τους αλγορίθμους διαχείρισής τους.
- Αναγνωρίζουν τα οφέλη της αξιοποίησης αντικειμενοστραφούς κώδικα στους τομείς της ασφάλειας, της επεκτασιμότητας και της συντηρησιμότητας.
- Αναγνωρίζουν τα οφέλη που παρέχει η χρήση της εικονικής μηχανής της Java (JVM) στον τομέα της μεταφερσιμότητας και της ασφάλειας του κώδικα.

# Στόχοι του μαθήματος - Δεξιότητες

- Γράφουν προγράμματα σε Java χρησιμοποιώντας τις βασικές δομές προγραμματισμού (μεταβλητές, συνθήκες, βρόχους, μεθόδους).
- Εφαρμόζουν την αντικειμενοστραφή προσέγγιση μέσω κλάσεων, αντικειμένων, μεθόδων και κληρονομικότητας.
- Σχεδιάζουν και κατασκευάζουν κλάσεις για την αναπαράσταση απλών και πιο σύνθετων οντοτήτων και επιλέγουν τις ανά περίπτωση κατάλληλες κλάσεις που παρέχει η γλώσσα.
- Αναπτύσσουν απλές και σύνθετες εφαρμογές που χρησιμοποιούν αρχεία, δυναμική διαχείριση μνήμης και εξαιρέσεις.
- Υλοποιούν αλγορίθμους ταξινόμησης, αναζήτησης και διαχείρισης δεδομένων.
- Αξιοποιούν προχωρημένες τεχνικές προγραμματισμού όπως η σχεδίαση ιεραρχιών κλάσεων, οι συλλογές, η διαχείριση σφαλμάτων με χρήση εξαιρέσεων, οι ροές δεδομένων και ο διαδικτυακός προγραμματισμός.
- Εξοικειώνονται με ένα μοντέρνο περιβάλλον ανάπτυξης εφαρμογών (IDE) για τη γλώσσα προγραμματισμού Java.
- Συνδυάζουν διαφορετικές δομές δεδομένων και μεθοδολογίες για τη βελτίωση της αποδοτικότητας του κώδικα.
- Εντοπίζουν και διορθώνουν σφάλματα (debugging) και προβλήματα απόδοσης (performance issues) στον κώδικα.
- Κρίνουν την ποιότητα και τη συντηρησιμότητα του κώδικα με βάση τις αρχές καλών πρακτικών προγραμματισμού.

# Στόχοι του μαθήματος - Στάσεις

- Αναπτύσσουν δεξιότητες κριτικής σκέψης και επίλυσης προβλημάτων με αναλυτικό και δομημένο τρόπο.
- Εξοικειώνονται με την αυτοκατευθυνόμενη μάθηση και τη συνεχή ενημέρωση για νέες εξελίξεις στον προγραμματισμό.
- Αναγνωρίζουν την αξία της συνεργατικότητας και λειτουργούν αποτελεσματικά σε ομαδικό περιβάλλον.
- Συνεργάζονται σε ομάδες ανάπτυξης λογισμικού χρησιμοποιώντας εργαλεία διαχείρισης κώδικα (π.χ. Git).
- Παρουσιάζουν τις εργασίες τους και αξιολογούν τις εργασίες των συμφοιτητών τους με συγκεκριμένα κριτήρια.

# Συγγράμματα

## Η γλώσσα προγραμματισμού JAVA - Αναλυτική Προσέγγιση



Horstmann Cay - 2021  
Εκδόσεις BROKEN HILL PUBLISHERS  
Κωδικός στον Εύδοξο: 94643857

## JAVA - Εκμάθηση με πρακτικά παραδείγματα



Joyce Farrell - 2018  
Εκδόσεις ΚΡΙΤΙΚΗ  
Κωδικός στον Εύδοξο: 77109703

# Περιεχόμενο μαθήματος



# Αξιολόγηση μαθήματος



# Αξιολόγηση, όπως ισχύει

## Τελική εξέταση

- δια ζώσης ~ 1:15' ώρα
- βαρύτητα 60% της τελικής βαθμολογίας, για όλους

## Ενδιάμεσες αξιολογήσεις (υποχρεωτικές, συνυπολογίζονται στον τελικό βαθμό)

- ισχύουν ενδιάμεσες αξιολογήσεις & ασκήσεις περασμένων ετών
- θα γίνουν πέντε (5) ενδιάμεσες αξιολογήσεις μέσω του `opencourses`
- κάθε αξιολόγηση θα αντιστοιχεί σε 10% της τελικής βαθμολογίας
- κρατάμε τις τέσσερις (4) καλύτερες → 40% της τελικής βαθμολογίας
- οι αξιολογήσεις θα είναι live και χρονομετρημένες, την ώρα των προγραμματιστικών ασκήσεων (15'-20' max)

# Αξιολόγηση, όπως ισχύει

## Παλιότεροι φοιτητές

- μπορούν να υποβάλλουν τις φετινές αξιολογήσεις και θα ισχύσει ο καλύτερος βαθμός ή
- κρατούν τις παλιές εργαστηριακές ασκήσεις (ακ. έτους από 2016-17 έως και 2024-25)
- **προσοχή (παλιοί):** η βαρύτητα είναι μεγαλύτερη από το 2021, άρα αν είχατε κάνει πχ μόνο μία από τις ασκήσεις του έτους σας... μάλλον συμφέρει να κάνετε πάλι τις φετινές!

# FAQ

**Υπάρχουν υποχρεωτικές παρουσίες στο εργαστήριο;**

Όχι, αλλά η συμμετοχή βοηθάει πάρα πολύ. (βασικά, αν χάσετε επαφή.. συνήθως κόβεστε)

**Μπορώ να μην πάρω μέρος στις ενδιάμεσες αξιολογήσεις και να περάσω;**

Ναι, αλλά μάλλον δε θα περάσεις! πρέπει να γράφεις πάνω από 4,5/6 στην τελική εξέταση.

**Είναι καλή ιδέα να μην πάρω μέρος στις ενδιάμεσες αξιολογήσεις;**

Όχι, μόλις το 6% όσων πέρασαν (τον Ιαν. 2021) δεν είχαν κάνει ενδιάμεσες αξιολογήσεις ή παλιότερες ασκήσεις εργαστηρίου

**Τελικά, θα περάσω;**

Το 2020-21 το ποσοστό επιτυχίας (Ιανουαρίου & Σεπτεμβρίου) ήταν >80% → άρα καλύτερα να εστιάσετε στο να μάθετε Java.. τότε, τα άλλα θα έρθουν μόνα τους

προγραμματιστικές ασκήσεις



# Πότε ξεκινάμε προγραμματιστικές ασκήσεις;

Στις 20/10 (δλδ την πρώτη Παρασκευή μετά τη σημερινή θεωρία)

Οι συναντήσεις στο Γαληνό έχουν κύριο ρόλο την υποστήριξη και επίλυση αποριών και προβλημάτων, κάθε εβδομάδα θα υπάρχουν προγραμματιστικές ασκήσεις στο <https://github.com/courses-ionio/java>

☞ **Bring your own device, [more than] welcome!**

Εξοικειωθείτε με ένα IDE, πχ VSCode

ή (αν δεν φέρνεται laptop) μπορείτε να δουλεύεται σε κάποιο cloud compiler, όπως πχ <https://onecompiler.com>, *αλλά δεν είναι το ίδιο*

# Προετοιμασία για προγραμματιστικές ασκήσεις 1/2

Kick-off task πρώτο (πριν την 1η συνάντηση στο εργαστήριο), δηλαδή τώρα:

Στο pc σας:

- εγκαταστήστε στο pc σας κάποιο OpenJDK-based, πχ 21 LTS, Java Development Kit (πηγή: <https://adoptium.net>)
- έλεγχος σε τερματικό (ανοίξτε και εκτελέστε):
  - `java -version`
  - `javac -version`
- εγκαταστήστε το VSCode IDE ή όποιο άλλο IDE θέλετε (no support!)
- προσθέστε και κάνετε το απαραίτητο configuration των extensions για υποστήριξη java

Θα χρειαστεί να εργαστείτε σε γραμμή εντολών για τη μεταγλώττιση και την εκτέλεση java κώδικα, εξοικειωθείτε:

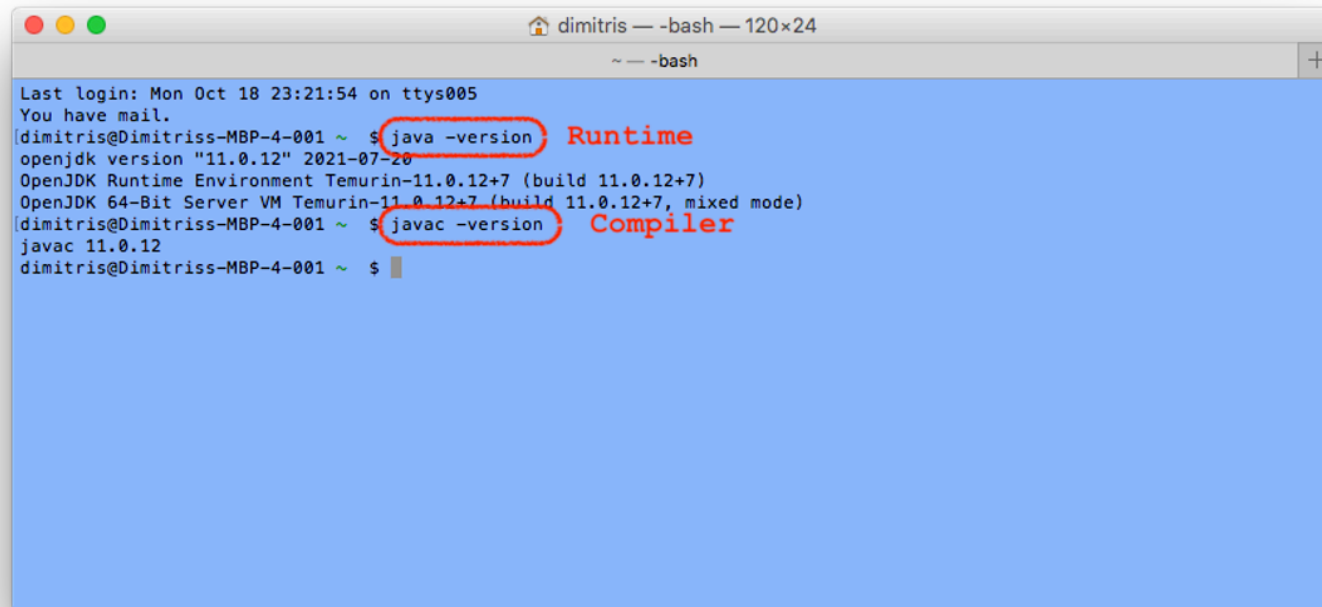
- <https://www.freecodecamp.org/news/command-line-for-beginners/>
- <https://www.cs.princeton.edu/courses/archive/spr05/cos126/cmd-prompt.html>
- <https://ubuntu.com/tutorials/command-line-for-beginners#3-opening-a-terminal>

# Στο pc σας

Θα πρέπει να μπορείτε να μεταγλωττίσετε και εκτελέσετε σε τερματικό κώδικα γραμμένο σε java (κατ' ελάχιστον) έκδοση 8· πχ εκτελέστε και ελέξτε:

```
java -version  
javac -version
```

“ *Java 8 support extended until at least 2030* ”



```
dimitris ~ -bash — 120x24  
~ -bash  
Last login: Mon Oct 18 23:21:54 on ttys005  
You have mail.  
[dimitris@Dimitriss-MBP-4-001 ~] $ java -version Runtime  
openjdk version "11.0.12" 2021-07-20  
OpenJDK Runtime Environment Temurin-11.0.12+7 (build 11.0.12+7)  
OpenJDK 64-Bit Server VM Temurin-11.0.12+7 (build 11.0.12+7, mixed mode)  
[dimitris@Dimitriss-MBP-4-001 ~] $ javac -version Compiler  
javac 11.0.12  
dimitris@Dimitriss-MBP-4-001 ~ $
```

# CodeCheck - anonymous, author-friendly autograder

## Εργαλείο προγραμματιστικών ασκήσεων

- συμπλήρωσης κώδικα
- puzzle κώδικα
- εκτέλεσης τεστ κώδικα
- πρόσβαση μέσω Firefox or Chrome

Θα δίνονται url με συγκεκριμένες ασκήσεις (δε χρειάζεται εγγραφή)

- Αυτοαξιολόγησης
- Αξιολογούμενες (ως μέρος των 5 ενδιάμεσων αξιολογήσεων)

τι  
γιατί  
πώς  
αποτέλεσμα



# Διαδικαστικός προγραμματισμός

Ακολουθία εντολών οι οποίες ορίζουν στον υπολογιστή τι πρέπει να κάνει βήμα-βήμα.

Το πρόγραμμα χωρίζεται σε **διαδικασίες** (ή ρουτίνες) [procedures, routines, subroutines, functions, ..].

## ✓ Εύκολα κατανοητός τρόπος προγραμματισμού.

- Απλά ορίζει βήμα-βήμα τι πρέπει να γίνει.

## ✗ Κώδικας δύσκολος να συντηρηθεί.

- Τα δεδομένα δεν αναπαριστούν απαραίτητα κάποια οντότητα, μπορεί να είναι πολύπλοκα και να έχουν εξαρτήσεις
- Μικρές αλλαγές σε ένα τύπο δεδομένων ή μια διαδικασία πρέπει να ελεγχθούν ότι δεν επηρεάζουν το σύνολο
- Δεν παρέχει απόκρυψη των δεδομένων

# Από τον διαδικαστικό στον αντικειμενοστραφή προγραμματισμό

Ο διαδικαστικός προγραμματισμός, παρότι απλός στην κατανόηση, παρουσιάζει σημαντικά προβλήματα όταν τα προγράμματα μεγαλώνουν:

## Προβλήματα διαδικαστικού:

- Δεδομένα και διαδικασίες είναι ξεχωριστά → δύσκολη διαχείριση
- Έλλειψη οργάνωσης → δύσκολη συντήρηση
- Δεν υπάρχει προστασία δεδομένων → λάθη και bugs
- Μικρή επαναχρησιμοποίηση κώδικα

## Η λύση: Αντικειμενοστραφής προγραμματισμός (Object-Oriented Programming)

- Οργάνωση του κώδικα γύρω από **αντικείμενα**
- Κάθε αντικείμενο περιλαμβάνει **δεδομένα** και **λειτουργίες** μαζί
- Στόχος: Αξιόπιστος, επαναχρησιμοποιήσιμος, και επεκτάσιμος κώδικας

# Αντικειμενοστραφής προγραμματισμός

Οργάνωση του κώδικα γύρω από αντικείμενα.

Ένα αντικείμενο περιλαμβάνει:

- Ιδιότητες (χαρακτηριστικά) – properties
- Συμπεριφορές (τι μπορεί να κάνει) – methods
- .. και είναι μοναδικό – identifier

Σκοπός - κώδικας ο οποίος να είναι:

- ✓ Αξιόπιστος
- ✓ Επαναχρησιμοποιήσιμος
- ✓ Επεκτάσιμος

# Αντικειμενοστραφής προγραμματισμός

Οι μέθοδοι ενός αντικειμένου μοιάζουν με τις διαδικασίες στον διαδικαστικό προγρ. εδώ όμως είναι μέρος του αντικειμένου.

Στον διαδικαστικό προγραμματισμό, τα δεδομένα και οι διαδικασίες είναι δυο ξεχωριστά πράγματα.

Στον αντικειμενοστραφή προγραμματισμό τα δεδομένα είναι μέρος του αντικειμένου ... και αυτό αποφασίζει πόσο προσβάσιμα από τρίτους είναι.

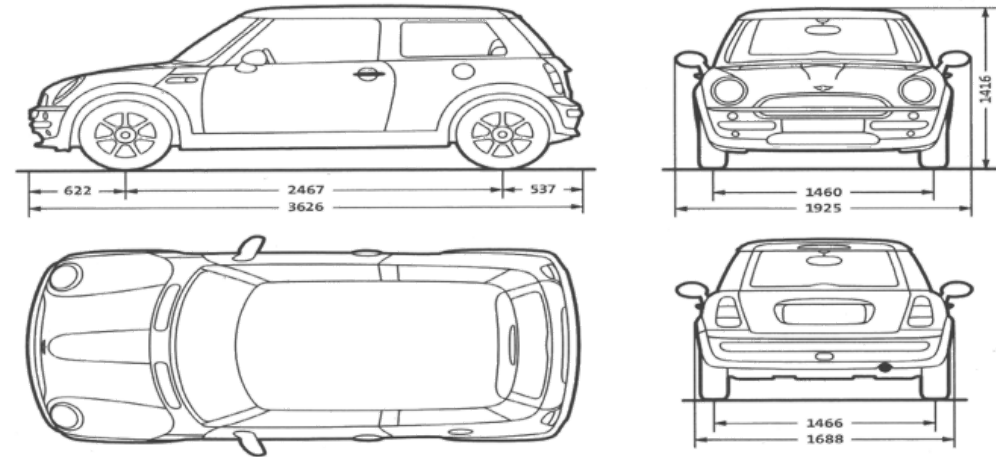
Η εκτέλεση του προγράμματος συνίσταται σε αντικείμενα τα οποία επικοινωνούν μεταξύ τους, και τα οποία ζητούν στην εκτέλεση ορισμένων εργασιών.

# Κλάσεις & Αντικείμενα (σύντομα)

Κάθε αντικείμενο (object) ανήκει σε μια κλάση (class)

Κλάση αναπαριστά μια συλλογή αντικειμένων με κοινές ιδιότητες και λειτουργικότητες

- Είναι ένας αφηρημένος τύπος δεδομένων
- Ορίζεται μια μόνο φορά
- Δημιουργούνται πολλά στιγμιότυπα της → **αντικείμενα**
- (μια κλάση συνήθως αναπαριστά ένα ουσιαστικό)



Βασικές αρχές:

- επαναχρησιμοποίηση
- ενθυλάκωση – απόκρυψη
- κληρονομικότητα



# Διαφορές διαδικαστικού – αντικειμενοστραφούς

Διαδικαστικός	Αντικειμενοστραφής
Top-down σχεδίαση	Σχεδίαση με βάση τα αντικείμενα
Πιο περίπλοκος κώδικας	Πιο περίπλοκη σχεδίαση
Πρόσβαση στα δεδομένα	Προστασία δεδομένων
Μικρή επαναχρησιμοποίηση κώδικα	Επαναχρησιμοποίηση κώδικα

# Java

## Αντικειμενοστραφής γλώσσα

### Ανεξάρτητη από συγκεκριμένη αρχιτεκτονική υπολογιστικού συστήματος

- Δυνατότητα εκτέλεσης σε οποιαδήποτε πλατφόρμα ή λειτουργικό.
- Write once, run anywhere

Μεταγλωττίσιμη και διερμηνευόμενη, χρησιμοποιώντας τα πλεονεκτήματα και των δύο μοντέλων παραγωγής κώδικα

### Απαλοιφή πρακτικών που οδηγούν σε μη ασφαλή κώδικα

- Αριθμητική με δείκτες
- Δέσμευση και αποδέσμευση μνήμης – αντικαθίσταται από συλλογή απορριμάτων

Διανέμεται δωρεάν (? 2019 update: <https://www.techrepublic.com/article/can-you-still-use-the-java-programming-language-for-free-what-you-need-to-know/>)

- [https://en.wikipedia.org/wiki/Java\\_\(programming\\_language\)#History](https://en.wikipedia.org/wiki/Java_(programming_language)#History)
- <https://adoptopenjdk.net/>

# Αδειοδότηση Java

## Σύντομο ιστορικό

- Η Java αναπτύχθηκε αρχικά από τη **Sun Microsystems** το 1995. Το 2010, η Sun εξαγοράστηκε από την **Oracle**, η οποία ανέλαβε τον έλεγχο τόσο της γλώσσας όσο και της αδειοδότησής της – αλλαγή που επηρέασε σημαντικά τους όρους εμπορικής χρήσης.

## Oracle JDK

- Δωρεάν για ανάπτυξη και προσωπική χρήση
- **Εμπορική χρήση / Production:** Απαιτείται εμπορική άδεια (Oracle Java SE Subscription)
- Από την έκδοση 17+: δωρεάν υπό την άδεια **NFTC** (No-Fee Terms and Conditions) – αλλά με περιορισμούς στην ανακατανομή

## OpenJDK

- Ανοιχτός κώδικας υπό την άδεια **GNU GPL v2 with Classpath Exception**
- Ελεύθερη χρήση για οποιοδήποτε σκοπό – ανάπτυξη, production, εμπορική χρήση
- Βασίζεται στον ίδιο κώδικα με το Oracle JDK

## Διανομές OpenJDK (συνιστώνται)

- **Eclipse Temurin (Adoptium)** – <https://adoptium.net>
- **Amazon Corretto, Microsoft Build of OpenJDK, Azul Zulu**

✓ Για production περιβάλλοντα, προτιμήστε μια διανομή OpenJDK ώστε να αποφύγετε περιορισμούς αδειοδότησης και κόστος.

# Java Virtual Machine (JVM)

Μια αφαιρετική υπολογιστική μηχανή η οποία επιτρέπει την εκτέλεση ενός προγράμματος γραμμένου σε Java.

**Προδιαγραφές JVM (specification)** – κοινές, περιγράφουν απαιτήσεις από κάθε υλοποίηση της JVM. Περιλαμβάνουν:

- Ορισμούς για σύνολο εντολών, καταχωρητών, μορφή εκτελέσιμων αρχείων, στοίβας, κ.λπ.
- Κοινές προδιαγραφές → (οδηγούν σε) διαλειτουργικότητα

**Υλοποιήσεις JVM (implemetations)** – ποικίλες σε κάθε πλατφόρμα (π.χ. Win, Mac, Linux)

- HotSpot (Sun/Oracle), OpenJDK, IcedTea (GNU), Eclipse OpenJ9, κ.α.

**Στιγμιότυπο JVM (instance)** – μια εκτελέσιμη διαδικασία η οποία εκτελεί ένα πρόγραμμα Java

- Κάθε πρόγραμμα εκτελείται σε ξεχωριστό στιγμιότυπο

# Βήματα συγγραφής, μεταγλώττισης και εκτέλεσης κώδικα Java

## 1. Συγγραφή πηγαίου κώδικα

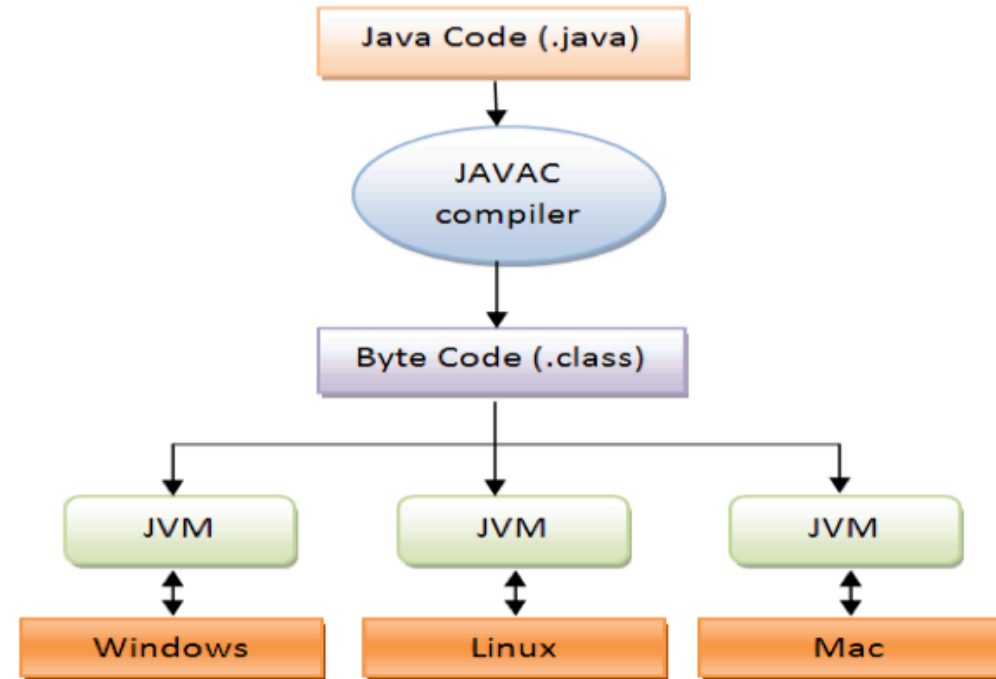
- Text editor ή Integrated Development Environment
- Επεξεργασία `.java` αρχείων

## 2. Μεταγλώττιση σε ενδιάμεσο κώδικα (bytecode)

- Compiler: `javac`
- Παραγωγή `.class` αρχείων

## 3. Εκτέλεση του ενδιάμεσου κώδικα από τη JVM

- Εντολή: `java <class>`



# Εκτέλεση του ενδιάμεσου κώδικα

(Θυμηθείτε: Στον ενδιάμεσο κώδικα [=bytecode] έχουμε φτάσει με μεταγλώττιση [=compile] του πηγαίου [=source])

Ο φορτωτής κλάσης (**class loader**) διαβάζει τα `.class` αρχεία που περιέχουν τα bytecodes από τον δίσκο και τα μεταφέρει στη μνήμη.

Ο **bytecode επαληθευτής** επιβεβαιώνει πως όλα τα bytecodes είναι έγκυρα και δεν παραβιάζουν τους περιορισμούς ασφάλειας της Java.

Η **εικονική μηχανή της Java (JVM)** διαβάζει τα bytecodes και τα διερμηνεύει σε μια γλώσσα που ο υπολογιστής μπορεί να καταλάβει ακριβώς όταν χρειάζεται (just-in-time, JIT).

Κατά τη διάρκεια της εκτέλεσης του προγράμματος, αποθηκεύει δεδομένα στην κύρια μνήμη.

✓ Ασφάλεια | ✓ Φορητότητα | ✓ Ταχύτητα εκτέλεσης (βελτίωση: just-in-time, hotspot compiler)

# JRE και JDK

## Java Runtime Environment (JRE)

Περιβάλλον εκτέλεσης προγραμμάτων που υλοποιεί την αρχιτεκτονική JVM, παρέχει API με βιβλιοθήκες έτοιμου λογισμικού και άλλα βοηθητικά εργαλεία εκτέλεσης προγραμμάτων.

## Java Development Kit (JDK)

Όλα τα προηγούμενα και επιπλέον

Εργαλεία για μεταγλώττιση (javac), τεκμηρίωση (javadoc), συσκευασία (jar) κ.α.

# Πρώτο πρόγραμμα σε Java

```
/*  
 * Τεκμηρίωση HelloWorld προγράμματος  
 * Αρχείο πηγαίου κώδικα HelloWorld.java  
 *  
 * Πρώτο πρόγραμμα σε Java. Εμφανίζει το κείμενο Hello, World στο τερματικό.  
 *  
 * Compilation: javac HelloWorld.java  
 * Execution:   java HelloWorld  
 *  
 *****/  
  
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello, World");  
    }  
}
```

# public class HelloWorld

Αποθήκευση πηγαίου κώδικα σε αρχείο με ίδιο όνομα και κατάληξη `.java`

→ `HelloWorld.java`

Μετά την εκτέλεση του `java compiler` θα παραχθεί το ενδιάμεσο `bytecode` αρχείο με ίδιο όνομα και κατάληξη `.class`

→ `HelloWorld.class`

# public static void main(String[] args)

## main

Μέθοδος από την οποία θα ξεκινήσει η εκτέλεση του προγράμματος

## public

Μπορεί να προσπελαστεί (κληθεί) από οποιονδήποτε, ακόμα και από το διερμηνέα της Java

## static

Δεν απαιτείται η δημιουργία στιγμιοτύπου της κλάσης για να γίνει η κλήση

## void

Η μέθοδος δεν επιστρέφει κάτι

## String args[]

Ορίσματα που μπορεί να δέχεται το πρόγραμμα

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

Πηγή: <https://docs.oracle.com/javase/7/docs/api/>

## System

The System class contains several useful class fields and methods

Among the facilities provided by the System class are standard input, standard output, and error output streams

## out

Αντικείμενο της κλάσης System

The "standard" output stream

## println

Μέθοδος `public void println(String x)`

Print a String and then terminate the line.

# Σχόλια κώδικα

Μιας γραμμής: `//`

```
// Prints "Hello, World" to the terminal window.
```

Πολλαπλών γραμμών: `/* ... */`

```
/*  
 * Τεκμηρίωση HelloWorld προγράμματος  
 * Αρχείο πηγαίου κώδικα HelloWorld.java  
 *  
 * Πρώτο πρόγραμμα σε Java. Εμφανίζει το κείμενο Hello, World στο τερματικό.  
 *  
 * Compilation: javac HelloWorld.java  
 * Execution:   java HelloWorld  
 *  
 *****/
```

# Μεταγλώττιση & εκτέλεση

Αρχείο πηγαίου κώδικα:

```
HelloWorld.java
```

Μεταγλώττιση (από τερματικό):

```
javac HelloWorld.java
```

Παράγει bytecode αρχείο: `HelloWorld.class`

Όχι `.exe`, είναι ενδιάμεσο αρχείο το οποίο αυτούσιο τρέχει σε πολλές πλατφόρμες

Εκτέλεση (από τερματικό):

```
java HelloWorld
```

Προσοχή, όχι: `java HelloWorld.class`

# Συνήθη σφάλματα

## javac: Command not found

- Δεν έχει εγκατασταθεί το JDK
- Δεν έχει οριστεί η μεταβλητή περιβάλλοντος path ή το JAVA\_HOME ώστε να εντοπίζεται ο compiler (javac)
- Δεν μπορεί να βρεθεί ο compiler

## HelloWorldApp.java:16: Method printl(java.lang.String) not found in class java.io.PrintStream.

- Λανθασμένο όνομα μεθόδου κατά την πληκτρολόγηση

## In class HelloWorldApp: main must be public and static

- Παράληψη μίας εκ των λέξεων: public ή static

## HelloWorld.java:12: error: class HelloWorldApp is public, should be declared in a file named HelloWorldApp.java

- Το αρχείο `.java` δεν έχει το ίδιο όνομα με την public class που περιέχει

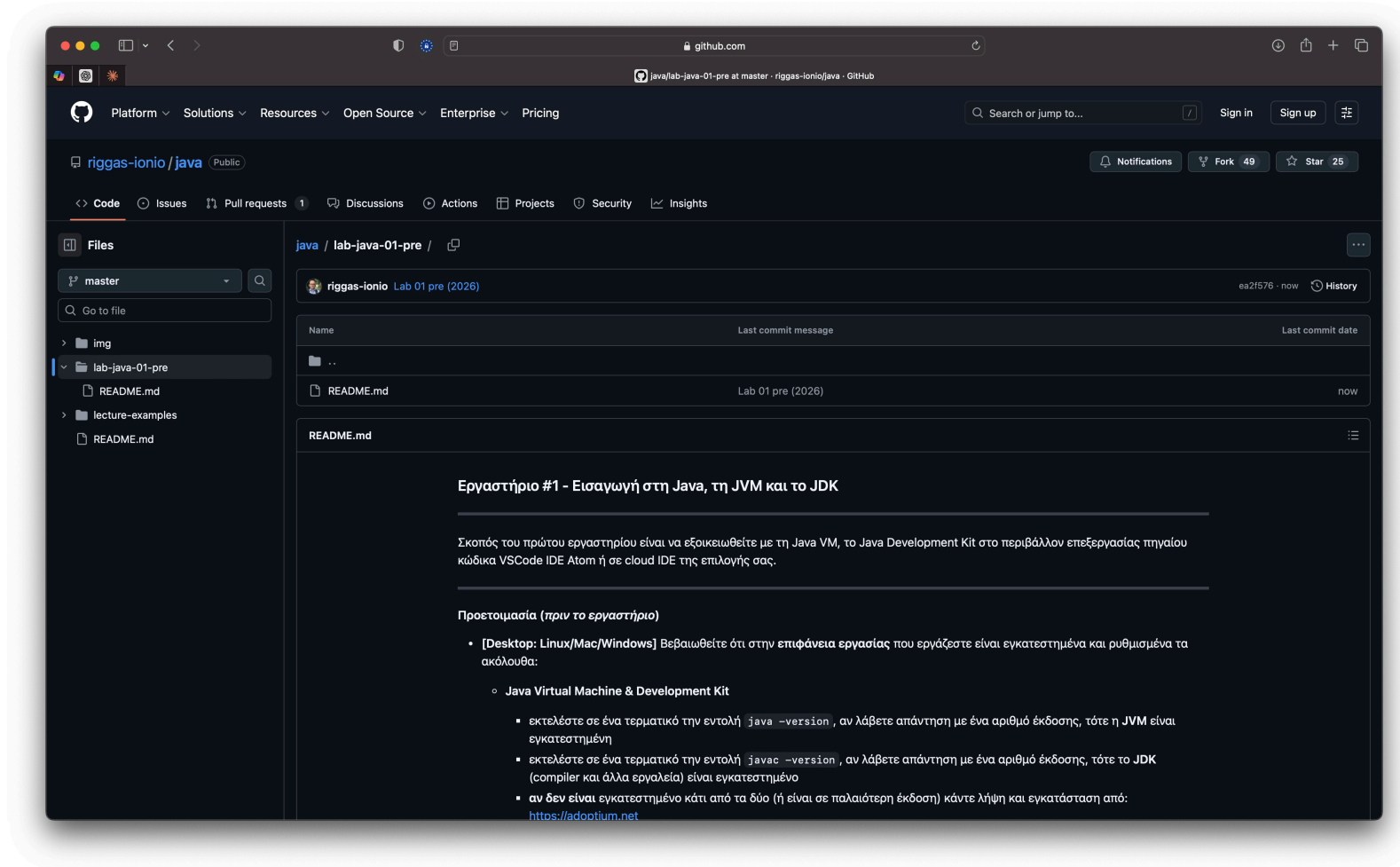
## Exception in thread "main" java.lang.NoClassDefFoundError: HelloWorldApp (wrong name: HelloWorld)

- Έχει γίνει rename το `.class` αρχείο από HelloWorld.class σε HelloWorldApp.class

μελέτη για το σπίτι  
πριν το εργαστήριο



# Προετοιμασία (πριν το εργαστήριο)



“ <https://github.com/riggas-ionio/java> „