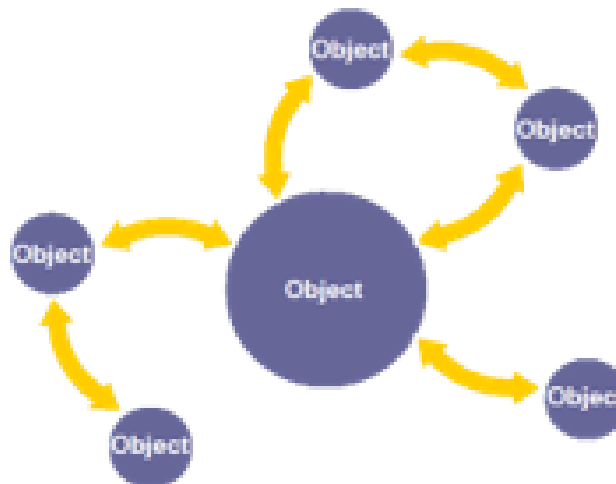


Αντικειμενοστραφής Προγραμματισμός

Ενότητα 2: Εισαγωγή στον Αντικειμενοστραφή Προγραμματισμό και την Java

Μ. Φειδάκης
ΕΔΙΠ ΠαΔΑ



Χ. Πατρικάκης
Καθηγητής, ΠαΔΑ

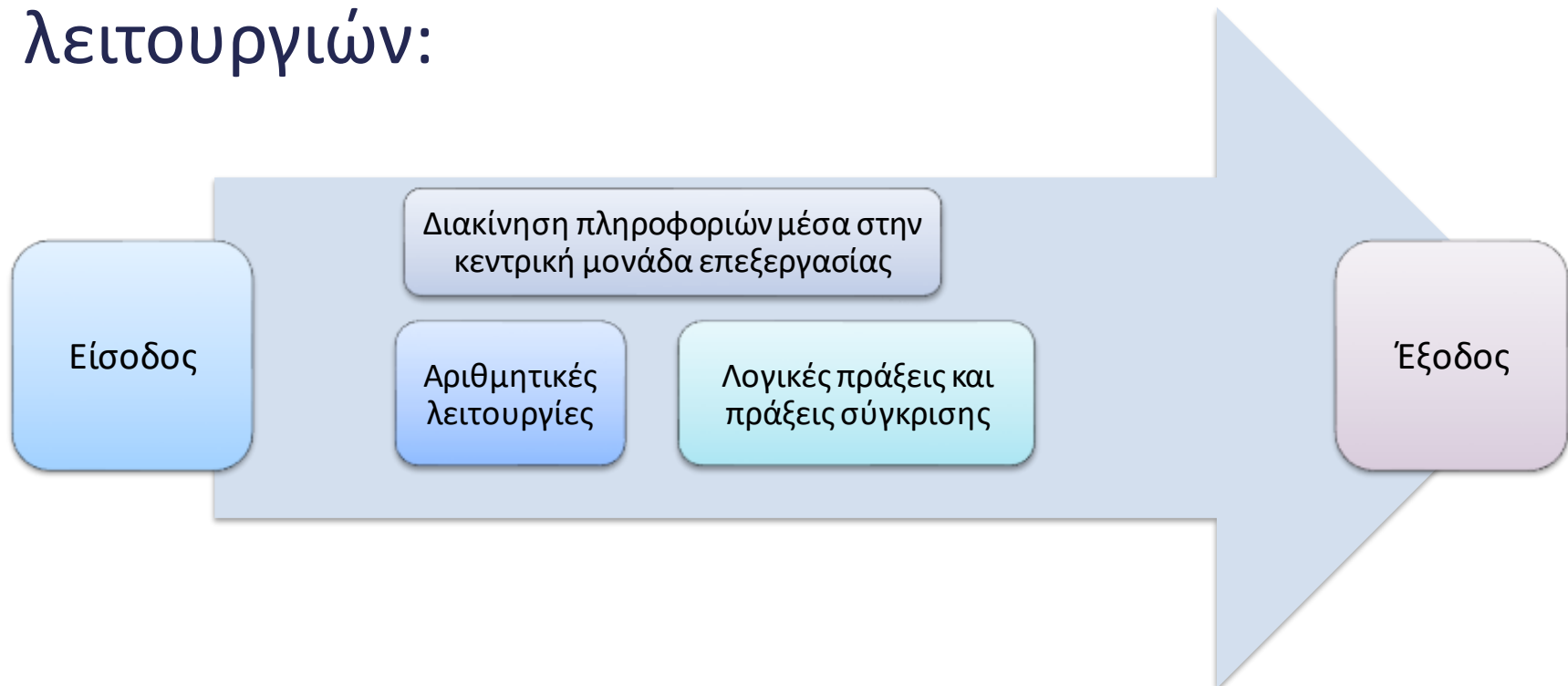
2α. Εισαγωγή στον Αντικειμενοστραφή Προγραμματισμό

1. Βασικές έννοιες του Προγραμματισμού
2. Γενιές γλωσσών προγραμματισμού
3. Είδη Προγραμματισμού
4. Αντικειμενοστραφής προγραμματισμός



Προγραμματισμός Η/Υ

Βασίζεται στην επίλυση προβλημάτων μέσω της συνδυασμένης χρήσης τεσσάρων θεμελιωδών λειτουργιών:



Βασικές έννοιες προγραμματισμού

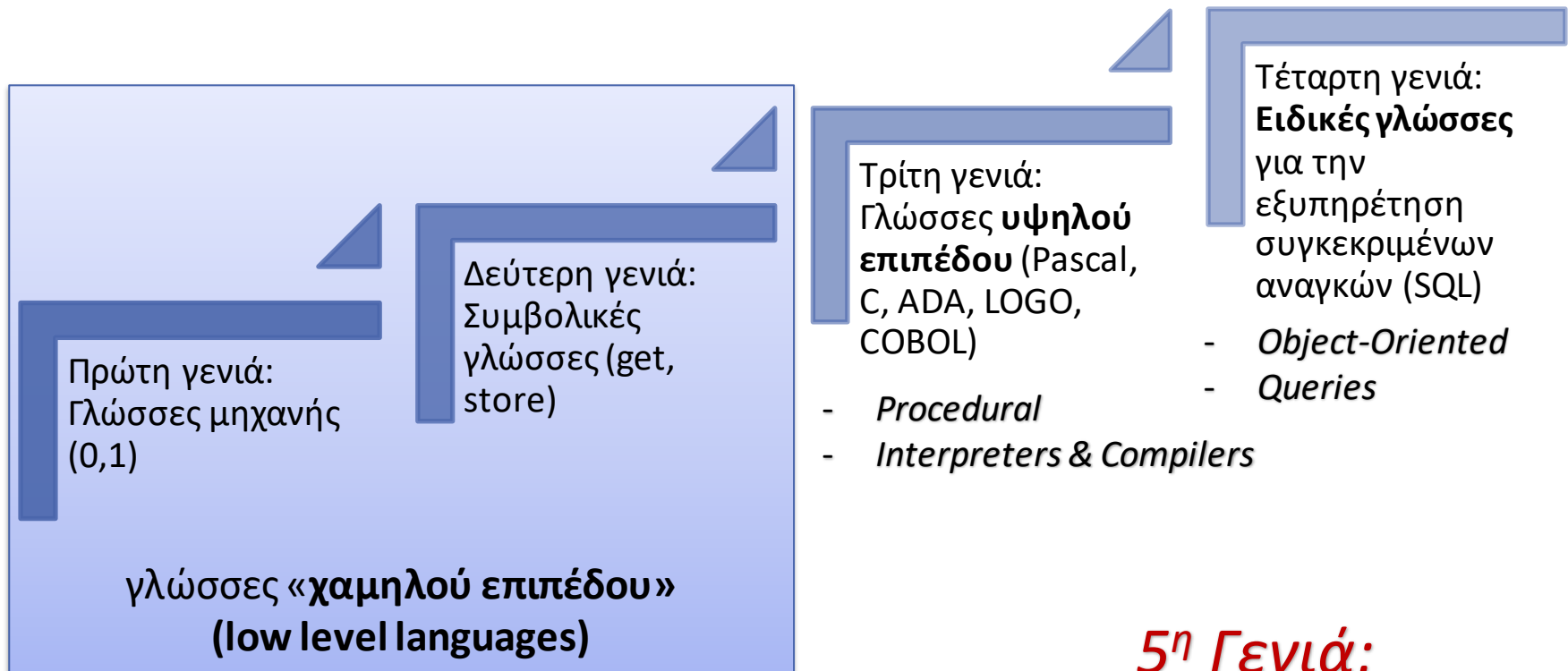
Πρόγραμμα – Ένα σύνολο εντολών που καθοδηγούν τον υπολογιστή στην εκτέλεση συγκεκριμένων λειτουργιών.

Γλώσσα Προγραμματισμού Υψηλού Επιπέδου – Γλώσσα προγραμματισμού το σύνολο των συμβόλων της οποίας προσιδιάζει στην φυσική (ανθρώπινη) γλώσσα.

Συντακτικό – Οι κανόνες που καθορίζουν τον τρόπο συνδυασμού των διαφόρων συμβόλων της γλώσσας προγραμματισμού έτσι ώστε να σχηματιστεί το πρόγραμμα.

Γλώσσα Μηχανής – Γλώσσα προγραμματισμού το σύνολο των συμβόλων της οποίας είναι άμεσα κατανοητό από τον επεξεργαστή του υπολογιστή.

Γενιές γλωσσών προγραμματισμού



Επεξεργασία κώδικα υψηλού επιπέδου

Μεταγλωττιστής (Compiler)

Μετατροπή από κώδικα υψηλού επιπέδου σε γλώσσα μηχανής. Εκτέλεση του προγράμματος που προκύπτει.

Διερμηνευτής (Interpreter)

Ο κώδικας υψηλού επιπέδου εκτελείται γραμμή προς γραμμή κατά τη διάρκεια εκτέλεσης του προγράμματος.

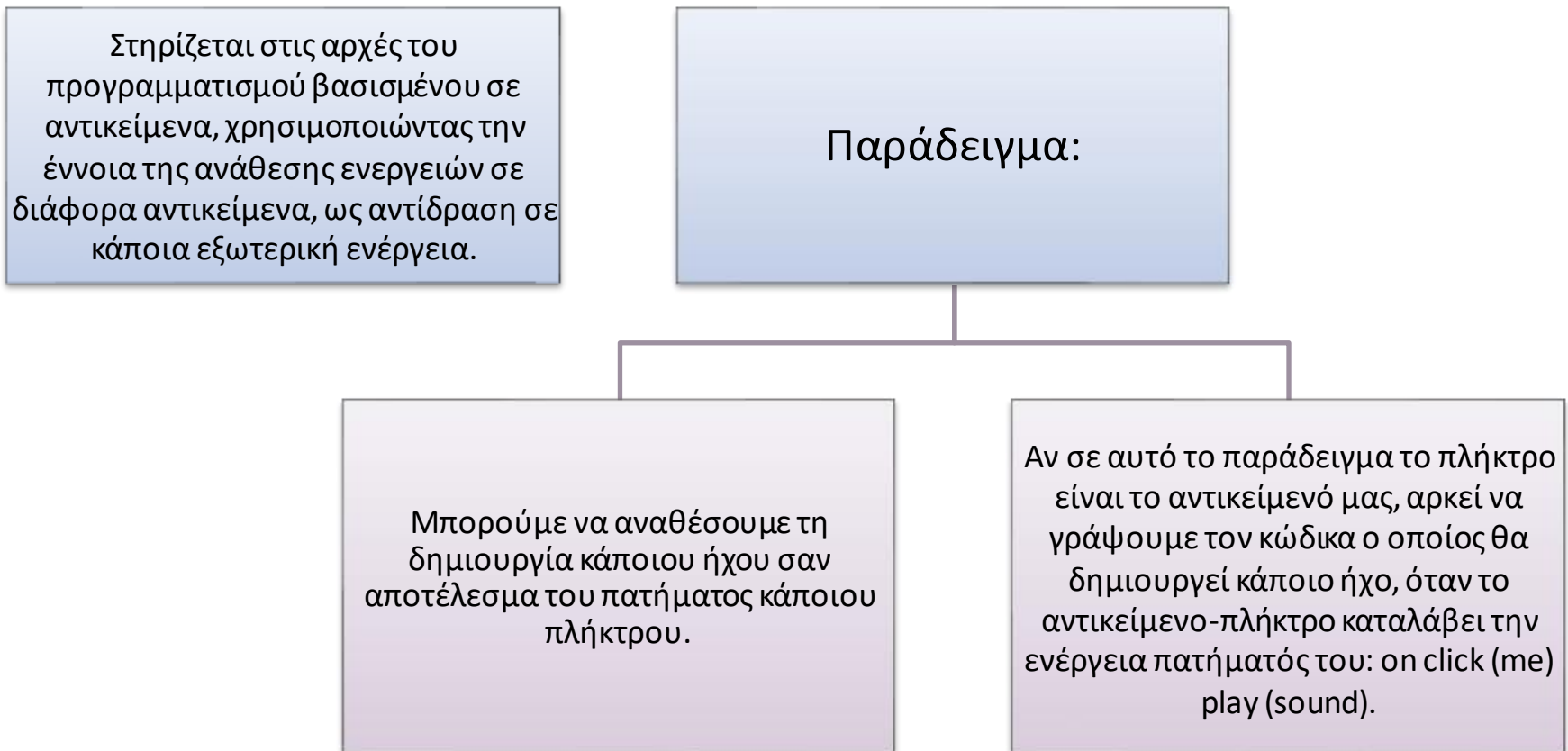
Διαδικασιακός (δομημένος) Προγραμματισμός

Δημιουργία μιας **διαδικασίας** ή αλλιώς ενός συνόλου εργασιών οι οποίες δίνονται με καθορισμένο τρόπο και σειρά στο πρόγραμμα για την εκπλήρωση ενός συγκεκριμένου στόχου.

Μια διαδικασία, από τη στιγμή που θα περιγραφεί μπορεί να χρησιμοποιηθεί και σαν **τμήμα** μιας άλλης διαδικασίας, χτίζοντας τελικά ένα σύνολο ενεργειών το οποίο υλοποιεί τον αλγόριθμο του προγράμματος που θέλουμε να δημιουργήσουμε.

Οι περισσότερες γλώσσες προγραμματισμού που χρησιμοποιούνται σήμερα για εφαρμογές γενικού τύπου είναι **γλώσσες διαδικασιακές**.

Προγραμματισμός βασισμένος σε γεγονότα (Event Driven Programming)



Αντικειμενοστραφής Προγραμματισμός (Object Oriented Programming)

Βασίζεται στη λογική ότι ένα πρόγραμμα είναι ένα σύνολο από **αντικείμενα** τα οποία επικοινωνούν και παράγουν τα επιδιωκόμενα αποτελέσματα.

Τα αντικείμενα αυτά έχουν δύο βασικά στοιχεία: **μεθόδους** και **ιδιότητες** (δεδομένα).

Η δημιουργία των προγραμμάτων βασίζεται στην εύρεση ή δημιουργία των **κατάλληλων** αντικειμένων και του ορισμού του τρόπου **λειτουργίας** και **αλληλεπίδρασης** μεταξύ τους για την επίτευξη του στόχου του προγράμματος.

Με τη χρήση των αντικειμένων υπάρχει η δυνατότητα για την **διανομή** των αντικειμένων και την **επαναχρησιμοποίησή** τους σε άλλα προγράμματα, διευκολύνοντας έτσι τη διαδικασία του προγραμματισμού.

Αντικειμενοστραφής Προγραμματισμός

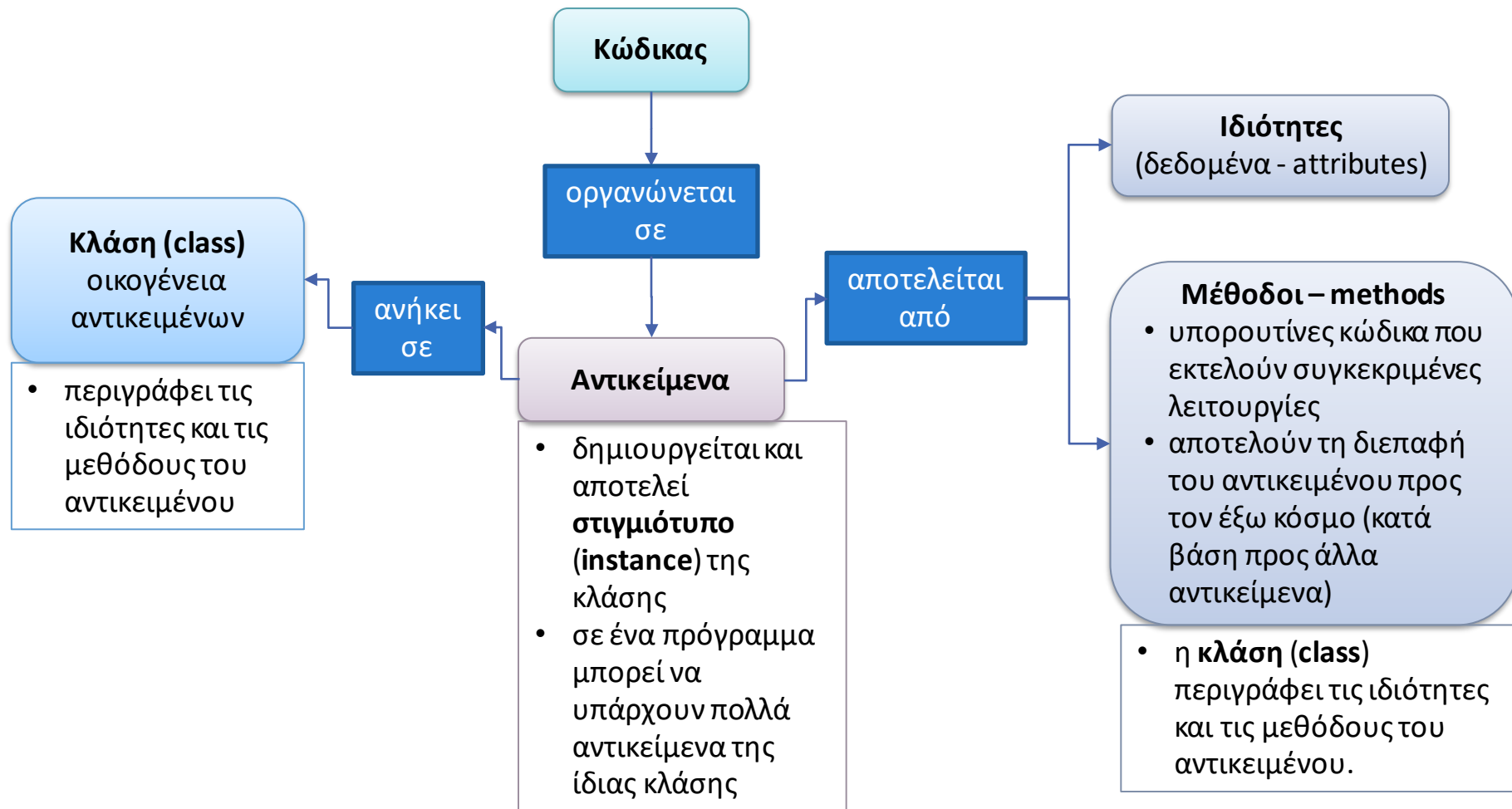
Τι σημαίνει ;

- Ο κώδικας είναι οργανωμένος σε κλάσεις αντικειμένων.
- Δημιουργία αντικειμένων βασισμένων στις παραπάνω κλάσεις.
- Δεν υπάρχουν:
 - Μεμονωμένες συναρτήσεις.
 - Μεμονωμένα τμήματα κώδικα.
- Η εκτέλεση του προγράμματος πραγματοποιείται με αρχικοποίηση αντικειμένων που ανταλλάσσουν μηνύματα (δυνατότητα για προγραμματισμό βασισμένο σε γεγονότα= event driven).

Αντικειμενοστραφείς γλώσσες προγραμματισμού

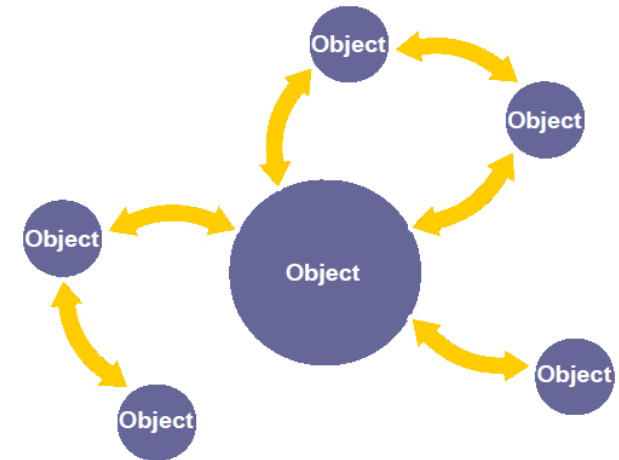


Αντικείμενα, Κλάσεις, Μέθοδοι, Ιδιότητες



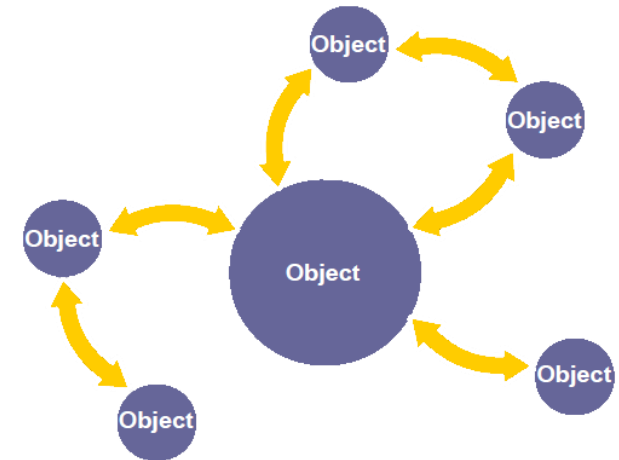
Διαδικαστικές Vs Αντικειμενοστραφείς

- Διαδικαστικές Γλώσσες
 - Διαβάζουν μία γραμμή τη φορά
 - η γλώσσα C είναι διαδικαστική
- Αντικειμενοστραφείς Γλώσσες
 - Διαβάζουν μία γραμμή τη φορά
 - Μοντελοποιούν αντικείμενα μέσω κώδικα
 - Δίνουν έμφαση στην αλληλεπίδραση αντικειμένων
 - Επιτρέπουν την αλληλεπίδραση χωρίς προκαθορισμένη σειρά
 - Η Java και η C ++ είναι αντικειμενοστραφείς γλώσσες



Προγραμματίζοντας με αντικείμενα

- Όταν έχετε μια ιδέα ή απαίτηση για ένα πρόγραμμα ...
 - Εξετάστε το είδος των αντικειμένων που μπορεί να υπάρχουν σε αυτό το πρόγραμμα.
 - Εξετάστε τις ιδιότητες και τις συμπεριφορές αυτών των τύπων αντικειμένων.
 - Εξετάστε πως αλληλοεπιδρούν τα αντικείμενα.



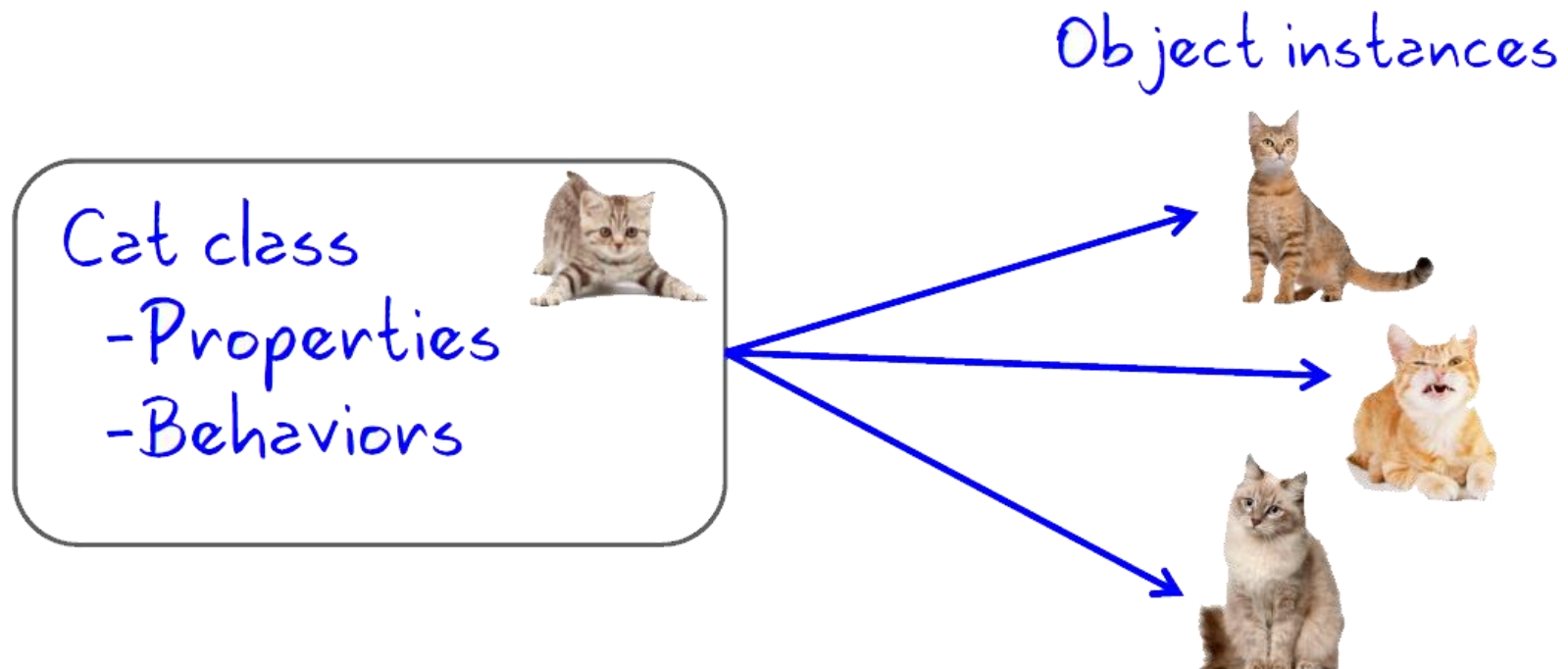
Παράδειγμα

- Ιδιότητες:
 - Όνομα
 - Ηλικία
 - Ράτσα
 - Αγαπημένο φαγητό
- Συμπεριφορές:
 - νιαούρισμα
 - Παίζω
 - Πλύση
 - Τρώω
 - Κυνήγι



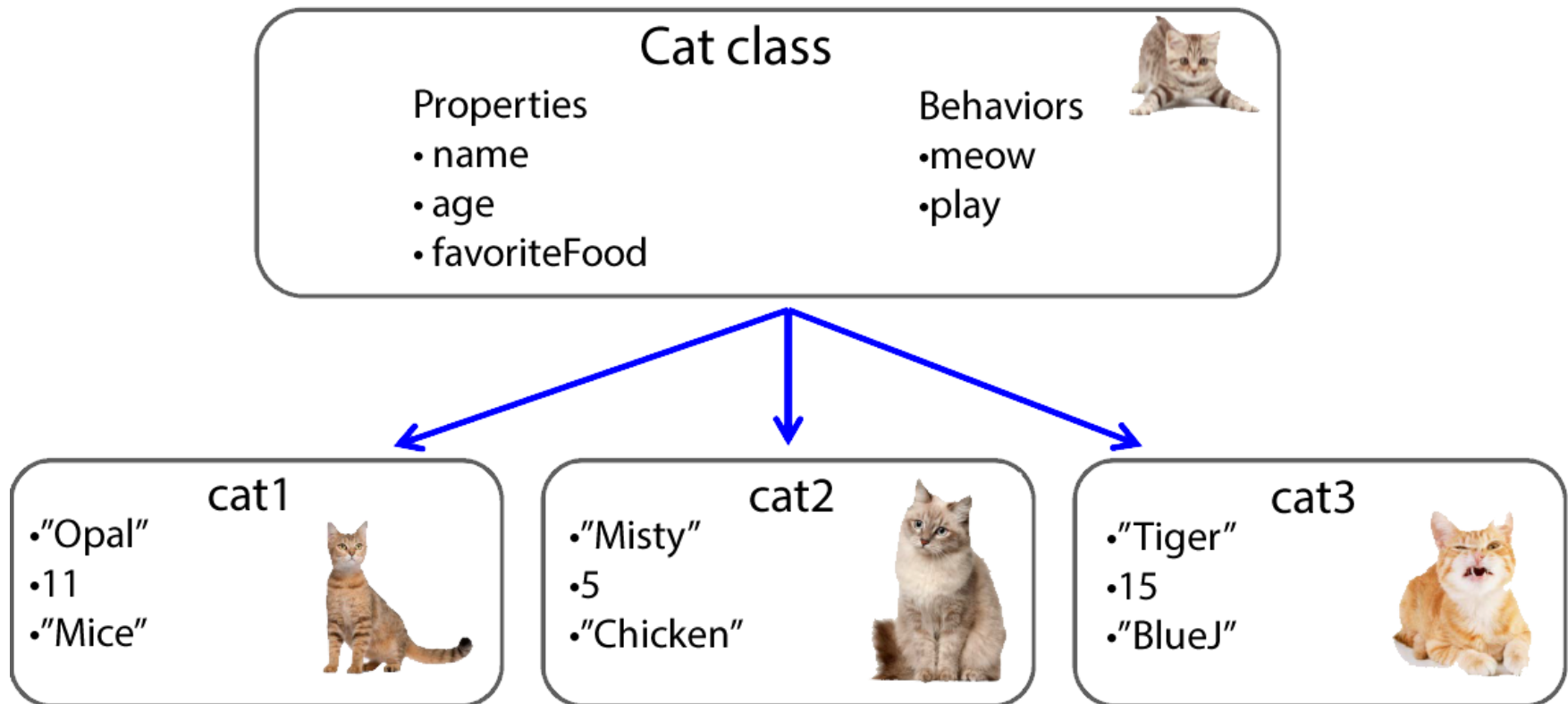
Κλάσεις (Classes) & Στιγμιότυπα (Instances)

- Κλάση = Ιδιότητες (properties) + Συμπεριφορές (behaviors)
- Η κλάση χρησιμοποιείται ως **Σχέδιο** ή **Συνταγή** ή «**Καλούπι**» για τη δημιουργία στιγμιότυπων αντικειμένων.



Κλάσεις (Classes) & Στιγμιότυπα (Instances)

- Όλα τα στιγμιότυπα «γάτας» μπορούν να νιαουρίζουν, παίζουν, τρώνε.



Μεθοδολογία ανάπτυξης προγραμμάτων σε αντικειμενοστραφές περιβάλλον

Μελετάμε το
πρόβλημα

Το αποσυνθέτουμε σε
κλάσεις που αντιστοιχούν
σε οντότητες του
πραγματικού κόσμου

- Επαναχρησιμοποίηση
έτοιμων κλάσεων
- Κάθε κλάση έχει μεθόδους
οι οποίες καλούν μεθόδους
άλλων κλάσεων.

Δημιουργούμε
αντικείμενα ανάλογα
με το πρόγραμμα που
θέλουμε να
αναπτύξουμε και τα
βάζουμε να
αλληλεπιδρούν

Χαρακτηριστικά Αντικειμενοστραφών Γλωσσών

Αφαίρεση (Abstraction)

- Ο προγραμματιστής δουλεύει σε υψηλό επίπεδο, με λιγότερες λεπτομέρειες και σαφώς καθορισμένες διεπαφές

Ενθυλάκωση (Encapsulation)

- Προστασία των δεδομένων και των μεθόδων

Κληρονομικότητα (Inheritance)

- Επιτρέπει την δημιουργία νέων κλάσεων βάσει προηγούμενων

Πολυμορφισμός (Polymorphism)

- Χρήσιμη ιδιότητα που πρέπει να υπάρχει σε κάθε αντικειμενοστραφή γλώσσα και επιτρέπει την αλλαγή της συμπεριφοράς κλάσεων

Οφέλη αντικειμενοστραφούς προγραμματισμού

- Ευκολότερη συντήρηση του κώδικα.
- Βελτίωση της αξιοπιστίας.
 - Αλλαγές στην εσωτερική δομή μιας μεθόδου δεν επηρεάζουν τυχόν άλλες κλάσεις που τυχόν την χρησιμοποιούν.
 - Ομοίως αλλαγές στην εσωτερική δομή ενός αντικειμένου επηρεάζουν μόνο το εν λόγω αντικείμενο.
- Περισσότερες δυνατότητες για επαναχρησιμοποίηση κώδικα.
- Δυνατότητα για εργασία σε ομάδες.
 - Δεν απαιτείται η μια ομάδα ανάπτυξης να ξέρει τη δομή των κλάσεων που δημιουργούνται από μια άλλη ομάδα. Αρκεί να ξέρει ποιες μεθόδους να καλέσει.

Ενότητα 2β: Εισαγωγή στη Java

1. Ιστορία Java
2. Προϊόντα & Τεχνολογίες Java
3. Platform-Dependent και Cross-Platform Independent γλώσσες
4. Java Runtime Environment (JRE)
5. Java Virtual Machine (JVM)
6. Java Development Kit (JDK)
7. Integrated Development Environment (IDE)
8. Διαδικασίας Εκτέλεσης προγράμματος Java
9. Ορολογία-λεξικό Java



Γιατί Java;

- Παγκόσμιο πρότυπο για την ανάπτυξη «ενσωματωμένων» και «φορητών» εφαρμογών, παιχνιδιών, περιεχόμενο Ιστού αλλά και επιχειρησιακό λογισμικό.
 1. Object-oriented
 2. Cross-platform
 3. Internet/Web based
 4. Free-Open Source
 5. Reusable



Java@Desktops

- Java ~ 1,1 δις
- JRE ~ 930 εκ. (downloads/year)
- JDK ~ 9,5 εκ. (downloads/year)



Java@Mobiles

- ~3 εκ. συσκευές



Java TV & Card

- 100% blu-ray players



#1 Development Platform in the Cloud

~30 Χρόνια Java

- 1995: Πρώτη έκδοση της Java



Ιστορία

- Ξεκίνησε το 1990 με το όνομα OAK από την εταιρεία Sun Microsystems
 - Διασύνδεση συσκευών με διαφορετικές CPU
 - Σχεδιάστηκε έχοντας κατά νου τη διαδραστική τηλεόραση!
 - Αρχηγός της ομάδας ανάπτυξης (Green Team) ήταν ο James Gosling
- Η έναρξη του WWW της έδωσε μεγάλη ώθηση
 - Ανάπτυξη Web Multimedia Components για ιστοσελίδες (Java Applets - small applications)
- 1995 → Java
 - Ενσωματώθηκε στον Netscape Navigator
- "Write Once, Run Anywhere" (WORA)
- Το όνομα προέρχεται:
 - Από μια ποικιλία καφέ που πουλούσε η τοπική καφετέρια (?)
 - Από κάποια από τα ονόματα των δημιουργών της γλώσσας (?): (James Gosling, Arthur Van Hoff, Andy Bechtolsheim)
- Πέρασε στην κυριότητα της Oracle το 2010



James Gosling
*The Father of
Java*

Java version history

Release	Year
JDK Beta	1995
JDK 1.0	1996
JDK 1.1	1997
J2SE 1.2	1998
J2SE 1.3	2000
J2SE 1.4	2002
J2SE 5.0	2004
Java SE 6	2006
Java SE 7	2011
Java SE 8	2014
Java SE 9	2017
Java SE 10 (18.3)	2018



ORACLE®



Duke
the Java Maskot

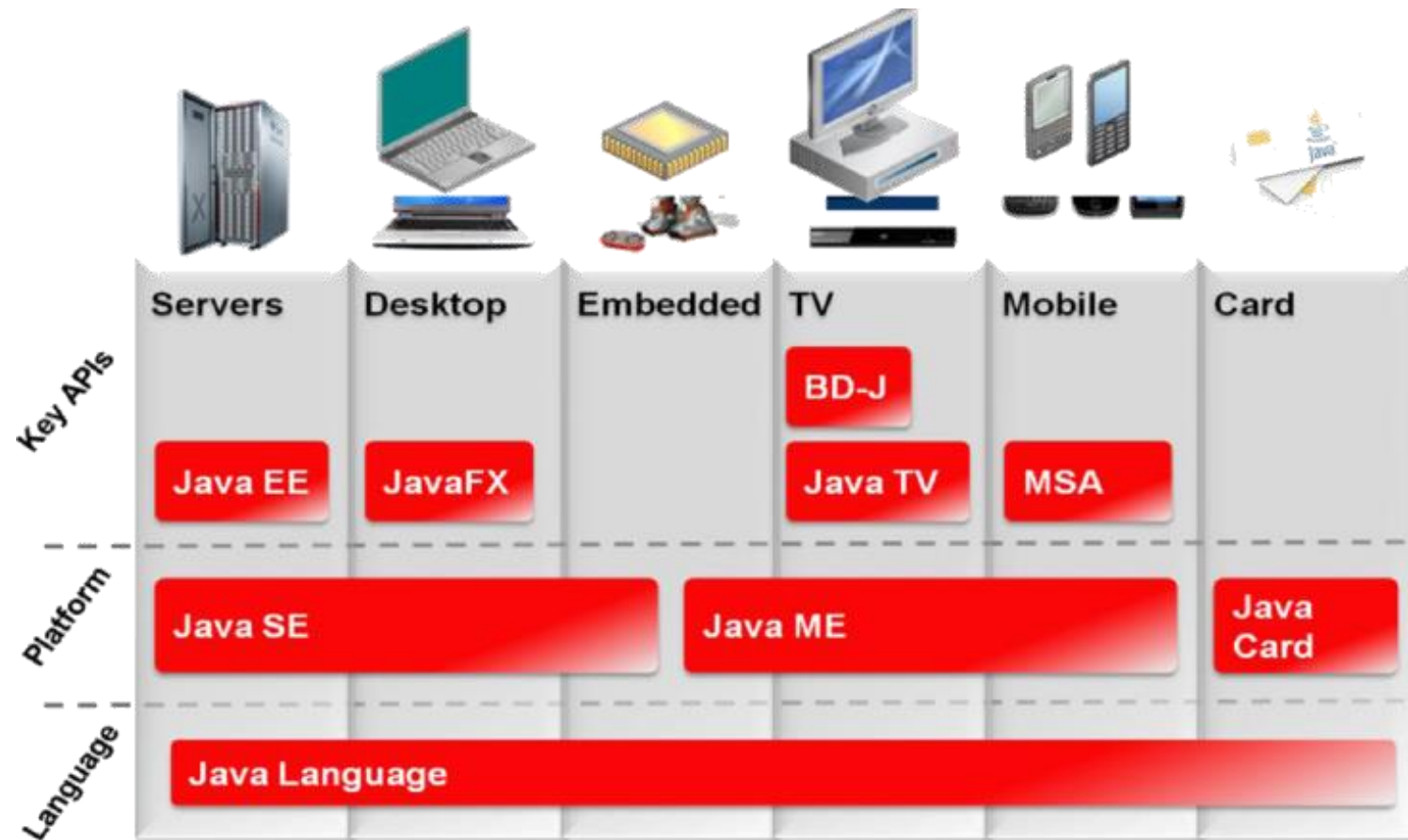
Source: https://en.wikipedia.org/wiki/Java_version_history

Ομάδες προϊόντων Java

1. Java Platform, Standard Edition (Java SE): Για desktops
2. Java Platform, Enterprise Edition (Java EE): Για μεγάλες επιχειρησιακές server-side/client-side καταναεμημένες εφαρμογές π.χ. για e-commerce εταιρίας
3. Java Platform, Micro Edition (Java ME): Για κινητά, PDAs, Smart TV, Smart cards, Raspberry Pi, κ.α.
4. Java Card: Για έξυπνες κάρτες (ταυτότητα, ασφάλεια, συναλλαγές, κάρτες SIM)



Συσκευές & Προϊόντα Java

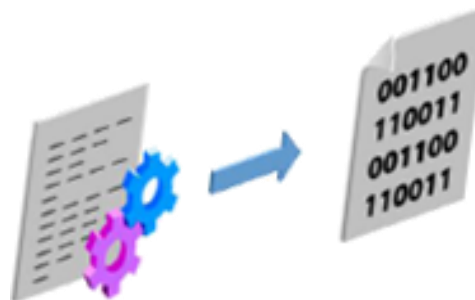


Notes: **BD-J:** Blu-ray Disk Java
MSA: Mobile Service Architecture

Java Programming

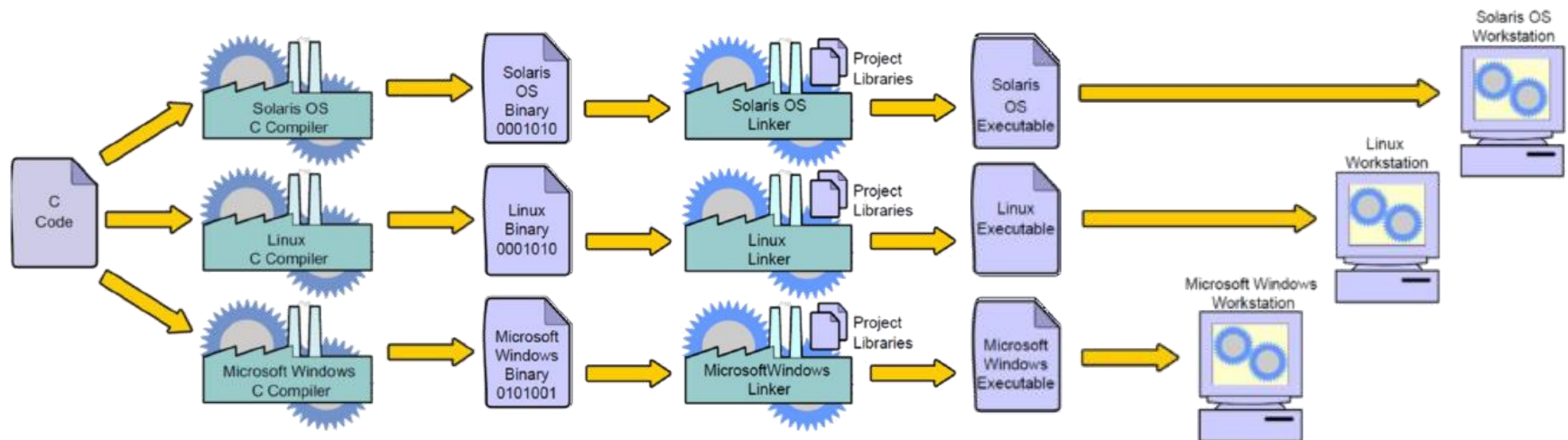
Τι είναι ένα πρόγραμμα;

- Σύνολο εντολών το οποίο τρέχει σε έναν Η/Υ ή οποιαδήποτε άλλη ψηφιακή συσκευή
 - Σε επίπεδο μηχανής → δυαδικές εντολές (11001010001,...)
(Κώδικας μηχανής)
- Τα περισσότερα προγράμματα είναι γραμμένα σε γλώσσα υψηλού επιπέδου (high-level code)
 - Πρέπει να μεταφραστούν σε γλώσσα μηχανής



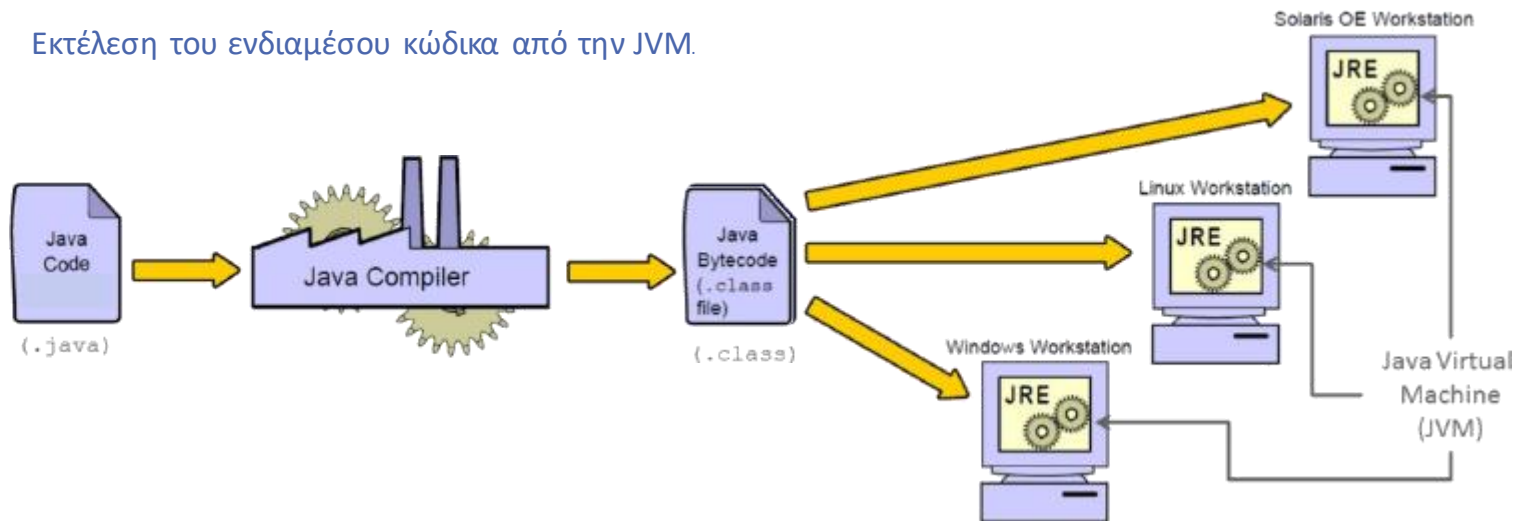
Platform-dependent γλώσσες

- Συνήθως οι γλώσσες (π.χ. C++) απαιτούν μεταγλώττιση (compiling) και σύνδεση (linking) του προγράμματος μέσω ενός εκτελέσιμου προγράμματος (executable) το οποίο εξαρτάται από την εκάστοτε πλατφόρμα (platform-dependent)



Java: Cross-Platform Independent γλώσσα

- Η Java εκτελείται σε έναν υποθετικό υπολογιστή που ονομάζεται Java Virtual Machine (JVM)
- Βήματα δημιουργίας και εκτέλεσης προγραμμάτων στη Java
 - Συγγραφή κώδικα.
 - Μεταγλώττιση σε ενδιάμεσο κώδικα (bytecode).
 - Εκτέλεση του ενδιάμεσου κώδικα από την JVM.



Java Runtime Environment (JRE)



- Η Java Virtual Machine (JVM):
 - Διερμηνεύει τον κώδικα Java
 - Φορτώνει τις κλάσεις Java (classes)
 - Εκτελεί τα προγράμματα Java
- Ένα πρόγραμμα Java, χρειάζεται επιπλέον ένα σύνολο από στάνταρ Java class βιβλιοθήκες (libraries)

Java Runtime Environment (JRE)


Java Runtime Environment (JRE)



- Περιλαμβάνει
 - Java Virtual Machine (JVM)
 - Java class libraries
- Σκοπός
 - Διαβάζει bytecode (.class)
 - Τρέχει τον ίδιο κώδικα (bytecode) παντού με JVM

Java Development Kit (JDK)



- Περιλαμβάνει
 - JRE 
 - Java Compiler
 - Πρόσθετα εργαλεία
- Σκοπός
 - Μεταγλωττίζει (compile) bytecode (.java → .class)

Integrated Development Environment (IDE)

- Παρέχει
 - Εξειδικευμένο επεξεργαστή κειμένου (text editor)
 - Εντοπισμό σφαλμάτων (debugging)
 - Διαχείριση Projects

- Παραδείγματα



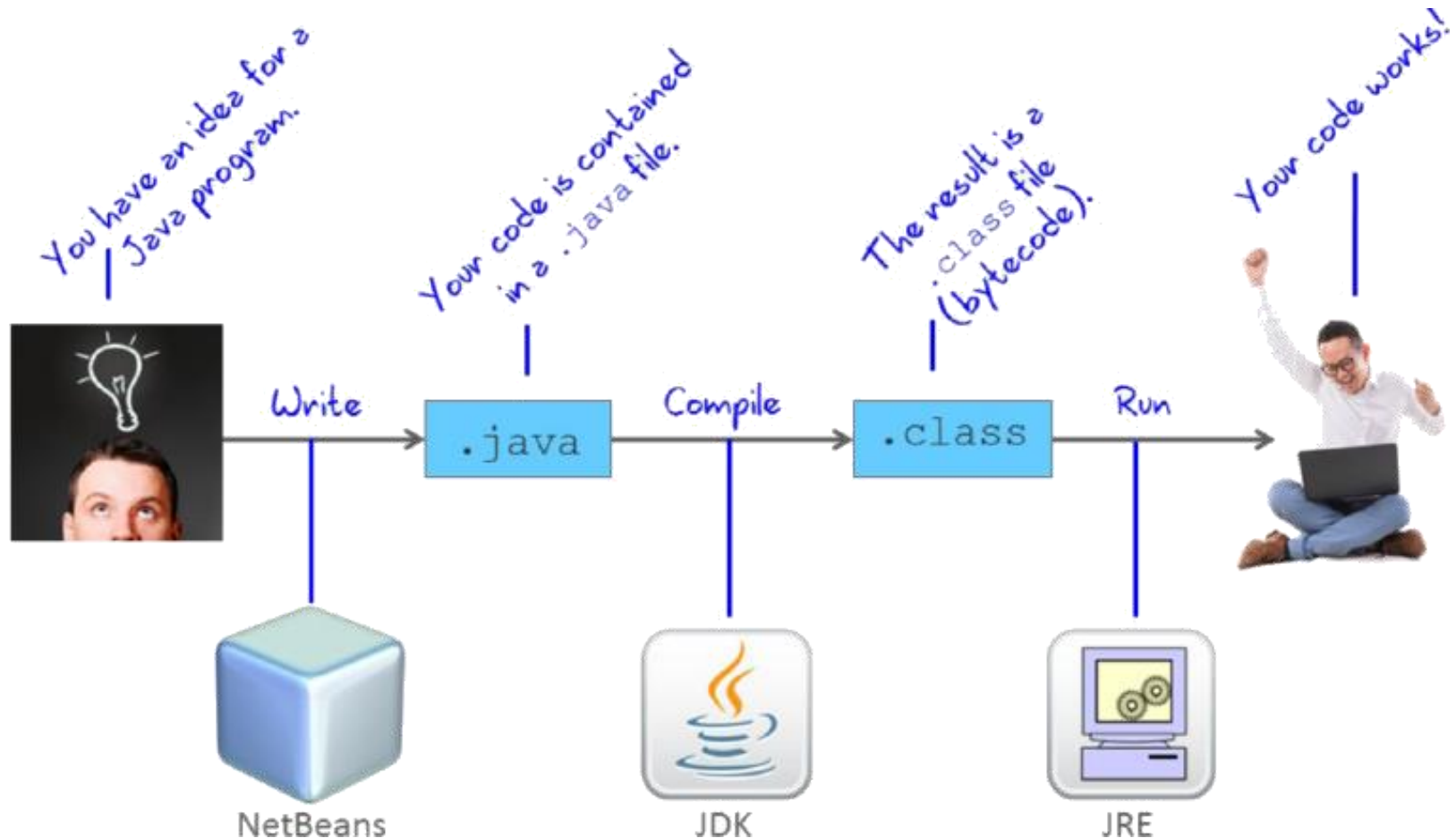
Κατηγορίες Προγραμμάτων Java

- **Java applications** – Αυτόνομα προγράμματα.
- **Java applets** – Προγράμματα ενσωματωμένα σε Ιστοσελίδα.
- **Java servlets** – Προγράμματα που εκτελούνται από διακομιστές Παγκοσμίου Ιστού (Web Servers) στο πλαίσιο εφαρμογών client-server.
- **Enterprise Java Beans** – Προγράμματα για ανάπτυξη εταιρικών εφαρμογών με την αρχιτεκτονική τριών επιπέδων (3-tier) J2EE.
- **Java για κινητά**
 - J2ME
- **JavaCard**
 - Έξυπνες κάρτες που προγραμματίζονται σε Java
 - Πιστωτικές
 - ATM
 - Ηλεκτρονικές κλειδαριές κ.α.

Ένα απλό πρόγραμμα σε Java: Hello.java

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

Εκτέλεση προγράμματος Java



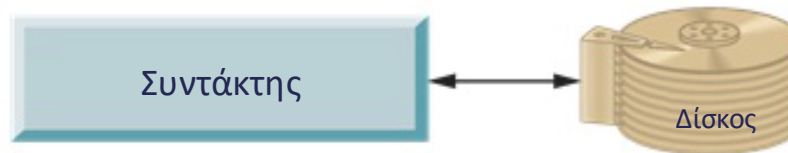
Εκτέλεση προγράμματος Java

- Συνήθως τα προγράμματα στη Java ακολουθούν πέντε στάδια:



Εκτέλεση προγράμματος Java

Στάδιο 1:
Επεξεργασία



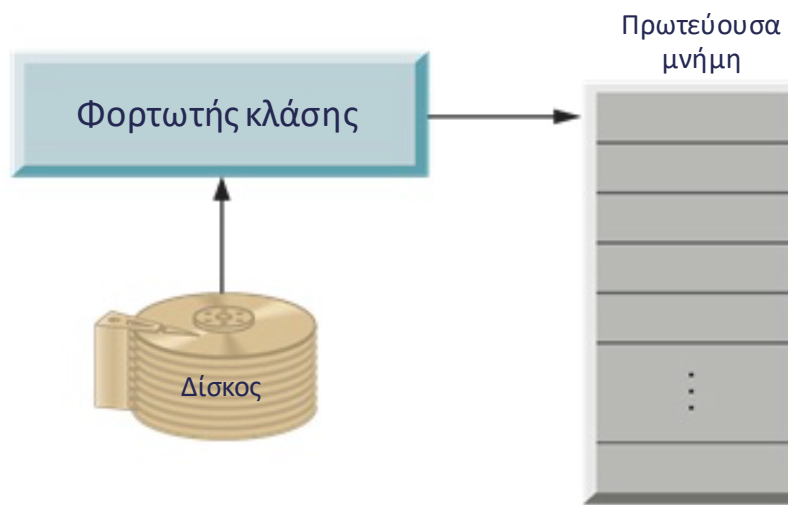
Το πρόγραμμα δημιουργείται με έναν συντάκτη και αποθηκεύεται στον δίσκο σε ένα αρχείο με κατάληξη **.java**

Στάδιο 2:
Μεταγλώττιση



Ο μεταγλωττιστής δημιουργεί bytecodes και τα αποθηκεύει στον δίσκο σε ένα αρχείο με κατάληξη **.class**

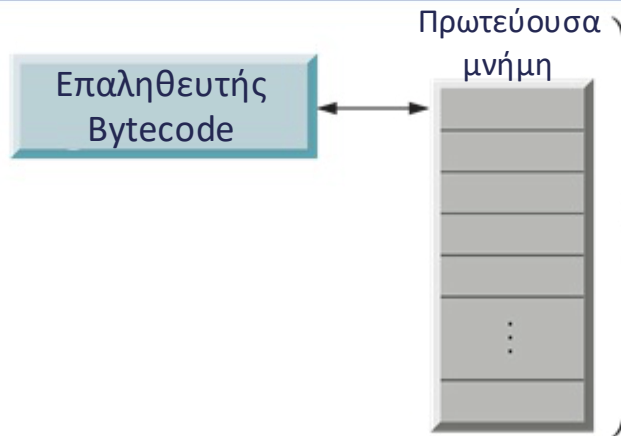
Στάδιο 3:
Φόρτωση



Ο φορτωτής κλάσης διαβάζει τα **.class** αρχεία που περιέχονται bytecodes από τον δίσκο και μεταφέρει τα bytecodes αυτά στη μνήμη

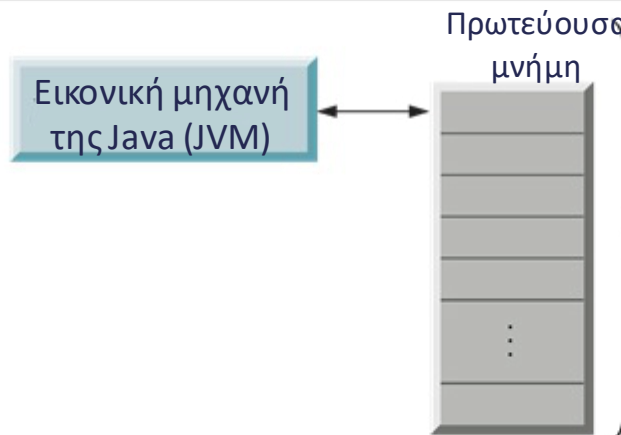
Εκτέλεση προγράμματος Java

Στάδιο 4:
Επαλήθευση



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

Στάδιο 5:
Εκτέλεση



Για την εκτέλεση του προγράμματος η εικονική μηχανή της Java (JVM) διαβάζει τα bytecodes και ακριβώς όταν χρειάζεται (just-in-time, JIT) τα μεταγλωττίζει (μεταφράζει) σε μια γλώσσα που ο υπολογιστής μπορεί να καταλάβει. Κατά τη διάρκεια της εκτέλεσης του προγράμματος, αποθηκεύει δεδομένα στην κύρια μνήμη.

Δημιουργία κώδικα Java

- Το στάδιο 1 αποτελείται από την επεξεργασία ενός αρχείου με ένα πρόγραμμα σύνταξης (ή απλά συντάκτης).
 - Γράφουμε ένα πρόγραμμα σε Java (πηγαίος κώδικας) με τον συντάκτη (editor)
 - Πραγματοποιούμε τις απαραίτητες διορθώσεις
 - Σώζουμε το πρόγραμμα

Ένα αρχείο με την κατάληξη .java υποδεικνύει ότι το αρχείο περιέχει πηγαίο κώδικα Java

- Συντάκτες σε Linux: vi, emacs, jedit, geany, gedit ...
- Συντάκτες σε Windows: Notepad, Notepad++ (www.notepad-plus-plus.org), EditPlus (www.editplus.com), TextPad (www.textpad.com), jEdit (www.jedit.org) ...

Μεταγλώττιση κώδικα σε Java

- Στάδιο 2
 - Με χρήση της εντολής `javac` (ο μεταγλωττιστής Java) μεταγλωττίζουμε το πρόγραμμα. Παράδειγμα: για μεταγλώττιση του προγράμματος με όνομα `Welcome.java`, πληκτρολογούμε

`javac Welcome.java`

- Αν το πρόγραμμα μεταγλωττιστεί επιτυχώς, δημιουργείται ένα `.class` αρχείο με όνομα `Welcome.class` που περιέχει την μεταγλωττισμένη έκδοση του προγράμματος.

Μεταγλώττιση κώδικα σε Java

- Ο μεταγλωττιστής μεταφράζει τον πηγαίο κώδικα Java σε bytecodes που αντιπροσωπεύουν τις διεργασίες προς εκτέλεση.
- Τα bytecodes εκτελούνται από την Εικονική Μηχανή της Java (JVM)
 - Τα Bytecodes είναι ανεξάρτητα από την πλατφόρμα
 - Τα Bytecodes είναι φορητά - μπορούν να εκτελεστούν σε οποιαδήποτε πλατφόρμα περιέχει μία εικονική μηχανή της Java (JVM) που καταλαβαίνει την έκδοση της Java στην οποία μεταγλωττίστηκαν.
- Η JVM επικαλείται από την εντολή `java` . Για παράδειγμα για να εκτελέσουμε μία Java εφαρμογή με όνομα `Welcome`, θα πληκτρολογήσουμε την εντολή:

`java Welcome`

Φόρτωση κλάσεων σε Java

- Στάδιο 3
 - Η JVM μεταφέρει το πρόγραμμα στην μνήμη για εκτέλεση
 - Ο φορτωτής των κλάσεων (class loader) λαμβάνει τα .class αρχεία που περιέχουν τα bytecodes του προγράμματος και τα μεταφέρει στην κύρια μνήμη.
 - Φορτώνει επίσης όσα .class αρχεία παρέχονται από την Java τα οποία χρησιμοποιεί το πρόγραμμά μας.
 - Τα .class αρχεία μπορούν να φορτωθούν από έναν δίσκο στο σύστημά μας ή και μέσω δικτύου.

Επαλήθευση bytecode σε Java

- Στάδιο 4
 - Καθώς οι κλάσεις φορτώνονται, ο επαληθευτής bytecode (Bytecode verifier) εξετάζει τα bytecodes τους: Εξασφαλίζει ότι είναι έγκυρα και δεν παραβιάζουν τους περιορισμούς ασφαλείας της Java.
 - Η Java επιβάλλει ισχυρή ασφάλεια ώστε να εξασφαλίσει ότι τα προγράμματα Java που φθάνουν από το δίκτυο δεν βλάπτουν τα αρχεία του συστήματός μας (όπως κάνουν οι ιοί στους υπολογιστές).

Σφάλματα στη Java

- Συντακτικά σφάλματα – Προκύπτουν όταν παραβιάζουμε τους κανόνες δόμησης του προγράμματος.
 - Ανιχνεύονται πάντοτε από το πρόγραμμα μεταγλώττισης.
- Σφάλματα εκτέλεσης – Λάθη εξαιτίας των οποίων το πρόγραμμα παράγει λανθασμένα αποτελέσματα ή διακόπτεται.
 - Η διακοπή εκτέλεσης στην Java συνοδεύεται από τη δημιουργία κατάστασης εξαίρεσης (exception).
- Λογικά σφάλματα – Λάθη στη λογική δημιουργίας του προγράμματος
 - Οδηγούν σε συμπεριφορά του προγράμματος αντίθετη με την αναμενόμενη.
 - Μπορεί να μην οδηγήσουν σε κάποιο πρόβλημα για αρκετό καιρό, και να διαφύγουν της προσοχής μας.

Το «λεξικό» της γλώσσας

abstract	assert	boolean	break	byte	case
catch	char	class	const	continue	default
double	do	else	enum	extends	false
final	finally	float	for	goto	if
implements	import	instanceof	int	interface	long
native	new	null	package	private	protected
public	return	short	static	strictfp	super
switch	synchronized	this	throw	throws	transient
true	try	void	volatile	while	

Το «λεξικό» της γλώσσας

abstract	assert	boolean	break	byte	case
catch	char	class	const	continue	default
double	do	else	enum	extends	false
final	finally	float	for	goto	if
implements	import	instanceof	int	interface	long
native	new	null	package	private	protected
public	return	short	static	strictfp	super
switch	synchronized	this	throw	throws	transient
true	try	void	volatile	while	

String?

Ονομασία μεταβλητών

- Ονόματα έως 255 χαρακτήρες
- Μπορούν να αρχίζουν από γράμμα, \$, _ (_hello, \$hello)
- Υποστηρίζονται Unicode χαρακτήρες (안녕하세요)
- Δεν μπορούμε να χρησιμοποιήσουμε κενά ή σύμβολα (hello!, hello there)
- Είναι case sensitive
- Δεν μπορούμε να χρησιμοποιήσουμε δεσμευμένες λέξεις

Χρήση ονομάτων στη Java

- Τα ονόματα των Packages πρέπει να είναι με μικρούς χαρακτήρες (lowercase).
 - Σε μικρά projects με λίγα packages μπορούμε να δίνουμε μικρά (αλλά πάντα με νόημα) ονόματα: `package pokeranalyzer`, `package mycalculator`
 - Σε εταιρίες λογισμικού και μεγάλα projects όπου τα packages μπορεί να εισαχθούν σε άλλες κλάσεις, τα ονόματα κανονικά υποδιαίρονται. Συνήθως ξεκινάμε με το όνομα της εταιρίας πριν περάσουμε σε διαίρεση σε layers ή features: `package com.mycompany.utilities`, `package org.bobscompany.application.userinterface`
- Κλάσεις: Τα ονόματα πρέπει να είναι σε CamelCase.
 - Προσπαθούμε να χρησιμοποιούμε ουσιαστικά, αφού μια κλάση συνήθως εκφράζει κάτι από τον πραγματικό κόσμο: `class Customer`, `class Account`
- Διεπαφές: Τα ονόματα πρέπει να είναι σε CamelCase.
 - Συνήθως έχουν ένα όνομα που περιγράφει την πράξη που μπορεί να πραγματοποιηθεί από την κλάση: `interface Comparable`, `interface Enumerable`
 - Σημείωση: Κάποιοι προγραμματιστές προτιμούν να ξεχωρίζουν τις διεπαφές, ξεκινώντας το όνομα με "I": `interface IComparable`, `interface IEnumerable`

Χρήση ονομάτων στη Java

- Μέθοδοι: Τα ονόματα πρέπει να είναι σε mixed case.
 - Χρησιμοποιούμε ρήματα για να περιγράψουμε το τι κάνει η μέθοδος:
`void calculateTax()`, `string getSurname()`.
- Μεταβλητές: Τα ονόματα πρέπει να είναι σε mixed case.
 - Τα ονόματα πρέπει να εκφράζουν τις τιμές που παίρνει η μεταβλητή:
`string firstName`, `int orderNumber`.
 - Χρησιμοποιούμε πολύ μικρά ονόματα όταν οι μεταβλητές έχουν πολύ μικρό χρόνο ζωής, όπως μέσα σε for loops: `for (int i=0; i<20;i++)`.
- Σταθερές: Τα ονόματα πρέπει να είναι με κεφαλαίους χαρακτήρες:
`static final int DEFAULT_WIDTH`, `static final int MAX_HEIGHT`.

Σε αυτήν την ενότητα μιλήσαμε για:

1. Βασικές έννοιες του Προγραμματισμού
2. Γενιές γλωσσών προγραμματισμού
3. Είδη Προγραμματισμού
4. Αντικειμενοστραφής προγραμματισμός
5. Ιστορία Java
6. Προϊόντα & Τεχνολογίες Java
7. Platform-Dependent και Cross-Platform Independent γλώσσες
8. Java Runtime Environment (JRE)
9. Java Virtual Machine (JVM)
10. Java Development Kit (JDK)
11. Integrated Development Environment (IDE)
12. Διαδικασίας Εκτέλεσης προγράμματος Java
13. Ορολογία-λεξικό Java

Βιβλιογραφία

- Harvey Deitel, Paul Deitel, «Java Προγραμματισμός, 8η έκδοση», Γκιούρδας, 2010.
- Γιώργος Λιακέας, «Εισαγωγή στη Java», Κλειδάριθμος, 2009.
- Χρήστος Κυτάγιας, Κώστας Δ. Κυτάγιας, Γιώργος Πρεζεράκος, Δημήτρης Κυτάγιας, «Αντικειμενοστραφής Προγραμματισμός με Java», Σύγχρονη Εκδοτική, 2013.
- Wanda Dann, Stephen Cooper, and Randy Pausch, «Learning to program with Alice» 3rd edition, Pearson Education, 2012.
- Michael Kölling, «Introduction to Programming with Greenfoot».
- «Object-Oriented Programming in Java with Games and Simulations», Pearson Education, August 2009.
- Οι σημειώσεις είναι βασισμένες και σε υλικό του καθηγητή του ΤΕΙ Πειραιά Γ. Πρεζεράκου.
- Oracle Academy/Java Foundations/Lesson 1 – Introduction
<http://academy.oracle.com>



Δρ. Μιχάλης Φειδάκης
m.feidakis@uniwa.gr

Καθ. Χαράλαμπος Ζ. Πατρικάκης
bpatr@uniwa.gr

