



ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΠΑΤΡΩΝ  
UNIVERSITY OF PATRAS

ΑΝΟΙΚΤΑ ακαδημαϊκά  
μαθήματα ΠΠ

# Οντοκεντρικός Προγραμματισμός

Ενότητα 5: Η ΓΛΩΣΣΑ C++  
**Διαφορές Java και C++**

**ΔΙΔΑΣΚΟΝΤΕΣ:** Ιωάννης Χατζηλυγερούδης, Χρήστος  
Μακρής

Πολυτεχνική Σχολή

Τμήμα Μηχανικών Η/Υ & Πληροφορικής

# ΔΙΑΦΟΡΕΣ JAVA – C++

# Διαφορές C++ - Java

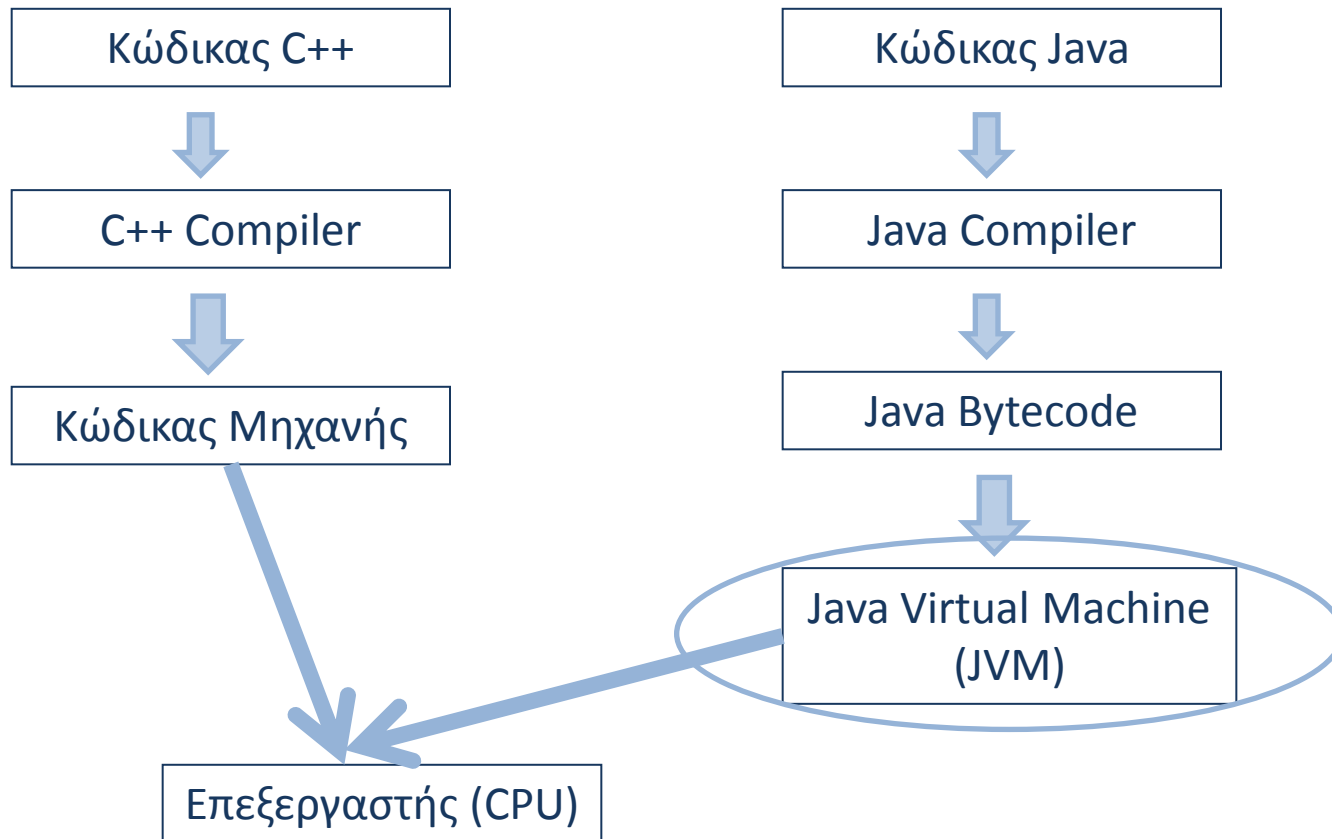
---

- Η Java δημιουργήθηκε (1991) αρκετά αργότερα από την C++ (1980+) και στηρίχθηκε σε μεγάλο βαθμό επάνω σε αυτήν.
- Για το λόγο αυτό το συντακτικό είναι σχεδόν παρόμοιο.
- Ωστόσο υπάρχουν **αρκετές διαφορές** μεταξύ της C++ και της Java



# Διαφορές C++ - Java

1. Η εμφανέστερη διαφορά είναι η ταχύτητα: Η Java που υφίσταται διερμηνεία (interpretation) καταλήγει να είναι **πιο αργή** από τη C++.



# Διαφορές C++ - Java

---

Ωστόσο με σύγχρονες τεχνικές στην κατασκευή μεταγλωττιστών για την Java (just-in-time compilers) και σύγχρονο χρονοπρογραμματισμό (scheduling) στη JVM, η διαφορά κατεβαίνει μέχρι το 10%.

2. Από την άλλη πλευρά η Java είναι ανεξάρτητη υπολογιστικής πλατφόρμας (επεξεργαστής-λειτουργικό).
3. Ο κώδικας Java μπορεί να εκτελεστεί μέσω web, ενώ ο κώδικας C++ όχι.



# Διαφορές C++ - Java

---

4. Στην Java **οτιδήποτε** πρέπει να βρίσκεται μέσα σε μία κλάση. Δεν υπάρχουν καθολικές συναρτήσεις ή καθολικές μεταβλητές.

Αν χρειάζεται το ισοδύναμο των καθολικών, χρησιμοποιούνται **static** μέθοδοι ή δεδομένα μέσα σε μία κλάση.

5. Στην Java δεν υπάρχουν επίσης **structs** ή **unions**
6. Στην Java όλες οι μέθοδοι **ορίζονται** (definitions) **μέσα στο σώμα της κλάσης**. Επομένως, αυτό θα φαινόταν στη C++ με το να δηλώσουμε όλες τις συναρτήσεις ως **inline** (αλλά δεν είναι). (Στη C++ μία member function που ορίζεται εξ'ολοκλήρου μέσα στην κλάση θεωρείται inline)



# Διαφορές C++ - Java

---

7. Οι ορισμοί των κλάσεων είναι περίπου ίδιοι για την Java και τη C++, αλλά στην Java δεν υπάρχει ελληνικό ερωτηματικό στο τέλος. Δεν υπάρχουν “δηλώσεις κλάσεων”, μόνο ορισμοί κλάσεων.
8. Δεν υπάρχει τελεστής εμβέλειας `::` (scope resolution operator) στην Java. Η Java χρησιμοποιεί τον τελεστή `.` για τα πάντα. Π.χ.

C++ `ClassName::staticMethod()` αναφορά σε static μέλος

Java `ClassName.staticMethod()` αναφορά σε static μέλος



# Διαφορές C++ - Java

---

9. Η Java έχει όπως και η C++ ορισμένους στοιχειώδεις τύπους. Στην Java είναι οι: boolean, char, byte, short, int, long, float, double. **Το μέγεθος των τύπων αυτών είναι καθορισμένο και ανεξάρτητο από την πλατφόρμα** (αρνητική επίπτωση στην ταχύτητα)

Ο έλεγχος τύπων (type-checking) είναι πιο "σφιχτός" στην Java. Για παράδειγμα, στην Java δεν μπορώ να έχω παραστάσεις που λειτουργούν ως συνθήκες ελέγχου και δεν είναι boolean. Π.χ. Η

```
if i ... (όπου i ακέραιος)
```

**Δεν είναι σωστή, θα προκαλέσει σφάλμα.**





# Διαφορές C++ - Java

---

10. Ο τύπος `char` στη Java χρησιμοποιεί το σύνολο χαρακτήρων 16-bit Unicode και κατά συνέπεια αναπαριστά τους περισσότερους εθνικούς χαρακτήρες αυτομάτως.
11. Στην Java αλφαριθμητικά μέσα σε `" "` μετατρέπονται αυτομάτως σε αντικείμενα τύπου `String`. Στην C++ τα αλφαριθμητικά δεν είναι αντικείμενα, αλλά πίνακες.

Στην standard C++ library όμως υπάρχει η κλάση `string`



# Διαφορές C++ - Java

---

12. Παρόλο που φαίνονται παρόμοιοι, οι πίνακες στη Java έχουν διαφορετική δομή και συμπεριφορά από ότι στη C++.

Υπάρχει μέλος/μεταβλητή δεδομένων (read-only) `length` που επιστρέφει το μέγεθος και run-time checking παράγει εξαίρεση αν "βγούμε" εκτός ορίων.

Μπορεί να ανατεθεί ένας πίνακας σε έναν άλλο (array handle is copied). Αντίθετα στη C/C++ οι δηλώσεις:

```
int A[3] = {2, 3, 1};
```

```
int B[3] = {3, 4, 5};
```

και η εντολή:

```
A = B;
```

παράγουν σφάλμα



# Διαφορές C++ - Java

---

13. Δεν χρειάζεται στη Java να έχει δηλωθεί κάτι προτού χρησιμοποιηθεί (και συνεπώς δεν χρειάζονται και forward class declarations).

Αν θέλετε να χρησιμοποιήσετε μία κλάση ή μία μέθοδο προτού αυτή οριστεί, απλώς τη χρησιμοποιείτε. Ο μεταγλωττιστής θα βρεί τον ορισμό (ακόμα και αν υπάρχει σε άλλο αρχείο).

14. Η Java δεν έχει preprocessor. Αν θέλετε να χρησιμοποιήσετε κλάσεις μίας βιβλιοθήκης, χρησιμοποιείτε import και το όνομα της βιβλιοθήκης.



# Διαφορές C++ - Java

---

15. Δεν υπάρχουν pointers στη Java με την έννοια της C/C++. Κατά τη δημιουργία ενός αντικειμένου επιστρέφεται πάντοτε μία αναφορά.

Οι αναφορές (reference type) στη C++ δεν είναι το ίδιο: Πρέπει να αρχικοποιηθούν κατά τη δημιουργία τους και δεν μπορούν να "δείχνουν" αλλού στην πορεία.

Στη C++ οι pointers μπορούν να δείχνουν οπουδήποτε στη μνήμη πράγμα επικίνδυνο. Στην Java δεν υπάρχει αριθμητική δεικτών.



# Διαφορές C++ - Java

---

16. Δεν υπάρχει εντολή `goto` στην Java.

Ο μόνος μηχανισμός άλματος χωρίς συνθήκη είναι η εντολή `break label`

17. Η Java δεν έχει templates ή κάποια άλλη υλοποίηση παραμετροποιήσιμων τύπων. Υπάρχει ένα set συλλογών: Vector, Stack και Hashtable που λαμβάνουν αναφορές Object αλλά οι συλλογές αυτές δεν είναι σχεδιασμένες με βάση την απόδοση όπως η C++ Standard Template Library. (Στην Java 1.5 υπάρχουν generics, επιτρέπεται δηλαδή η δήλωση δομών όπου καθορίζεται ο τύπος των στοιχείων της χωρίς να χρειάζεται casting για την προσπέλαση ενός στοιχείου)



# Διαφορές C++ - Java

---

18. Η Java κάνει αυτόματη περισυλλογή αχρησιμοποίητου χώρου μνήμης και αποδέσμευση του (**garbage collector** – automatic memory management).

Στην Java όλα τα αντικείμενα συλλέγονται από τον garbage collector όταν δοθεί η αναφορά null **σε όλες τις αναφορές** του αντικειμένου.

Στη C++ πρέπει να καταμετρώνται όλες οι αναφορές (pointers) με τιμή null και ο καθαρισμός του χώρου μνήμης πρέπει να γίνεται ρητά με την εντολή delete



# Διαφορές C++ - Java

---

19. Η Java έχει ενσωματωμένη υποστήριξη για multithreading. Η κλάση Thread επιτρέπει τη δημιουργία νέων νημάτων (επικαλύπτοντας την run() μέθοδο).
20. Τα προσδιοριστικά ορατότητας (public, private, protected) δεν εφαρμόζονται στην Java σε μπλοκ κώδικα. Κάθε δήλωση μέλους μιας κλάσης λαμβάνει το δικό της προσδιοριστικό. (Αν δεν υπάρχει προσδιοριστικό τα μέλη είναι προσπελάσιμα στο ίδιο πακέτο).
21. Στην Java χρησιμοποιείται η δεσμευμένη λέξη **extends** για την κληρονομικότητα (αντί :) και η **super** για τον καθορισμό μεθόδων που θα κληθούν από την γονική κλάση (μόνο όμως μεθόδων που βρίσκονται ένα επίπεδο πιο πάνω).



# Διαφορές C++ - Java

---

22. Η Java παρέχει τη δυνατότητα **διασυνδέσεων** μέσω της δεσμευμένης λέξης **interface** ως ισοδύναμο μιας **αφηρημένης** κλάσης βάσης.
23. Η Java δεν υποστηρίζει πολλαπλή κληρονομικότητα.

Μπορεί ωστόσο να προσομοιώσει κάτι τέτοιο μέσω διασυνδέσεων (interfaces).





# Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στο πλαίσιο του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Αθηνών**» έχει χρηματοδοτήσει μόνο την αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



# Σημείωμα Ιστορικού Εκδόσεων Έργου

---

Το παρόν έργο αποτελεί την έκδοση 1.0.1



# Σημείωμα Αναφοράς

---

Copyright: Πανεπιστήμιον Πατρών, Ιωάννης Χατζηλυγερούδης, 2015.  
«Οντοκεντρικός Προγραμματισμός». Έκδοση: 1.0.1 Πάτρα 2015. Διαθέσιμο  
από τη δικτυακή διεύθυνση:

<https://eclass.upatras.gr/courses/CEID1105/>



# Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά, Μη Εμπορική Χρήση Παρόμοια Διανομή 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



[1] <http://creativecommons.org/licenses/by-nc-sa/4.0/>

Ως **Μη Εμπορική** ορίζεται η χρήση:

- που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
- που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
- που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο

Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

# Διατήρηση Σημειωμάτων

---

Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:

- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.



# Σημείωμα Χρήσης Έργων Τρίτων

---

- Οι διαφάνειες βασίζονται στο βιβλίο «C++ How to Program, 8th Edition, Harvey M. Deitel, Paul J. Deitel, Prentice Hall.»

