

C++ vs Java

Τμήμα Πληροφορικής και Τηλεπικοινωνιών (Άρτα)

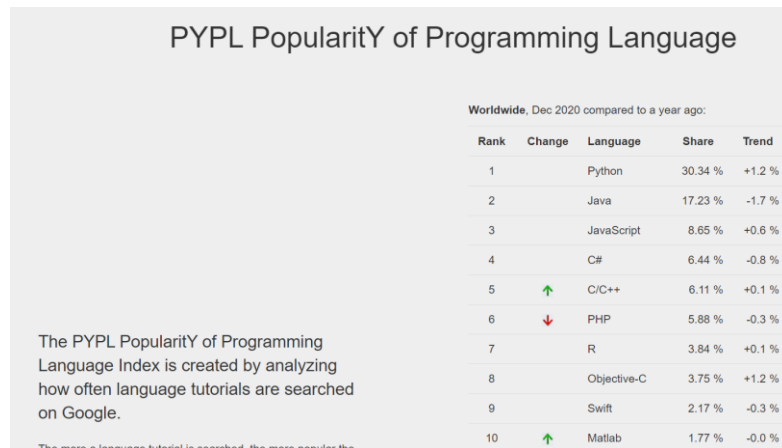
Πανεπιστήμιο Ιωαννίνων

Γκόγκος Χρήστος

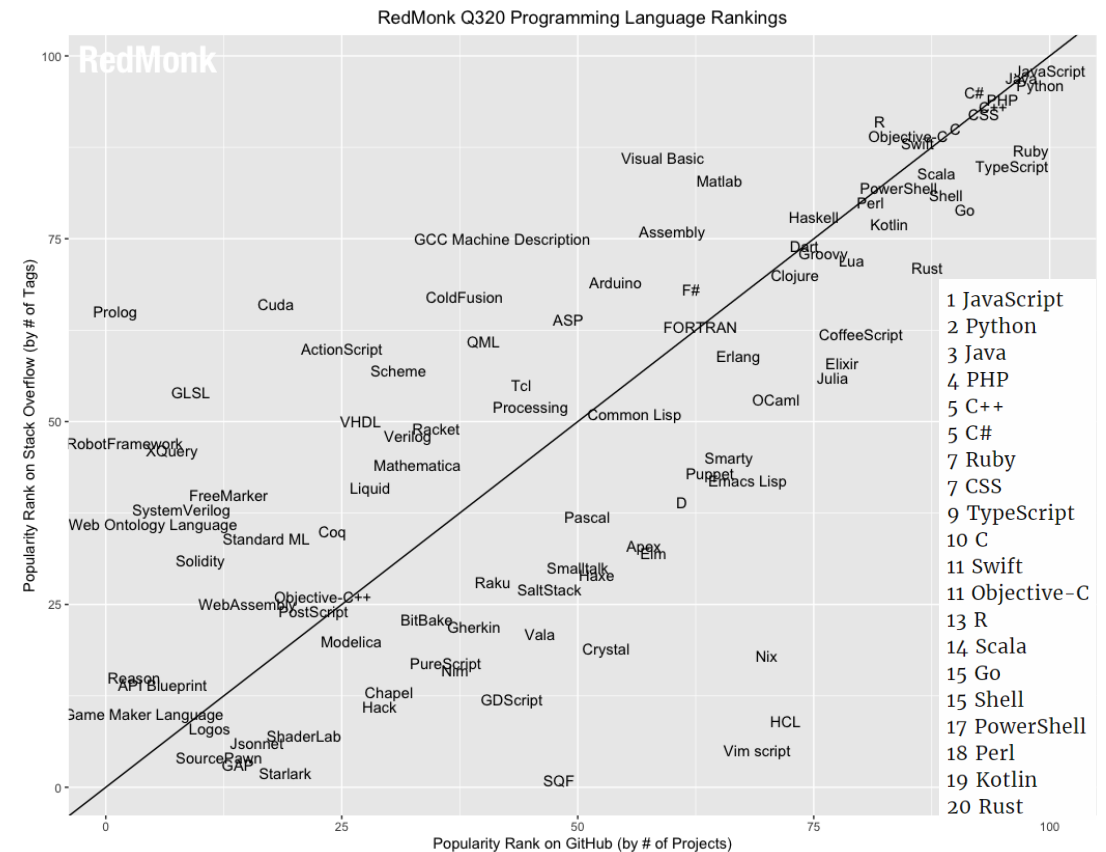
Δημοφιλείς γλώσσες προγραμματισμού (2020)

Dec 2020	Dec 2019	Change	Programming Language	Ratings	Change
1	2	▲	C	16.48%	+0.40%
2	1	▼	Java	12.53%	-4.72%
3	3		Python	12.21%	+1.90%
4	4		C++	6.91%	+0.71%
5	5		C#	4.20%	-0.60%
6	6		Visual Basic	3.92%	-0.83%
7	7		JavaScript	2.35%	+0.26%
8	8		PHP	2.12%	+0.07%
9	16	▲	R	1.60%	+0.60%
10	9	▼	SQL	1.53%	-0.31%

<https://www.tiobe.com/tiobe-index/>



<https://pypl.github.io/PYPL.html>



<https://redmonk.com/sograzy/2020/07/27/language-rankings-6-20/>

Αντικειμενοστρέφεια

- Τόσο η C++ όσο και η Java υποστηρίζουν τον αντικειμενοστραφή προγραμματισμό
 - Στην Java τα πάντα(;) είναι αντικείμενα και δεν επιτρέπει σε δεδομένα ή συναρτήσεις να υπάρχουν εκτός κλάσεων
 - Η C++ υποστηρίζει και το μοντέλο διαδικασιακού προγραμματισμού (procedural programming)
- Η Java αντιμετωπίζει διαφορετικά τους πρωτογενείς τύπους (π.χ. int, double, char κλπ) από τα αντικείμενα
- Η Java είναι απλούστερη από τη C++ και περισσότερο ασφαλής
- Η Java υποστηρίζει πολυμορφισμό αυτόματα ενώ στη C++ χρειάζεται να δηλώνονται οι συναρτήσεις μέλη ως virtual
- Η Java υποστηρίζει διεπαφές (interfaces) έναν τρόπο επιβολής υλοποίησης συγκεκριμένων συναρτήσεων από κλάσεις

Σύγκριση χαρακτηριστικών C++ και Java (1/2)

C++

- Υποστηρίζει δείκτες, πολλαπλή κληρονομικότητα, υπερφόρτωση τελεστών
- Ο προγραμματιστής είναι υπεύθυνος έτσι ώστε να μην βγαίνει εκτός ορίων πινάκων
- Η διαχείριση μνήμης είναι ευθύνη του προγραμματιστή
- Είναι γρήγορη
- Ο κώδικας μεταγλωττίζεται (το εκτελέσιμο δεν εκτελείται σε άλλη πλατφόρμα)

Java

- Δεν υποστηρίζει δείκτες, πολλαπλή κληρονομικότητα και υπερφόρτωση τελεστών
- Εμπεριέχει ελέγχους για όρια πινάκων
- Διαθέτει αυτόματο garbage collection
- Είναι αρκετά γρήγορη
- Είναι cross-platform (ο κώδικας αρχικά μεταγλωττίζεται σε bytecode και στη συνέχεια διερμηνεύεται από το JVM) - **WORA(Write Once and Run Everywhere)**

Σύγκριση χαρακτηριστικών C++ και Java (2/2)

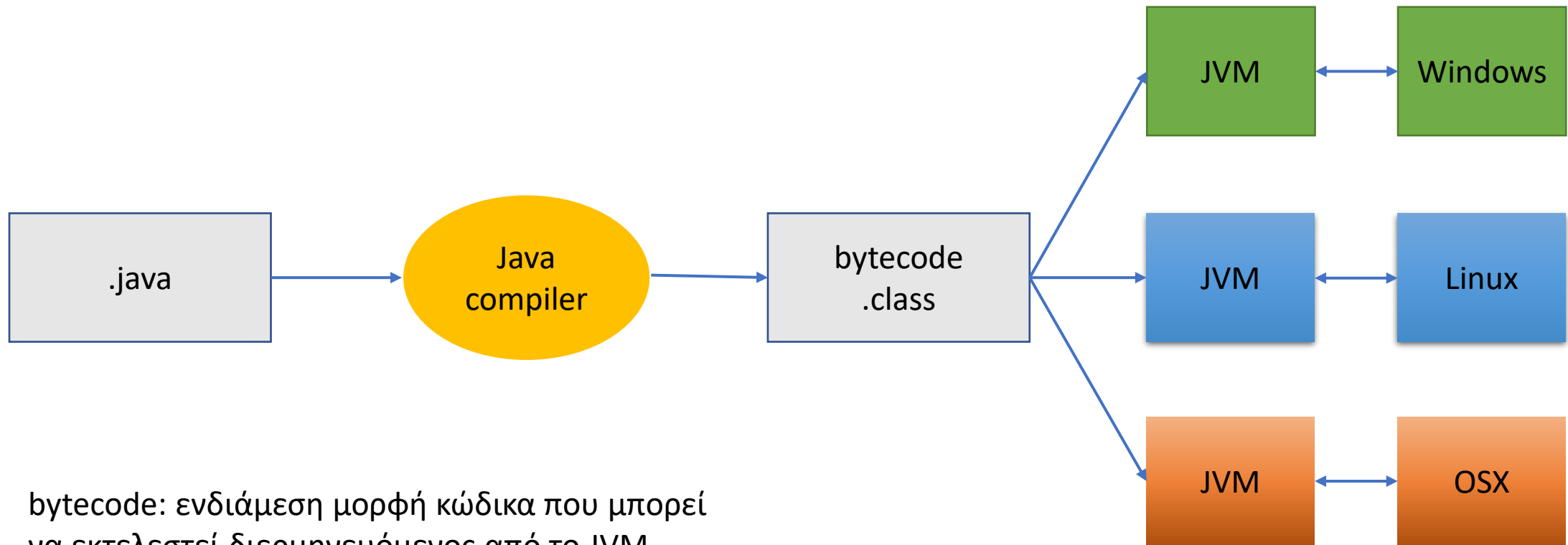
C++

- Υποστηρίζει πέρασμα με τιμή (call by value) και πέρασμα με αναφορά (call by reference)
- Χρησιμοποιεί templates
- Στην ίδια τη γλώσσα υπάρχει μικρότερος αριθμός βιβλιοθηκών
- Είναι system level
- Χρησιμοποιείται όπου απαιτείται ταχύτατη εκτέλεση κώδικα (π.χ. επιστημονικές εφαρμογές, ανάπτυξη παιχνιδιών)

Java

- Υποστηρίζει μόνο πέρασμα με τιμή (call by value)
- Χρησιμοποιεί generics
- Διαθέτει βιβλιοθήκες για μεγάλο εύρος λειτουργιών (π.χ. GUIs, Network programming, JDBC κ.α.)
- Βρίσκεται υψηλότερα από το system level
- Χρησιμοποιείται για ανάπτυξη εφαρμογών σε κινητές συσκευές (Android)

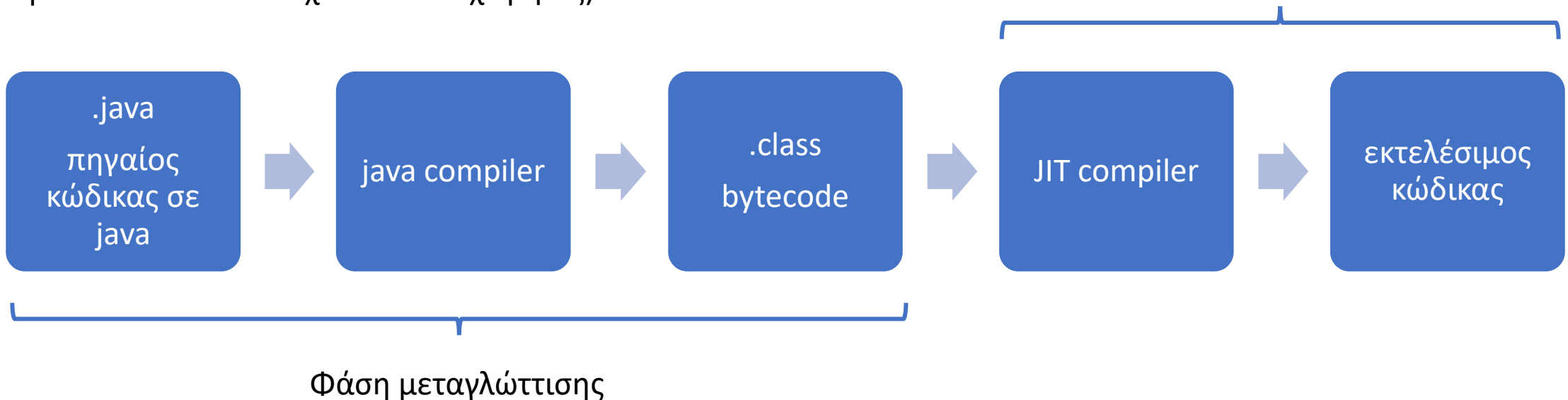
JVM (Java Virtual Machine) - μεταφερσιμότητα κώδικα



bytecode: ενδιάμεση μορφή κώδικα που μπορεί να εκτελεστεί διερμηνευόμενος από το JVM

Java – επιτάχυνση εκτέλεσης με τον JIT compiler

Ο JIT compiler μεταγλωττίζει ακολουθίες bytecode σε κώδικα μηχανής και βελτιστοποιήσεις (π.χ. ελάττωση των προσπελάσεων στη μνήμη, διατηρώντας δεδομένα που προσπελούνται συχνά σε καταχωρητές)



Java: Garbage collection (1/2)

- Η συλλογή απορριμμάτων στη Java είναι η διαδικασία της αυτόματης διαχείρισης μνήμης (δέσμευσης/αποδέσμευσης μνήμης)
- Ο GC (Garbage Collector) εντοπίζει τα αντικείμενα που δεν χρησιμοποιούνται, τα διαγράφει και ελευθερώνει τη μνήμη που κατελάμβαναν
- Η δυναμική δέσμευση μνήμης γίνεται (στο heap) με τον τελεστή new και η μνήμη παραμένει δεσμευμένη έως ότου να πάψουν να υπάρχουν αναφορές προς τη μνήμη αυτή
- Η συλλογή απορριμμάτων γίνεται αυτόματα καθώς εκτελείται το πρόγραμμα

Java: Garbage collection (2/2)

- Το JVM της Oracle, HotSpot, διαθέτει 4 Garbage Collectors (Serial, Parallel, Concurrent Mark Sweep, Garbage First) με διαφορετικά χαρακτηριστικά και επιδόσεις
- Η βασική ιδέα λειτουργίας των Garbage Collectors είναι ότι:
 - Σε πρώτη φάση εντοπίζουν τα αντικείμενα για τα οποία δεν υπάρχει αναφορά και τα σημειώνουν ως «έτοιμα προς αποκομιδή»
 - Σε δεύτερη φάση διαγράφει τα «έτοιμα προς αποκομιδή» αντικείμενα
 - Προαιρετικά, η μνήμη αναδιατάσσεται μετά τη διαγραφή έτσι ώστε τα εναπομείναντα αντικείμενα να βρίσκονται σε συνεχόμενες θέσεις στην αρχή του heap

Παραδείγματα κώδικα

Υλοποιήσεις σε C++ και σε Java

Hello World σε C++ και σε Java

C++

hello.c

```
#include <iostream>
using namespace std;
int main()
{
    cout << "Hello World" << endl;
}
```

```
$ g++ hello.c
$ ./a.out
Hello World
```

Java

Hello.java

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

```
$ javac Hello.java
$ java Hello
Hello World
```

Κλάσεις και αντικείμενα

C++

dogs.c

```
#include <iostream>
using namespace std;
class dog
{
private:
    string name;
    string species;
public:
    dog(string n, string s) : name(n), species(s) {}
    void make_sound()
    {
        cout << "Dog " << name << " of species " << species
            << " barks" << endl;
    }
};
int main()
{
    dog obj("Adelle", "Maltese");
    obj.make_sound();
}
```

```
$ g++ dogs.c
```

```
$ ./a.out
```

```
Dog Adelle of species Maltese barks
```

Java

Dog.java

```
public class Dog {
    private String name;
    private String species;
    public Dog(String n, String s) {
        name = n;
        species = s;
    }
    public void makeSound() {
        System.out.println("Dog " + name + " of species "
            + species + " barks");
    }
    public static void main(String[] args) {
        Dog obj = new Dog("Adelle", "Maltese");
        obj.makeSound();
    }
}
```

```
$ javac Dog.java
```

```
$ java Dog
```

```
Dog Adelle of species Maltese barks
```

Παραδείγματα κώδικα σε C++ και Java

Κώδικας σε C++

- <https://chgogos.github.io/oop/preparation/prepare01.cpp>
- <https://chgogos.github.io/oop/preparation/prepare02.cpp>
- <https://chgogos.github.io/oop/preparation/prepare03.cpp>

Κώδικας σε Java

- <https://chgogos.github.io/oop/lab2020-2021/lab06/prepare01a/>
- <https://chgogos.github.io/oop/lab2020-2021/lab06/prepare01b/>
- <https://chgogos.github.io/oop/lab2020-2021/lab06/prepare01c/>

Εκφωνήσεις: <https://chgogos.github.io/oop/preparation/proodos.pdf>

Αναφορές

- <https://www.tiobe.com/tiobe-index/>
- <https://pypl.github.io/PYPL.html>
- <https://redmonk.com/>
- <https://stackify.com/what-is-java-garbage-collection/>
- https://www.ibm.com/support/knowledgecenter/SSYKE2_8.0.0/com.ibm.java.vm.80.doc/docs/jit_overview.html