

# C++ vs. Java

Τμήμα Πληροφορικής και Τηλεπικοινωνιών (Άρτα)

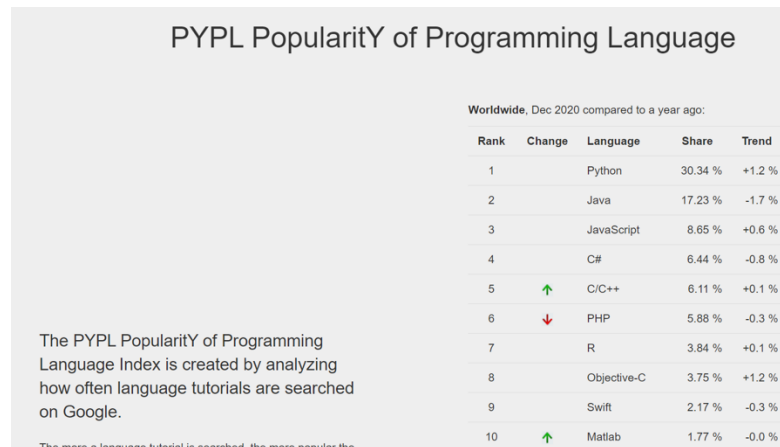
Πανεπιστήμιο Ιωαννίνων

Γκόγκος Χρήστος

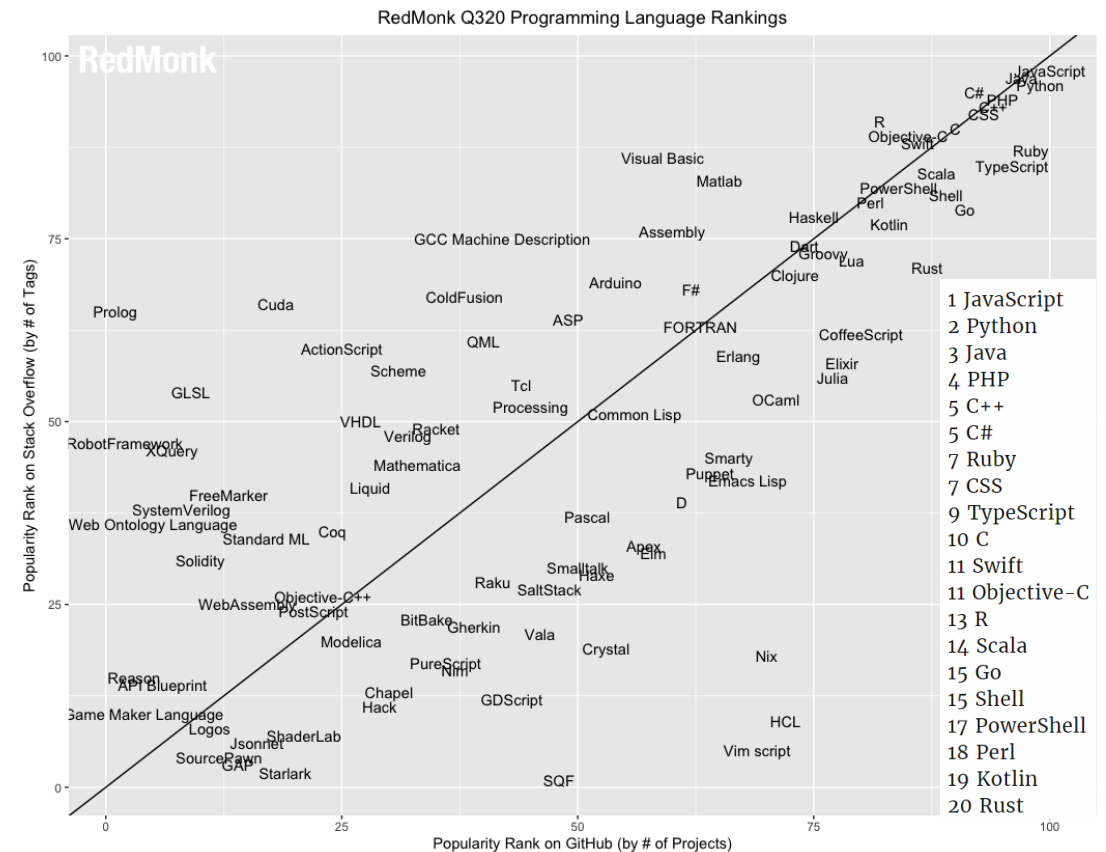
# Δημοφιλείς γλώσσες προγραμματισμού (2020)

| 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/sogrady/2020/07/27/language-rankings-6-20/>

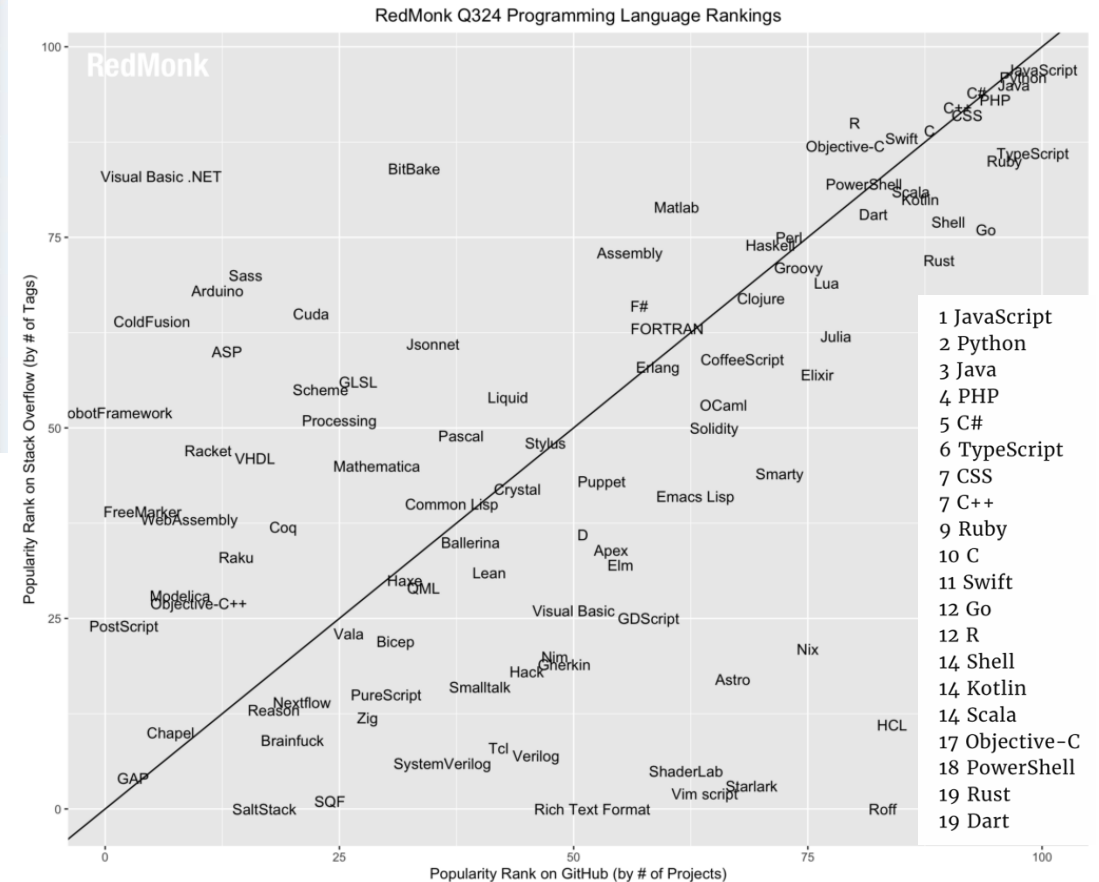
# Δημοφιλείς γλώσσες προγραμματισμού (2025)

| Jan 2025 | Jan 2024 | Change | Programming Language   | Ratings | Change |
|----------|----------|--------|--|---------|--------|
| 1        | 1        |        |  Python       | 23.28%  | +9.32% |
| 2        | 3        | ▲      |  C++          | 10.29%  | +0.33% |
| 3        | 4        | ▲      |  Java         | 10.15%  | +2.28% |
| 4        | 2        | ▼      |  C            | 8.86%   | -2.59% |
| 5        | 5        |        |  C#           | 4.45%   | -2.71% |
| 6        | 6        |        |  JavaScript   | 4.20%   | +1.43% |
| 7        | 11       | ▲▲     |  Go           | 2.61%   | +1.24% |
| 8        | 9        | ▲      |  SQL          | 2.41%   | +0.95% |
| 9        | 8        | ▼      |  Visual Basic | 2.37%   | +0.77% |
| 10       | 12       | ▲      |  Fortran      | 2.04%   | +0.94% |

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

| Worldwide, Jan 2025 : |        |             |         |              |
|-----------------------|--------|-------------|---------|--------------|
| Rank                  | Change | Language    | Share   | 1-year trend |
| 1                     |        | Python      | 29.8 %  | +1.7 %       |
| 2                     |        | Java        | 15.35 % | -0.5 %       |
| 3                     |        | JavaScript  | 7.96 %  | -1.0 %       |
| 4                     |        | C/C++       | 7.17 %  | +0.3 %       |
| 5                     |        | C#          | 6.36 %  | -0.3 %       |
| 6                     |        | R           | 4.63 %  | +0.0 %       |
| 7                     |        | PHP         | 3.87 %  | -0.6 %       |
| 8                     |        | TypeScript  | 2.84 %  | -0.0 %       |
| 9                     | ▲▲     | Rust        | 2.8 %   | +0.5 %       |
| 10                    |        | Objective-C | 2.65 %  | +0.2 %       |

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



<https://redmonk.com/sogrady/2024/09/12/language-rankings-6-24/>

# Αντικειμενοστρέφεια

- Τόσο η C++ όσο και η Java υποστηρίζουν τον αντικειμενοστραφή προγραμματισμό
  - Στην Java τα πάντα(;) είναι αντικείμενα και η γλώσσα 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 κ.λπ.) 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)

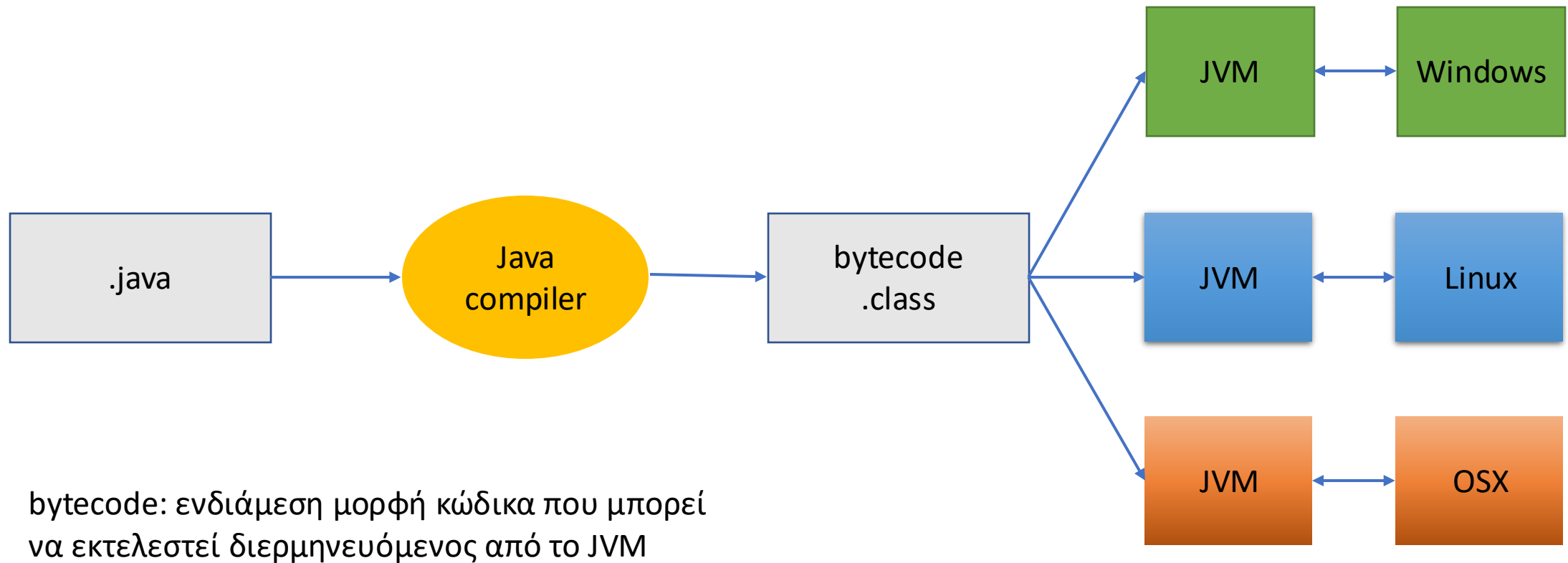
## C++

- 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

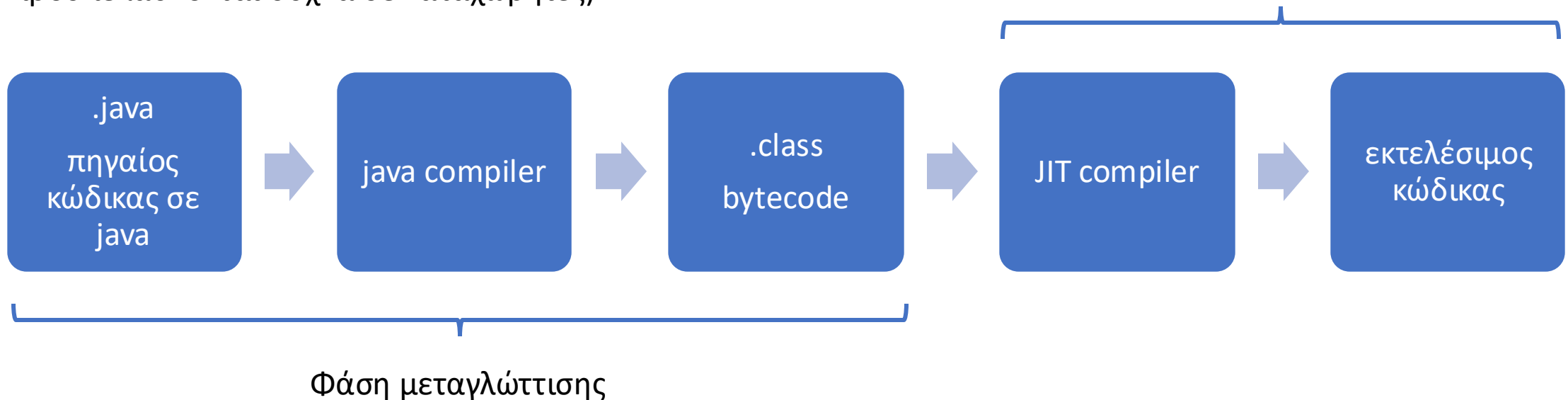
- Υποστηρίζει μόνο πέρασμα με τιμή (call by value)
- Χρησιμοποιεί generics
- Διαθέτει βιβλιοθήκες για μεγάλο εύρος λειτουργιών (π.χ. GUIs, Network programming, JDBC κ.α.)
- Βρίσκεται υψηλότερα από το system level
- Χρησιμοποιείται για ανάπτυξη εφαρμογών σε κινητές συσκευές (Android)

# JVM (Java Virtual Machine) - μεταφερσιμότητα κώδικα



# Java – επιτάχυνση εκτέλεσης με τον JIT compiler

Ο JIT compiler μεταγλωττίζει ακολουθίες bytecode σε κώδικα μηχανής και πραγματοποιεί 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.cpp

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

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

[https://github.com/chgogos/oop/blob/master/variuous/CPP\\_VS\\_JAVA/helloworld\\_cpp/hello.cpp](https://github.com/chgogos/oop/blob/master/variuous/CPP_VS_JAVA/helloworld_cpp/hello.cpp)

## Java

Hello.java

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

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

από την Java 11 και μετά

ή

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

[https://github.com/chgogos/oop/blob/master/variuous/CPP\\_VS\\_JAVA/helloworld\\_java/Hello.java](https://github.com/chgogos/oop/blob/master/variuous/CPP_VS_JAVA/helloworld_java/Hello.java)

# Hello World σε Java 22 (ως preview feature)

Hello22.java

```
void main() {  
    System.out.println("Hello World");  
}
```

```
$ java --enable-preview --source 22 Hello22.java  
Hello World
```

[https://github.com/chgogos/oop/blob/master/variuous/CPP\\_VS\\_JAVA/helloworld\\_java/Hello22.java](https://github.com/chgogos/oop/blob/master/variuous/CPP_VS_JAVA/helloworld_java/Hello22.java)

# Έλεγχος ορίων (bounds check)

## C++

bounds.cpp

```
#include <iostream>
using namespace std;
int main()
{
    int a[] = {0, 1, 2, 3, 4, 5};
    for (int i = 0; i < 10; i++)
    {
        cout << a[i] << " ";
    }
}
```

```
$ g++ bounds.cpp
$ ./a.out
0 1 2 3 4 5 16 7 0 0
```

## Java

Bounds.java

```
public class Bounds {
    public static void main(String[] args) {
        int a[] = new int[] { 0, 1, 2, 3, 4, 5 };
        for (int i = 0; i < 10; i++) {
            System.out.print(a[i] + " ");
        }
    }
}
```

```
$ java Bounds.java
0 1 2 3 4 5 Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException:
Index 6 out of bounds for length 6
    at Bounds.main(Bounds.java:5)
```

[https://github.com/chgogos/oop/blob/master/various/CPP\\_VS\\_JAVA/bounds\\_check\\_java/Bounds.java](https://github.com/chgogos/oop/blob/master/various/CPP_VS_JAVA/bounds_check_java/Bounds.java)

[https://github.com/chgogos/oop/blob/master/various/CPP\\_VS\\_JAVA/bounds\\_check\\_cpp/bounds.cpp](https://github.com/chgogos/oop/blob/master/various/CPP_VS_JAVA/bounds_check_cpp/bounds.cpp)

# Κλάσεις και αντικείμενα

## C++

dogs.cpp

```
#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.cpp
```

```
$ ./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();
    }
}
```

```
$ java Dog.java
```

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

[https://github.com/chgogos/oop/blob/master/various/CPP\\_VS\\_JAVA/simple\\_java/Dog.java](https://github.com/chgogos/oop/blob/master/various/CPP_VS_JAVA/simple_java/Dog.java)

[https://github.com/chgogos/oop/blob/master/various/CPP\\_VS\\_JAVA/simple\\_cpp/dog.cpp](https://github.com/chgogos/oop/blob/master/various/CPP_VS_JAVA/simple_cpp/dog.cpp)

# Παραδείγματα κώδικα σε C++ και Java

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

## Απαντήσεις στις ασκήσεις 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>

## Απαντήσεις στις ασκήσεις 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/>



# 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](https://www.ibm.com/support/knowledgecenter/SSYKE2_8.0.0/com.ibm.java.vm.80.doc/docs/jit_overview.html)