

Αντικειμενοστρεφής Προγραμματισμός (Object-Oriented Programming)

(CEID_NNY106)

Java as an OOP Language

Java

High-level programming language



Java as Extension of C

Difference Between C and Java

In the following table, we have described some key differences between C and Java.

S.	Basis	C	Java
1	Language Level	C is a middle-level language.	Java is a high-level language.
2	Paradigm	C is a structural and procedure-oriented programming language.	Java is an object-oriented programming language.
3	Approach	It follows the top-down approach to design the application.	It follows the bottom-up approach to design the application.
4	Translation	It is a compiled language.	It is an interpreted language.
5	Platform Dependency	It is platform dependent .	It is platform- independent .
6	Keywords	There are 32 keywords in C.	Java has 50 keywords.

Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible.

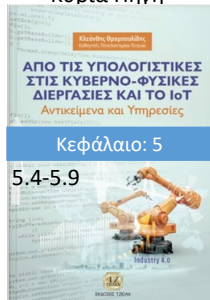
[Wikipedia](#)

Designed by: James Gosling

First appeared: May 23, 1995; 27 years ago

Paradigm: Multi-paradigm: generic, object-oriented (class-based), functional, imperative, reflective, concurrent

Κύρια Πηγή



Kleanthis Thramboulidis
Prof. of Software and System Engineering
University of Patras

<https://sites.google.com/site/thramboulidiskleanthis/>

Οργάνωση Διάλεξης

- Τύποι Δεδομένων
- Τελεστές
- Προτάσεις ροής ελέγχου
- Functions vs Methods
- Λοιπές διαφορές της Java από την C

Τύποι Δεδομένων

- διαθέτει μικρό σύνολο πρωτογενών τύπων
- υποστηρίζει τη δημιουργία οποιουδήποτε σύνθετου διαμέσου των μηχανισμών
 - της **κλάσης**,
 - του **interface** και
 - του **πίνακα**

Data Type	Default Value (for fields)
byte	0
short	0
int	0
long	0L
float	0.0f
double	0.0d
char	'\u0000'
String (or any object)	null
boolean	false

<https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html>

Πρωτογενείς Τύποι

υιοθετεί τους τύπους της C με τις παρακάτω διαφοροποιήσεις

- υποστηρίζει επιπλέον τους τύπους **byte** και **boolean**
- καθορίζει αυστηρά το μέγεθος κάθε τύπου
- δεν υποστηρίζει απρόσημους αριθμούς
 - Java 8 adds support for unsigned ints and longs
- δεν υποστηρίζει τον τύπο του δείκτη
- ορίζει για κάθε τύπο μια default τιμή
 - Η τιμή αυτή αποδίδεται σε κάθε δήλωση μεταβλητής του τύπου, αν ο προγραμματιστής δεν ορίσει αρχική τιμή
 - **local/block variables don't get the default values (should be initialized).**

Πρωτογενείς Τύποι

Τύπος	Περιλαμβάνει	Αρχική τιμή	Μέγεθος	Ελάχιστη Τιμή	Μέγιστη Τιμή
boolean	true or false	false	1 bit	N.A	N.A
char	Unicode character	\u0000	16 bits	\u0000	\uFFFF
byte	signed integer	0	8 bits	-128	127
short	signed integer	0	16 bits	-32768	32767
int	signed integer	0	32 bits	-2147483648	2147483647
long	signed integer	0	64 bits	-9223372036854775808	9223372036854775807
float	IEEE 754 floating-point	0.0	32 bits	$\pm 3.40282347E+38$	$\pm 1.40239846E-45$
double	IEEE 754 floating-point	0.0	64 bits	$\pm 1.79769313486231570E+308$	$\pm 4.94065645841246544E-324$

Ο τύπος Boolean

- οι μεταβλητές τύπου **Boolean** δεν μπορούν
 - να χρησιμοποιηθούν ως ακέραιοι αριθμοί και
 - να μετατραπούν από και προς άλλον τύπο
- έμμεση μετατροπή του ακεραίου i σε boolean b ,
 $b=(i!=0);$
- έμμεση μετατροπή του boolean b σε ακέραιο i
 $i=(b)?1:0;$

Δηλώσεις μεταβλητών

- Γενικά ακολουθεί τη σύνταξη της C, C++
 - `<τύπος> <όνομα μεταβλητής>`

π.χ. `int num;`
 `char ch;`
 `float fl;`
- Όπως και στη C++ μπορούν να δηλώνονται και μέσα στον κώδικα


```
for(int i = 0 ; i<20 ; i++)
```

Τύποι αναφοράς (reference types)

- **Class, Array, Enumeration, Interface, and Annotation**
- χρησιμοποιούνται για να προσδιορίσουν τον τύπο αναφορών.
- Η αναφορά (reference) μας δίνει τη δυνατότητα αναφοράς σε στιγμιότυπο
- Αντιγραφή αντικειμένου


```
Circle c1 = new Circle();
```

~~`c2 = c1;` ?~~

```
Circle c2 = c1.clone(); // assuming that Circle implements the Clonable
// interface
```
- Έλεγχος ισότητας


```
c1 == c2
```
- αναφορά που δεν αναφέρεται σε στιγμιότυπο ή πίνακα, έχει τιμή *null*.

Αλφαριθμητικά

- θεώρηση της **C**
 - τα αλφαριθμητικά είναι πίνακες χαρακτήρων
- θεώρηση της **Java**
 - το αλφαριθμητικό είναι αντικείμενο με σαφώς ορισμένη συμπεριφορά που περιγράφεται από την κλάση **String**
 - κάθε αλφαριθμητικό είναι στιγμιότυπο αυτής της κλάσης και συμπεριφέρεται όπως αυτή ορίζει

Methods	
Modifier and Type	Method and Description
char	charAt(int index) Returns the char value at the specified index.
int	codePointAt(int index) Returns the character (Unicode code point) at the specified index.
int	codePointBefore(int index) Returns the character (Unicode code point) before the specified index.
int	codePointCount(int beginIndex, int endIndex) Returns the number of Unicode code points in the specified text range of the string.
int	compareTo(String anotherString) Compares two strings lexicographically.
int	compareToIgnoreCase(String str) Compares two strings lexicographically, ignoring case differences.
String	concat(String str) Concatenates the specified string to the end of this string.
boolean	contains(CharSequence s) Returns true if and only if this string contains the specified sequence of characters.
boolean	contentEquals(CharSequence cs) Compares this string to the specified CharSequence.
boolean	contentEquals(StringBuffer sb) Compares this string to the specified StringBuffer.
static String	copyValueOf(char[] data) Returns a String that represents the character sequence in the array specified by the argument.
static String	copyValueOf(char[] data, int offset, int count) Returns a String that represents the character sequence in the array specified by the arguments.
boolean	endsWith(String suffix) Tests if this string ends with the specified suffix.

© 2023 Κλεάνθης Θραμπουλίδης

Java as Extension of C

Διαφάνεια 10

Array

- δημιουργία πίνακα
int [] x; // δημιουργία αναφοράς σε πίνακα ακααράιων
x = new int [10]; // δημιουργία πίνακα και ανάθεση του στην αναφορά
Circle[] circles = new Circle[12];
- αναφορά στο μέγεθος του πίνακα
<όνομα αναφοράς πίνακα>.length
- πρόσβαση στα στοιχεία πίνακα
for (int i = 0; i < circles.length; i++)
circles[i] = new Circle(point[i], radius[i]);

```
public class Point {
    public int x = 0;
    public int y = 0;
    //constructor
    public Point(int a, int b) {
        x = a;
        y = b;
    }
}
```

© 2023 Κλεάνθης Θραμπουλίδης

Java as Extension of C

Διαφάνεια 11

Τελεστές 1/2

- **δεν υποστηρίζει**
 - τον τελεστή κόμμα (,) για συνδυασμό δύο εκφράσεων σε μία
 - τους τελεστές δεικτών * and & ,
 - τον τελεστή sizeof
- **υποστηρίζει τους παρακάτω νέους τελεστές**
 - **+, instanceof, >>>, &** και |

Operators in Java

- Unary Operator,
- Arithmetic Operator,
- Shift Operator,
- Relational Operator,
- Bitwise Operator,
- Logical Operator,
- Ternary Operator and.
- Assignment Operator.

<https://docs.oracle.com/javase/tutorial/java/nutsandbolts/opsummary.html>

Τελεστές 2/2

- **δυναδικός +** (string concatenation)
 - `string + string` `string + int` `string + float`
 - **==, !=**
 - επιτρέπουν τη σύγκριση ανάμεσα σε αναφορές
 - **λογικοί &&, &, ||, |, !**
 - εκτός απο τον **short-circuit** υπολογισμό (&&,||) προσφέρονται και οι **&, |** τελεστές που υπολογίζουν και τα δύο ορίσματα
 - **instanceof**
 - ελέγχει αν ένα στιγμιότυπο είναι του δεδομένου τύπου ή υποκλάσης του
- if (someObject instanceof Stack)**
((Stack)someObject).push(element);

Εφαρμοστική σειρά :

applicative order calculate operands, next apply operator
Υπολογισμός περιορισμένης έκτασης: short circuit evaluation calculate left operand and only if required calculates the right one.

Προτάσεις ελέγχου ροής

■ Επανάληψης

- for
- while
- do ... while

γενικά
διατηρείται η
σύνταξη της C

■ διακλάδωσης υπό συνθήκη

- if if ... else
- switch (see also **switch expression** in Java 13)

■ διακλάδωσης χωρίς συνθήκη

- break
- continue

multi-level
break/continue

Προτάσεις if/else, while και do/while

- διατηρείται η C σύνταξη
- η έκφραση στην οποία βασίζονται οι υπό συνθήκη προτάσεις πρέπει να είναι τύπου boolean και όχι ακέραιου τύπου ή τύπου αναφοράς

```
int i = 10;
while( i-- ) {           // λάθος
    Circle c1= new Circle();
    if ( c1 ) {          // λάθος
        int j;
        do { ...
            } while ( j ); // λάθος
    }
}
```

```
int i= 10;
while( i-- >0 ) {
    Circle c1= new Circle();
    if ( c1 != null ) {
        int j;
        do { ....
            } while ( j != 0 );
    }
}
```

Προτάσεις break, continue

break [label];

```
label_one:
for(int i = 0; i < len ; i++) {
  if(i < len/2) {
    while (e1) {
      if(e2)
        break label_one;
      :
    }
    :
  }
  :
}
```

continue [label];

```
label_one:
for( i = 0; i < len ; i++) {
  :
  while(e1) {
    :
    if(e2)
      continue label_one;
    :
  }
  :
}
```

Σταθερές

- Οι σταθερές έχουν τύπο
 - Δηλώνονται όπως και οι μεταβλητές με προσθήκη δύο προσδιοριστών
 - static
 - final
 - Όπως και στις C/C++, συνηθίζεται να γράφονται με κεφαλαία
- static final** double PI = 3,14...;

Απαριθμητικός τύπος

```
public enum Day {
    SUNDAY, MONDAY, TUESDAY, WEDNESDAY,
    THURSDAY, FRIDAY, SATURDAY
}
```

```
public class EnumTest {
    Day day;
    main() {
        ....
        switch (day) {
            case MONDAY:
                ...
            case TUESDAY :
                ...
        }
    }
}
```

- An *enum* type is a **special data type** that enables for a variable to be a **set of predefined constants**.
- The variable must be equal to one of the values that have been predefined for it.
- Common examples include compass directions (values of NORTH, SOUTH, EAST, and WEST) and the days of the week.
- Because they are constants, the names of an enum type's fields are in uppercase letters.

<https://docs.oracle.com/javase/tutorial/java/javaOO/enum.html>

var keyword

- **var** allows you to **declare variables without their type**, e.g.

- `var str = "Java"`

- instead of `String str = "Java"`

- `var list = new ArrayList<String>();`

- `// infers ArrayList<String>`

- `var stream = list.stream();`

- `// infers Stream<String>`

- `var bos = new ByteArrayOutputStream();`

```
default Stream<E> stream()
```

Returns a sequential Stream with this collection as its source.

Enhanced switch (without break)

```
var event = Event.PLAY;
switch (event) {
    case PLAY -> System.out.println("PLAY event!");
    case STOP -> System.out.println("STOP event");
    default -> System.out.println("Unknown event");
};
```

```
javac --enable-preview --release 12 Example.java
```

To run the generated class file, you'll need to pass the --enable-preview flag to the Java launcher:

```
switch (event) {
    case PLAY -> {
        System.out.println("PLAY event!");
        counter++;
    }
    case STOP -> System.out.println("STOP event");
};
```

```
java --enable-preview Example
```

Enhanced switch functionality is currently only available as a [preview feature](#).

switch Expression

```
var log = switch (event) {
    case PLAY -> "User has triggered the play button";
    case STOP -> "User needs a break";
    default -> "No event to log";
};
```

```
Day day = Day.WEDNESDAY;
System.out.println(
    switch (day) {
        case MONDAY, FRIDAY, SUNDAY -> 6;
        case TUESDAY -> 7;
        case THURSDAY, SATURDAY -> 8;
        case WEDNESDAY -> 9;
        default -> throw new IllegalStateException(
```

<https://docs.oracle.com/>

At the moment, in Java 12, the switch cases support only switching on enum, String, byte, short, char, int, and their wrapper classes. However, in the future there may well be more sophisticated forms and support for structural pattern matching on arbitrary “switchable” types. [</article>](#)

Functions vs Methods

- A method defines behavior like a function.
- However:
 - Methods are defined inside a class definition (establish a relationship between the class and the method)
 - The invocation of a method is different from the call of a function.
- **Java has no functions**
- C++ and Python have both methods and functions

Λοιπές διαφορές από την C 1/2

- δυνατότητα δήλωσης τοπικών μεταβλητών σε κάθε μπλοκ κώδικα
- επιτρέπει υπερφόρτωση μεθόδων
- Δεν υποστηρίζει καθολικές μεταβλητές (τις υποστηρίζει όμως έμμεσα)
- δεν υποστηρίζει τη λέξη κλειδί **typedef**
- δεν υποστηρίζει **struct**, **union** και **bitfields**
 - Υποστηρίζει Bitwise and Bit Shift Operators

Λοιπές διαφορές από την C 2/2

- δεν υποστηρίζει μεταβλητό αριθμό παραμέτρων
 - see Arbitrary Number of Arguments i.e., (Point... corners)-> (Point[] corners)
- αντιμετωπίζει διαφορετικά τα ορίσματα γραμμής διαταγών
- δεν χρησιμοποιεί το **void** σε περίπτωση που μια μέθοδος δεν δέχεται παραμέτρους
- δεν διαθέτει προ-επεξεργαστή
- υποστηρίζει **exception handling**
- υποστηρίζει πολλαπλά νήματα (**multiple threads**)
- Διαθέτει **Garbage Collector**
- υποστηρίζει γραφική διεπαφή (**awt, swing, ...**)