

Αντικειμενοστρεφής Προγραμματισμός

Η Java ως επέκταση της C

Kleanthis Thramboulidis
Prof. of Software and System Engineering
<https://sites.google.com/site/thramboulidiskleanthis/>

1

Η Java ως επέκταση της C

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

© Κλ. Θραμπουλίδης

Java extends C

2

Τύποι Δεδομένων

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

© Κλ. Θραμπουλίδης

Java extends C

3

3

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).

© Κλ. Θραμπουλίδης

Java extends C

4

4

Οι βασικοί τύποι της Java

| 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 | ≈3.40282347E+38 | ≈1.40239846E-45 |
| double | IEEE 754 floating-point | 0.0 | 64 bits | ≈1.79769313486231570E+308 | ≈4.94065645841246544E-324 |

© Κλ. Θραμπουλίδης

Java extends C

5

5

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;

© Κλ. Θραμπουλίδης

Java extends C

6

6

Δηλώσεις μεταβλητών

- Γενικά ακολουθεί τη σύνταξη της 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++)`

© Κλ. Θραμπουλιάνης

Java extends C

7

7

Τύποι αναφοράς

- τα αντικείμενα και οι πίνακες στην Java ονομάζονται και τύποι αναφοράς (**reference types**) γιατί η διαχείρισή τους βασίζεται στην έννοια της αναφοράς
- Αντιγραφή αντικειμένου
`Circle c1 = new Circle();`
`c2 = c1; ?`
`c2 = c1.clone();`
- Έλεγχος ισότητας
`c1 == c2`
- αναφορά που δεν αναφέρεται σε αντικείμενο ή πίνακα, έχει τιμή **null**.

Python

```
>>> p1 = Point()
>>> p1.x = 3.0
>>> p1.y = 4.0
>>> import copy
>>> p2 = copy.copy(p1)

>>> p1 is p2
False
```

© Κλ. Θραμπουλιάνης

Java extends C

8

8

Τύπος Πίνακα (1)

- δημιουργία πίνακα
`int [] x; // δημιουργία αναφοράς`
`// δημιουργία αντικειμένου και`
`//ανάθεση του στην αναφορά`
`x = new int [10];`
- αναφορά στο μέγεθος του πίνακα
`<όνομα αναφοράς πίνακα>.length`

© Κλ. Θραμπουλιάνης

Java extends C

9

9

Τύπος Πίνακα (2)

- πρόσβαση στα στοιχεία πίνακα

```
Circle[] circles = new Circle[12];
```

```
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;
    }
}
```

Java extends C

10

10

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

© Κλ. Θραμπουλιάνης

Java extends C

11

11

Τελεστές (1)

- 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 `|`

© Κλ. Θραμπουλιάνης

Java extends C

12

12

Τελεστές

(2)

- **δυναδικός +** (string concatenation)
`string + string` `string + int` `string + float`
- **==, !=**
 - επιτρέπουν τη σύγκριση ανάμεσα σε αναφορές
- **λογικοί &&, &, ||, !, !**
 - εκτός από τον **short-circuit** υπολογισμό (&&,||) προσφέρονται και οι **&, |** τελεστές που υπολογίζουν και τα δύο ορίσματα

© Κλ. Θραμπουλιάνης

Java extends C

13

13

Τελεστές

(3)

- **instanceof**
 - ελέγχει αν ένα αντικείμενο είναι του δεδομένου τύπου ή υποκλάση του

π.χ.

if (someObject instanceof Stack)
((Stack)someObject).push(element);

```
Python
>>> isinstance(p, Point)
>>> hasattr(p, 'x')
```

© Κλ. Θραμπουλιάνης

Java extends C

14

14

Προτάσεις ελέγχου ροής

- **Επανάληψης**
 - for
 - while
 - do ... while
- **διακλάδωσης υπό συνθήκη**
 - if if ... else
 - switch (see also **switch expression** in Java 13)
- **διακλάδωσης χωρίς συνθήκη**
 - break
 - continue

γενικά
διατηρείται η
σύνταξη της C

multi-level
break/continue

© Κλ. Θραμπουλιάνης

Java extends C

15

15

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 );
    }
}
```

© Κλ. Θραμπουλιάνης

Java extends C

16

16

Προτάσεις break, continue

break [label]; **continue [label];**

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

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

© Κλ. Θραμπουλιάνης

Java extends C

17

17

Σταθερές

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

© Κλ. Θραμπουλιάνης

Java extends C

18

18

Αντιμετώπιση απαριθμητικών

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

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

© Κλ. Θραμπουλιάδης

Java extends C

19

19

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>s`
- `var bos = new ByteArrayOutputStream();`

© Κλ. Θραμπουλιάδης

Java extends C

20

20

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:

java --enable-preview Example

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

© Κλ. Θραμπουλιάδης

Java extends C

21

21

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";
};
```

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>](#)

© Κλ. Θραμπουλιάδης

Java extends C

22

22

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

© Κλ. Θραμπουλιάδης

Java extends C

23

23

Λοιπές διαφορές από την C (1)

- 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

© Κλ. Θραμπουλιάδης

Java extends C

24

24

Λοιπές διαφορές από την C(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, ...**)