3. Μαθαίνοντας την Java

Το αλφάβητο της Java

- Σχόλια
- Μπλοκ Εντολών
- Μεταβλητές
- Τύποι Δεδομένων
- Τελεστές
- Πίνακες

Σχόλια

```
// Σχόλια μιας γραμμής

/* Σχόλια πολλών γραμμών...
... Κι άλλα σχόλια
*/

/**
 * Σχόλια για τεκμηρίωση του προγράμματος (Javadoc)
 */
```

Μπλοκ εντολών

ΟΡΙΣΜΟΣ:

Μπλοκ Εντολών είναι οτιδήποτε περικλείεται μεταξύ δύο αγκίστρων { }.

```
class A{
// σώμα κλάσης

    public void firstMethod(){
        // σώμα μεθόδου
    }
    private void secondMethod(){
        // σώμα μεθόδου
    }
}
```

Κάθε μεταβλητή που δηλώνεται μέσα σε ένα μπλοκ εντολών έχει ισχύ ***μόνο εντός του μπλοκ****.*

Μεταβλητές

ΟΡΙΣΜΟΣ:

Μεταβλητή είναι μια μονάδα δεδομένων που φέρει σαφώς προσδιορισμένο όνομα και χαρακτηρίζεται από έναν συγκεκριμένο τύπο.

Δήλωση μεταβλητών:

<τύπος> <όνομα0>, <όνομα1>, ..., <όνομαN>;

Παράδειγμα:

```
float a, c, d;
int b;
```

Ονομασία μεταβλητών

Κανόνες για τα ονόματα μεταβλητών:

1. Να είναι έγκυροι ***προσδιοριστές.***
2. Να μην είναι α) ***λέξη-κλειδί,*** β) ***δυαδική σταθερά*** (**true, false**), ή γ) η ***δεσμευμένη*** λέξη **null.**
3. Να έχουν μοναδική ονομασία στο περιβάλλον που βρίσκονται β (π.χ. μέσα σε ένα μπλοκ εντολών).

Ονόματα μεταβλητών

Σωστό	Λάθος	Αιτία
HelloWorld	**Hello World**	**Περιέχει κενό**
Hi_Mom	**Hi_Mom!**	**Περιέχει θαυμαστικό**
heyDude3	**3heyDude**	**Ξεκινάει με αριθμό**
tall	**short**	**Είναι λέξη-κλειδί**
poundage	**#age**	**Ξεκινάει με #**

*Ένας π**ροσδιοριστής** είναι **έγκυρος** όταν αποτελείται από μια αλφαριθμητική έκφραση (γράμματα και αριθμούς) στην οποία ο πρώτος χαρακτήρας είναι γράμμα, ή ένα από τα σύμβολα _ (underscore), **$** (δολάριο).*

Λέξεις-κλειδιά

Java Keywords

abstract	float	public
boolean	for	return
break	if	short
byte	implements	static
case	import	super
catch	instanceof	switch
char	int	synchronized
class	interface	this
continue	long	throw
default	native	throws
do	new	transient
double	null	try
else	operator	void
extends	package	volatile
final	private	while
finally	protected	

Τύποι δεδομένων (πρωταρχικοί)

Τύπος	**Μέγεθος (bits)**
boolean	8
byte	8
char	16
short	16
int	32
long	64
float	32
double	64

Προσδιοριστές σταθερών

Προσδιοριστής	**Τύπος Δεδομένων**
178	int
8864L	long
37.266	double
37.266D	double
87.363F	float
26.77e3	double
‘c‘	char
true	boolean
false	boolean

Χαρακτήρες Διαφυγής

Χαρακτήρας	Περιγραφή
\n	Αλλαγή γραμμής
\t	Στηλοθέτης (tab)
\b	Οπισθοδρόμηση (backspace)
\r	Επαναφορά δρομέα (return)
\f	Αλλαγή σελίδας
\’	Απλό εισαγωγικό
\”	Διπλό εισαγωγικό

Τελεστές

- Αριθμητικοί
- Τελεστές σύγκρισης
- Λογικοί τελεστές
- Τελεστές καταχώρησης

Αριθμητικοί τελεστές

Τελεστής	Χρήση	Περιγραφή
+	op1 + op2	Προσθέτει τους op1 και op2
-	op1 - op2	Αφαιρεί τον op2 από τον op1
*	op1 * op2	Πολλαπλασιάζει τον op1 επί op2
/	op1 / op2	Διαιρεί τον op1 από τον op2
%	op1 % op2	Υπολογίζει το υπόλοιπο της ακέραιης διαίρεσης των op1 by op2

Συντετμημένοι αριθμητικοί τελεστές

Τελεστής	Χρήση	Περιγραφή
++	op++	Αυξάνει τον op κατά 1, επιστρέφει την τιμή του op πριν αυξηθεί
++	++op	Αυξάνει τον op κατά 1, επιστρέφει την τιμή του op αφού αυξηθεί
--	op--	Μειώνει τον op κατά 1, επιστρέφει την τιμή του op πριν μειωθεί
--	--op	Μειώνει τον op κατά 1, επιστρέφει την τιμή του op αφού μειωθεί

Τελεστές σύγκρισης

Τελεστής	Χρήση	Επιστρέφει true όταν...
>	op1 > op2	ο op1 είναι μεγαλύτερος από τον op2
>=	op1 >= op2	ο op1 είναι μεγαλύτερος ή ίσος από τον op2
<	op1 < op2	ο op1 είναι μικρότερος από τον op2
<=	op1 <= op2	ο op1 είναι μικρότερος ή ίσος από τον op2
==	op1 == op2	ο op1 είναι ίσος με τον op2
!=	op1 != op2	ο op1 είναι διαφορετικός από τον op2

Λογικοί τελεστές

Τελεστής	Χρήση	Επιστρέφει `true` όταν...
`&&`	`op1 && op2`	Οι `op1` και `op2` έχουν και οι δύο την τιμή `true`
`||`	`op1 || op2`	ένας από τους `op1` , `op2` έχει την τιμή `true`
`!`	`! op`	ο `op` έχει την τιμή `false`
`&`	`op1 & op2`	οι `op1` και `op2` έχουν και οι δύο την τιμή `true`,
`|`	`op1 | op2`	ένας από τους `op1` , `op2` έχει την τιμή `true`
`^`	`op1 ^ op2`	οι op1 και op2 είναι διαφορετικοί – δηλαδή όταν ένας από τους δύο έχει την τιμή true, αλλά όχι και οι δύο μαζί

Δυαδικοί τελεστές

Τελεστής	Χρήση	Λειτουργία
`>>`	`op1 >> op2`	Ολισθαίνει τα bits του `op1` προς τα δεξιά κατά `op2` θέσεις
`<<`	`op1 << op2`	Ολισθαίνει τα bits του `op1` προς τα αριστερά κατά `op2` θέσεις
`&`	`op1 & op2`	Επιστρέφει το δυαδικό `and`
`|`	`op1 | op2`	Επιστρέφει το δυαδικό `or`
`^`	`op1 ^ op2`	Επιστρέφει το δυαδικό `xor`

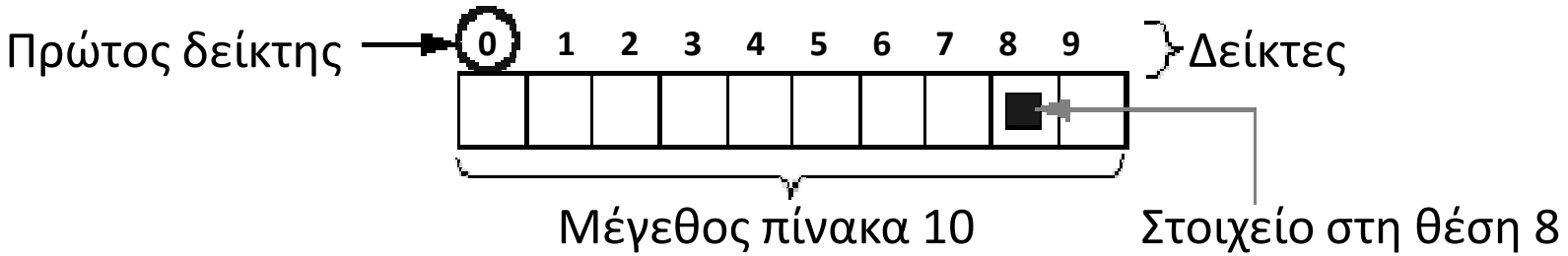
Τελεστές καταχώρησης

Τελεστής	Χρήση	Ισοδύναμο με:
+=	op1 += op2	op1 = op1 + op2
-=	op1 -= op2	op1 = op1 - op2
*=	op1 *= op2	op1 = op1 * op2
/=	op1 /= op2	op1 = op1 / op2
%=	op1 %= op2	op1 = op1 % op2
&=	op1 &= op2	op1 = op1 & op2
|=	op1 |= op2	op1 = op1 | op2
^=	op1 ^= op2	op1 = op1 ^ op2
<<=	op1 <<= op2	op1 = op1 << op2
>>=	op1 >>= op2	op1 = op1 >> op2

Πίνακες

ΟΡΙΣΜΟΣ:

Πίνακας είναι μια δομή που αποθηκεύει δεδομένα **ίδιου τύπου** σε **συνεχείς θέσεις μνήμης**.



Δήλωση πινάκων

ΔΗΛΩΣΗ ΠΙΝΑΚΑ

<τύπος>[] όνομα_πίνακα; // μονοδιάστατος
<τύπος>[][] όνομα_πίνακα; // 2-διάστατος, κλπ.

Παραδείγματα

```
int[] anArray;
float[][] anArrayOfFloats;
boolean[] anArrayOfBooleans;
Object[] anArrayOfObjects;
String[] anArrayOfStrings;
```

*Στην Java η **δήλωση** ενός πίνακα **δε δεσμεύει** μνήμη για αυτόν, δηλαδή **δεν τον δημιουργεί** (όπως γίνεται στη C)*

Δημιουργία πινάκων

ΔΗΜΙΟΥΡΓΙΑ ΠΙΝΑΚΑ

<όνομα_πίνακα> = new <τύπος>[<μέγεθος_πίνακα>];

Παραδείγματα

```
anArray = new int[10]; // create an array of integers
anArrayOfFloats = new float[12][20]; // 2 dimensions
```

*Για συντομία μπορούμε να **δηλώσουμε** και να **δημιουργήσουμε** έναν πίνακα σε μια γραμμή, π.χ:*

```
int[] anArray = new int[10];
```

Προσπέλαση στοιχείων πίνακα

Προσπέλαση στοιχείων

```
anArray[0] = 5;
int a = anArray[0] + 3; // Το a είναι ίσο με 8
```

Μέγεθος πίνακα

```
int len = anArray.length;
```

Αρχικοποίηση πίνακα

```
int[] a ={ 2, 3, 5, 1, 0, 15, 22, 97 };
String[] names ={“George”, “Vicky”, “John”, “Mary”};
```