Η Java με πολύ απλά λόγια

ΚΕΦΑΛΑΙΟ 2 • Βασικά Στοιχεία της Γλώσσας | Ενότητα 2.1 | Αναλυτικές έγχρωμες σημειώσεις

Πώς θα διαβάσουμε; Πρώτα μια ιστορία που καταλαβαίνουμε, έπειτα ο επιστημονικός όρος, μετά ένα μικρό παράδειγμα Java. Κρατάμε όλη την ορολογία του βιβλίου.

1. Τι είναι η Java;

Φαντάσου έναν υπολογιστή σαν ένα ρομπότ που περιμένει οδηγίες. Αν του πεις απλώς «κάνε κάτι ωραίο», δεν ξέρει τι να κάνει. Αν όμως του γράψεις σαφείς εντολές, τις εκτελεί. Η Java είναι μια γλώσσα προγραμματισμού: ένας τρόπος να γράφουμε αυτές τις εντολές.

Αντικειμενοστρεφής γλώσσα προγραμματισμού

Φαντάσου ένα εργοστάσιο παιχνιδιών. Έχει ένα ΣΧΕΔΙΟ αυτοκινήτου (κλάση / class). Με το σχέδιο φτιάχνει ένα κόκκινο και ένα μπλε αυτοκίνητο (αντικείμενα / objects). Κάθε αυτοκίνητο έχει δικά του χαρακτηριστικά, π.χ. χρώμα και ταχύτητα (πεδία / attributes). Αυτός είναι ένας απλός τρόπος να σκεφτόμαστε την αντικειμενοστρεφή Java. Οι πρωτογενείς τύποι δεδομένων είναι εξαίρεση: δεν είναι αντικείμενα.

Θυμάμαι Κλάση = σχέδιο. Αντικείμενο = κάτι συγκεκριμένο που δημιουργήθηκε από το σχέδιο. Πεδία = χαρακτηριστικά του αντικειμένου.

2. APIs (βιβλιοθήκες) και αρχή της ελάσσονος προσπάθειας

Θέλεις να χτίσεις ένα κάστρο LEGO. Δεν κατασκευάζεις μόνος σου κάθε τουβλάκι: χρησιμοποιείς έτοιμα. Στη Java, οι βιβλιοθήκες παρέχουν έτοιμες κλάσεις και λειτουργίες. Τα APIs (Application Programming Interfaces) περιγράφουν πώς χρησιμοποιούμε αυτές τις δυνατότητες. Έτσι μπορούμε, για παράδειγμα, να εμφανίσουμε ένα παράθυρο διαλόγου χωρίς να το σχεδιάσουμε από την αρχή.

Η ιδέα «Δεν επανεφευρίσκουμε τον τροχό» ονομάζεται εδώ αρχή της ελάσσονος προσπάθειας: πριν γράψουμε νέο κώδικα, εξετάζουμε αν υπάρχει έτοιμη λύση. Αν δημιουργήσουμε κάτι χρήσιμο, μπορούμε να το μοιραστούμε με την κοινότητα προγραμματιστών.

```
import javax.swing.JOptionPane;

JOptionPane.showMessageDialog(null, "Γεια σου!");
```

Το JOptionPane είναι έτοιμη κλάση βιβλιοθήκης. Η showMessageDialog εμφανίζει διαγνωστικό/ενημερωτικό μήνυμα σε παράθυρο διαλόγου. Το παράδειγμα δείχνει τη χρήση της, όχι ένα ολόκληρο αυτόνομο πρόγραμμα.

3. Μεταβλητές (variables): τα κουτάκια της μνήμης

Φαντάσου ένα κουτί με ετικέτα «ηλικία». Μέσα βάζεις τον αριθμό 12. Αύριο μπορείς να βάλεις 13. Στον υπολογιστή αυτό το κουτί λέγεται μεταβλητή: μια ονομασμένη θέση αποθήκευσης πληροφορίας. Η τιμή μπορεί να αλλάζει.

```
int age = 12;
age = 13;
```

Το int λέει ποιος είναι ο τύπος δεδομένων, το age είναι το όνομα μεταβλητής, το = είναι τελεστής ανάθεσης και το 12 η αρχική τιμή. Το ; τελειώνει την εντολή. Η δεύτερη εντολή αλλάζει την τιμή, δεν δημιουργεί δεύτερη μεταβλητή.

Δήλωση, ανάθεση τιμής και αρχικοποίηση

```
int number;           // δήλωση μεταβλητής
number = 6;           // ανάθεση τιμής
int counter = 6;      // δήλωση και αρχικοποίηση
```

Στη Java δηλώνουμε τον τύπο πριν χρησιμοποιήσουμε τη μεταβλητή. Η δήλωση μπορεί να γίνει σε διαφορετικά σημεία του κώδικα, αλλά πριν από τη χρήση της. Μια τοπική μεταβλητή πρέπει επίσης να έχει πάρει τιμή πριν τη διαβάσουμε.

4. Βασικοί πρωτογενείς τύποι δεδομένων

Δεν χωράνε όλα τα πράγματα στο ίδιο κουτί: άλλα κουτιά είναι για ολόκληρους αριθμούς, άλλα για δεκαδικούς, άλλα για γράμματα και άλλα για απαντήσεις ΝΑΙ/ΟΧΙ. Ο τύπος δεδομένων καθορίζει τι είδους τιμή επιτρέπεται να αποθηκεύσουμε. Οι οκτώ πρωτογενείς τύποι δεν είναι αντικείμενα κάποιας κλάσης.

Τύπος	Τι αποθηκεύει	Μέγεθος / εύρος
byte	Μικρό ακέραιο	1 byte · −128 έως 127
short	Ακέραιο περιορισμένου εύρους	2 bytes · −32768 έως 32767
int	Ακέραιο	4 bytes · -2^{31} έως $2^{31}-1$
long	Εκτεταμένο ακέραιο	8 bytes · -2^{63} έως $2^{63}-1$
float	Πραγματικό κινητής υποδιαστολής	4 bytes · περίπου 1.4×10^{-45} έως 3.4×10^{38} σε θετικές τιμές
double	Πραγματικό διπλής ακρίβειας	8 bytes · περίπου 4.9×10^{-324} έως 1.8×10^{308} σε θετικές τιμές
boolean	Λογική τιμή	true / false · χωρίς ορισμένο σταθερό μέγεθος σε bytes
char	Μονάδα χαρακτήρα Unicode (UTF-16)	2 bytes

Οι τέσσερις ακέραιοι: byte, short, int, long

Σκέψου τέσσερα κουτιά, από μικρό μέχρι τεράστιο. Όλα κρατούν ακέραιους αριθμούς, δηλαδή χωρίς δεκαδικό μέρος. Όσο μεγαλύτερο είναι το εύρος, τόσο μεγαλύτερους αριθμούς μπορούμε να αποθηκεύσουμε.

```
byte lives = 3;

short students = 250;

int population = 1000000;

long stars = 9000000000L;
```

Το L στο τέλος του μεγάλου αριθμού δηλώνει κυριολεκτική τιμή τύπου long.

Οι πραγματικοί αριθμοί: float και double

Μια πίτσα μπορεί να κοστίζει 8.50 ευρώ και ένα παιδί να έχει ύψος 1.42 μέτρα. Αυτοί είναι πραγματικοί αριθμοί (κινητής υποδιαστολής). Ο double προσφέρει μεγαλύτερη ακρίβεια από τον float. Στη Java γράφουμε τελεία αντί για δεκαδικό κόμμα.

```
float height = 1.42f;

double price = 8.50;
```

Το f δηλώνει κυριολεκτική τιμή τύπου float. Τα εύρη στον πίνακα είναι προσεγγιστικά για θετικές μη μηδενικές τιμές· υπάρχουν και αντίστοιχες αρνητικές.

Ο boolean: σαν διακόπτης ΝΑΙ/ΟΧΙ

Φαντάσου έναν διακόπτη φωτός: είτε είναι αναμμένος είτε σβηστός. Ο λογικός τύπος boolean παίρνει μόνο δύο τιμές: true (αληθής) και false (ψευδής).

```
boolean lightOn = true;

boolean raining = false;
```

Ο char: ένας χαρακτήρας

Φαντάσου ένα κουτάκι όπου βάζεις ένα γράμμα, ψηφίο ή σύμβολο. Αυτός είναι ο τύπος char. Γράφουμε την τιμή σε μονά εισαγωγικά. Ο char αποθηκεύει μία μονάδα κώδικα UTF-16· ορισμένα σύμβολα Unicode χρειάζονται δύο τέτοιες μονάδες.

```
char initial = 'M';

char symbol = '@';

char greek = 'Ω';
```

5. Τελεστής ανάθεσης (=)

Αν γράψω στην ετικέτα ενός κουτιού «πόντοι» και βάλω μέσα 50, έχω αναθέσει την τιμή 50. Ο τελεστής ανάθεσης = κάνει ακριβώς αυτό: θέτει τιμή σε μεταβλητή. Δεν σημαίνει «είναι ίσο με»!

```
int points = 50;
points = 70;
```

Πρόσεξε = σημαίνει ανάθεση τιμής. == σημαίνει έλεγχος ισότητας. Αυτά είναι δύο διαφορετικά πράγματα.

6. Αριθμητικοί τελεστές

Είναι τα εργαλεία των μαθηματικών πράξεων. Χρησιμοποιούμε + για πρόσθεση, − για αφαίρεση, * για πολλαπλασιασμό, / για πηλίκο διαίρεσης και % για υπόλοιπο ακέραιας διαίρεσης.

Τελεστής	Όνομα	Παράδειγμα → αποτέλεσμα
+	Πρόσθεση	18 + 10 → 28
-	Αφαίρεση	18 - 10 → 8
*	Πολλαπλασιασμός	18 * 10 → 180
/	Πηλίκο διαίρεσης	18 / 10 → 1
%	Υπόλοιπο ακέραιας διαίρεσης	18 % 10 → 8

Γιατί το 18 / 10 δίνει 1;

Έχεις 18 μπισκότα και θέλεις να φτιάξεις πακέτα των 10. Μπορείς να φτιάξεις 1 ολόκληρο πακέτο. Αν η Java διαιρεί δύο ακέραιους, κρατά το ακέραιο πηλίκο, δηλαδή 1. Αν ένας αριθμός είναι πραγματικός, μπορεί να εμφανιστεί δεκαδικό αποτέλεσμα.

```
18 / 10       // 1: ακέραια διαίρεση
18.0 / 10     // 1.8: διαίρεση με πραγματικό
```

Τι κάνει το %;

Από τα 18 μπισκότα, αφού φτιάξουμε ένα πακέτο των 10, περισσεύουν 8. Αυτό το 8 είναι το υπόλοιπο ακέραιας διαίρεσης.

```
18 % 10       // 8
20 % 10       // 0
7 % 2         // 1
```

Χρήσιμο κόλπο: ένας ακέραιος είναι άρτιος αν το υπόλοιπο της διαίρεσής του με το 2 είναι μηδέν.

7. Σχεσιακοί τελεστές: οι συγκρίσεις

Σκέψου δύο παιδιά που συγκρίνουν τις ηλικίες τους: «Είσαι μεγαλύτερος από εμένα;» Η απάντηση μπορεί να είναι μόνο αληθής ή ψευδής. Οι σχεσιακοί τελεστές συγκρίνουν δύο τιμές και επιστρέφουν boolean.

Σύμβολο	Περιγραφή	Παράδειγμα → αποτέλεσμα
==	Ισότητα	18 == 10 → false
<	Μικρότερο	18 < 10 → false
>	Μεγαλύτερο	18 > 10 → true
<=	Μικρότερο ή ίσο	18 <= 10 → false
>=	Μεγαλύτερο ή ίσο	18 >= 10 → true
!=	Διαφορετικό	18 != 10 → true

```
int age = 12;
boolean sameAge = (age == 12);   // true
boolean older = (age > 15);     // false
```

8. Λογικοί τελεστές: οι κανόνες με ΝΑΙ και ΟΧΙ

Σκέψου κανόνες για μια σχολική εκδρομή. Μερικές φορές θέλουμε να ισχύουν ΔΥΟ συνθήκες, άλλες φορές αρκεί ΜΙΑ. Οι λογικοί τελεστές συνδυάζουν λογικές τιμές.

! — όχι (NOT)

Αν η πρόταση «βρέχει» είναι αληθής, η πρόταση «δεν βρέχει» είναι ψευδής. Το ! αντιστρέφει την τιμή.

```
boolean raining = true;
boolean dry = !raining;      // false
```

&& — και (AND)

Για να μπεις στο γήπεδο χρειάζεσαι ΚΑΙ εισιτήριο ΚΑΙ άδεια. Αν λείπει ένα από τα δύο, δεν μπαίνεις. Το && είναι true μόνο όταν και οι δύο συνθήκες είναι true.

```
boolean ticket = true;
boolean permission = false;
boolean canEnter = ticket && permission; // false
```

|| — ή (OR)

Μπορείς να διαλέξεις φρούτο αν υπάρχει μήλο Ή μπανάνα. Αρκεί να υπάρχει τουλάχιστον ένα. Το || είναι true αν μία ή και οι δύο συνθήκες είναι true.

```
boolean apple = false;
```

```
boolean banana = true;
boolean hasFruit = apple || banana; // true
```

^ — αποκλειστικό ή (XOR)

Έχεις δικαίωμα να διαλέξεις ακριβώς ΕΝΑ από δύο παιχνίδια. Αν διαλέξεις και τα δύο, ο κανόνας δεν τηρείται. Το XOR είναι true μόνο όταν οι δύο λογικές τιμές είναι διαφορετικές.

```
boolean chooseBall = true;
boolean choosePuzzle = false;
boolean valid = chooseBall ^ choosePuzzle; // true
```

Πίνακας αλήθειας

Ο πίνακας αλήθειας δείχνει όλους τους δυνατούς συνδυασμούς. Διάβαζε κάθε γραμμή σαν να απαντάς ΝΑΙ ή ΟΧΙ σε δύο ερωτήσεις.

A	B	!A	A && B	A || B	A ^ B
true	true	false	true	true	false
true	false	false	false	true	true
false	true	true	false	true	true
false	false	true	false	false	false

Λεπτομέρεια ορολογίας Το σχολικό βιβλίο δείχνει επίσης A & B και A | B. Για boolean τιμές δίνουν τα ίδια λογικά αποτελέσματα με && και ||, αλλά οι & και | δεν κάνουν short-circuit evaluation.

9. Short-circuit evaluation (αξιολόγηση βραχυκυκλώματος)

Φαντάσου ότι για να βγεις διάλειμμα πρέπει να έχει τελειώσει το μάθημα ΚΑΙ να έχει χτυπήσει το κουδούνι. Αν το μάθημα δεν έχει τελειώσει, ξέρεις ήδη ότι δεν μπορείς να βγεις. Δεν χρειάζεται να εξετάσεις το δεύτερο ερώτημα. Αυτό είναι short-circuit evaluation: η Java δεν αξιολογεί το δεύτερο μέρος όταν το πρώτο αρκεί για το αποτέλεσμα.

```
false && (5 > 2)    // false, το δεύτερο μέρος δεν χρειάζεται
true || (5 < 2)     // true, το δεύτερο μέρος δεν χρειάζεται
```

Συμβαίνει με && όταν το πρώτο είναι false και με || όταν το πρώτο είναι true. Ο τελεστής ^ δεν λειτουργεί έτσι.

10. Μεταβλητές στιγμιοτύπου (instance variables) και πεδία (attributes)

Επιστρέφουμε στα αυτοκίνητα παιχνιδιών. Η κλάση Car είναι το σχέδιο. Κάθε αυτοκίνητο-αντικείμενο έχει τη δική του ταχύτητα και το δικό του χρώμα. Οι μεταβλητές που ανήκουν σε κάθε συγκεκριμένο αντικείμενο λέγονται μεταβλητές στιγμιοτύπου (instance variables), και είναι πεδία (attributes) του αντικειμένου. Καταλαμβάνουν χώρο στη μνήμη.

```
class Car {
    private int speed;
    private String color;
}
```

Η λέξη private είναι τροποποιητής πρόσβασης: το πεδίο χρησιμοποιείται απευθείας μόνο μέσα στην κλάση. Η public δίνει ευρύτερη πρόσβαση. Ο String είναι κλάση για κείμενο και όχι πρωτογενής τύπος δεδομένων.

Γενική σύνταξη δήλωσης μεταβλητής στιγμιοτύπου

```
private int speed;
public boolean available;
```

Η σειρά είναι: τροποποιητής πρόσβασης (π.χ. public/private) + τύπος δεδομένων + όνομα μεταβλητής + ;. Οι τροποποιητές πρόσβασης δεν είναι πάντοτε υποχρεωτικοί.

11. Κανόνες ονοματοδοσίας

Όπως δεν γράφουμε δύο διαφορετικά ονόματα στο ίδιο θρανίο, έτσι δίνουμε σε κάθε μεταβλητή ένα κατανοητό όνομα. Προτιμάμε περιγραφικά ονόματα. Συνήθως τα ονόματα μεταβλητών ξεκινούν με μικρό γράμμα και τα ονόματα κλάσεων με κεφαλαίο.

Σωστό	Λάθος	Γιατί
studentAge	student age	Δεν επιτρέπονται κενά
age2	2age	Δεν αρχίζουμε με ψηφίο
number	int	Δεν χρησιμοποιούμε δεσμευμένες λέξεις
Student (κλάση)	—	Σύμβαση: κεφαλαίο στην αρχή της κλάσης

Το βιβλίο συμβουλεύει να μη δίνουμε ελληνικά ονόματα στις μεταβλητές και, αν χρειάζεται, να χρησιμοποιούμε greeklish. Η Java τεχνικά επιτρέπει ελληνικά γράμματα, αλλά τα αγγλικά ονόματα είναι συνηθέστερη πρακτική.

12. Ένα μικρό παράδειγμα που ενώνει όσα μάθαμε

Η Άννα έχει 12 πόντους και ο Νίκος 10. Θέλουμε να δούμε ποιος έχει περισσότερους και αν και οι δύο έχουν τουλάχιστον 10 πόντους.

```
int annaPoints = 12;
int nikosPoints = 10;
boolean annaWins = annaPoints > nikosPoints;
boolean bothPassed = annaPoints >= 10 && nikosPoints >= 10;
```

1. Οι δύο πρώτες εντολές δηλώνουν ακέραιες μεταβλητές και αναθέτουν τιμές. 2. Ο σχεσιακός τελεστής > ελέγχει αν 12 > 10, άρα annaWins = true. 3. Ο τελεστής && συνδυάζει δύο συγκρίσεις: και οι δύο είναι αληθείς, άρα bothPassed = true.

13. Επανάληψη με απλά λόγια

Όρος του βιβλίου	Πώς να τον θυμάμαι
Μεταβλητή	Κουτί με όνομα και περιεχόμενο
Τύπος δεδομένων	Τι είδος πράγματος χωρά στο κουτί
Τελεστής ανάθεσης =	Βάζω τιμή στο κουτί
Αριθμητικός τελεστής	Κάνω μαθηματική πράξη
Σχεσιακός τελεστής	Συγκρίνω δύο πράγματα
Λογικός τελεστής	Ενώνω κανόνες ΝΑΙ/ΟΧΙ
Πίνακας αλήθειας	Βλέπω όλους τους συνδυασμούς
Short-circuit evaluation	Δεν κάνω περιττό δεύτερο έλεγχο
Κλάση / αντικείμενο	Σχέδιο / συγκεκριμένο παιχνίδι
Μεταβλητή στιγμιοτύπου	Χαρακτηριστικό συγκεκριμένου αντικειμένου
API / βιβλιοθήκη	Έτοιμα εργαλεία προγραμματισμού

14. Ασκήσεις εμπέδωσης

1. Ποια είναι η διαφορά ανάμεσα σε κλάση και αντικείμενο;

Απάντηση: .....

2. Τι είναι τα APIs (βιβλιοθήκες) και τι σημαίνει «Δεν επανεφευρίσκουμε τον τροχό»;

Απάντηση: .....

3. Γράψε μια δήλωση ακέραιας μεταβλητής age με τιμή 11.

Απάντηση: .....

4. Ποιος τύπος δεδομένων ταιριάζει: 15, 2.75, true, 'K';

Απάντηση: .....

5. Τι αποτέλεσμα δίνουν 19 / 5 και 19 % 5;

Απάντηση: .....

6. Τι διαφορά έχουν = και ==;

Απάντηση: .....

7. Υπολόγισε: 8 > 3, 8 != 8, 8 <= 10.

Απάντηση: .....

8. Αν A=true και B=false, βρες A && B, A || B, A ^ B και !A.

Απάντηση: .....

9. Πότε γίνεται short-circuit evaluation με τον &&;

Απάντηση: .....

10. Τι είναι μια μεταβλητή στιγμιοτύπου και τι σημαίνει private;

Απάντηση: .....

11. Διόρθωσε τα ονόματα: 2score, my age, int.

Απάντηση: .....

12. Εξήγησε γιατί boolean success = (10 >= 10); δίνει true.

Απάντηση: .....

15. Ενδεικτικές απαντήσεις

1. Η κλάση είναι σχέδιο, το αντικείμενο συγκεκριμένο στιγμιότυπο.

2. Έτοιμες λειτουργίες και τρόποι χρήσης τους· επαναχρησιμοποιούμε υπάρχουσες λύσεις.

3. int age = 11;

4. int, double, boolean, char αντίστοιχα.

5. 19 / 5 = 3 και 19 % 5 = 4.

6. = ανάθεση, == σύγκριση ισότητας.

7. true, false, true.

8. false, true, true, false.

9. Όταν το πρώτο μέρος του && είναι false, δεν αξιολογείται το δεύτερο.

10. Πεδίο συγκεκριμένου αντικειμένου· private σημαίνει πρόσβαση μέσα στην κλάση.

11. score2, myAge, number.

12. Επειδή 10 >= 10 είναι αληθές.