

Ο κώδικας είναι ένα απλό πρόγραμμα Java που κάνει **ανάλυση βαθμών μιας τάξης**. Ζητάει από τον χρήστη πόσοι μαθητές υπάρχουν, διαβάζει τους βαθμούς τους και στο τέλος υπολογίζει **μέσο όρο, μεγαλύτερο βαθμό, μικρότερο βαθμό και πόσοι μαθητές έχουν βαθμό πάνω από τον μέσο όρο**.

1. Εισαγωγή της Scanner

```
import java.util.Scanner;
```

Η εντολή αυτή εισάγει την κλάση `Scanner`, η οποία χρησιμοποιείται για να παίρνει δεδομένα από τον χρήστη μέσω του πληκτρολογίου.

2. Δημιουργία της κλάσης

```
public class GradesAnalysis {
```

Δημιουργείται μια κλάση με όνομα `GradesAnalysis`.

Στη Java, ο κώδικας ενός προγράμματος οργανώνεται μέσα σε κλάσεις.

3. Η μέθοδος `main`

```
public static void main(String[] args) {
```

Η `main` είναι η **κύρια μέθοδος** του προγράμματος. Από εδώ ξεκινάει η εκτέλεση του κώδικα.

4. Δημιουργία του Scanner

```
Scanner input = new Scanner(System.in);
```

Δημιουργείται ένα αντικείμενο `Scanner` με όνομα `input`.

Το:

```
System.in
```

σημαίνει ότι τα δεδομένα θα διαβάζονται από την **είσοδο του πληκτρολογίου**.

Έτσι, για παράδειγμα, όταν χρησιμοποιούμε:

```
input.nextInt();
```

το πρόγραμμα περιμένει από τον χρήστη να πληκτρολογήσει έναν ακέραιο αριθμό.

5. Εισαγωγή του αριθμού των μαθητών

```
System.out.print("Πόσους μαθητές έχει η τάξη; ");  
int n = input.nextInt();
```

Αρχικά εμφανίζεται το μήνυμα:

Πόσους μαθητές έχει η τάξη;

Ο χρήστης, για παράδειγμα, μπορεί να πληκτρολογήσει:

5

Ο αριθμός αποθηκεύεται στη μεταβλητή *n*.

Άρα:

n = 5

6. Δημιουργία πίνακα βαθμών

```
int[] grades = new int[n];
```

Δημιουργείται ένας πίνακας ακεραίων με όνομα *grades*.

Το μέγεθος του πίνακα είναι *n*, δηλαδή όσοι είναι οι μαθητές.

Αν:

n = 5

τότε δημιουργείται ένας πίνακας με 5 θέσεις:

```
grades[0]  
grades[1]  
grades[2]  
grades[3]  
grades[4]
```

Προσοχή: Στη Java οι θέσεις ενός πίνακα ξεκινούν από το 0, όχι από το 1.

7. Εισαγωγή των βαθμών

```
for (int i = 0; i < n; i++) {  
    System.out.print("Δώσε τον βαθμό του μαθητή " + (i + 1) + ": ");  
    grades[i] = input.nextInt();  
}
```

Εδώ υπάρχει ένας *for* βρόχος που επαναλαμβάνεται *n* φορές, δηλαδή μία φορά για κάθε μαθητή.

Η μεταβλητή:

```
int i = 0;
```

ξεκινάει από το 0.

Κάθε φορά που ολοκληρώνεται ο βρόχος:

```
i++
```

αυξάνει το i κατά 1.

Η συνθήκη:

$i < n$

σημαίνει ότι ο βρόχος συνεχίζεται όσο το i είναι μικρότερο από το n .

Για παράδειγμα, αν έχουμε 5 μαθητές, οι τιμές του i θα είναι:

$0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4$

Στη συνέχεια:

```
grades[i] = input.nextInt();
```

διαβάζει τον βαθμό και τον αποθηκεύει στον αντίστοιχο πίνακα.

Για παράδειγμα, αν ο χρήστης δώσει:

```
15
18
12
17
10
```

ο πίνακας θα είναι:

```
grades = [15, 18, 12, 17, 10]
```

Το:

$(i + 1)$

χρησιμοποιείται μόνο για την εμφάνιση του αριθμού του μαθητή, ώστε να βλέπουμε:

```
Δώσε τον βαθμό του μαθητή 1:
Δώσε τον βαθμό του μαθητή 2:
Δώσε τον βαθμό του μαθητή 3:
...
```

ενώ ο πίνακας εσωτερικά συνεχίζει να χρησιμοποιεί θέσεις $0, 1, 2, \dots$

8. Αρχικοποίηση των μεταβλητών

```
int sum = 0;
int max = grades[0];
int min = grades[0];
```

Δημιουργούνται τρεις μεταβλητές.

sum

```
int sum = 0;
```

Θα χρησιμοποιηθεί για να αποθηκεύσει το **άθροισμα όλων των βαθμών**.

Ξεκινάει από 0, επειδή δεν έχουμε προσθέσει ακόμα κανέναν βαθμό.

max

```
int max = grades[0];
```

Το **max** θα κρατάει τον **μεγαλύτερο βαθμό**.

Αρχικά θεωρούμε ότι ο μεγαλύτερος βαθμός είναι ο πρώτος βαθμός του πίνακα.

min

```
int min = grades[0];
```

Αντίστοιχα, το **min** θα κρατάει τον **μικρότερο βαθμό**.

Και αυτός αρχικά θεωρείται ότι είναι ο πρώτος βαθμός.

9. Υπολογισμός αθροίσματος, μέγιστου και ελάχιστου

```
for (int grade : grades) {  
    sum += grade;  
  
    if (grade > max)  
        max = grade;  
  
    if (grade < min)  
        min = grade;  
}
```

Εδώ χρησιμοποιείται ένας **enhanced for loop**, γνωστός και ως **for - each**.

```
for (int grade : grades)
```

σημαίνει:

Για κάθε στοιχείο του πίνακα **grades**, βάλε την τιμή του στη μεταβλητή **grade**.

Για παράδειγμα, αν:

```
grades = [15, 18, 12, 17, 10]
```

η μεταβλητή **grade** θα πάρει διαδοχικά τις τιμές:

```
15  
18  
12  
17  
10
```

Υπολογισμός αθροίσματος

```
sum += grade;
```

είναι συντομογραφία του:

```
sum = sum + grade;
```

Άρα οι υπολογισμοί γίνονται:

```
sum = 0
sum = 0 + 15 = 15
sum = 15 + 18 = 33
sum = 33 + 12 = 45
sum = 45 + 17 = 62
sum = 62 + 10 = 72
```

Άρα:

```
sum = 72
```

Εύρεση μέγιστου βαθμού

```
if (grade > max) max = grade;
```

Ελέγχει αν ο τρέχων βαθμός είναι μεγαλύτερος από τον βαθμό που έχουμε μέχρι εκείνη τη στιγμή ως `max`.

Για παράδειγμα:

```
max = 15
```

έρχεται ο βαθμός 18.

Επειδή:

```
18 > 15
```

γίνεται:

```
max = 18
```

Με αυτόν τον τρόπο στο τέλος το `max` περιέχει τον μεγαλύτερο βαθμό.

Εύρεση ελάχιστου βαθμού

```
if (grade < min) min = grade;
```

Ελέγχει αντίστοιχα αν ο τρέχων βαθμός είναι μικρότερος από τον μέχρι τώρα μικρότερο.

Με τους βαθμούς:

```
15, 18, 12, 17, 10
```

στο τέλος:

```
min = 10
```

10. Υπολογισμός μέσου όρου

`double average = (double) sum / n;`

Εδώ υπολογίζεται ο **μέσος όρος**.

Ο τύπος είναι:

μέσος όρος = άθροισμα βαθμών / αριθμός μαθητών

Στο παράδειγμά μας:

`sum = 72`
`n = 5`

οπότε:

`average = 72 / 5 = 14.4`

Γιατί υπάρχει το `(double)`;

Το:

`(double) sum`

μετατρέπει προσωρινά το `sum` από `int` σε `double`.

Αυτό είναι σημαντικό ώστε η διαίρεση να γίνει ως **δεκαδική διαίρεση**.

Για παράδειγμα, θέλουμε:

`72 / 5 = 14.4`

και όχι ακέραιο αποτέλεσμα.

11. Μέτρηση μαθητών πάνω από τον μέσο όρο

`int aboveAverageCount = 0;`

Δημιουργείται μια μεταβλητή που θα μετράει πόσοι μαθητές έχουν βαθμό μεγαλύτερο από τον **μέσο όρο**.

Ξεκινάει από:

`0`

Στη συνέχεια:

```
for (int grade : grades) {  
    if (grade > average) aboveAverageCount++;  
}
```

Το πρόγραμμα περνάει ξανά από όλους τους βαθμούς.

Ελέγχει:

```
if (grade > average)
```

δηλαδή:

Είναι ο βαθμός μεγαλύτερος από τον μέσο όρο;

Αν ναι:

```
aboveAverageCount++;
```

Το ++ αυξάνει την τιμή κατά 1.

Για το παράδειγμά μας:

Βαθμοί:	15	18	12	17	10
Μέσος όρος:	14.4				

Έχουν μεγαλύτερο βαθμό από το 14.4 οι:

15
18
17

Άρα:

```
aboveAverageCount = 3
```

12. Εμφάνιση αποτελεσμάτων

```
System.out.printf("\nΜέσος όρος: %.2f\n", average);
```

Εμφανίζει τον μέσο όρο.

Το:

```
%.2f
```

σημαίνει ότι ο αριθμός θα εμφανιστεί με **2 δεκαδικά ψηφία**.

Άρα:

```
14.4
```

θα εμφανιστεί ως:

```
14.40
```

Το:

```
\n
```

σημαίνει αλλαγή γραμμής.

Στη συνέχεια:

```
System.out.println("Μέγιστος βαθμός: " + max);
```

εμφανίζει τον μεγαλύτερο βαθμό.

```
System.out.println("Ελάχιστος βαθμός: " + min);
```

εμφανίζει τον μικρότερο βαθμό.

Και:

```
System.out.println("Μαθητές πάνω από τον μέσο όρο: " + aboveAverageCount);
```

εμφανίζει πόσοι μαθητές έχουν βαθμό μεγαλύτερο από τον μέσο όρο.

13. Κλείσιμο του Scanner

```
input.close();
```

Κλείνει το αντικείμενο Scanner, αφού πλέον δεν χρειάζεται να διαβάσουμε άλλα δεδομένα.

Συνολικά τι κάνει το πρόγραμμα;

Αν ο χρήστης εισάγει:

Πόσους μαθητές έχει η τάξη; 5

Δώσε τον βαθμό του μαθητή 1: 15
Δώσε τον βαθμό του μαθητή 2: 18
Δώσε τον βαθμό του μαθητή 3: 12
Δώσε τον βαθμό του μαθητή 4: 17
Δώσε τον βαθμό του μαθητή 5: 10

το πρόγραμμα αποθηκεύει τους βαθμούς:

[15, 18, 12, 17, 10]

και υπολογίζει:

Άθροισμα = 72
Μέσος όρος = 14.40
Μέγιστος βαθμός = 18
Ελάχιστος βαθμός = 10
Μαθητές πάνω από τον μέσο όρο = 3

Άρα η τελική έξοδος θα είναι:

Μέσος όρος: 14.40
Μέγιστος βαθμός: 18
Ελάχιστος βαθμός: 10
Μαθητές πάνω από τον μέσο όρο: 3

Η λογική του προγράμματος σε 6 βήματα

1. Ρωτάει πόσοι μαθητές υπάρχουν.
2. Δημιουργεί έναν πίνακα για να αποθηκεύσει τους βαθμούς.
3. Διαβάζει όλους τους βαθμούς από το πληκτρολόγιο.
4. Υπολογίζει άθροισμα, μέγιστο και ελάχιστο.
5. Υπολογίζει τον μέσο όρο και μετράει τους μαθητές που είναι πάνω από αυτόν.
6. Εμφανίζει όλα τα αποτελέσματα.