2. ΕΙΣΑΓΩΓΗ ΣΤΗ JAVA

Η γλώσσα προγραμματισμού Java

Βασικό συντακτικό, ορισμός μεταβλητών, έλεγχος ροής

ΜΕΤΑΒΛΗΤΕΣ - ΑΝΑΘΕΣΕΙΣ

Μεταβλητές

- Τα προγράμματα μας χρησιμοποιούν δεδομένα για να κάνουν υπολογισμούς. Τα δεδομένα αυτά αποθηκεύονται σε μεταβλητές.
- Μέσα στο πρόγραμμα μας, μία μεταβλητή χαρακτηρίζεται από:
  - Το όνομα της (πως θα αναφερόμαστε σε αυτή στο πρόγραμμα)
  - Την τιμή της (τα δεδομένα που κρατάει)
  - Τον τύπο της (τι είδους δεδομένα κρατάει)
- Η Java είναι strongly typed γλώσσα: κάθε μεταβλητή θα πρέπει να έχει ένα τύπο ο οποίος δεν αλλάζει.

Ορισμός/Δήλωση μεταβλητών

- Ορισμός μεταβλητής

```
<τυπος> <όνομα μεταβλητής> [ = τιμή];
```

- Ο ορισμός της μεταβλητής γίνεται μόνο μία φορά, πριν ή όταν θα την χρησιμοποιήσουμε για πρώτη φορά.
- Με τον ορισμό της μεταβλητής:
  - Δεσμεύουμε ένα όνομα για τη μεταβλητή και το όνομα αντιστοιχίζεται σε μια θέση μνήμης.
  - Καθορίζουμε τον τύπο της μεταβλητής
    - Ο τύπος της μεταβλητής είναι είτε ένας πρωταρχικός τύπος, είτε μια κλάση
  - (Προαιρετικά αλλά πολύ συχνά) Αναθέτουμε μια αρχική τιμή στην μεταβλητή.
- Αφού ορίσουμε μια μεταβλητή μπορούμε να την χρησιμοποιήσουμε μέσα στο πρόγραμμα, είτε για να διαβάσουμε την τιμή της, είτε για να της αναθέσουμε τιμή.
- Η χρήση μπορεί να γίνει πολλές φορές, ο ορισμός μόνο μία! Δεν μπορούμε να ορίσουμε την μεταβλητή ξανά ή να της αλλάξουμε τον τύπο.

Παραδείγματα

Πρωταρχικός τύπος int για ακέραιες μεταβλητές

```
int firstInt = 2;
int secondInt = 10;
```

Ορισμός ακέραιων μεταβλητών με αρχικοποίηση

Πρωταρχικός τύπος double για πραγματικές μεταβλητές

```
double myDouble = 2.5;
```

Ορισμός πραγματικής μεταβλητής με αρχικοποίηση

Πρωταρχικός τύπος boolean για λογικές μεταβλητές

```
boolean myBoolean = true;
```

Ορισμός λογικής μεταβλητής με αρχικοποίηση

Υπάρχουσα κλάση String για αλφαριθμητικά

```
String myString = "Hello";
```

Ορισμός String μεταβλητής με αρχικοποίηση

```
int sum;
sum = fistInt + secondInt;
```

Ορισμός μεταβλητής χωρίς αρχικοποίηση. Δεν θα έχει τιμή μέχρι να της δώσουμε

```
int sumAgain = fistInt + secondInt;
```

Ορισμός μεταβλητής και αρχικοποίηση με πράξη

Παράδειγμα 2

- Φτιάξτε ένα πρόγραμμα που τυπώνει το λόγο δύο ακεραίων.

Division.java

```
class Division
{
  public static void main(String args[])
  {
    int enumerator = 30;
    int denominator = 10;
    double division = enumerator/denominator;
    System.out.println(division);
  }
}
```

Η println μετατρέπει τον πραγματικό αριθμό σε String

Δήλωση δύο ακεραίων μεταβλητών και μιας πραγματικής
Οι ακέραιες μεταβλητές αρχικοποιούνται με μια σταθερά, η πραγματική με το αποτέλεσμα της διαίρεσης των δύο ακεραίων

Division-Err1.java

ΛΑΘΟΣ!

```
class Division
{
  public static void main(String args[])
  {
    int enumerator = 30;
    int denominator = 10;
    double division = enumerator/denominator;
    int enumerator = 40;
    System.out.println(division);
  }
}
```

Το μήνυμα λάθους μας λέει ότι ορίσαμε δύο φορές τη μεταβλητή με το όνομα enumerator

```
>javac Division-Err1.java
Division-Err1.java:8: error: variable enumerator is already
defined in method main(String[])
                int enumerator = 40;
                    ^
1 error
```

Το ^ μας δείχνει σε ποιο σημείο είναι το λάθος.

Division-Err2.java

```
class Division
{
  public static void main(String args[])
  {
    int enumerator = 30;
    int denominator = 10;
    division = enumerator/denominator;
    System.out.println(division);
  }
}
```

ΛΑΘΟΣ!

Το μήνυμα λάθους μας λέει ότι ο compiler δεν ξέρει τι σημαίνει το σύμβολο division γιατί δεν έχουμε ορίσει την μεταβλητή

```
>javac Division-Err2.java
Division-Err2.java:7: error: cannot find symbol
                division = enumerator/denominator;
                ^
  symbol:   variable division
  location: class Division
Division-Err2.java:8: error: cannot find symbol
                System.out.println(division);
                                   ^
  symbol:   variable division
  location: class Division
2 errors
```

Τύποι μεταβλητών

- Ο τύπος της μεταβλητής είναι είτε ένας πρωταρχικός τύπος, είτε μια κλάση
- Οι τύποι int, double και boolean είναι πρωταρχικοί (βασικοί) τύποι (primitive types)
- Εκτός από τους βασικούς τύπους, όλοι οι άλλοι τύποι (π.χ., String) είναι κλάσεις
- Όταν ο τύπος της μεταβλητής είναι μια κλάση αυτή η μεταβλητή είναι ένα αντικείμενο της κλάσης.

Πρωταρχικοί τύποι

Όνομα τύπου	Τιμή	Μνήμη
boolean	true/false	1 byte
char	Χαρακτήρας (Unicode)	2 bytes
byte	Ακέραιος	1 byte
short	Ακέραιος	2 bytes
int	Ακέραιος	4 bytes
long	Ακέραιος	8 bytes
float	Πραγματικός	4 bytes
double	Πραγματικός	8 bytes

Όταν ορίζουμε μια μεταβλητή δεσμεύεται ο αντίστοιχος χώρος στη μνήμη.
Το όνομα της μεταβλητής αντιστοιχίζεται με αυτό το χώρο στη μνήμη.

Πρωταρχικοί τύποι

Όνομα τύπου	Τιμή	Μνήμη
boolean	true/false	1 byte
char	Χαρακτήρας (Unicode)	2 bytes
byte	Ακέραιος	1 byte
short	Ακέραιος	2 bytes
int	Ακέραιος	4 bytes
long	Ακέραιος	8 bytes
float	Πραγματικός	4 bytes
double	Πραγματικός	8 bytes

Όταν ορίζουμε μια μεταβλητή δεσμεύεται ο αντίστοιχος χώρος στη μνήμη.
Το όνομα της μεταβλητής αντιστοιχίζεται με αυτό το χώρο στη μνήμη.

Ανάθεση

```
class Division
{
  public static void main(String args[])
  {
    int enumerator = 32;
    int denominator = 10;
    double division = enumerator/denominator;
    System.out.println(division);
  }
}
```

Έκφραση που αποτιμάται σε μια τιμή

μεταβλητή

Ο τύπος της έκφρασης και της μεταβλητής θα πρέπει να συμφωνούν

Ανάθεση/Εκχώρηση

- Το γενικό συντακτικό μιας ανάθεσης

```
<Μεταβλητή> = <έκφραση που αποτιμάται σε μια τιμή>
```

- Στο αριστερό μέρος έχουμε πάντα μια μεταβλητή.
  - Μπορούμε να έχουμε και τη δήλωση της μεταβλητής
- Το δεξί μέρος είναι μια έκφραση που μας δίνει μια τιμή. Η έκφραση μπορεί να είναι:
  - Μία σταθερά
  - Μια μεταβλητή
  - Μια πράξη μεταξύ μεταβλητών ή/και σταθερών
  - Η κλήση μιας μεθόδου που επιστρέφει μια τιμή.
- Πρέπει να υπάρχει συμβατότητα μεταξύ του τύπου της μεταβλητής (αριστερό μέρος) και του τύπου της τιμής που μας δίνει η έκφραση (δεξί μέρος).
- Πρώτα γίνεται ο υπολογισμός της τιμής στο δεξί μέρος και μετά εκχωρείται η τιμή στην μεταβλητή.

Division-Err3.java

ΛΑΘΟΣ!

```
class Division
{
  public static void main(String args[])
  {
    int enumerator = 30;
    int denominator = 10;
    double division = 0;
    enumerator/denominator = division;
    System.out.println(division);
  }
}
```

```
>javac Division-Err3.java
Division-Err3.java:8: error: unexpected type
                enumerator/denominator = division;
                           ^
  required: variable
  found:    value
```

Το μήνυμα λάθους μας λέει ότι ο compiler περίμενε να βρει μια μεταβλητή αλλά βρήκε μια τιμή.

Συμβατότητα

- Στην ανάθεση κατά κανόνα, η τιμή του δεξιού μέρους θα πρέπει να είναι ίδιου τύπου με την μεταβλητή του αριστερού μέρους.
- Υπάρχουν εξαιρέσεις όταν υπάρχει συμβατότητα μεταξύ τύπων

- byte → short → int → long → float → double
  - Μια τιμή τύπου T μπορούμε να την αναθέσουμε σε μια μεταβλητή τύπου που εμφανίζεται δεξιά του T.

- (Σε αντίθεση με την C) ο τύπος boolean δεν είναι συμβατός με τους ακέραιους.

Compatiblity.java

ΛΑΘΟΣ!

```
class Compatibility
{
   public static void main(String args[])
   {
        int intVariable = 30;
        double doubleVariable = intVariable;
        intVariable = doubleVariable;
   }
}
```

Αυτή η ανάθεση είναι αποδεκτή: μπορούμε να αναθέσουμε ακέραια τιμή σε πραγματική μεταβλητή

Αυτή η ανάθεση προκαλεί λάθος: δεν μπορούμε να αναθέσουμε πραγματική τιμή σε ακέραια μεταβλητή

```
>javac Compatibility.java
Compatibility.java:7: error: incompatible types: possible
lossy conversion from double to int
                intVariable = doubleVariable;
                              ^
1 error
```

Το μήνυμα λάθους μας λέει κάναμε μια ανάθεση μη συμβατής τιμής.

Casting – Division2.java

```
class Division
{
  public static void main(String args[])
  {
    int enumerator = 32;
    int denominator = 10;
    double division = enumerator/(double)denominator;
    System.out.println(division);
  }
}
```

Μετατροπή τύπου (type casting): **(double)denominator** μετατρέπει την τιμή της μεταβλητής **denominator** σε **double**

Αν δεν γίνει η μετατροπή, η διαίρεση μεταξύ ακεραίων μας δίνει πάντα ακέραιο.