ΒΑΣΙΚΑ ΣΤΟΙΧΕΙΑ ΤΗΣ ΓΛΩΣΣΑΣ JAVA

Δημήτρης Κ. Ιακωβίδης

Στόχος αυτής της ενότητας είναι η γνωριμία με τα βασικά στοιχεία της γλώσσας Java, τα οποία επιτρέπουν τη δημιουργία στοιχειωδών εκτελέσιμων εφαρμογών όπως εκείνες που αναπτύσσονται με τη γλώσσα C. Στη γλώσσα Java έχουν κληροδοτηθεί από τη C βασικά στοιχεία της, όπως είναι η σύνταξη, τύποι μεταβλητών και οι βασικές εντολές, κάνοντας πολύ εύκολη τη μεταφορά/μετάφραση ενός προγράμματος από γλώσσα C σε γλώσσα Java.

2.1 Μεταβλητές

Μέσα στη συνάρτηση main μπορούν να δηλωθούν **μεταβλητές** (variables) με τον ίδιο τρόπο που δηλώνονται και σε ένα πρόγραμμα C. Ομοίως μπορούμε να αναθέσουμε σε αυτές **τιμές** (values) και να τις **αρχικοποιήσουμε** (initialization), όπως εικονίζεται στο Σχ. 1. Η δήλωση μιας μεταβλητής σε ένα πρόγραμμα Java μπορεί να γίνει σε οποιοδήποτε σημείο του προγράμματος, όμως μόνο μια φορά.

Οι διαφορετικοί τύποι μεταβλητών της γλώσσας Java, πολλοί από τους οποίους έχουν κληροδοτηθεί από τη γλώσσα C και τη C++, ονομάζονται **πρωταρχικοί τύποι** (primitive types) και συνοψίζονται στον Πίνακα 1. Παρατηρείστε ότι το πρώτο γράμμα τους είναι πεζό.

```
1    class Test
2    {
3          public static void main(String args[])
4          {
5                int x;       // declaration
6                int y = 1;   // declaration and initialization
7
8                x = 5;       // assignment
9          }
10   }
```

Σχήμα 1. Χρήση μεταβλητών στη γλώσσα Java.

Πίνακας 1. Πρωταρχικοί τύποι μεταβλητών της γλώσσας Java.

Τύπος	*Περιγραφή*	*Μέγεθος (bytes)*	*Παράδειγμα*
byte	Ακέραιος πολύ μικρός	1	byte x = 5;
short	Ακέραιος μικρός	2	short x = 5;
int	Ακέραιος	4	Int x = 5;
long	Ακέραιος μεγάλος	8	long x = 5L;
float	Δεκαδικός	4	float x = 5.2F;
double	Δεκαδικός διπλής ακρίβειας	8	double x = 5.2;
boolean	Λογικές τιμές (μόνο true ή false, όχι 1 ή 0)	1	boolean x = true;
char	Χαρακτήρας (κωδικοποίηση Unicode)	2	char x = 'a';

Οι πρωταρχικοί τύποι μεταβλητών στη Java είναι πάντα προσημασμένοι (singed). Για παράδειγμα, ο τύπος byte έχει μέγεθος 1 byte = 8 bit, δηλαδή χωράει 2^8 ακεραίους αριθμούς. Όμως αυτοί οι αριθμοί δεν είναι από το 0 έως το 255 (ή 2^8-1) όπως διαισθητικά θα περιμέναμε, αλλά το διάστημά τους είναι κεντραρισμένο στο μηδέν, δηλαδή λαμβάνουν τιμές από το -128 (ή -2^{8-1}) έως το +127 (ή $+2^{8-1}$-1).

Πέρα από αυτούς, τους πρωταρχικούς τύπους μεταβλητών, στη Java υπάρχει και άλλο ένα είδος μεταβλητών που το όνομά τους ξεκινά με κεφαλαίο γράμμα. Οι τύποι αυτοί λέγονται τύποι κλάσης (class types). Αντιπροσωπευτικό παράδειγμα είναι ο τύπος String, ο οποίος χρησιμοποιείται για την αποθήκευση αλφαριθμητικών.

Η εκτύπωση μεταβλητών στην κονσόλα γίνεται και πάλι με τη System.out.println, όπως ακριβώς γίνεται και η εκτύπωση αλφαριθμητικών. Οι μεταβλητές μπορούν να εκτυπωθούν σε συνδυασμό με αλφαριθμητικά χρησιμοποιώντας τον τελεστή +. Αν μια έκφραση αποτελείται από πολλούς τελεστές +, και ένας τουλάχιστον από τους τελεστέους είναι αλφαριθμητικό, τότε οι μη αλφαριθμητικοί τελεστέοι μετατρέπονται σε

αλφαριθμητικά και πραγματοποιείται η πράξη της συνένωσης αλφαριθμητικών. Παραδείγματα εικονίζονται στο Σχ. 2.

```
1    class Test
2    {
3          public static void main(String args[])
4          {
5                String name = "Dimitris";
6                int age = 20;
7
8                System.out.println("My name is " + name + " and
9                I am " + age + " years old");
10
11               System.out.println(5 + 2);
12               // The output is: 7
13
14               System.out.println("Result = " + 5 + 2);
15               // The output is: Result = 52
16
17               System.out.println("Result = " + (5 + 2));
18               // The output is: Result = 7
19
20         }
21   }
```

Σχήμα 2. Εκτύπωση μεταβλητών στη κονσόλα με τη χρήση της System.out.println.

Εναλλακτικά της System.out.println μπορεί να χρησιμοποιηθεί και η **System.out.printf** η οποία είναι κληροδοτημένη από τη γλώσσα C και συντάσσεται με τον ίδιο ακριβώς τρόπο που συντάσσεται και στη C. Ως παράδειγμα δίνεται το Σχ. 3 με το οποίο εκτυπώνεται ότι και με τον κώδικα του Σχ. 2, αλλά χρησιμοποιώντας τη System.out.printf αντί της System.out.println.

Παρατηρείστε ότι για την εκτύπωση των ακεραίων χρησιμοποιείται το σύμβολο %d. Για την εκτύπωση αλφαριθμητικών χρησιμοποιείται το %s, ενώ για την εκτύπωση δεκαδικών αριθμών χρησιμοποιείται το σύμβολο %f. Θα επιλέγαμε να χρησιμοποιήσουμε τη System.out.printf για την εκτύπωση στην κονσόλα στην περίπτωση που επιθυμούμε να μορφοποιήσουμε εύκολα δεκαδικούς αριθμούς, π.χ. να ρυθμίσουμε το **πλήθος των δεκαδικών ψηφίων** που θα εκτυπωθεί. Αν χρησιμοποιήσουμε το σύμβολο %.2f τότε ο δεκαδικός αριθμός θα εκτυπωθεί με δύο δεκαδικά ψηφία, αν χρησιμοποιήσουμε το %.3f, θα εκτυπωθεί με τρία δεκαδικά ψηφία, κοκ.

```
1    class Test
2    {
3          public static void main(String args[])
4          {
5                String name = "Dimitris";
6                int age = 20;
7
8                System.out.printf("My name is %s and I am %d
9                years old", name, age);
10
11               System.out.printf("%d", 5 + 2);
12               // The output is: 7
13
14               System.out.printf("Result = %d%d", 5, 2);
15               // The output is: Result = 52
16
17               System.out.printf("Result = %d", (5 + 2));
18               // The output is: Result = 7
19
20         }
21   }
```

Σχήμα 3. Εκτύπωση μεταβλητών στη κονσόλα με τη χρήση της System.out.printf.

2.2 Μετατροπή τύπων

Σε κάθε ανάθεση τιμής ο τύπος αριστερά πρέπει να είναι ίδιος με τον τύπο δεξιά από το σύμβολο της ανάθεσης (=), π.χ. αν έχουμε x = y, ο τύπος της μεταβλητής x πρέπει να είναι ίδιος με τον τύπο της μεταβλητής y. Αν δεν είναι, θα πρέπει το δεξί μέρος να το μετατρέψουμε με σαφή τρόπο στον ίδιο τύπο με το αριστερό μέρος, χρησιμοποιώντας τη **μετατροπή τύπου** (type casting). Μετατροπές τύπων πραγματοποιούνται μόνο μεταξύ πρωταρχικών τύπων και όχι μεταξύ τύπων κλάσης. Παραδείγματα εικονίζονται στο Σχ. 4.

Πρέπει να σημειωθεί ότι αυτό που ονομάζουμε «μετατροπή τύπου» στην πραγματικότητα δε σημαίνει ότι μετατρέπεται μόνιμα ένας τύπος μεταβλητής σε έναν άλλο. Η μεταβλητή που βρίσκεται στα δεξιά του συμβόλου της ανάθεσης υποδύεται ότι είναι του τύπου που βρίσκεται στην παρένθεση. Για παράδειγμα στο Σχ. 4, η μεταβλητή y **υποδύεται** ότι είναι ακέραια, «φορώντας τη μάσκα» "(int)".

Η μετατροπή τύπου είναι ένας τελεστής και μπορεί να αποτελεί μέρος οποιασδήποτε έκφρασης πέραν της ανάθεσης, μαζί με άλλους τελεστές που παρουσιάζονται στην επόμενη ενότητα π.χ. 4+(int)x-2(int)z κλπ.

```
1    class Test
2    {
3          public static void main(String args[])
4          {
5                int x;
6                double y = 5.2;
7
8                x = (int)y; // type casting to int
9
10               System.out.println(x);
11         }
12   }
```

Σχήμα 4. Μετατροπή τύπου.

2.3 Τελεστές

Με τους τελεστές πραγματοποιούνται πράξεις μεταξύ μεταβλητών. Το σύνολο των τελεστών της γλώσσας Java εικονίζεται στον Πίνακα 2.

Πίνακας 2. Βασικοί τελεστές της γλώσσας Java.

Τελεστής	*Περιγραφή*	*Παράδειγμα*
=	Ανάθεση τιμής	int x = 5;
+, -, *, /, %	Βασικές αριθμητικές πράξεις. Το % σημαίνει υπόλοιπο διαίρεσης (modulo).	int x = 5 + 2;
	Ο τελεστής + πραγματοποιεί και συνένωση αλφαριθμητικών.	String s = “Hello ” + “ Java”;
()	Παρενθέσεις: Οι πράξεις σε παρένθεση πραγματοποιούνται πρώτες	int x = (5 + 2) * 3;
==	Σύγκριση: ίσο	boolean y = (x == 5);

!=	Σύγκριση: διάφορο	boolean y = (x == 5);
>, >=, <, <=	Συγκρίσεις: μεγαλύτερο, μεγαλύτερο ή ίσο κλπ	boolean y = (x >= 5);
&, &&	Λογικό και	boolean y = true & false;
I, II	Λογικό ή	boolean y = true | false;
!	Λογικό όχι	boolean y = !true;
++, --	Προσαύξηση ή μείωση κατά ένα	x++; ή ++x;
+=, -=, *=, /=, %=	Ανάθεση και πράξη	x+=2; (ισοδυναμεί με x = x + 2;)
?:	Τριαδικός	String s = (x >= 5)?"yes":"no";

Προσοχή χρειάζεται στη χρήση του **τελεστή της διαίρεσης (/)**. Όταν ο διαιρετέος και ο διαιρέτης είναι ακέραιοι, τότε πραγματοποιείται ακέραια διαίρεση, δηλαδή το αποτέλεσμα είναι ακέραιος αριθμός, π.χ. 5/2 δίνει 2 και όχι 2.5, και το υπόλοιπο της ακέραιας διαίρεσης δίνεται από την πράξη 5%2. Αντιθέτως, αν ένας από τους δύο είναι δεκαδικός τότε το αποτέλεσμα είναι δεκαδικός π.χ. 5/2.0 δίνει 2.5. Επίσης, ο παρονομαστής πρέπει να είναι πάντοτε διαφορετικός του μηδενός. Για τον έλεγχο αυτής της συνθήκης, συστήνεται η χρήση διακλάδωσης, όπως περιγράφεται στην ακόλουθη ενότητα.

2.4. Διακλαδώσεις

Οι διακλαδώσεις (branches) επιτρέπουν την εκτέλεση τμημάτων κώδικα υπό συνθήκη. Ως παράδειγμα δίνεται ο κώδικας του Σχ. 5, όπου ελέγχεται αν η τιμή της μεταβλητής x είναι μεγαλύτερη ή όχι από το 2. Περισσότερες από μια συνθήκες μπορούν να συνδυαστούν με τη χρήση των λογικών τελεστών (και, ή, όχι), π.χ. ((x > 5) && (x < 10)) εάν επιθυμούμε η τιμή του x να είναι ανάμεσα στο 5 και στο 10.

```
1    class Test
2    {
3          public static void main(String args[])
4          {
```

```
5                int x = 5;
6
7                if (x > 2)
8                {
9                      System.out.println("Greater than 2.");
10               }
11               else
12               {
13                     System.out.println("Smaller than 2.");
14               }
15         }
16   }
```

Σχήμα 5. Διακλάδωση με if – else.

Πολλαπλές διακλαδώσεις μπορούν να πραγματοποιηθούν συνδυάζοντας περισσότερες εκφράσεις if – else τη μία μετά την άλλη ή χρησιμοποιώντας διακλαδώσεις switch – case, όπως εικονίζεται στο Σχ. 6. Στο παράδειγμα αυτό, αν η μεταβλητή x λάβει μια ακέραια τιμή από 1 έως 3 θα ενεργοποιηθεί ο αντίστοιχος κλάδος case και θα εκτυπώσει την αντίστοιχη επιλογή. Σε διαφορετική περίπτωση, θα ενεργοποιηθεί ο κλάδος default και θα εκτυπώσει "You selected something else." Αξίζει να σημειωθεί ότι κάθε κλάδος τερματίζεται με την εντολή **break**. Αν δεν υπάρχει break, τότε η ροή του προγράμματος συνεχίζεται προς τους επόμενους κλάδους.

```
1    class Test
2    {
3          public static void main(String args[])
4          {
5                int x = 5;
6
7                switch(x)
8                {
9                      case 1:
10                     System.out.println("You selected 1.");
11                     break;
12
13                     case 2:
14                     System.out.println("You selected 2.");
15                     break;
16
17                     case 3:
18                     System.out.println("You selected 3.");
19                     break;
20
21                     default:
22                     System.out.println("You selected
```

```
23                      something else.");
24                  }
25          }
26  }
```

Σχήμα 6. Πολλαπλή διακλάδωση με switch – case.

2.5 Βρόχοι

Για την επανάληψη ενός τμήματος κώδικα πολλές φορές, χρησιμοποιούνται οι βρόχοι (loops). Εάν είναι γνωστό εκ των προτέρων το πλήθος των επαναλήψεων που πρέπει να πραγματοποιηθούν, τότε χρησιμοποιείται ο βρόχος for, όπως εικονίζεται στο Σχ. 7.

Προσέξτε πως η μεταβλητή i ορίζεται εντός του βρόχου, ως τοπική μεταβλητή. Αυτό είναι προαιρετικό, και σημαίνει πως η μεταβλητή i δεν υπάρχει έξω από τα όρια του βρόχου, άρα δε μπορούμε να τη χρησιμοποιήσουμε εκτός βρόχου. Γενικά, σε οποιοδήποτε μπλοκ κώδικα, δηλαδή κώδικα που βρίσκεται μέσα σε αγκύλες { }, μπορούμε να δηλώνουμε τοπικές μεταβλητές όπως είναι η μεταβλητή i, π.χ. για λόγους εκσφαλμάτωσης (debugging).

Η έκφραση i++ έχει ως αποτέλεσμα την προσαύξηση της μεταβλητής i κατά μία μονάδα σε κάθε επανάληψη. Αν επιθυμούμε μείωση κατά μια μονάδα, αυτό επιτυγχάνεται με i--, ή αν επιθυμούμε μεγαλύτερο βήμα αύξησης ή μείωσης τότε χρησιμοποιούμε π.χ. i+=2, i-=2 αντίστοιχα.

```
1   class Test
2   {
3           public static void main(String args[])
4           {
5                   for (int i = 1; i <= 10; i++)
6                   {
7                           System.out.println(i);
8                   }
9           }
10  }
```

Σχήμα 7. Βρόχος for που εμφανίζει τις τιμές από 1 έως 10.

Εναλλακτικά, μπορούν να χρησιμοποιηθούν οι βρόχοι while και do – while, όπως εικονίζονται στα Σχ. 8 και 9, αντίστοιχα.

```
1    class Test
2    {
3          public static void main(String args[])
4          {
5                int i = 1;
6                while(i<=10)
7                {
8                      System.out.println(i++);
9                }
10         }
     }
```

Σχήμα 8. Βρόχος while που εμφανίζει τις τιμές από 1 έως 10.

```
1    class Test
2    {
3          public static void main(String args[])
4          {
5                int i = 1;
6                do
7                {
8                      System.out.println(i++);
9                } while(i<10);
10         }
     }
```

Σχήμα 9. Βρόχος do – while που εμφανίζει τις τιμές από 1 έως 10.

2.6 Διακοπή εκτέλεσης βρόχων

Για τη διακοπή ενός βρόχου χρησιμοποιείται όπως και στην περίπτωση της switch – case η εντολή **break**. Υπάρχει δυνατότητα η ροή ενός βρόχου να διακοπεί και προσωρινά (για μια επανάληψη) με τη χρήση της εντολής **continue**.

```
1    class Test
2    {
3          public static void main(String args[])
4          {
5                for (int i = 1; i <= 10; i++)
6                {
7                      if (i == 5) break;
8                      System.out.println(i);
9                }
10         }
11   }
```

Σχήμα 10. Μόνιμη διακοπή ενός βρόχου με τη **break**. Το παράδειγμα θα εκτυπώσει από το 1 έως το 4.

```
1    class Test
2    {
3          public static void main(String args[])
4          {
5                for (int i = 1; i <= 10; i++)
6                {
7                      if (i == 5) continue;
8                      System.out.println(i);
9                }
10         }
     }
```

Σχήμα 11. Προσωρινή διακοπή ενός βρόχου με την **continue**. Το παράδειγμα θα εκτυπώσει από το 1 έως το 10 εκτός του 5.

2.7 Διακοπή προγράμματος

Η διακοπή της εκτέλεσης του προγράμματός μας δε γίνεται με τις break και continue. Εάν βρισκόμαστε μέσα στη main τότε οριστική διακοπή μπορεί να επέλθει με τη χρήση της εντολής **return**; .

Γενικότερος τρόπος οριστικής διακοπής του προγράμματός μας από οποιοδήποτε σημείο του προγράμματός μας επιτυγχάνεται με τη χρήση της

System.exit(0);

Ασκήσεις

1. Να υλοποιηθούν τα παραπάνω παραδείγματα.

2. Να κατασκευαστεί πρόγραμμα που να εκτυπώνει την προπαίδεια ενός αριθμού. Ο αριθμός θα δίνεται ως τιμή ακέραιας μεταβλητής x, π.χ. x = 5. Η προπαίδεια θα πρέπει να εκτυπώνεται στην παρακάτω μορφή:

1*5=5

2*5=10

...

10*5=50

3. Στην παραπάνω άσκηση να αντικαταστήσετε την πράξη του πολλαπλασιασμού με τις πράξεις: /, + και %. Τι παρατηρείτε; Πως εξηγείτε την έξοδο;

4. Στην παραπάνω άσκηση πως θα μπορούσαμε να κάνουμε τη διαίρεση για να μας δίνει δεκαδικούς αριθμούς στην έξοδο;

5. Να κατασκευάσετε ένα πρόγραμμα που να σχεδιάζει ένα παραλληλόγραμμο n γραμμών και m στηλών με αστερίσκους *. Π.χ. αν n=3 και m=2 να τυπώνει στην οθόνη:

**

**

**

6. Να τροποποιήσετε το παραπάνω πρόγραμμα ώστε να σχεδιάζει τρίγωνα, είτε ορθογώνια με κορυφή επάνω, είτε ορθογώνια με κορυφή κάτω, είτε ισοσκελή κλπ. Για παράδειγμα ένα τρίγωνο με κορυφή επάνω είναι το εξής:

*

**

***

****

7. Να χρησιμοποιήσετε τη switch – case για να δίνετε έναν αριθμό από το 1 έως το 12 στην είσοδο και να λαμβάνεται ολογράφως το όνομα του αντίστοιχου μήνα στην έξοδο.