

# Αντικειμενοστραφής Προγραμματισμός

## Ενότητα 3: Μεταβλητές (Variables)

**Μ. Φειδάκης**  
ΕΔΙΠ ΠαΔΑ



**Χ. Πατρικάκης**  
Καθηγητής, ΠαΔΑ

# Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons και βασίστηκε στο πρόγραμμα Java Foundations της Oracle Academy.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.

# Περιεχόμενα



1. Μεταβλητές (τύποι, δήλωση, ονομασία, λάθη, σταθερές)
2. Αριθμητικά δεδομένα (ακέραιοι, πραγματικοί)
3. Χαρακτήρες-Συμβολοσειρές-Ακολουθίες διαφυγής
4. Μετατροπές (explicit/implicit typecast)

# Μεταβλητές

- Τοποθεσίες στη μνήμη του υπολογιστή, τις οποίες έχουμε ονομάσει και στις οποίες αποθηκεύεται μία τιμή.

Παράδειγμα:

```
String a = "a";  
String b = "letter b";  
int k = 15;  
a = "letter a";  
String c = a + "and" + b;
```

# Μεταβλητές

- Θυμηθείτε τις μεταβλητές σε μια εξίσωση.
- Μπορούμε να τις δώσουμε οποιαδήποτε αξία.

Έστω  $y = -2x + 5$

Εάν  $x=0$ , τότε  $y=5$

Εάν  $x=2$ , τότε  $y=1$

- Παρομοίως και στη Java:

```
String x = "Alex";  
System.out.println("My name is " + x);
```



"My name is Alex"

# Γιατί μεταβλητές;

- Χωρίς μεταβλητές

```
System.out.println("My name is Alex");  
System.out.println("Alex is so cool!");  
System.out.println("Hooray Alex!");  
System.out.println("Please enjoy Alex  
Appreciation "  
+ "Day! My name is Alex. I know how excited "  
+ "everyone is to start appreciating Alex on Alex"  
+ "Appreciation Day! Alex, Alex, Alex! Yay"  
+ "Alex!!! That's me! Alex is the best date ever!");
```

- Με μεταβλητές

```
String x = "Sam";  
System.out.println("My name is "+x);  
System.out.println(x+" is so cool!");  
System.out.println("Hooray "+x+"!");  
System.out.println("Please enjoy "+x+"  
Appreciation "  
+ "Day! My name is "+x+" . I know how excited "  
+ "everyone is to start appreciating "+x+" on "+x+"  
+ "Appreciation Day! "+x+" , "+x+" , "+x+"! Yay"+ x  
+ "!!! That's me! "+x+" is the best date ever!");
```

Τι αλλαγές πρέπει να κάνουμε στον παραπάνω κώδικα ώστε να εκτυπώνει το αποτέλεσμα και για την *Rita*, και στις 2 περιπτώσεις.

## Άσκηση 3.1 – Μεταβλητές

- Ανοίξτε και τρέξτε το “Variables02”.
- Ακολουθήστε τα βήματα της άσκησης /\* στα σχόλια\*/
- Εκτελέστε το πρόγραμμα μεταξύ κάθε βήματος και παρατηρήστε την έξοδο.
- Το πρόγραμμά σας θα πρέπει να παράγει τις ακόλουθες εξόδους:
  - -Μετά το βήμα 1)

```
puppy  
puppy
```
  - -Μετά το βήμα 2)

```
kitty  
kitty
```
  - -Μετά το βήμα 3)

```
kitty  
bunny
```

# Τύποι Δεδομένων

- Όλοι οι τύποι..

	Τύπος τιμών	Μήκος σε bytes	Πεδίο τιμών	Αρχική τιμή
<b>byte</b>	Ακέραιος (Integer)	1	-128 . . . 127	0
<b>short</b>		2	-32768 . . . 32767	0
<b>int</b>		4	$-2^{31} \dots 2^{31}-1$ / $0 \dots 2^{32}-1$ (Java SE8)	0
<b>long</b>		8	$-2^{63} \dots 2^{63}-1$ / $0 \dots 2^{64}-1$ (Java SE8)	0
<b>float</b>	Κινητής υποδιαστολής (floating point number)	4	single-precision 32-bit IEEE 754 floating point	0
<b>double</b>		8	double-precision 32-bit IEEE 754 floating point	0
<b>char</b>	Χαρακτήρας (Single character)	2	<code>\u0000</code> (0) . . . <code>\uffff</code> (65.535)	0
<b>boolean</b>	Λογική τιμή	?	false - true	False



# Τύποι Δεδομένων

- *Συνήθεις τύποι..*

Type	Keyword	Example Values
Boolean	<code>boolean</code>	true, false
Integer	<code>int</code>	1, -10, 20000
Double	<code>double</code>	1.0, -10.0005, 3.141
Char	<code>char</code>	'a', '&'
String	<code>String</code>	"Alex", "I ate too much dinner."

# Δήλωση μεταβλητής

- Πρέπει να δηλώσετε τον τύπο της μεταβλητής χρησιμοποιώντας λέξεις-κλειδιά.
- Αφού δηλώσετε μια μεταβλητή ...
  - Αυτή η μεταβλητή υπάρχει..
  - Δεν χρειάζεται να τη δηλώσετε ξανά.
- Μπορείτε να δηλώσετε και ορίσετε μια μεταβλητή σε μία γραμμή:
- ή να δηλώσετε μια μεταβλητή σε μία γραμμή και να την ορίσετε σε μία άλλη

*type*   *name*

```
boolean bool;  
int x;  
double y;  
String z;
```

*type*   *name*   *value*

```
boolean bool = true;
```

*type*   *name*

```
boolean bool;  
int x;  
double y;  
...  
bool = true;
```

*name*   *value*

# Δήλωση μεταβλητής – Συνήθη λάθη

1. Ακατάλληλες τιμές σε μία μεταβλητή

```
int intVar1 = true;  
int z = "Puppies!";
```

2. Μη δήλωση μεταβλητής

```
intVar3 = 3; //intVar3 ?
```

3. Λάθος γραφή μεταβλητής

```
double doubleVar2;  
doublevAr2 = 2.1;
```

4. Διπλή δήλωση της ίδια μεταβλητής

```
double doubleVar3;  
double doubleVar3 = 3.1;
```

5. Χρήση μεταβλητής χωρίς προηγούμενη εκχώρηση τιμής

```
double doubleVar4;  
System.out.println (doubleVar4);
```

$y = -2x + 5$

$x = \text{"Puppies!"};$

$y = -2 (\text{"Puppies"}) + 5$

$y = ???$

Παράλειψη αρχικοποίησης

→ Είναι σωστό το παρακάτω;

```
double doubleVar1, doubleVar2, doubleVar3 = 3.1;
```

# Ονομασία Μεταβλητών

```
int dsfdsfspoop = 20;           // «Τι θέλει να πει ο ποιητής;»  
int x = 20;                     /* συνήθως μόνο για testing ή δείκτης επανάληψης για μικρούς  
                                βρόγχους (loops) */  
int x = 20;                     // Ποιο x?  
double x = 22.0;  
System.out.println(x);  
boolean 1337Hacker = true;      /* οι μεταβλητές δεν μπορούν να ξεκινούν από αριθμό */  
int continue = 20;              /* Οι δεσμευμένες λέξεις δεν μπορούν να χρησιμοποιηθούν για  
                                όνομα μεταβλητής - για αυτό και έχουν χρώμα μπλε στο  
                                NetBeans */
```

## Καλά Ονόματα!

- myVariable ✓
- int studentAge = 20; ✓
- String myCatchPhrase = "Enjoy Alex Appreciation Day!"; ✓

# Μεταβλητές και Σταθερές

- Μία μεταβλητή
  - Μπορεί να παίρνει διαφορετικές τιμές κατά τη διάρκεια του προγράμματος.
  - Μπορεί να δηλωθεί σε κάποιο σημείο και να πάρει (και να ξαναπάρει) τιμή αλλού.  
  
`int numberOfHoursInADay= 67;`  
`numberOfHoursInADay= 24;`
- Μία σταθερά
  - Παίρνει μία τιμή (μπορεί και μετά τη δήλωσή της).
  - Δηλώνεται σε ένα σημείο του κώδικα και δεν μπορεί να ξαναδηλωθεί.
  - Η σύμβαση λέει ότι πρέπει να χρησιμοποιούνται κεφαλαία γράμματα και το σύμβολο `_` για την ονομασία της.  
  
`final int NUMBER_OF_HOURS_IN_A_DAY = 24;`

# Άσκηση 3.2 – Διόρθωση Μεταβλητών

- Ανοίξτε και τρέξτε το “Variables03”.
- Υπάρχουν 6 λάθη σε αυτό το πρόγραμμα.
- Μπορείτε να διορθώσετε αυτά τα λάθη έτσι ώστε το πρόγραμμα να παράγει την ακόλουθη έξοδο;

```
bool = true  
intVar1 = 1  
intVar2 = 2  
intVar3 = 3  
doubleVar1 = 1.1  
doubleVar2 = 2.1  
doubleVar3 = 3.1  
doubleVar4 = 4.1  
stringVar1 = 11  
stringVar2 = 22
```

# Αριθμητικά δεδομένα

# Ακέραιοι

**Πότε τους χρησιμοποιούμε;**

	Τύπος	Μήκος	Πιθανές τιμές	Ελάχιστη τιμή	Μέγιστη τιμή
<b>Ποτέ</b> →	Byte	8 bits	$2^8$ , ή... 256	$-2^7$ , ή... -128	$2^7-1$ , ή... 127
<b>Ποτέ</b> →	short	16 bits	$2^{16}$ , ή... 65.535	$-2^{15}$ , ή... -32.768	$2^{15}-1$ , ή... 32.767
<b>Συχνά</b> →	int	32 bits	$2^{32}$ , ή... 4.294.967.296	$-2^{31}$ , ή... -2.147.483.648	$2^{31}-1$ , ή... 2.147.483.647
<b>Σπάνια</b> →	long	64 bits	$2^{64}$ , ή... 18.446.744.073.709.551.616	$-2^{63}$ , ή... -9.223.372.036. 854.775.808L	$2^{63}-1$ , ή... 9.223.372.036. 854.775.807L



# Μαθηματικοί τελεστές

Πράξη	Τελεστής	Παράδειγμα	Σχόλια
Πρόσθεση	+	<code>sum = num1 + num2;</code>	If num1 is 10 and num2 is 2, sum is 12
Αφαίρεση	−	<code>diff = num1 − num2;</code>	If num1 is 10 and num2 is 2, diff is 8
Πολλαπλασιασμός	*	<code>prod = num1 * num2;</code>	If num1 is 10 and num2 is 2, prod is 20
Διαίρεση	/	<code>quot = num1 / num2;</code>	If num1 is 31 and num2 is 6, quot is 5 → Το υπόλοιπο απορρίπτεται → Η διαίρεση με 0 επιστρέφει σφάλμα
Υπόλοιπο	%	<code>num1 = 31; num2 = 6; mod = num1 % num2; mod is 1</code>	Το υπόλοιπο του πρώτου αριθμού διαιρούμενο με τον δεύτερο αριθμό

# Εκχώρηση αριθμών

- Βρείτε το x?

Εντολές	Τιμές του x	Οθόνη
<pre>int x = 20; x = 25; x = 5 + 3; System.out.println(x);</pre>	<pre>20 25 8</pre>	<pre>8</pre>
<pre>int x = 20; x = 25; x = 5 + 3; x = x + 1; x += 1; x++; System.out.println(x);</pre>	<pre>20 25 8 9 10 11</pre>	<pre>11</pre>

συντόμευση += για αύξηση κατά γ

συντόμευση ++ για αύξηση κατά 1

Εντολές	Τιμές του x, γ	Οθόνη
<pre>int γ = 20; int x = γ; γ++; System.out.println(x); System.out.println(γ);</pre>	<pre>x      γ 20     20       21</pre>	<pre>20 21</pre>

Η αλλαγή του γ δεν επηρεάζει το x

# Τελεστές και εκχωρήσεις

Πράξη	Τελεστής	Παράδειγμα Εάν, int a = 6, b = 2;	Αποτέλεσμα
Εκχώρηση	=	a = b	a = 2
Πρόσθεση & εκχώρηση	+=	a += b	a = 8
Αφαίρεση & εκχώρηση	-=	a -= b	a = 4
Πολλαπλασιασμός & εκχώρηση	*=	a *= b	a = 12
Διαίρεση & εκχώρηση	/=	a /= b	a = 3
Αύξηση κατά 1	++	a++; (=> a= a + 1;)	a = 7
Προ-αύξηση (++ μεταβλητή)		b= ++a;	a=7 b=7
Μετα-αύξηση (μεταβλητή++)		b= a++;	b=6 a=7
Μείωση κατά 1	--	a--; (=> a= a - 1;)	a = 5
Προ-μείωση (-- μεταβλητή)		b= --a;	a=5 b=5
Μετα-μείωση (μεταβλητή--)		b= a--;	b=6 a=5

# Τι εμφανίζει το παρακάτω πρόγραμμα;

```
1.  int count=15;  
2.  int a, b, c, d;  
3.  a = count++;  
4.  b = count;  
5.  c = ++count;  
6.  d = count;  
7.  System.out.println(a + ", " + b + ", " + c + ", " + d);
```

# Τι εμφανίζει το παρακάτω πρόγραμμα;

```
1.  int count=15;  
2.  int a, b, c, d;  
3.  a = count++;  
4.  b = count;  
5.  c = ++count;  
6.  d = count;  
7.  System.out.println(a + ", " + b + ", " + c + ", " + d);
```

count	a	b	c	d
15				
16	15			
		16		
17			17	
				17

Output

15, 16, 17, 17

# Άσκηση 3.3 - Chickens Project

1. Ανοίξτε και επεξεργαστείτε το “Chickens01”.
2. Διαβάστε την ιστορία και υπολογίστε /εμφανίστε το `totalEggs` που συλλέγονται μεταξύ της Δευτέρας και της Τετάρτης:
  - i. Τα κοτόπουλα του αγρότη Brown πάντα παράγουν **`eggsPerChicken`** ακριβώς στο μεσημέρι, τα οποία συλλέγει εκείνη την ημέρα.
  - ii. Τη Δευτέρα, ο Brown έχει **`chickenCount`** κοτόπουλα.
  - iii. Την Τρίτη το πρωί, ο Brown κερδίζει 1 κοτόπουλο.
  - iv. Την Τετάρτη το πρωί, μία αλεπού τρώει τα μισά κοτόπουλα!
  - v. Πόσα αυγά θα συλλέξει ο αγρότης Brown αν ξεκινήσει με ...
    - a) `eggsPerChicken = 5`, `chickenCount = 3`
    - b) `eggsPerChicken = 4`, `chickenCount = 8`

# Άσκηση 3.3 - Chickens Project

1. Ανοίξτε και επεξεργαστείτε το “Chickens01”.
2. Διαβάστε την ιστορία και υπολογίστε /εμφανίστε το `totalEggs` που συλλέγονται μεταξύ της Δευτέρας και της Τετάρτης:
  - i. Τα κοτόπουλα του αγρότη Brown πάντα παράγουν **eggsPerChicken** ακριβώς στο μεσημέρι, τα οποία συλλέγει εκείνη την ημέρα.
  - ii. Τη Δευτέρα, ο Brown έχει **chickenCount** κοτόπουλα.
  - iii. Την Τρίτη το πρωί, ο Brown κερδίζει 1 κοτόπουλο.
  - iv. Την Τετάρτη το πρωί, μία αλεπού τρώει τα μισά κοτόπουλα!
  - v. Πόσα αυγά θα συλλέξει ο αγρότης Brown αν ξεκινήσει με ...
    - a) `eggsPerChicken = 5`, `chickenCount = 3`
    - b) `eggsPerChicken = 4`, `chickenCount = 8`

45	σενάριο (a)
84	σενάριο (b)

# Πραγματικοί

Τύπος	Μήκος	Παράδειγμα	Πότε χρησιμοποιούνται;
float	32 bits	public float pi = 3.141592F;	Σχεδόν ποτέ
double	64 bits	public double pi = 3.141592;	Συχνά

- Προσέξτε τα παρακάτω:

(α)

```
int x = 9/2;  
System.out.println(x); //prints 4
```

(β)

```
double x = 9/2;  
System.out.println(x); //prints 4.0
```

- Γιατί συμβαίνει αυτό;



# Πραγματικοί

Τύπος	Μήκος	Παράδειγμα	Πότε χρησιμοποιούνται;
float	32 bits	public float pi = 3.141592F;	Σχεδόν ποτέ
double	64 bits	public double pi = 3.141592;	Συχνά

- Προσέξτε τα παρακάτω:

(α)

```
int x = 9/2;  
System.out.println(x); //prints 4
```

(β)

```
double x = 9/2;  
System.out.println(x); //prints 4.0
```

- Γιατί συμβαίνει αυτό;
  - Στο (α) η Java κόβει το .5
  - Στο (β) μετατρέπει τον x σε double.

Η διαίρεση ("/") λειτουργεί διαφορετικά στους ακέραιους και στους πραγματικούς (οι ακέραιοι αριθμοί δεν **στρογγυλοποιούνται**, αλλά **περικόπτονται**)

# Πραγματικοί

Τύπος	Μήκος	Παράδειγμα	Πότε χρησιμοποιούνται;
float	32 bits	public float pi = 3.141592F;	Σχεδόν ποτέ
double	64 bits	public double pi = 3.141592;	Συχνά

- Προσέξτε τα παρακάτω:

(α)

```
int x = 9/2;  
System.out.println(x); //prints 4
```

(β)

```
double x = 9/2;  
System.out.println(x); //prints 4.0
```

- Γιατί συμβαίνει αυτό;
  - Στο (α) η Java κόβει το .5
  - Στο (β) μετατρέπει τον x σε double.

Η διαίρεση ("/") λειτουργεί διαφορετικά στους ακέραιους και στους πραγματικούς (οι ακέραιοι αριθμοί δεν **στρογγυλοποιούνται**, αλλά **περικόπτονται**)

- Λύση:

- Χρήση double στην πράξη

(γ)

```
double x = 9/2.0;  
System.out.println(x); //prints 4.5
```

## Άσκηση 3.4 - Chickens Project

1. Ανοίξτε και επεξεργαστείτε το “Chickens02”.
2. Διαβάστε την ιστορία και υπολογίστε /εμφανίστε τις ζητούμενες τιμές:
  - i. Τη Δευτέρα, ο αγρότης Fred συλλέγει 100 αυγά
  - ii. Την Τρίτη, ο Fred συλλέγει 121 αυγά
  - iii. Την Τετάρτη, ο Fred συλλέγει 117 αυγά
  - iv. Ποιος είναι μέσος ημερήσιος αριθμός συλλογής αυγών **dailyAverage** και ποιος ο μηνιαίος **monthlyAverage**;
  - v. Εάν ένα αυγό μπορεί να πωληθεί με κέρδος 0,18€, ποιο είναι το συνολικό μηνιαίο κέρδος του αγρότη Fred για όλα τα αυγά;

## Άσκηση 3.4 - Chickens Project

Η έξοδος του προγράμματος θα πρέπει να είναι της μορφής:

```
Output  
Daily Average: 112.66666666666667  
Monthly Average: 3380.0  
Profit: 608.4€
```

# Προτεραιότητας Εκτέλεσης Πράξεων

Οι πράξεις εκτελούνται κατά προτεραιότητα με την ακόλουθη σειρά.

1. Παρενθέσεις '()'
2. Αύξηση/μείωση '++' ή '--'
3. Πολλαπλασιασμός/διαίρεση '\*' ή '/' (από αριστερά προς τα δεξιά)
4. Πρόσθεση/αφαίρεση '+' ή '-' (από αριστερά προς τα δεξιά)

Τελεστές της ίδιας προτεραιότητας εκτελούνται από αριστερά προς τα δεξιά.

# Προτεραιότητας Εκτέλεσης Πράξεων: *Παραδείγματα*

```
int x = 10 +20 +30 / 3;
```

```
int x = 10 +20 +(30 / 3);
```

```
int x = (10 +20 +30) / 3;
```

```
double x = 3 / 2 + 1;
```

```
double y = 3 / (2+1);
```

```
int x = (((25 -5) * 4) / (2 -10)) + 4;
```

```
int x = 25 -5 * 4 / 2 -10 + 4;
```

# Προτεραιότητας Εκτέλεσης Πράξεων: *Παραδείγματα*

<code>int x = 10 +20 +30 / 3;</code>	<code>x=40</code>
<code>int x = 10 +20 +(30 / 3);</code>	<code>x=40</code>
<code>int x = (10 +20 +30) / 3;</code>	<code>x=20</code>
<code>double x = 3 / 2 + 1;</code>	<code>x = 2.0</code>
<code>double y = 3 / (2+1);</code>	<code>y = 1.0</code>
<code>int x = (((25 -5) * 4) / (2 -10)) + 4;</code>	<code>x = -6</code>
<code>int x = 25 -5 * 4 / 2 -10 + 4;</code>	<code>x = 9</code>

# Χαρακτήρες



# Τύπος δεδομένων

Τύπος	Μήκος	Πιθανές τιμές
char	16 bits	'a..z', 'A..Z', ...

- π.χ. `char shirtSize = 'M';` // μονά εισαγωγικά για απλούς χαρακτήρες
- Μπορείτε να ενώσετε χαρακτήρες για να δημιουργήσετε προτάσεις.

```
char letter1 = 'H';  
char letter2 = 'e';  
char letter3 = 'l';  
char letter4 = 'l';  
char letter5 = 'o';  
System.out.println(letter1 + letter2 + letter3 + letter4 + letter5);  
// Οι μεγάλες προτάσεις θα ήταν λίγο επώδυνο να κωδικοποιηθούν
```

# Τύπος δεδομένων

Τύπος	Μήκος	Πιθανές τιμές
char	16 bits	'a..z', 'A..Z', ...

- π.χ. `char shirtSize = 'M';` // μονά εισαγωγικά για απλούς χαρακτήρες
- Μπορείτε να ενώσετε χαρακτήρες για να δημιουργήσετε προτάσεις.

```
char letter1 = 'H';  
char letter2 = 'e';  
char letter3 = 'l';  
char letter4 = 'l';  
char letter5 = 'o';  
System.out.println(letter1 + letter2 + letter3 + letter4 + letter5);  
// Οι μεγάλες προτάσεις θα ήταν λίγο επώδυνο να κωδικοποιηθούν
```

- Όμως υπάρχει και καλύτερος τρόπος

```
String greeting = "Hello World!";  
//Παρατηρήστε τα διπλά εισαγωγικά  
System.out.println(greeting);
```

# Χαρακτήρες (chars) Vs Συμβολοσειρές (strings)

- Οι χαρακτήρες (chars):
  - είναι για ένα μόνο χαρακτήρα
  - χρησιμοποιούν μονά εισαγωγικά
  - δεν μπορούν να χειριστούν πολλούς χαρακτήρες
  - `char` είναι δεσμευμένη λέξη της Java (γίνεται μπλε)
    - `char shirt1Size = 'S';` ✓
    - `char shirt2Size = 'M';` ✓
    - `char shirt3Size = 'L';` ✓
    - `char shirt4Size = 'XL';` ✗
    - `char shirt5Size = "XXL";` ✗
- Οι συμβολοσειρές (strings)
  - είναι για πολλούς χαρακτήρες
  - χρησιμοποιούν διπλά εισαγωγικά
  - είναι αντικείμενα ειδικού σκοπού, όχι δεσμευμένη λέξη (έχει κεφαλαίο το πρώτο γράμμα)
  - όπως όλα τα αντικείμενα έχουν **ιδιότητες** (String properties) & **συμπεριφορές** (String behaviors)
    - `String shirt6Size = "XXXL";` ✓

# Παραδείγματα

- Δήλωση και αρχικοποίηση μία μεταβλητής

```
int intVar = 300;
```

```
String stringVar = "Three Hundred";
```

- Δήλωση και αρχικοποίηση πολλών μεταβλητών

```
int x, y, z;
```

```
String xString, yString, zString;
```

- Αρχικοποίηση δηλωμένων μεταβλητών

```
xString = "One";
```

Δηλώστε και χρησιμοποιήστε τα Strings όπως τους άλλους τύπους δεδομένων τη Java

# Συνδυασμός Strings

- `String combinedLiterals = "I want to"+ " buy a shirt.";`  
`System.out.println(combinedLiterals);` → I want to buy a shirt.
- `String var1 = "This shirt has";`  
`String var2 = " too many buttons.";`  
`String combinedVariables = var1 + var2;`  
`System.out.println(combinedVariables);` → This shirt has too many buttons.
- `String greet1 = "Hello";`  
`String greet2 = "World";`  
`String message1 = greet1 +" "+greet2 +"!";`  
`String message2 = greet1 +" "+greet2 +" "+2016 +"!";`  
`System.out.println(message2);` → Hello World 2016!  
`System.out.println(greet1 +" "+greet2 +" "+2016 +"!");` → Hello World 2016!

# Άσκηση 3.5 - Shopping Cart 1

1. Ανοίξτε και επεξεργαστείτε το “ShoppingCart01”.
2. Δηλώστε και αρχικοποιήστε τη String μεταβλητή **custName**.
3. Δηλώστε και αρχικοποιήστε τη String μεταβλητή **itemDesc**.
4. Δηλώστε τη String μεταβλητή **message**.
5. Εκχωρήστε στο **message** μία τιμή που να περιέχει **custName**, **itemDesc** και μία φράση όπου το αποτέλεσμα να αντιστοιχεί σε μια πλήρη πρόταση, π.χ. “**Alex wants to buy a shirt**”
6. Εκτυπώστε το μήνυμα.

## Hints!

- ShoppingCart properties:
  - Κάτοχος
  - Προϊόντα που περιέχει
  - Περιγραφή
- ShoppingCart behaviors:
  - Εκτύπωση μηνύματος

# Strings & Numbers

- Οι συμβολοσειρές (Strings) μπορούν να περιέχουν αριθμούς:
  - `String totalPrice = "Total: $" + 3;`
  - `System.out.println(totalPrice);` `//Total: $3`
- Προσοχή όμως:
  - `String totalPrice = "Total: $" + 3 + 2 + 1;`
  - `System.out.println(totalPrice);` `//Total: $321`
- Καλύτερα παρενθέσεις για αριθμούς:
  - `String totalPrice = "Total: $" + (3 + 2 + 1);`
  - `System.out.println(totalPrice);` `//Total: $6`

## *The Underscore “\_”*

- Με την Java SE7, μπορείτε να χρησιμοποιείτε **underscore “\_”** για να κάνετε μεγάλους αριθμούς πιο ευανάγνωστους:
  - `int x = 123456789;` → `int x = 123_456_789;`



## Άσκηση 3.6 - Shopping Cart 2

1. Ανοίξτε και επεξεργαστείτε το “ShoppingCart02”.
2. Δηλώστε και αρχικοποιήστε τις αριθμητικές μεταβλητές:
  - **price(double)**
  - **tax(double)**
  - **quantity(int)**
3. Δηλώστε την αριθμητική μεταβλητή: **totalPrice(double)**:
4. Προσθέστε μια τιμή, υπολογιζόμενη από τη **price**, το **tax** και την **quantity**.
5. Αλλάξτε το μήνυμα έτσι ώστε να περιέχει και την ποσότητα (π.χ. “Ο Alex θέλει να αγοράσει 2 πουκάμισα.”)
6. Εκτυπώστε ένα άλλο μήνυμα για το συνολικό κόστος.

### Hints!

- ShoppingCart properties:
  - Κάτοχος
  - Περιγραφή
  - Προϊόντα που περιέχει
  - Τιμή προϊόντος
  - Ποσότητα προϊόντος
  - ΦΠΑ
  - Συνολικό κόστος
- ShoppingCart behaviors:
  - Εκτύπωση μηνύματος
  - Εκτύπωση συνόλου

### Output

Ο Alex θέλει να αγοράσει 2 πουκάμισα.  
Συνολικό κόστος με φόρο: 28.52 €

# Ακολουθίες διαφυγής

- Ένας χαρακτήρας μετά από το `'\'` ονομάζεται ειδικός χαρακτήρας και δημιουργεί μία ακολουθία διαφυγής (escape sequence) με ιδιαίτερη σημασία για τον μεταγλωττιστή.
- Παραδείγματα

Ειδικός χαρακτήρας	Περιγραφή
<code>\t</code>	new tab
<code>\b</code>	backspace
<code>\n</code>	new line
<code>\r</code>	Αλλαγή γραμμής
<code>\f</code>	Form feed
<code>\'</code>	Απλό εισαγωγικό
<code>\"</code>	Διπλά εισαγωγικά
<code>\\</code>	Backslash

– `System.out.println("This is the first line. \n" + "This is the second line.");`

Output:  
This is the first line.  
This is the second line.

– `System.out.println("The cat said \"Meow!\" to me.");`

Output:  
The cat said "Meow!" to me.

# Εκτύπωση αλλαγής γραμμής println

- Όταν γράφουμε κείμενο σε νέα γραμμή, δεν σημαίνει και ότι θα εκτυπωθεί σε μια νέα γραμμή:

```
System.out.println("This is the first line." +  
"This is NOT the second line.");
```

Output  
This is the first line. This is NOT the second line.

- Αυτό γίνεται με 2 τρόπους:

1. Escape sequences:

```
System.out.println("This is the first line. \n" +  
"This is the second line.");
```

2. Println:

```
System.out.println("This is the first line.");  
System.out.println("This is the second line.");
```

Output  
This is the first line.  
This is the second line.

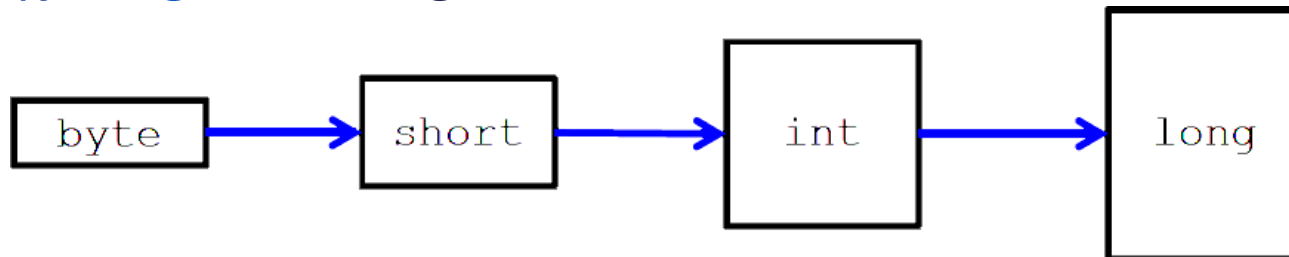
# Ιεραρχία τύπων δεδομένων και implicit typecast

- Οι τύποι δεδομένων στη Java είναι ιεραρχημένοι.
- Το νόημα είναι να μπορεί ένας αριθμός που ανήκει σε τύπο πιο ψηλά στην ιεραρχία, να συγκρατήσει την ακρίβεια του χαμηλότερου στη μετατροπή (άρα όσο πιο ψηλά, τόσο μεγαλύτερη ακρίβεια ενός τύπου)
- Η ιεραρχία είναι:
  1. `double`
  2. `float`
  3. `long`
  4. `int`
  5. `short`
  6. `byte`
- Implicit typecast: Αναβαθμίζει τον τύπο, σύμφωνα με την ιεραρχία

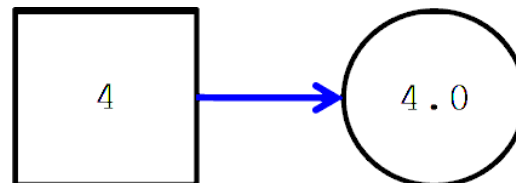
# Προβιβασμοί (Promotion)/ implicit typecast

- Αυτόματοι

- Εκχώρηση μικρότερου τύπου σε μεγαλύτερο, π.χ. `long intToLong = 6;`



- Εκχώρηση ακεραίου σε αριθμό κινητής υποδιαστολής, π.χ. `double intToDouble = 4;`



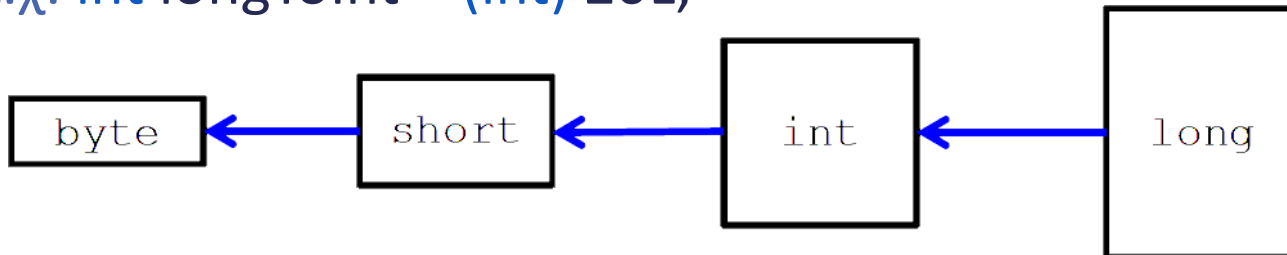
# Προβιβασμοί (Promotion): Προβλήματα

Πιθανό πρόβλημα	Λύση
<pre>int num1 = 55555; int num2 = 66666; long num3; num3 = num1 * num2;</pre>	<pre>int num1 = 55555; long num2 = 66666; long num3; num3 = num1 * num2;</pre>
Λάθος → //num3 is -591337666	//num3 is 3703629630 ← Σωστό

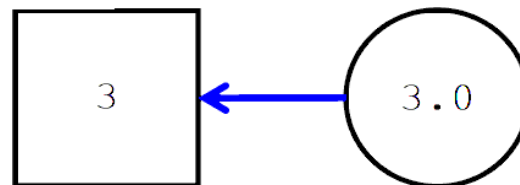
Πιθανό πρόβλημα	Λύση
<pre>int num1 = 7; int num2 = 2; double num3; num3 = num1 / num2;</pre>	<pre>int num1 = 7; double num2 = 2; double num3; num3 = num1 / num2;</pre>
Λάθος → //num3 is 3.0	//num3 is 3.5 ← Σωστό

# Μετατροπές (Casting)

- Πότε;
  - Εκχώρηση μεγαλύτερου τύπου σε μικρότερο,  
π.χ. `int longToInt = (int) 20L;`



- Εκχώρηση αριθμού κινητής υποδιαστολής σε ακέραιο  
π.χ. `short doubleToShort = (short) 3.0;`



# Μετατροπές (Casting): Προβλήματα

Πιθανό πρόβλημα		Λύση	
<pre>int myInt; long myLong = 123987654321L; myInt = (int)myLong;</pre>		<pre>int myInt; long myLong = 99L; myInt = (int)myLong;</pre>	
Λάθος →	<pre>// Number is "chopped" // myInt is -566397263</pre>	<pre>// myInt is 99 //no data loss, only zeroes.</pre>	← Σωστό



# Παράδειγμα

Ποιά από τα παρακάτω επιτρέπονται;

- **int x = 3.5;**
- **float x = 3;**
- **long i = 3;**
- **byte x = 155;**
- **double d = 3.14159F;**

# Παραδείγματα

Ποιά από τα παρακάτω επιτρέπονται;

- **int x = 3.5;**

Δεν επιτρέπεται γιατί το 3.5 δεν χωράει στο int

- **float x = 3;**

Επιτρέπεται γιατί το 3 είναι τύπου int που χωράει στο float

- **long i = 3;**

Επιτρέπεται γιατί το 3 είναι τύπου int που χωράει στο long

- **byte x = 155;**

Δεν επιτρέπεται γιατί το 155 είναι πολύ μεγάλο για να χωρέσει ένα byte ( $\leq 127$ )

- **double d = 3.14159F;**

Επιτρέπεται γιατί το 3.14159F είναι τύπου float, που χωράει στο double

## Άσκηση 3.7 - Casting

- Ανοίξτε και επεξεργαστείτε το “Casting01”.
- Δηλώστε και αρχικοποιήστε ένα **byte** με τιμή 128
  - Παρατηρήστε την “διαμαρτυρία” του Compiler.
  - Βάλτε στα σχόλια τη λάθος γραμμή.
- Δηλώστε και αρχικοποιήστε ένα **short** με με τιμή 128:
  - Δημιουργήστε μια εντολή εκτύπωσης που μετατρέπει αυτό το **short** σε ένα **byte**.
- Δηλώστε και αρχικοποιήστε ένα **byte** με τιμή 127.
  - Προσθέστε 1 σε αυτήν τη μεταβλητή και εκτυπώστε την.
  - Προσθέστε ξανά 1 σε αυτήν τη μεταβλητή και εκτυπώστε ξανά.

## Άσκηση 3.8 - Casting

- Ανοίξτε και επεξεργαστείτε το “Casting02”.
- Υπάρχουν πολλά σφάλματα σε αυτό το πρόγραμμα. Θα πρέπει να είστε σε θέση να διορθώσετε αυτά τα σφάλματα χρησιμοποιώντας τις γνώσεις σας για...
  - Τύπους δεδομένων
  - Προβιβασμούς
  - Μετατροπές

# Μετατροπή συμβολοσειρών σε αριθμητικά δεδομένα

- Η μετατροπή κειμένου σε αριθμό είναι μορφή ανάλυσης (parsing).

- String to an int:

```
int intVar1 = Integer.parseInt("100");
```

- String to a double:

```
double doubleVar2 = Double.parseDouble("2.72");
```

## Άσκηση 3.9 - Parsing

- Ανοίξτε και επεξεργαστείτε το “Parsing01” project.
- Δηλώστε και αρχικοποιήστε 3 **Strings** με τα ακόλουθα δεδομένα

String Variable	Description	Example Values
shirtPrice	Text to be converted to an int:	"15"
taxRate	Text to be converted to a double:	"0.05"
gibberish	Gibberish	"887ds7nds87dsfs"

- Μετατρέψτε και πολλαπλασιάστε το **shirtPrice** \* **taxRate** για να βρείτε τον φόρο.
  - Εκτυπώστε αυτήν την τιμή.
- Προσπαθήστε να μετατρέψετε το **taxRate** ως **int**.
  - Τι λέει το μήνυμα σφάλματος;
- Προσπαθήστε να μετατρέψετε το **gibberish** ως **int**.
  - Τι λέει το μήνυμα σφάλματος;

# Εισαγωγή δεδομένων από χρήστη

- Υπάρχουν πολλοί τρόποι για να πάρουμε δεδομένα από χρήστες (user input):
  - Κουμπιά (φυσικά ή εικονικά)
  - Αναγνώριση φωνής
  - Παράθυρα διαλόγου
- Η Java προσφέρει διάφορους τρόπους όπως:
  - JavaFX (διάδοχος του Swing)
  - Scanner

# Εισαγωγή δεδομένων με αντικείμενα Scanner

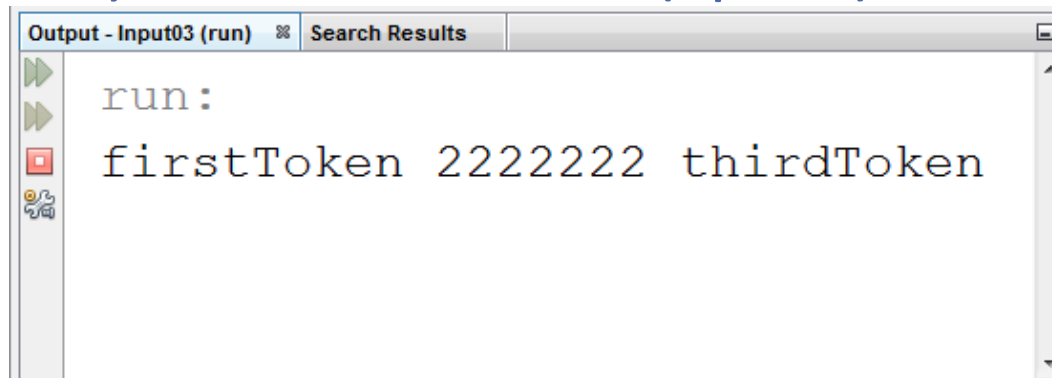
- Ένα αντικείμενο Scanner ανοίγει μια ροή για τη συλλογή δεδομένων εισόδου:
  - System.in διαβάζει για να ροές δεδομένων εισόδου από το πληκτρολόγιο.
  - Το Scanner μπορεί επίσης να χρησιμοποιηθεί χωρίς IDE.
- Καλό είναι να κλείνετε τη ροή εισόδου όταν τελειώσετε.

```
public static void main(String[] args) {  
    Scanner sc = new Scanner(System.in);  
    System.out.println(sc.nextInt());  
    System.out.println(sc.next());  
    sc.close();  
}
```



# Ανάγνωση εισόδου με Scanner

- Τα αντικείμενα Scanner αναζητούν αναζήτηση για λεκτικές μονάδες (tokens).
- Τα tokens διαχωρίζονται με έναν συγκεκριμένο οριοθέτη (delimiter).
  - Συνήθως είναι ένα “κενό” (Space).



```
Output - Input03 (run)  Search Results
run:
firstToken 2222222 thirdToken
```

# Μέθοδοι της κλάσης Scanner

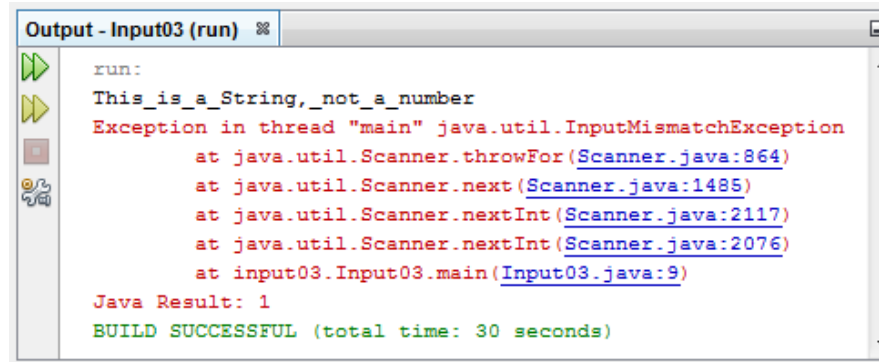
- `nextInt ()`: διαβάζει το επόμενο token ως `int`.
- `nextDouble ()`: διαβάζει το επόμενο token ως πραγματικό.
- `next()`: διαβάζει το επόμενο token ως `String`
- `nextLine()`: διαβάζει την είσοδο ως `String` μέχρι ο χρήστης να αλλάξει γραμμή

```
public static void main(String[] args) {  
    Scanner sc = new Scanner(System.in);  
    int x = sc.nextInt();  
    double y = sc.nextDouble();  
    String z = sc.next();  
    String input = sc.nextLine();  
    sc.close();  
}
```

## Άσκηση 3.12 – Input 3

- Ανοίξτε το Input03.java
- Δημιουργήστε ένα Scanner object για να:
  - Βρίσκει και εκτυπώνει το άθροισμα τριών ακέραιων αριθμών που εισάγει ο χρήστης.
    - Δοκιμάστε να εισάγετε λιγότερα από 3 tokens.
    - Δοκιμάστε να εισάγετε 1 token που δεν μπορεί να αναλυθεί ως int.

# Πιθανά λάθη: InputMismatch Exception

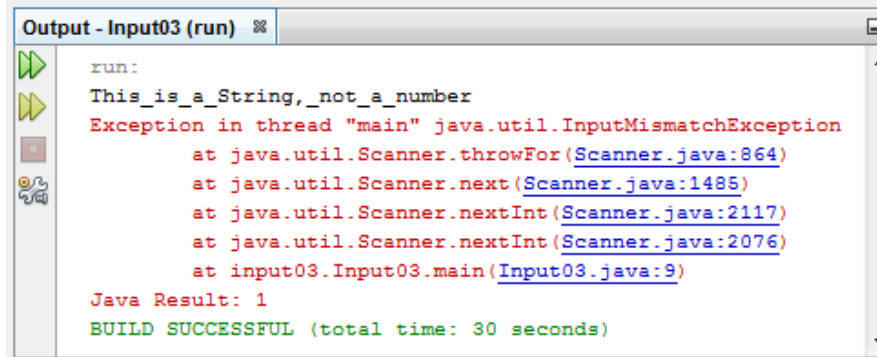


```
Output - Input03 (run) %
run:
This_is_a_String,_not_a_number
Exception in thread "main" java.util.InputMismatchException
    at java.util.Scanner.throwFor(Scanner.java:864)
    at java.util.Scanner.next(Scanner.java:1485)
    at java.util.Scanner.nextInt(Scanner.java:2117)
    at java.util.Scanner.nextInt(Scanner.java:2076)
    at input03.Input03.main(Input03.java:9)
Java Result: 1
BUILD SUCCESSFUL (total time: 30 seconds)
```

- Συμβαίνει επειδή τα δεδομένα εισόδου δεν μπορούν να αναλυθούν (parse) όπως ήταν αναμενόμενο, π.χ.

```
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.println(sc.nextInt());
    sc.close();
}
```

# Πιθανά λάθη: IllegalStateException



```
Output - Input03 (run)
run:
This_is_a_String,_not_a_number
Exception in thread "main" java.util.InputMismatchException
    at java.util.Scanner.throwFor(Scanner.java:864)
    at java.util.Scanner.next(Scanner.java:1485)
    at java.util.Scanner.nextInt(Scanner.java:2117)
    at java.util.Scanner.nextInt(Scanner.java:2076)
    at input03.Input03.main(Input03.java:9)
Java Result: 1
BUILD SUCCESSFUL (total time: 30 seconds)
```

- Συμβαίνει επειδή τα δεδομένα εισόδου προσπελάστηκαν αφού είχε κλείσει η ροή εισόδου, π.χ.

```
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    sc.close();
    System.out.println(sc.nextInt());
}
```

# Σε αυτήν την ενότητα μιλήσαμε για:

1. Μεταβλητές (τύποι, δήλωση, ονομασία, λάθη, σταθερές)
2. Αριθμητικά δεδομένα (ακέραιοι, πραγματικοί)
3. Χαρακτήρες-Συμβολοσειρές-Ακολουθίες διαφυγής
4. Μετατροπές (explicit/implicit typecast)
5. Δεδομένα Εισόδου (input data)



**Δρ. Μιχάλης Φειδάκης**  
Ε.ΔΙ.Π.  
m.feidakis@uniwa.gr

**Δρ. Χαράλαμπος Ζ. Πατρικάκης**  
Καθηγητής  
bpatr@uniwa.gr

