



ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΠΑΤΡΩΝ  
UNIVERSITY OF PATRAS

ΑΝΟΙΚΤΑ ακαδημαϊκά  
μαθήματα ΠΠ

# Οντοκεντρικός Προγραμματισμός

Ενότητα 2: Η ΓΛΩΣΣΑ JAVA  
**Βιβλιοθήκες**

**ΔΙΔΑΣΚΟΝΤΕΣ:** Ιωάννης Χατζηλυγερούδης, Χρήστος  
Μακρής

Πολυτεχνική Σχολή

Τμήμα Μηχανικών Η/Υ & Πληροφορικής

# **ΒΙΒΛΙΟΘΗΚΗ JAVA**

# ΒΑΣΙΚΗ ΒΙΒΛΙΟΘΗΚΗ JAVA

---

- Ένα σύνολο κλάσεων και διεπαφών οργανωμένων σε πακέτα
- Βασικά πακέτα
  - ✓ `java.applet`
  - ✓ `java.awt`
  - ✓ `java.io`
  - ✓ `java.lang`
  - ✓ `java.math`
  - ✓ `java.net`
- Για να χρησιμοποιήσουμε κλάσεις ή μεθόδους κλάσεων από τα πακέτα της βιβλιοθήκης (πλην αυτών του `java.lang`), πρέπει να τις εισάγουμε στο πρόγραμμά μας με την εντολή **import**



# ΚΛΑΣΗ System (1)

```
public final class System extends Object
```

- Η κλάση **System** περιέχει τη βασική μεταβλητή εξόδου **out**  
(μεταβλητή κλάσης)
- Η **out** είναι τύπου **PrintStream** (μια άλλη κλάση του πακέτου)

Η γνωστή εντολή εκτύπωσης στην οθόνη

**System.out.println**

ουσιαστικά αποτελεί αποστολή μηνύματος (println) σ' ένα στιγμιότυπο της **PrintStream**, του οποίου αναφορά είναι η μεταβλητή **out** της **System**. Το στιγμιότυπο δημιουργείται αυτόματα από το σύστημα.



# ΚΛΑΣΗ System (2)

---

- Η `println` είναι μέθοδος στιγμιοτύπων της κλάσης `PrintStream`
- Αν η `println` ήταν μέθοδος κλάσης, δεν θα χρειαζόταν να στείλουμε μήνυμα σε στιγμιοτύπο της `PrintStream`

Π.χ. οι μέθοδοι `toString`, `valueOf` είναι μέθοδοι κλάσης. Οπότε μπορώ να στείλω κατ' ευθείαν μήνυμα στην κλάση τους.



# ΚΛΑΣΗ Double (1)

---

```
public final class Double extends Number
                        implements Comparable
```

- Προσοχή!!! Άλλο η κλάση `Double`,  
άλλο ο πρωτογενής τύπος `double`
- Περιέχει ένα απλό πεδίο (μεταβλητή) τύπου `double`



# ΚΛΑΣΗ Double (2)

---

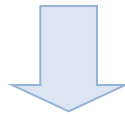
## Μέθοδοι (κλάσης)

**static String toString (double d)**

(δημιουργεί στιγμιότυπο της String που είναι η αλφαριθμητική αναπαράσταση του d, επιστρέφει αναφορά στο στιγμιότυπο)

Π.χ. 

```
double d1 = 1821;
String myString;
myString = Double.toString(d1);
System.out.println(myString);
```



**"1821"**



# ΚΛΑΣΗ Double (3)

---

**static Double valueOf (String s)**

(δημιουργεί στιγμιότυπο της Double με αρχική τιμή την αριθμητική αναπαράσταση του s)

Π.χ. 

```
Double myDouble;  
String s1 = "1821";  
myDouble = Double.valueOf(s1);  
System.out.println(myDouble);
```



**"1821"**



# ΔΗΜΙΟΥΡΓΙΑ ΣΤΙΓΜΙΟΤΥΠΟΥ (ΧΩΡΙΣ NEW)

Με τη χρήση μεθόδου που επιστρέφει αναφορά σε στιγμιότυπο του ζητούμενου τύπου.

Π.χ.

```
Double myDouble = Double.valueOf("2000")
```

## Προσοχή!!!

```
Double myDouble;
```

```
double d1 = myDouble.doubleValueOf();
```



```
Double myDouble;
```

```
myDouble = new Double ("2000");
```

```
double d1 = myDouble.doubleValueOf();
```

Η `doubleValueOf()` είναι μέθοδος στιγμιότυπου και επιστρέφει `double`.

Δεν αναφέρεται σε συγκεκριμένο στιγμιότυπο

# ΚΛΑΣΗ `String` (1)

---

```
public final class String extends Object
    implements Serializable, Comparable
```

(Όλες οι αλφαριθμητικές σταθερές, π.χ. "abc", υλοποιούνται σαν στιγμιότυπά της)

## Μέθοδοι

```
int length ()
```

(επιστρέφει τον αριθμό χαρακτήρων)

```
char charAt(int index)
```

(επιστρέφει τον χαρακτήρα στη θέση index)



# ΚΛΑΣΗ String (2)

---

**indexOf(char ch)**

(επιστρέφει την πρώτη θέση που βρίσκεται ο ch, αλλιώς -1)

**lastIndexOf(char ch)**

(επιστρέφει την τελευταία θέση που βρίσκεται ο ch, αλλιώς -1)

**equals(String s)**

(επιστρέφει 'true', αν το ίδιο αντικείμενο με αυτό που καλεί τη μέθοδο, αλλιώς 'false')

**replace(char oldChar, char newChar)**

(επιστρέφει ένα νέο String, ίδιο με αυτό που καλεί, αλλά με newChar όπου oldChar)



# ΚΛΑΣΗ `StringBuffer`

---

## Μέθοδοι

`char setCharAt(int x, char newChar)`

(αλλάζει τον χαρακτήρα στη θέση x με τον newChar)

`replace(char oldChar, char newChar)`

(αλλάζει με newChar όπου oldChar)

**Προσοχή!!!** Διαφορά από την αντίστοιχη της String:  
Δεν δημιουργεί νέο String,  
αλλάζει αυτό που καλεί τη μέθοδο.



# Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στο πλαίσιο του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Αθηνών**» έχει χρηματοδοτήσει μόνο την αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



# Σημείωμα Ιστορικού Εκδόσεων Έργου

---

Το παρόν έργο αποτελεί την έκδοση 1.0.



# Σημείωμα Αναφοράς

---

Copyright: Πανεπιστήμιον Πατρών, Ιωάννης Χατζηλυγερούδης, 2015.  
«Οντοκεντρικός Προγραμματισμός». Έκδοση: 1.0.1 Πάτρα 2015. Διαθέσιμο  
από τη δικτυακή διεύθυνση:

<https://eclass.upatras.gr/courses/CEID1105/>



# Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά, Μη Εμπορική Χρήση Παρόμοια Διανομή 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



[1] <http://creativecommons.org/licenses/by-nc-sa/4.0/>

Ως **Μη Εμπορική** ορίζεται η χρήση:

- που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
- που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
- που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο

Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.



# Διατήρηση Σημειωμάτων

---

Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:

- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.



# Σημείωμα Χρήσης Έργων Τρίτων

---

