



Αντικειμενοστρεφής Προγραμματισμός

Ενότητα 5: Πιο Εξελιγμένη Συμπεριφορά

Γρηγόρης Τσουμάκας, Επικ. Καθηγητής
Τμήμα Πληροφορικής



Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «Ανοικτά Ακαδημαϊκά Μαθήματα στο Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης» 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.



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης





ΑΡΙΣΤΟΤΕΛΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ

ΑΝΟΙΚΤΑ
ΑΚΑΔΗΜΑΙΚΑ
ΜΑΘΗΜΑΤΑ



Πιο Εξελιγμένη Συμπεριφορά



Τα παραδείγματα κώδικα που χρησιμοποιούνται σε κάποιες από τις ακόλουθες διαφάνειες μπορούν να βρεθούν στον παρακάτω σύνδεσμο:
<http://users.auth.gr/greg/oor.zip>

Περιεχόμενα

- Μεταβλητές και μέθοδοι κλάσεων.
- Αλληλεπίδραση μέσω της γραμμής εντολών.
- Κλάσεις βιβλιοθήκης.



Σταθερές Μεταβλητές

- Χρήση λέξης κλειδί *final* πριν τον τύπο.
 - Αν ανατεθεί τιμή στην μεταβλητή, τότε αυτή η τιμή δεν μπορεί να αλλάξει.
- Παράδειγμα.

```
final int X = 3;
```

```
X = 5; // σφάλμα
```

```
final int Y;
```

```
Y = 5;
```

```
Y = 3; // σφάλμα
```



Μεταβλητές Κλάσεων

- Μεταβλητές κλάσεων ή στατικές μεταβλητές.
 - Ανά πάσα στιγμή υπάρχει ένα και μόνο αντίγραφο τους ανεξάρτητα από τον αριθμό των αντικειμένων.
- Δήλωση.
 - Με τη λέξη κλειδί *static* πριν από τον τύπο τους.
- Αρχικοποίηση.
 - Μαζί με την δήλωση με τον τελεστή =
 - Σε στατικό μπλοκ κώδικα αρχικοποίησης.



Παράδειγμα 1

```
public class Product {  
    private int id;  
  
    private static int numProducts = 0;  
  
    public Product() { id = ++numProducts; }  
  
    public int getId() { return id; }  
  
    public static void main(String[] args) {  
        Product p1 = new Product();  
        Product p2 = new Product();  
        System.out.println(p2.getId());  
    }  
}
```

Product.java



Παράδειγμα 2

```
import java.util.ArrayList;

public class StaticBlock {

    private static ArrayList<String> names;

    static {
        names = new ArrayList<>();
        names.add("greg");
        names.add("nick");
    }
}
```

StaticBlock.java



Σταθερές Κλάσεων

- Σταθερές κλάσεων.
 - Μια συνηθισμένη χρήση μεταβλητών κλάσεων είναι ο ορισμός σταθερών πεδίων.
 - Δεν υπάρχει λόγος να υπάρχουν αντίγραφα μιας σταθερής μεταβλητής σε κάθε αντικείμενο.



Παράδειγμα

```
public class FreeFall {  
    private double height;  
    private static final double GRAVITY = 9.78;  
    public FreeFall(double h) { height = h; }  
    public double heightAfterSeconds(int t) {  
        return height-0.5*GRAVITY*t*t;  
    }  
    public double timeToVelocity(double v) {  
        return v/GRAVITY;  
    }  
}
```

FreeFall.java



Μέθοδος Κλάσης (1/2)

- Μέθοδος κλάσης ή αλλιώς στατική μέθοδος.
 - Μπορεί να κληθεί χωρίς να υπάρχει στιγμιότυπο.
- Δήλωση.
 - Με την προσθήκη της λέξης κλειδί *static* πριν από τον τύπο της επιστρεφόμενης τιμής μιας μεθόδου.
- Κλήση.
 - Όνομα_κλάσης.όνομα_στατικής_μεθόδου.



Παράδειγμα

```
public class StaticMethod {  
    public static void hello() {  
        System.out.println("hello");  
    }  
}
```

```
-----  
public class StaticMethodApp {  
    public static void main(String[] args) {  
        StaticMethod.hello();  
    }  
}
```

StaticMethod.java, StaticMethodApp.java



Μέθοδος Κλάσης (2/2)

- Περιορισμοί.
 - Δεν μπορεί να προσπελάσει μη στατικά πεδία, αφού αυτά σχετίζονται με συγκεκριμένα αντικείμενα.
 - Δεν μπορεί να προσπελάσει μη στατικές μεθόδους, αφού αυτές μπορεί να σχετίζονται με μη στατικά πεδία.



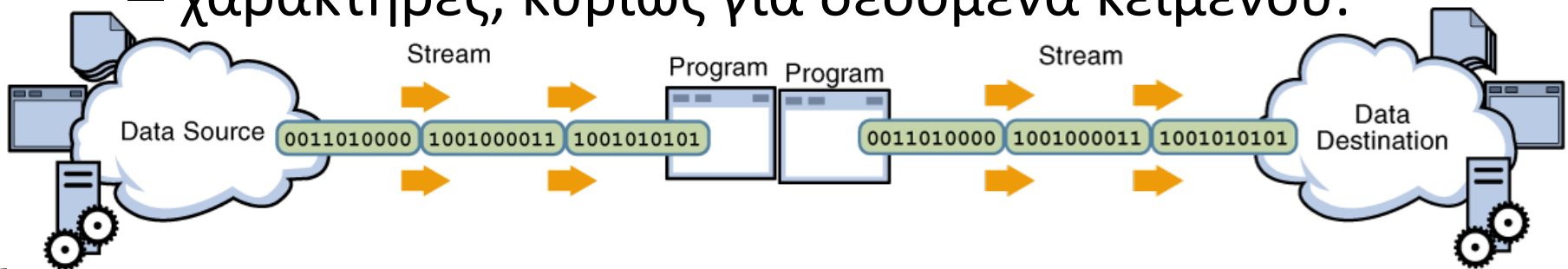
Περιεχόμενα

- Μεταβλητές και μέθοδοι κλάσεων.
- Αλληλεπίδραση μέσω της γραμμής εντολών.
- Κλάσεις βιβλιοθήκης.



Ρεύματα (Streams)

- Ένας ορισμός του ρεύματος.
 - Αφαιρετική έννοια για κάτι το οποίο παράγει (είσοδος) ή καταναλώνει (έξοδος) δεδομένα.
- Είδη ρευμάτων.
 - byte, κυρίως για δυαδικά δεδομένα.
 - χαρακτήρες, κυρίως για δεδομένα κειμένου.



Προκαθορισμένα Ρεύματα

- System.in
 - Προκαθορισμένο ρεύμα εισόδου.
 - Συνήθως αντιστοιχεί στο πληκτρολόγιο.
- System.out
 - Προκαθορισμένο ρεύμα φυσιολογικής εξόδου.
 - Συνήθως αντιστοιχεί στην έξοδο προς την οθόνη.
- System.err
 - Προκαθορισμένο ρεύμα εξόδου για σφάλματα.
 - Συνήθως αντιστοιχεί στην έξοδο προς την οθόνη ή προς κάποιο αρχείο καταγραφής (log) σφαλμάτων.



Μορφοποιημένη Έξοδος

- Η κλάση *PrintStream*.
 - Τα *System.out* και *System.err* είναι αντικείμενα της κλάσης *PrintStream*, η οποία επιτρέπει τη μορφοποιημένη έξοδο.
- Μέθοδοι.
 - *print, println*.
 - *format* ή *printf* (όπως η *printf* της C).



Συμβολοσειρά Μορφοποίησης (1/3)

- Δομή συμβολοσειράς μορφοποίησης.
 - %[θέση\$] [σημαίες] [πλάτος] [.ακρίβεια] τύπος
- **Θέση.**
 - Για επαναχρησιμοποίηση της ίδιας παραμέτρου σε διαφορετικά σημεία, ίσως με διαφορετικό τύπο.
 - Για αλλαγή της σειράς εμφάνισης των παραμέτρων.
- **Πλάτος.**
 - Ελάχιστο πλάτος εμφάνισης (δεξιά στοίχιση).
- **Σημαία: '-'**
 - Αριστερή στοίχιση μέσα στο ελάχιστο πλάτος.



Συμβολοσειρά Μορφοποίησης (2/3)

- Δομή συμβολοσειράς μορφοποίησης.
 - %[θέση\$] [σημαίες] [πλάτος] [.ακρίβεια] τύπος.
- Διάφορα.
 - %% (%), %n (ορθή αλλαγή γραμμής ανεξάρτητα από το λειτουργικό σύστημα).
- Τύπος String %s, %S.
 - Ακρίβεια: μέγιστο πλάτος κειμένου.
- Τύπος char %c, %C.
- Τύπος boolean %b, %B.



Συμβολοσειρά Μορφοποίησης (3/3)

- Δομή συμβολοσειράς μορφοποίησης.
 - %[θέση\$] [σημαίες] [πλάτος] [.ακρίβεια] τύπος.
- Ακέραιος τύπος %d, %x, %X, %o.
- Πραγματικός τύπος %f, %e, %E.
 - Ακρίβεια: αριθμός δεκαδικών (στρογγυλοποίηση).
- Σημαίες για αριθμούς.
 - '+'/' ': προσθέτει ένα '+'/' ' εμπρός από θετικούς.
 - '0': γεμίζει με 0 αντί κενών το ελάχιστο πλάτος.
 - ',': εμφανίζει διαχωριστή χιλιάδων.

UsingPrintf.java



Είσοδος από το Πληκτρολόγιο

- Δύσχρηστο το *System.in*.
 - Χρήση βοηθητικής κλάσης *java.util.Scanner*.
- Αρχικοποίηση της *Scanner* με το *System.in*.
 - `Scanner aScanner = new Scanner(System.in);`
- Μέθοδοι για ανάγνωση δεδομένων.
 - `nextByte()`, `nextShort()`, `nextInt()`, `nextLong()`.
 - `nextFloat()`, `nextDouble()`, `nextBoolean()`.
 - `nextLine()` για ανάγνωση κειμένου.

ScannerTest.java



Παράδειγμα 1

```
Scanner input = new Scanner(System.in);  
int number1, number2, sum;  
System.out.printf("Δώσε ακέραιο: ");  
number1 = input.nextInt();  
System.out.printf("Δώσε ακέραιο: ");  
number2 = input.nextInt();  
sum = number1 + number2;  
System.out.printf("Άθροισμα: %d\n", sum);
```

AddInt.java



Παράδειγμα 2

```
Scanner scanner = new Scanner(System.in) ;  
System.out.print("Δώσε αριθμό: ") ;  
int x = scanner.nextInt() ;  
  
System.out.print("Δώσε όνομα: ") ;  
  
//scanner.nextLine() ;  
  
String name = scanner.nextLine() ;  
System.out.print("Δώσε όνομα: ") ;  
name = scanner.nextLine() ;  
System.out.print("Δώσε αριθμό: ") ;  
x = scanner.nextInt() ;
```

ReadText.java



Τοπικές Ρυθμίσεις (1/2)

- Η κλάση Locale.
 - Αντιπροσωπεύει μια γλώσσα και μια γεωγραφική περιοχή (προαιρετικά).
- Τρέχουσες τοπικές ρυθμίσεις.
 - `Locale default = Locale.getDefault();`
- Δημιουργία.
 - `Locale usa = new Locale("en", "US");`



Τοπικές Ρυθμίσεις (2/2)

- Οι κλάσεις `PrintStream` και `Scanner`.
 - Λαμβάνουν υπόψη το προκαθορισμένο `Locale`.
 - Δέχονται ως παράμετρο ένα εναλλακτικό `Locale`.
- Παράδειγμα.

```
Locale l = new Locale("en", "US");  
System.out.printf(l, "%.2f", 3.1415);  
Scanner s = new Scanner(System.in);  
s.useLocale(locale);
```

AddDouble.java



Κλάση Console

- Αλληλεπίδραση μέσω αντικειμένου της.
 - `Console c = System.console();`
 - Αν επιστρέψει null, τότε δεν είναι διαθέσιμο.
- Μέθοδοι.
 - `String readLine(String message);`
 - `char[] readPassword(String message);`



Παράδειγμα

```
Console c = System.console();  
  
if (c == null) {  
    System.out.println("No console.");  
    return;  
}  
  
String s;  
  
do {  
    String l = c.readLine("Login: ");  
    char[] p = c.readPassword("Password: ");  
    s = String.copyValueOf(p);  
} while (!s.equals("java"));  
  
System.out.println("Welcome!");
```

Password.java



Ορίσματα Γραμμής Εντολών

- Παράδειγμα ορισμάτων γραμμής εντολών.

```
javac MyProgram.java.
```

```
java MyProgram argument1 "argument 2".
```

- Στο NetBeans.

- Δεξί κλικ στο project -> properties -> Run.

- Πρόσβαση μέσω της παραμέτρου της main.

- `public static void main(String[] args).`

Arguments.java

- Τα στοιχεία του πίνακα αντιστοιχούν στα ορίσματα που έχουν δοθεί στη γραμμή εντολών.



Περιεχόμενα

- Μεταβλητές και μέθοδοι κλάσεων.
- Αλληλεπίδραση μέσω της γραμμής εντολών.
- Κλάσεις βιβλιοθήκης.



Εκδόσεις της Java

Έκδοση	Ημερομ.	Κλάσεις	Πακέτα	Νέα χαρακτηριστικά
Java 1.0	01-1996	212	8	
Java 1.1	02-1997	504	22	Εσωτερικές κλάσεις
Java 1.2	12-1998	1520	59	Java 2, collections, Swing, J2ME, J2EE, ...
Java 2 1.3	05-2000	1842	76	HotSpot, Sound, ...
Java 2 1.4	02-2002	2991	135	XML, GUI, Images, ...
Java 2 5.0	09-2004	3562	166	generics, enum, for/in, autoboxing, ...
Java 6	12-2006	3777	202	Web services, XML, DB, Desktop, ...
Java 7	07-2011	4024	209	Υποστήριξη πρόσφατων πρωτοκόλλων δικτύου, καρτών γραφικών, ...



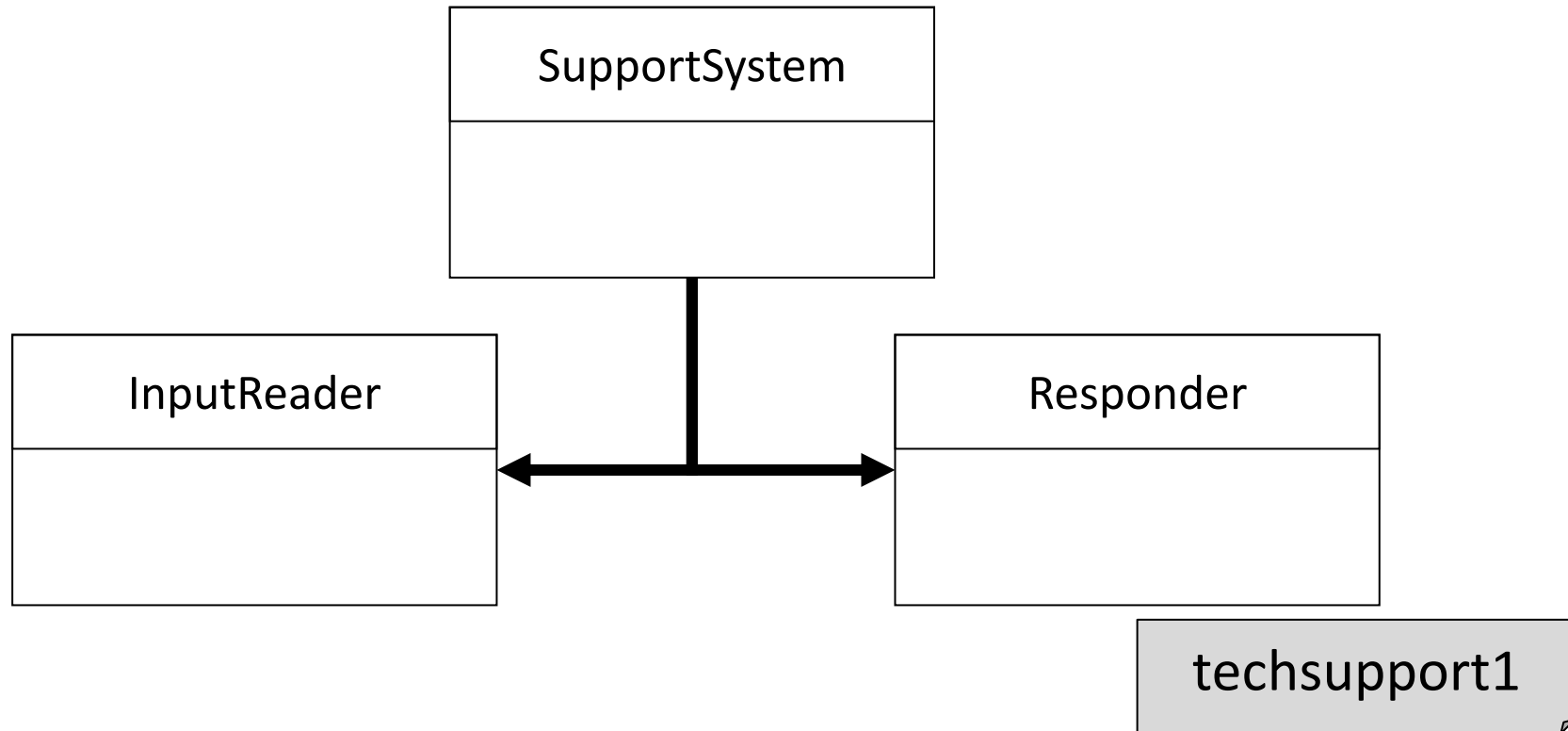
Καλός Προγραμματιστής Java

- Πρέπει να γνωρίζει τα ονόματα ορισμένων από τις σημαντικότερες κλάσεις βιβλιοθήκης.
 - String, System, ArrayList, Iterator, Scanner, ...
 - Άλλες που θα δούμε: HashSet, HashMap, Random.
- Πρέπει να γνωρίζει πώς μπορεί να μάθει περισσότερα για άλλες κλάσεις βιβλιοθήκης.
 - Τι δημόσιες μεθόδους προσφέρουν; Τι παραμέτρους δέχονται; Τι ακριβώς κάνουν;



Παράδειγμα

- Σύστημα αυτόματης τεχνικής υποστήριξης.



Τεκμηρίωση Κλάσεων Βιβλιοθήκης

- Αφορά στη **διασύνδεση** τους (αφαίρεση).
 - Όνομα, γενική περιγραφή, λίστα κατασκευαστών και μεθόδων, περιγραφή, παράμετροι και επιστρεφόμενες τιμές τους.
- Διατίθεται online καθώς και για download.
 - <http://download.oracle.com/javase/7/docs>



Πηγαίος Κώδικας Κλάσεων Βιβ.

- Αφορά στην **υλοποίηση** τους.
- Περιλαμβάνεται στη διανομή του JDK.
 - Αρχείο *src.zip* στο βασικό φάκελο εγκατάστασης.
- Κάθε σοβαρό IDE, επιτρέπει την πρόσβαση στον πηγαίο κώδικα κλάσεων βιβλιοθήκης.
 - Στο *NetBeans*, κάνετε δεξί κλικ επάνω στην κλάση ή μέθοδο και επιλέγετε *Navigate -> Go to Source*.



Συγγραφή Τεκμηρίωσης

- Συνήθως αγνοούμε την τεκμηρίωση.
 - Δύσκολη η κατανόηση των κλάσεων από εσάς και άλλους έπειτα από κάποιο χρονικό διάστημα.
 - Δύσκολη η ανάγνωση πηγαίου κώδικα.
- Γενική αρχή.
 - Η τεκμηρίωση πρέπει να είναι αρκετά λεπτομερής ώστε να μπορεί κάποιος να χρησιμοποιήσει την κλάση χωρίς να χρειάζεται να δει την υλοποίηση.



`/ javadoc */`**

- Πάνω από τον ορισμό της κλάσης γράφουμε.
 - Μια γενική περιγραφή της κλάσης.
 - Τα ονόματα των συγγραφέων με `@author`.
 - Τον αριθμό έκδοσης της κλάσης με `@version`.
- Πάνω από τον ορισμό μιας μεθόδου.
 - Περιγραφή του σκοπού και της λειτουργίας της.
 - Περιγραφή της κάθε παραμέτρου με `@param`.
 - Περιγραφή της επιστρεφόμενης τιμής με `@return`.



Χρήση javadoc στο NetBeans

- Ανάλυση της τεκμηρίωσης.
 - Δεξί κλικ σε πακέτο/κλάση: Tools->Analyze Javadoc.
 - Μας αναφέρει πιθανές ελλείψεις στην τεκμηρίωση.
- Παραγωγή τεκμηρίωσης.
 - Δεξί κλικ επάνω σε ένα project: Generate Javadoc
 - Ιστοσελίδες με τεκμηρίωση, όπως αυτές του Java API, δημιουργούνται στον υποφάκελο *dist/javadoc/*
 - Απαιτείται ο κώδικας να είναι σε κάποιο πακέτο.



Μήκος Συμβολοσειράς

- Μέθοδος `int length()`.
- Παράδειγμα.

```
String str1 = "hello";  
  
int length = str1.length();  
  
System.out.println(length)  
  
// 5
```



Χαρακτήρας σε Δοθείσα Θέση

- Μέθοδος `char charAt(int index)`.
- Παράδειγμα.

```
String str = "hello";
```

```
for (int i=0; i<str.length(); i++)
```

```
    System.out.println(str.charAt(i));
```

```
// h
```

```
// e
```

```
// l
```

```
// l
```

```
// o
```



Αφαίρεση Λευκών Κενών

- Μέθοδος String trim().
- Παράδειγμα.

```
String str1 = " \t Hello World! \n";
```

```
String str2 = str1.trim();
```

```
System.out.print(str2 + str2);
```

```
// Hello World!Hello World!
```



Μετατροπή σε Πεζά/Κεφαλαία

- Μέθοδος String toLowerCase().
- Μέθοδος String toUpperCase().
- Παράδειγμα.

```
String str1 = "Hello World!";  
String str2 = str1.toLowerCase();  
System.out.println(str2);  
// hello world!
```



Αμετάλλακτα Αντικείμενα

- Αμετάλλακτα αντικείμενα (immutable objects).
 - Αντικείμενα κλάσεων δίχως μεθόδους μετάλλαξης.
 - Από τη στιγμή που θα δημιουργηθούν και έπειτα, η κατάσταση τους δεν μπορεί να αλλάξει.
- Τα αντικείμενα της *String* είναι αμετάλλακτα.
 - Δεν υπάρχουν μέθοδοι αλλαγής των χαρακτήρων.
 - Μέθοδοι που "φαίνεται" να αλλάζουν το αντικείμενο, στην πραγματικότητα δημιουργούν και επιστρέφουν ένα νέο.



Έλεγχος Ισότητας

- Μέθοδος `boolean equals(String str)`.
 - Ο τελεστής `==` ελέγχει αν οι θέσεις μνήμης που είναι αποθηκευμένες στις μεταβλητές είναι ίδιες!
- Παράδειγμα.

```
String s1 = new String("ABC");  
String s2 = new String("ABC");  
System.out.printf("%b ", s1 == s2);  
System.out.printf("%b ", s1.equals(s2));  
s1 = "ABC"; s2 = "ABC";  
System.out.printf("%b ", s1 == s2);  
  
// false true true
```

StringEquality.java



Βελτίωση Παραδείγματος

- Τροποποιήστε τη μέθοδο *start* της κλάσης *SupportSystem* ώστε η έξοδος να
 - Γίνεται και με " bye", "bye ", "Bye ", "BYE", " bye" .
 - Μην γίνεται όταν ακολουθούν και άλλες λέξεις.
- Χρήση *trim* και *toLowerCase*.
- Χρήση *equals* αντί για *startsWith*.



Διαχωρισμός σε Τμήματα

- Μέθοδος `String[] split(String s)`.
 - Ως παράμετρο παίρνει μια συμβολοσειρά διαχωρισμού της συμβολοσειράς που την καλεί.
 - Επιστρέφει πίνακα `String` με τα τμήματα.
- Παράδειγμα.

```
String s = "ένα-δύο-τρία-τέσσερα-πέντε";  
String[] tokens = s.split("-");  
for (String token : tokens)  
    System.out.println(token);
```

Split.java



Σύγκριση με Άλλη Συμβολοσειρά

- Μέθοδος `int compareTo(String str)`.
 - Επιστρέφει 0 αν ισούνται, >0 αν η καλούσα έπεται λεξικογραφικά της `str`, και <0 αν προηγείται.

- Παράδειγμα.

```
String s1 = "Hello", s2 = "Bye bye!";
```

```
int result = s1.compareTo(s2);
```

```
if (result < 0)
```

```
    System.out.println("προηγείται η " + s1);
```

```
else
```

```
    System.out.println("προηγείται η " + s2);
```



Συνένωση Συμβολοσειρών

- Μέθοδος String concat(String str).
- Παράδειγμα.

```
String str1 = "Βοηθήστε";
```

```
String str2 = " με!";
```

```
String str3 = str1.concat(str2);
```

```
System.out.println(str3); // Βοηθήστε με!
```

```
String str4 = "Java ".concat(" 1.7");
```

```
System.out.println(str4); // Java 1.7
```



Συνένωση με τον Τελεστή Πρόσθεσης

- Η συνένωση γίνεται εναλλακτικά με τον τελεστή "+" της πρόσθεσης.
 - Μοναδική περίπτωση υπερφόρτωσης τελεστή στη Java.
- Παράδειγμα.

```
String str1 = "ίσα";
```

```
str1 = str1 + " " + str1;
```

```
System.out.println(str1); // ίσα ίσα
```

```
String str2 = "Java";
```

```
String str3 = str2 + " 1.7";
```

```
System.out.println(str3); // Java 1.7
```



Η Κλάση `StringBuilder`

- Είπαμε πως τα `Strings` είναι αμετάλλακτα.
 - Τι κάνουμε αν θέλουμε να κάνουμε αλλαγές;
- Χρησιμοποιούμε την κλάση `StringBuilder`.
 - `append(String s) // int, char, κτλ`
 - `setCharAt(int index, char ch)`
 - `deleteCharAt(int index)`
 - `insert(int index, String s) // int, κτλ`
 - `delete(int from, int to)`
 - `replace(int from, int to, String s)`
 - `setLength(int l)`

`StringConcatenation.java`



Συμβολοσειρά από Μεταβλητές

- Στατική μέθοδος `String format(String s, ...)`.
 - Λειτουργεί όπως η μέθοδος *format* του *System.out*
- Παράδειγμα.

```
int x = 3;
```

```
int y = 2;
```

```
int z = x+y;
```

```
String s;
```

```
s = String.format("%d + %d = %d", x, y, z);
```



Τυχαίοι Αριθμοί (1/2)

- Η κλάση *java.util.Random*.
 - Γεννήτρια ψευδοτυχαίων αριθμών.
- Αρχικοποίηση.
 - `Random r1 = new Random(); // ρολόι`
`r1.setSeed(756L) // χρήστης`
 - `Random r2 = new Random(12L); // χρήστης`
- Λήψη τυχαίου πραγματικού αριθμού.
 - `float f = r1.nextFloat(); // [0..1)`
 - `double d = r1.nextDouble(); // [0..1)`



Τυχαίοι Αριθμοί (2/2)

- Μέθοδος λήψης τυχαίας λογικής τιμής.

```
boolean b = r1.nextBoolean(); // true/false
```

- Μέθοδοι λήψης τυχαίου ακέραιου αριθμού.

```
int i1 = r1.nextInt(); // όλο το εύρος
```

```
int i2 = r1.nextInt(5); // 0, 1, 2, 3, 4
```

```
long l = r1.nextLong(); // όλο το εύρος
```

- Τυχαίος αριθμός κανονικής κατανομής.

– Μέση τιμή 0.0 και τυπική απόκλιση 1.0.

```
double gauss = r1.nextGaussian(); RandomNums.java
```



Βελτίωση Παραδείγματος

- Τροποποιήστε την κλάση *Responder* ώστε να επιστρέφει τυχαία μία απάντηση μέσα από μια λίστα με έτοιμες απαντήσεις.
 - Χρήση `ArrayList` με απαντήσεις.
 - Χρήση `Random` για τυχαίο αριθμό από 0 έως το πλήθος των απαντήσεων μείον ένα.
 - Χρήση `get` της `ArrayList` με αυτόν τον αριθμό.

techsupport2



Σύνολο (Set)

- Συλλογή αντικειμένων με τις εξής ιδιότητες.
 - Κάθε διαφορετικό στοιχείο υπάρχει μία μόνο φορά.
 - Τα στοιχεία δεν διατηρούνται στη σειρά με την οποία προστίθενται στη συλλογή.
- Η κλάση *java.util.HashSet* υλοποιεί ένα σύνολο.
 - Διαθέτει μεθόδους *add*, *remove*, *size*, αλλά όχι *get*.
 - Χρήση *for-each* ή *Iterator* για να το διατρέξουμε.
 - Χρησιμοποιεί πίνακα κατακερματισμού για αποθήκευση των αντικειμένων.

SetMethods.java



Χάρτης (Map)

- Αναπαριστά συλλογές από ζεύγη αντικειμένων.
 - Αντικείμενο **κλειδί** και αντικείμενο **τιμή**.
 - Επιτρέπει την ανάκτηση του αντικειμένου τιμή με βάση το αντικείμενο κλειδί.
 - Δεν επιτρέπεται η ύπαρξη του ίδιου κλειδιού παραπάνω από μία φορές.
- Η κλάση *java.util.HashMap* υλοποιεί ένα χάρτη.
 - Χρησιμοποιεί πίνακα κατακερματισμού για αποθήκευση των κλειδιών.



Βασικές Μέθοδοι HashMap (1/2)

- V put(K key, V value);
 - Επιστρέφει *null* αν δεν υπήρχε το *key*, αλλιώς την προηγούμενη τιμή που ήταν αποθηκευμένη.
- V get(Object key);
 - Επιστρέφει *null* αν δεν υπήρχε το *key*, αλλιώς την τιμή που είναι αποθηκευμένη.
- V remove(Object key);
 - Επιστρέφει *null* αν δεν υπήρχε το *key*, αλλιώς την τιμή που ήταν αποθηκευμένη.



Βασικές Μέθοδοι HashMap (2/2)

- Έλεγχος για ύπαρξη κλειδιού και τιμής.

```
boolean containsKey(Object key) ;
```

```
boolean containsValue(Object value) ;
```

- Λήψη συλλογής κλειδιών και τιμών.

```
keySet () ; // επιστρέφει σύνολο κλειδιών
```

```
values () ; // επιστρέφει συλλογή τιμών
```

- Λήψη συνόλου ζευγών κλειδιών και τιμών τύπου Map.Entry<K, V>, με μεθόδους getKey() και getValue()

```
entrySet () ;
```

MapMethods.java



Βελτίωση Παραδείγματος

- Τροποποιήστε το σύστημα τεχνικής υποστήριξης ώστε να επιστρέφει μια κατάλληλη απάντηση ανάλογα με τις λέξεις που δίνει ο χρήστης.
 - HashMap για αντιστοίχιση λέξεων σε απαντήσεις.
 - Χρήση μεθόδου split της κλάσης String για να σπάσουμε την ερώτηση του χρήστη σε λέξεις και προσθήκη τους σε HashSet.
 - Αναζήτηση λέξεων χρήστη στο HashMap. `techsupport3`



Κλάσεις Περιτύλιξης (1/2)

- Θετικό χαρακτηριστικό θεμελιωδών τύπων.
 - Αποδοτικότητα (ταχύτητα), καθώς δεν αποτελούν αντικείμενα (δεν κληρονομούν την κλάση Object).
- Πρόβλημα θεμελιωδών τύπων.
 - Ορισμένες προηγμένες δυνατότητες της Java (π.χ. οι συλλογές) εφαρμόζονται μόνο σε αντικείμενα.
- Η λύση στο παραπάνω πρόβλημα.
 - Κλάσεις περιτύλιξης (type wrappers), οι οποίες ενσωματώνουν τους θεμελιώδεις τύπους.



Κλάσεις Περιτύλιξης (2/2)

- Βρίσκονται στο java.lang
 - Character, Boolean, Float, Double.
 - Byte, Short, Integer, Long,
- Είναι **αμετάτρεπτες** κλάσεις.
 - Άπαξ και δημιουργηθεί ένα αντικείμενο τους, δεν γίνεται να αλλάξει το περιεχόμενό του.
 - Δεν μπορούμε να περάσουμε ως παράμετρο ένα τέτοιο αντικείμενο σε μια μέθοδο και αυτή να αλλάξει το περιεχόμενό του.



Αρχικοποίηση

- Όλες οι αριθμητικές κλάσεις αρχικοποιούνται με δομητή που παίρνει ως παράμετρο.
 - Μια τιμή του αντίστοιχου θεμελιώδη τύπου.
 - Μια αναπαράσταση της τιμής αυτής ως συμβολοσειρά.
- Παραδείγματα.
 - `Integer age = new Integer(23);`
 - `Double money = new Double("700.70");`



Μετατροπή σε Θεμελιώδη Τύπο

- Μέθοδοι για μετατροπή σε επιθυμητό θεμελιώδη αριθμητικό τύπο.
 - `byte` `byteValue()`, `short` `shortValue()`.
 - `int` `intValue()`, `long` `longValue()`.
 - `float` `floatValue()`, `double` `doubleValue()`.
- Παράδειγμα.

```
Double sum = new Double(72.34);  
  
int x = sum.intValue();
```



Αυτόματη (Από)Πλαισίωση

- Από θεμελιώδη τύπο σε κλάση (boxing).

```
Integer age = new Integer(23);
```

- Από κλάση σε πρωτόγονο (unboxing).

```
Double money = new Double(700.70);
```

```
double m = money.doubleValue();
```

- Αυτοματοποιήσεις (από Java 5 και έπειτα).

```
Integer x1 = 3, x2; // boxing
```

```
int y1 = x1; // unboxing
```

```
x2 = x1 + (x1 / 2); // unboxing, boxing
```



Από Συμβολοσειρά σε Αριθμό

- Διατίθενται στατικές μέθοδοι για μετατροπή συμβολοσειρών σε θεμελιώδεις τύπους.

```
String str1 = "1.5";
```

```
double a = Double.parseDouble(str1);
```

```
float b = Float.parseFloat(str1);
```

```
String str2 = "5";
```

```
int c = Integer.parseInt(str2);
```

```
long d = Long.parseLong(str2);
```

```
short e = Short.parseShort(str2);
```

```
byte f = Byte.parseByte(str2);
```



Από Αριθμό σε Συμβολοσειρά

- Διατίθεται στατική μέθοδος *toString* για μετατροπή σε συμβολοσειρά.

```
String strFloat = Float.toString(0.01f);  
System.out.println(strFloat) // 0.01
```

- Εναλλακτικά η μετατροπή μπορεί να γίνει με τη στατική μέθοδο *valueOf* της *String*

```
String strDouble = String.valueOf(3.14);  
System.out.println(strDouble) // 3.14
```



Παράδειγμα

```
double d = 858.48;
```

```
String s = Double.toString(d);
```

```
int dot = s.indexOf('.');
```

```
System.out.print(dot + " ψηφία πριν και ");
```

```
System.out.println((s.length() - dot - 1) +  
" μετά το σημείο των δεκαδικών.");
```





Τέλος Ενότητας

Επεξεργασία: Εμμανουήλ Ρήγας
Θεσσαλονίκη, Εαρινό Εξάμηνο 2013-2014



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ