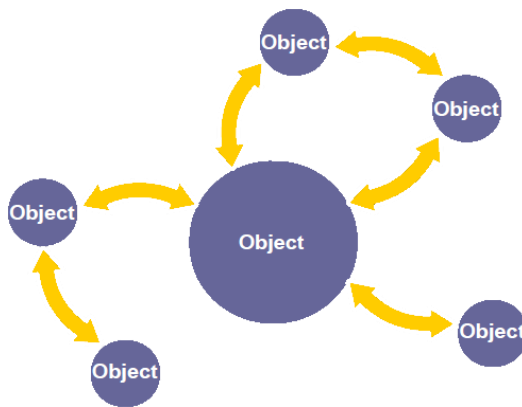


Αντικειμενοστραφής Προγραμματισμός

Ενότητα 4: Μέθοδοι, κλάσεις και βιβλιοθήκες

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. Δημιουργία-δομή κλάσης
2. Βιβλιοθήκες Java
3. Η κλάση String
4. Η κλάση Random
5. Η κλάση Math

Κλάσεις (Classes)

- Κατηγορίες Κλάσεων:
 1. Κλάσεις που θα γράψετε εσείς..
 2. Κλάσεις που έχει γράψει κάποιος άλλος..
 3. Κλάσεις που ανήκουν στην Java

Άσκηση 4.1 : Alex Birthday 1

Είναι τα γενέθλια του Alex! Έχετε οργανώσει ένα πάρτυ οκτώ φίλων για να γιορτάσετε σε ένα τοπικό εστιατόριο. Όταν έρχεται ο λογαριασμός, κανείς δεν είναι σίγουρος τι οφείλει. Γνωρίζετε μόνο ότι: ΦΠΑ (5%) και φιλοδώρημα: 15%. Τι πρέπει να πληρώσει ο καθένας;

- Ανοίξτε και επεξεργαστείτε το “Tip01”.
- Στον πίνακα δίπλα φαίνεται τι οφείλει ο καθένας:
- Το πρόγραμμα θα πρέπει να τυπώνει στο τέλος:

```
person1: 12.0€  
person2: 14.4€  
person3: 10.8€  
person4: 9.6€  
person5: 8.4€  
person6: 18.0€  
person7: 13.2€  
person8: 36.0€
```

Person 1: 10€
Person 2: 12€
Person 3: 9€
Person 4: 8€
Person 5: 7€
Person 6: 15€ (Alex)
Person 7: 11€
Person 8: 30€

Άσκηση 4.1 - Λύση 1

```
public class Tip01 {  
  
    public static void main(String[] args) {  
  
        double person1 = 10*1.2;  
        System.out.println("person1: "+person1);  
  
        double person2 = 12*1.2;  
        System.out.println("person2: "+person2);  
  
        double person3 = 10.8*1.2;  
        System.out.println("person3: "+person3);  
  
    }  
}
```

/* Για τον 1^ο
10+10*0.15+10*0.05=
10*(1+0.20)=
10*1.2 */

/* Αντίστοιχα για τον 2^ο
12*1.2 */

/* και για τον 3^ο
10.8*1.2 */

// και αν είχαμε 100!!!
// αν κάναμε λάθος το φόρο!

Άσκηση 4.1 - Λύση 2

```
public class Tip01 {  
    static double tax = 0.05;  
    static double tip = 0.15;  
    public static void main(String[] args) {  
        double person1 = 10;  
        double total1 = person1*(1 + tax + tip);  
        System.out.println ("person1: "+total1);  
  
        double person2 = 12;  
        double total2 = person2*(1 + tax + tip);  
        System.out.println ("person2: "+total2);  
  
        double person3 = 10.8;  
        double total3 = person3*(1 + tax + tip);  
        System.out.println ("person3: "+total3);  
    }  
}
```

← variables **tax**, **tip**

Οι μεταβλητές
προσφέρουν ευελιξία

Επαναλαμβανόμενες
συμπεριφορές

Μπορεί να
χρησιμοποιηθεί μία
μέθοδος

Χρήση μεθόδων

- Εάν ...
 - Πολλές παρόμοιες γραμμές κώδικα (συμπεριλαμβανομένων υπολογισμών).
 - Πρέπει να περιγράψουμε τη συμπεριφορά ενός αντικειμένου.

Κύρια Μέθοδος (main)

- Γνωστή και ως **οδηγός** (driver):
 - Καθοδηγεί τα συμβάντα (events) ενός προγράμματος.
 - Παρέχει πρόσβαση σε πεδία και μεθόδους ή σε άλλες κλάσεις.
- Δεν περιγράφει τη συμπεριφορά κάποιου συγκεκριμένου αντικειμένου.
 - Πρέπει να είναι ξεχωριστά από τις διάφορες κλάσεις αντικειμένων.
 - Μόνο μία κύρια μέθοδο για κάθε εφαρμογή.


Δημιουργία κλάσης αντικειμένων

```
public class Calculator{  
    public static void main(String args[]){  
        double tax = 0.05;  
        double tip = 0.15;  
        double person1 = 10;  
        double total1 = person1*(1+tax+tip);  
        System.out.println(total1);  
    }  
}
```

Βήματα:

Δημιουργία κλάσης αντικειμένων

```
public class Calculator{  
    //Fields  
    public double tax = 0.05;  
    public double tip= 0.15;  
    public double originalPrice= 10;  
  
    public static void main(String args[]){  
        //double tax = 0.05;  
        //double tip = 0.15;  
        //double person1 = 10;  
        double total1 = originalPrice *(1 +tax +tip);  
        System.out.println(total1);  
    }  
}
```




Βήματα:

1. Μετακίνηση πεδίων από την κύρια μέθοδο

Δημιουργία κλάσης αντικειμένων

```
public class Calculator{  
    //Fields  
    public double tax = 0.05;  
    public double tip= 0.15;  
    public double originalPrice= 10;  
  
    //Methods  
    public void findTotal(){  
        double total1 = originalPrice *(1 +tax +tip);  
        System.out.println(total1);  
    }  
    public static void main(String args[]){  
        //double tax = 0.05;  
        //double tip = 0.15;  
        //double person1 = 10;  
        //double total1 = person1*(1 +tax +tip);  
        //System.out.println(total1);  
    }  
}
```



Βήματα:

1. Μετακίνηση πεδίων από την κύρια μέθοδο
2. Μετακίνηση επαναλαμβανόμενων συμπεριφορών από την κύρια μέθοδο

Δημιουργία κλάσης αντικειμένων

```
public class Calculator{  
    //Fields  
    public double tax = 0.05;  
    public double tip= 0.15;  
    public double originalPrice= 10;  
  
    //Methods  
    public void findTotal(){  
        double total1 = originalPrice *(1 +tax +tip);  
        System.out.println(total1);  
    }  
    //public static void main(String args[]){  
        //double tax = 0.05;  
        //double tip = 0.15;  
        //double person1 = 10;  
        //double total1 = person1*(1 +tax +tip);  
        //System.out.println(total1);  
    //}  
}
```

Βήματα:

1. Μετακίνηση πεδίων από την κύρια μέθοδο
2. Μετακίνηση επαναλαμβανόμενων συμπεριφορών από την κύρια μέθοδο
3. Κατάργηση κύριας μεθόδου

Δημιουργία κλάσης αντικειμένων

```
public class Calculator{  
    //Fields  
    public double tax = 0.05;  
    public double tip= 0.15;  
    public double originalPrice= 10;  
  
    //Methods  
    public void findTotal(){  
        double total1 = originalPrice *(1 +tax +tip);  
        System.out.println(total1);  
    }  
}
```

Βήματα:

1. Μετακίνηση πεδίων από την κύρια μέθοδο
2. Μετακίνηση επαναλαμβανόμενων συμπεριφορών από την κύρια μέθοδο
3. Κατάργηση κύριας μεθόδου

Τέλος!

Δημιουργία κλάσης αντικειμένων

```
public class Calculator{  
    //Fields  
    public double tax = 0.05;  
    public double tip= 0.15;  
    public double originalPrice= 10;  
  
    //Methods  
    public void findTotal(){  
        double total1 = originalPrice *(1 +tax +tip);  
        System.out.println(total1);  
    }  
}
```

Βήματα:

1. Μετακίνηση πεδίων από την κύρια μέθοδο
2. Μετακίνηση επαναλαμβανόμενων συμπεριφορών από την κύρια μέθοδο
3. Κατάργηση κύριας μεθόδου

Τέλος!

Και η κύρια main() μέθοδος;;;

Δημιουργία κλάσης αντικειμένων

```
public class Calculator{  
    //Fields  
    public double tax = 0.05;  
    public double tip= 0.15;  
    public double originalPrice= 10;  
  
    //Methods  
    public void findTotal(){  
        double total1 = originalPrice *(1 +tax +tip);  
        System.out.println(total1);  
    }  
}
```

```
public class CalculatorTest {  
    public static void main(String args[]){  
        //Create Calculator object instance  
        Calculator calc = new Calculator();  
        calc.tip= 0.10;    //Altering a field  
        calc.findTotal(); //Calling a method  
    }  
}
```

Βήματα:

1. Μετακίνηση πεδίων από την κύρια μέθοδο
2. Μετακίνηση επαναλαμβανόμενων συμπεριφορών από την κύρια μέθοδο
3. Κατάργηση κύριας μεθόδου

Τέλος!

Και η κύρια main() μέθοδος;;;

Σε μια άλλη κλάση, π.χ. test class η οποία:

- Δημιουργεί αντικείμενα
- Καλεί πεδία και μεθόδους ενός αντικειμένου χρησιμοποιώντας την τελεία “.”

Μεταβλητές αντικειμένων

- Όπως και στους βασικούς τύπους (primitives), για τα αντικείμενα χρησιμοποιούνται μεταβλητές.
- Για τον ορισμό και την αρχικοποίησή τους απαιτείται η λέξη **new**.
 - Αυτό ονομάζεται **instantiating** (δημιουργία αντικειμένου).
 - Στα αντικείμενα τύπου **String**, δεν απαιτείται **instantiating**.

```
int age = 22;  
String str = "Happy Birthday!";  
Scanner sc = new Scanner();  
Calculator calc = new Calculator();
```

type name value

Ο τελεστής “.”

- Τοποθετείται μετά το όνομα μιας μεταβλητής **αντικειμένου** για πρόσβαση στα **πεδία** ή τις **μεθόδους** του.

```
public class CalculatorTest {  
    public static void main(String args[]) {  
        Calculator calc = new Calculator();  
        calc.printTip();           //prints 0.15  
        calc.tip = 0.10;  
        calc.printTip();           //prints 0.10  
    }  
}
```

```
public class Calculator{  
    public double tip= 0.15; //initialized value 0.15  
    public void printTip(){  
        System.out.println(tip);  
    }  
}
```

Άσκηση 4.2 : Alex Birthday 2

- Εισαγάγετε και ανοίξτε το Tip02
- Συμπληρώστε τη μέθοδο **findTotal ()**, η οποία θα πρέπει να:
 - Υπολογίζει σύνολο με βάση τα πεδία: **tax**, **tip**, και **originalPrice** .
 - Εκτυπώνει σύνολο ενός ατόμου.
- Από την κύρια μέθοδο:
 - Δημιουργήστε ένα **Calculator** object με όνομα **calc**.
 - Αποκτήστε πρόσβαση στα πεδία και στις μεθόδους του αντικειμένου για να εκτυπώσετε το σύνολο για κάθε άτομο.
- Αλλάξτε τις τιμές στα πεδία **tip** και **tax**.

Ορίσματα Μεθόδων (arguments)

- Στο πρόγραμμα **Calculator**:

```
public class Calculator{  
    public double tip= 0.15; //initialized value 0.15  
    public void printTip(){  
        System.out.println(tip);  
    }  
}
```

- Χρειάζονται 2 γραμμές για κάθε άτομο!
- Επίσης είναι επικίνδυνο να γράφουμε κώδικα ο οποίος αλλάζει πεδία απευθείας!
- Μπορούμε ωστόσο να περάσουμε **ορίσματα (arguments)** σε μία μέθοδο

Ορίσματα (arguments) & Παράμετροι (parameters)

- **Όρισμα** είναι μια τιμή που περνά κατά τη κλήση μιας μεθόδου:

```
Calculator calc = new Calculator();  
calc.calculate(3, 2.0); //εκτυπώνει 1.5
```

Arguments

- **Παράμετρος** είναι μια μεταβλητή που ορίζεται στη δήλωση μεθόδου:

```
public void calculate (int x, double y){  
    System.out.println (x/y);  
}
```

Parameters

Ορίσματα Μεθόδων (arguments)

- Η μέθοδος **calculate** είναι γραμμένη για να δεχθεί δύο ορίσματα:

- Το 1^ο τύπου **int**
- Το 2^ο τύπου **double**,

```
public void calculate(int x, double y){  
    System.out.println(x/y); //prints 1.5  
}
```

- Κλήση **calculate** →

```
Calculator calc = new Calculator();  
calc.calculate(3, 2.0);
```

- Τι θα γίνει εάν αλλάξουμε τη σειρά;

```
Calculator calc = new Calculator();  
calc.calculate(2.0, 3);
```

- Compiler error:

- **int x cannot be assigned a double value!**
- Η σειρά των ορισμάτων έχει σημασία!

Άσκηση 4.3 : Alex Birthday 3

- Ανοίξτε το Tip03
- Από την κύρια μέθοδο:
 - Χρησιμοποιήστε ένα αντικείμενο **Calculator** και κάντε κλήση της **FindTotal** () με τις κατάλληλες παραμέτρους για να εκτυπώσετε το σύνολο για κάθε άτομο.
 - Παρατηρήστε τη μέθοδο **findTotal** () στην κλάση **Calculator**. Πόσες παραμέτρους δέχεται αυτή η μέθοδος;
- Σε ποιον ανήκει το κάθε σύνολο;
 - Τροποποιήστε τη μέθοδο **findTotal** () ώστε να δέχεται μία παράμετρο **name** τύπου **String** και παράλληλα να την εκτυπώνει στη έξοδο.
 - Τι μήνυμα εμφανίζεται και πως μπορεί να διορθωθεί;

Παραδείγματα

```
public void calculate0() {  
    System.out.println("No parameters");  
}
```

```
public void calculate1(int x) {  
    System.out.println(x/2.0);  
}
```

```
public void calculate2(int x, double y) {  
    System.out.println(x/y);  
}
```

```
public void calculate3(int x, double y, int z) {  
    System.out.println(x/y +z);  
}
```


Παραδείγματα

```
public void calculate0() {  
    System.out.println("No parameters");  
}
```

```
public void calculate1(int x) {  
    System.out.println(x/2.0);  
}
```

*calculate1(3)
→ 1.5*

```
public void calculate2(int x, double y) {  
    System.out.println(x/y);  
}
```

*calculate2(3, 2.0)
→ 1.5*

```
public void calculate3(int x, double y, int z) {  
    System.out.println(x/y + z);  
}
```

*calculate3(3, 2.0, 2)
→ 3.5*

Εμβέλεια (scope)

- Οι παράμετροι της μεθόδου είναι μεταβλητές που έχουν εμβέλεια μέσα στη μέθοδο {μπλοκ κώδικα μεθόδου}

```
public void calculate2(int x, double y){  
    System.out.println(x/y);  
}
```

*Εμβέλεια
του x*

```
public void calculate(){  
    System.out.println(x/2.0);  
}
```

*Εκτός εμβέλειας του x
//τι είναι το x;*

Μεταβλητές εκτός εμβέλειας

- Πως θα βγάλω το `total` έξω από τη μέθοδο;

```
public class Calculator {  
    public double tax = 0.05;  
    public double tip= 0.15;  
  
    public void findTotal(double price, String name) {  
        double total = price*(1+tax+tip);  
        System.out.println(name +": $" +total);  
    }  
}
```

```
public class CalculatorTest{  
    public static void main(String args[]){  
        Calculator calc = new Calculator();  
        System.out.println(calc.findTotal(10)+ calc.findTotal(12));  
    }  
}
```

Μεταβλητές εκτός εμβέλειας

- Πως θα βγάλω το `total` έξω από τη μέθοδο;

```
public class Calculator {  
    public double tax = 0.05;  
    public double tip= 0.15;  
  
    public void findTotal(double price, String name) {  
        double total = price*(1+tax+tip);  
        System.out.println(name +": $" +total);  
    }  
}
```

```
public class CalculatorTest{  
    public static void main(String args[]){  
        Calculator calc = new Calculator();  
        System.out.println(calc.findTotal(10)+ calc.findTotal(12));  
    }  
}
```

- Όμως: `'void' type not allowed here` //μήνυμα λάθους

Τύποι μεθόδου

- Οι μέθοδοι τύπου **void** δεν επιστρέφουν τιμές (δεν αποθηκεύονται τιμές μετά την κλήση μιας **void** μεθόδου).
- Οι μέθοδοι, όπως και οι μεταβλητές, μπορούν και επιστρέφουν τιμές διαφόρων τύπων π.χ. `byte`, `short`, `int`, `long`, `double`, `String`, `boolean`, κ.α.:
- Πως;
 - Δηλώστε τη μέθοδο να είναι κάποιου τύπου, και όχι **void**.
 - Χρησιμοποιήστε τη λέξη-κλειδί **return** μέσα σε μια μέθοδο, ακολουθούμενη από μια τιμή.

```
public String returnString() {  
    return "Hello";  
}
```

```
public int sum(int x, int y) {  
    return x + y;  
}
```

```
public boolean isGreater(int x, int y) {  
    return x > y;  
}
```

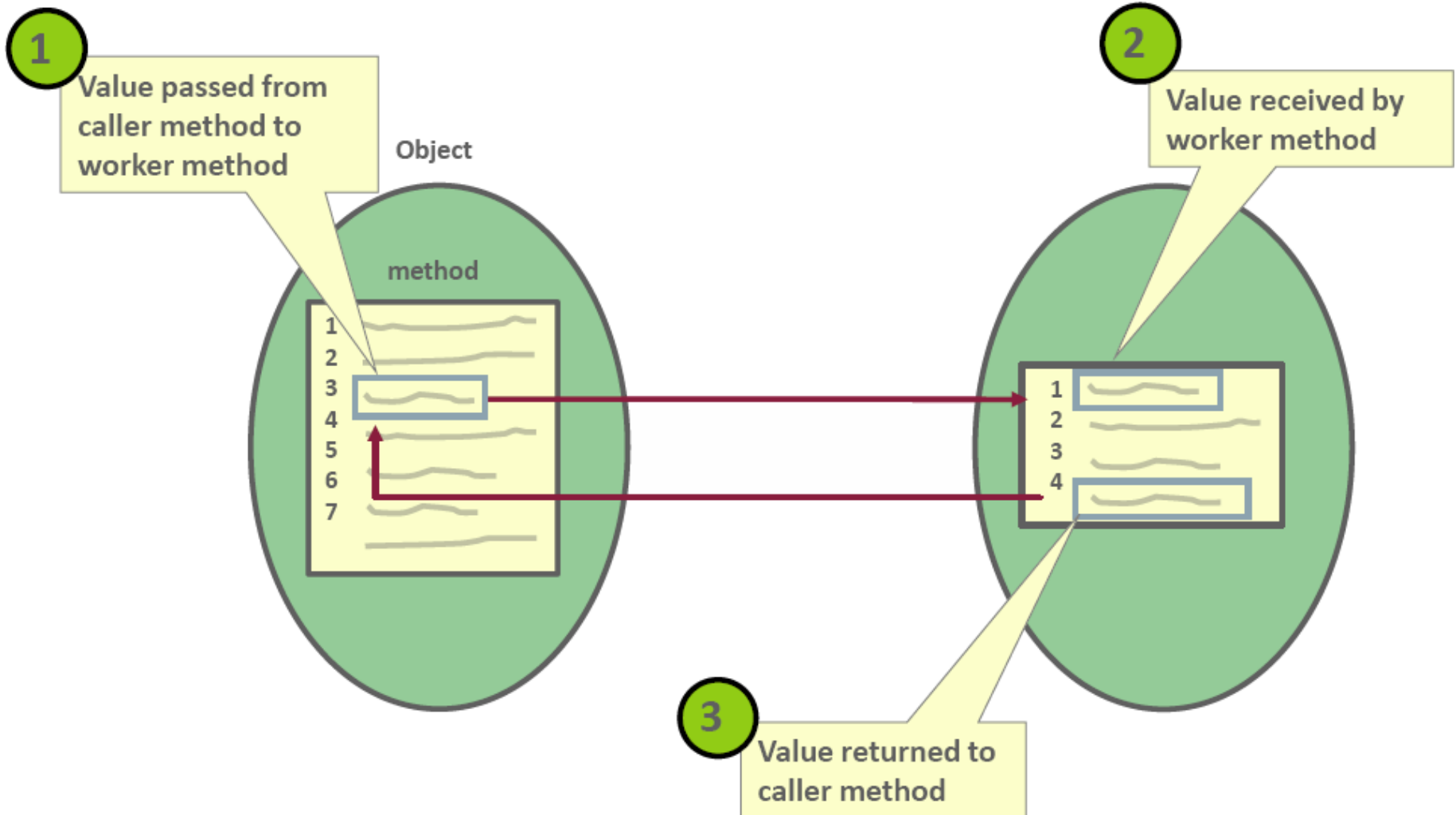
Χρήση παραμέτρων

- Τι διαφορά έχουν τα παρακάτω:

```
public static void main(String[] args){  
    int num1 = 1, num2 = 2;  
    int result = num1 + num2;  
    System.out.println(result);  
}
```

```
public class Athroisma {  
    public static int sum(int x, int y) {  
        return(x + y);  
    }  
  
    public static void main(String[] args) {  
        int num1 = 1, num2 = 2;  
        int result = sum(num1,num2);  
        System.out.println(result);  
    }  
}
```

Χρήση παραμέτρων



Άσκηση 4.4 : Alex Birthday 4

- Ανοίξτε το Tip03 ή το Tip04
- Βρείτε και εκτυπώστε το σύνολο όλου του πίνακα, συμπεριλαμβανομένου του φόρου και του φιλοδώρηματος.
 - Θα πρέπει να επεξεργαστείτε το `findTotal ()` ώστε να επιστρέφει την υπολογιζόμενη τιμή του.
- Ο `Person8` ξέχασε το πορτοφόλι του. Τροποποιήστε το `findTotal ()` ώστε το κόστος των γευμάτων να μοιράζεται εξίσου σε όλους (ο Alex είναι το τιμώμενο πρόσωπο!).
- Υπολογίστε πάλι το σύνολο όλου του πίνακα. Ο αριθμός αυτός δεν θα έπρεπε να είχε αλλάξει;

Συνοψίζοντας...

Method return type
Method name
Parameters
Implementation

```
public double calculate(int x, double y) {  
    double quotient = x/y;  
    return quotient;  
}
```

Βιβλιοθήκες Java

Βιβλιοθήκες Java

- Γιατί θα πρέπει να ανακαλύψουμε τον τροχό;
Αντί να ξαναγράψουμε τον ίδιο κώδικα Java για διάφορα προγράμματα...
...μπορούμε να χρησιμοποιήσουμε τη βιβλιοθήκη που παρέχεται από την Java, η οποία οργανώνει τον κώδικα που χρησιμοποιείται συχνά.
Αυτή η βιβλιοθήκη καλείται βιβλιοθήκη κλάσης Java.

Η τεκμηρίωση της βιβλιοθήκης:

<https://docs.oracle.com/javase/8/docs/api/>

Πακέτα στη βιβλιοθήκη κλάσεων Java

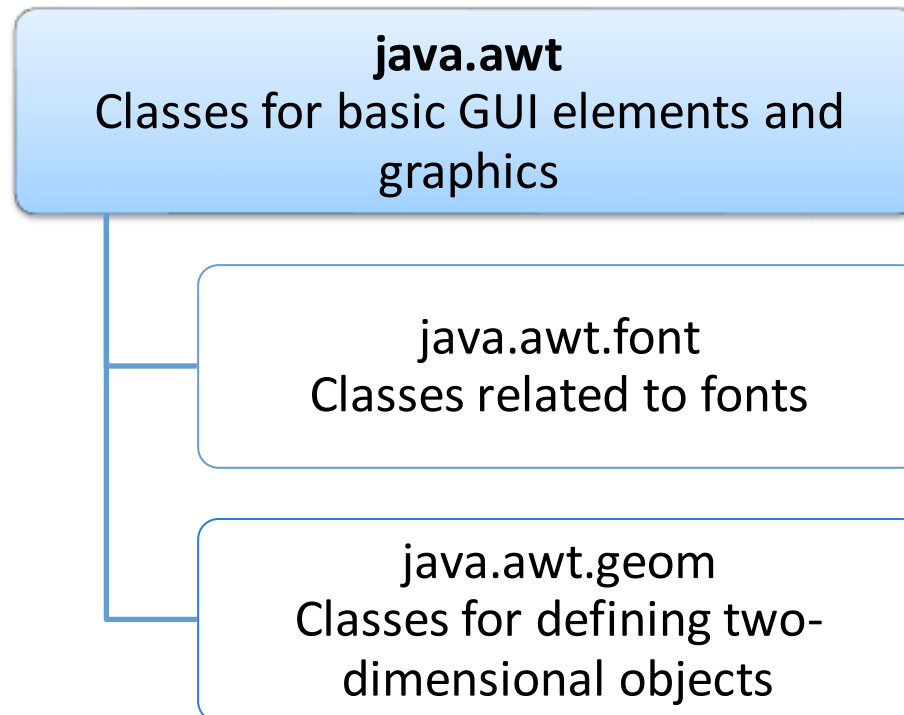
Οι κλάσεις της βιβλιοθήκης κλάσεων Java είναι οργανωμένες σε πακέτα.

Ένα πακέτο περιέχει μια ομάδα σχετικών κλάσεων.

Πακέτο (Package)	Σκοπός (Περιεχόμενο)
java.lang	Περιέχει κλάσεις που είναι θεμελιώδεις για το σχεδιασμό της γλώσσας Java
javax.swing	Περιέχει κλάσεις για την κατασκευή στοιχείων GUI
java.net	Περιέχει κλάσεις για εφαρμογές δικτύωσης
java.time	Περιέχει κλάσεις για ημερομηνίες, ώρες, στιγμιότυπα και διάρκεια

Πακέτα στη βιβλιοθήκη κλάσεων Java

Η τεράστια συλλογή των κλάσεων Java είναι οργανωμένη σε ιεραρχία δένδρου, η οποία επιτρέπει τα πακέτα (packages) να χωριστούν σε υποπακέτα (subpackages)



Χρήση κλάσης πακέτου

- Για πρόσβαση σε μία κλάση ενός πακέτου θα πρέπει να οριστεί το πλήρες όνομα:

– π.χ. `java.util.Scanner`
Package *Class Name*

Ωστόσο, έτσι δημιουργούνται μεγάλα ονόματα!

```
public static void main(String[] args) {  
    int num;  
    java.util.Scanner keyboard = new java.util.Scanner(System.in);  
    System.out.print("Enter a number");  
    num= keyboard.nextInt();  
    System.out.println("The number you entered is " + num);  
}
```

Χρήση *import*

- Μπορούμε να αποφύγουμε το πλήρες όνομα μίας κλάσης κατηγορίας με την εντολή ***import*** (***import*** package.className), π.χ.

Package *Class Name*

```
import java.util.Scanner;  
public class AddNums {  
    //class code goes here  
}
```

Πρόσβαση σε όλες τις κλάσεις ενός πακέτου

- Μπορείτε να εισαγάγετε όλες τις κλάσεις ενός πακέτου χρησιμοποιώντας τον χαρακτήρα “*” (wildcard) στην εντολή εισαγωγής import.

```
import java.util.Date;  
import java.util.Calendar;  
  
public class DisplayDate {  
    //class definition here  
}
```


Πρόσβαση σε όλες τις κλάσεις ενός πακέτου

- Μπορείτε να εισαγάγετε όλες τις κλάσεις ενός πακέτου χρησιμοποιώντας τον χαρακτήρα “*” (wildcard) στην εντολή εισαγωγής import.

```
import java.util.*;  
  
public class DisplayDate {  
    //class definition here  
}
```

java.lang

- Τα παραπάνω δεν ισχύουν για τις κλάσεις του πακέτου java.lang...

```
public class DisplayOutput {  
    public static void main(String[] args) {  
        System.out.println("Hello, how are you today?");  
    }  
}
```

...καθώς το java.lang εισάγεται **αυτόματα** σε όλα τα προγράμματα Java.

Η κλάση String

Strings

- Ακολουθίες χαρακτήρων (συμβολοσειρές)
 - “AaBb123,...?, etc.”
- Δεν αποτελούν μεταβλητές ενός βασικού τύπου μεταβλητών δεδομένων (primitive data type) όπως π.χ int, double, boolean, κ.α.
- Αλλά, αντικείμενα της κλάσης java.lang.String:
 - Π.χ. String s1= “Hello, World”;
- Τεκμηρίωση της κλάσης String:
 - <https://docs.oracle.com/javase/8/docs/api/>



Methods of the
String class

Τεκμηρίωση της κλάσης String

```
public int charAt(String str)
```

Name of the
method

Data type of the parameter that
must be passed into the method

Return type of
the method

Method Summary	
Methods	
Modifier and Type	Method and Description
char	charAt (int index) Returns the char value at the specified index.
int	codePointAt (int index) Returns the character (Unicode code point) at the specified index.
int	codePointBefore (int index) Returns the character (Unicode code point) before the specified index.
int	codePointCount (int beginIndex, int endIndex) Returns the number of Unicode code points in the specified text range of this String.
int	compareTo (String anotherString) Compares two strings lexicographically.
int	compareToIgnoreCase (String str) Compares two strings lexicographically, ignoring case differences.
String	concat (String str) Concatenates the specified string to the end of this string.

Μέθοδοι (χαρακτηριστικές) του αντικειμένου String

Μέθοδος	Λειτουργία
<code>charAt(int index)</code>	Επιστρέφει το χαρακτήρα στη θέση <code>index</code>
<code>compareTo(String anotherString)</code>	Συγκρίνει δύο strings. Επιστρέφει: < 0, όταν <code>string</code> που καλεί τη μέθοδο είναι λεξικογραφικά πρώτο. == 0, όταν τα 2 strings είναι λεξικογραφικά ίδια > 0, όταν η παράμετρος που περνάει στη μέθοδο είναι λεξικογραφικά πρώτη
<code>contains(CharSequence s)</code>	Επιστρέφει <code>true</code> μόνο εάν το <code>string</code> περιέχει τη σειρά χαρακτήρων <code>s</code>
<code>indexOf(int ch)</code>	Επιστρέφει τη θέση στην οποία θα συναντήσει για πρώτη φορά το χαρακτήρα <code>ch</code> στο <code>string</code>
<code>isEmpty()</code>	Επιστρέφει <code>true</code> , εάν το <code>string</code> είναι κενό (δηλαδή έχει μηδενικό μήκος)
<code>length()</code>	Επιστρέφει το μήκος ενός <code>string</code>
<code>replace(char oldChar, char newChar)</code>	Αντικαθιστά όλες τις εμφανίσεις του <code>oldChar</code> με <code>newChar</code> .
<code>replaceFirst(String pattern, String replacement)</code>	Αντικαθιστά μόνο την πρώτη εμφάνιση του <code>String pattern</code> στο <code>String</code>
<code>toLowerCase() / toUpperCase()</code>	Μετατρέπει όλους τους χαρακτήρες στο <code>string</code> σε <code>lowercase/uppercase</code>
<code>str.substring(int beginIdx)</code>	Επιστρέφει το <code>substring</code> από το <code>beginIdx</code> ως το τέλος του <code>str</code>
<code>str.substring(int beginIdx, int endIdx)</code>	Επιστρέφει το <code>substring</code> από το <code>beginIdx</code> ως το <code>endIdx</code>

Παραδείγματα

Κώδικας Java	Αποτέλεσμα
<pre>String name= "Mike.W"; System.out.println(name.length());</pre>	
<pre>String str= "Hello, World"; str.indexOf(char l)</pre>	
<pre>String str="Susan"; System.out.println(str.charAt(0)); System.out.println(str.charAt(3));</pre>	
<pre>String phoneNum= "404-543-2345"; int idx1 = phoneNum.indexOf('-'); System.out.println("index of first dash: "+ idx1); int idx2 = phoneNum.indexOf('-', idx1); System.out.println("second dash idx: "+idx2);</pre>	
<pre>String greeting = "Hello, World!"; String sub = greeting.substring(0, 5); String tail = greeting.substring(7);</pre>	

Παραδείγματα

Κώδικας Java	Αποτέλεσμα
<pre>String name= "Mike.W"; System.out.println(name.length());</pre>	6
<pre>String str= "Hello, World"; str.indexOf(char l)</pre>	2
<pre>String str="Susan"; System.out.println(str.charAt(0)); System.out.println(str.charAt(3));</pre>	S a
<pre>String phoneNum= "404-543-2345"; int idx1 = phoneNum.indexOf('-'); System.out.println("index of first dash: "+ idx1); int idx2 = phoneNum.indexOf('-', idx1); System.out.println("second dash idx: "+idx2);</pre>	index of first dash: 3 second dash idx: 7
<pre>String greeting = "Hello, World!"; String sub = greeting.substring(0, 5); String tail = greeting.substring(7);</pre>	Hello World!

Παραδείγματα

Κώδικας Java	Αποτέλεσμα
<pre>String str= "Using String replace to replace character"; String newString =str.replace("r", "R"); System.out.println(newString);</pre>	Using StRing Replace to Replace chaRacterR
<pre>String replace = "String replace with replaceFirst"; String newString = replace.replaceFirst("re", "RE"); System.out.println(newString);</pre>	String REplacewith replaceFirst
<pre>String s1 = "Susan"; String s2 = "Susan"; String s3 = "Robert"; System.out.println(s1.compareTo(s2)); System.out.println(s1.compareTo(s3)); System.out.println(s3.compareTo(s1));</pre>	<pre>0 // s1 είναι ίδιο με s2 1 // το 'S' ακολουθεί το 'R' -1 // το 'R' προηγείται του 'S'</pre>

StringBuffer

- Για να έχουμε τη δυνατότητα να τροποποιήσουμε ένα αλφαριθμητικό, χρησιμοποιούμε την κλάση StringBuffer.
 - Τα αντικείμενα τύπου StringBuffer μπορούν να περιέχουν αχρησιμοποίητο χώρο για χαρακτήρες (ενώ τα τύπου String δεν μπορούν).
 - `StringBuffer sb = new StringBuffer(10)` → κρατάει χώρο για 10 χαρακτήρες στο αντικείμενο sb.
- Οι μέθοδοι `length()` και `capacity()` μας δείχνουν το μήκος και τη χωρητικότητα ενός αντικειμένου τύπου StringBuffer.
- Μπορούμε να μετατρέψουμε ένα αντικείμενο StringBuffer σε String, με χρήση της μεθόδου `toString()`.
- Η StringBuffer παρέχει και τη δυνατότητα αντιστροφής χαρακτήρων με τη μέθοδο `reverse()`.

Άσκηση 4.5: StringsEx

- Ανοίξτε το ShoppingCart.java.
 - Χρησιμοποιήστε τη μέθοδο **indexOf** για να πάρετε τη θέση του χαρακτήρα διαστήματος (“_”) μέσα στο **custName**. Εκχωρήστε το στο **spaceIdx**.
 - Χρησιμοποιήστε τη μέθοδο **substring** και **spaceIdx** για να πάρετε το τμήμα του ονόματος του **custName**.
 - Εκχωρήστε το στο **firstName** και εκτυπώστε το.

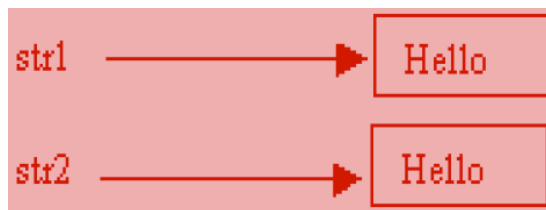
Τα Strings είναι αμετάβλητα

- Το String object είναι αμετάβλητο, δηλαδή, αφού δημιουργηθεί, η τιμή του δεν μπορεί να αλλάξει.
- Για αυτό και η Java μπορεί να τα επεξεργαστεί αποτελεσματικά.
- Για παράδειγμα

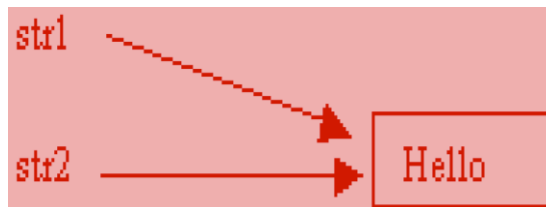
– Εάν

```
String str1 = "Hello";  
String str2 = "Hello";
```

– τότε



– ωστόσο



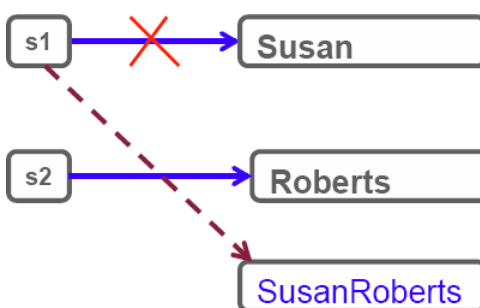
Το JRE γνωρίζει ότι τα δύο Strings είναι ίδια οπότε δεσμεύει την ίδια θέση μνήμης και για τα δύο αντικείμενα.

Συνένωση (Concatenating) Strings

- Μπορεί να γίνει με δύο τρόπους:

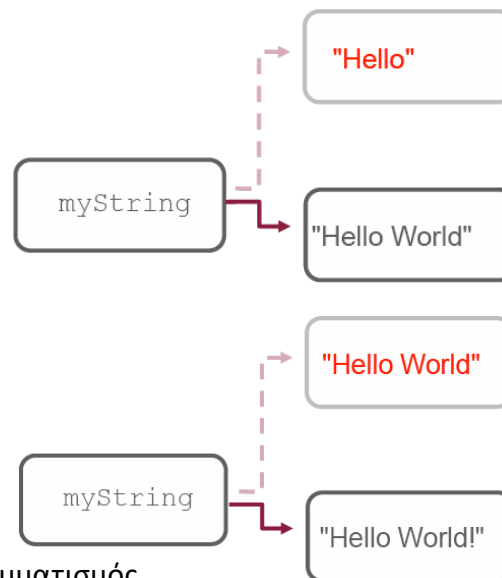
– με τον τελεστή “+”

```
String s1 = "Susan";
String s2 = "Roberts";
s1 = s1 + s2;
System.out.println(s1);
```



– με τη μέθοδο concat ()

```
String myString = "Hello";
myString = myString.concat("World");
...
myString = myString + "!"
```



Άσκηση 4.6: NameMaker

- Ανοίξτε το NameMaker.java.
 - Δηλώστε τις μεταβλητές String: firstName, middleName, lastName και fullName
 - Το πρόγραμμα θα πρέπει να ζητάει από τους χρήστες να εισάγουν το πρώτο, το μεσαίο και το επώνυμο τους...

```
Enter your first name: [firstName]
Enter your middle name: [middleName]
Enter your last name: [lastName]
```

- ...και να εκτυπώνει το αποτέλεσμα ως

```
Your full name is <firstName middleName lastName>
```

→ Τι πιστεύετε ότι είναι προτιμότερο για αυτό το σενάριο; Η μέθοδος `concat()` ή το άθροισμα strings `“+”`;

concat () Vs “+”

- Ο τελεστής “+”:
 - Μπορεί να λειτουργήσει μεταξύ ενός string αντικειμένου και των char, int, double ή float τύπων μεταβλητών.
 - Μετατρέπει την τιμή σε string πριν από την ένωση/άθροισμα.
- Η μέθοδος concat ():
 - Μπορεί να καλείται μόνο σε strings.
 - Ελέγχει για συμβατότητα των τύπων δεδομένων και αναπαράγει σφάλμα μεταγλώττισης (compile time error) αν δεν ταιριάζουν.

Η κλάση Random



Τυχαίοι Αριθμοί

- Η κλάση `java.util.Random` χρησιμοποιείται για την παραγωγή τυχαίων αριθμών για διάφορες εφαρμογές π.χ. μοίρασμα τράπουλας, λοταρία, κ.α.
- Περιέχει μεθόδους που επιστρέφουν τυχαίους αριθμούς τύπου `double`, `boolean`, `float`, `long`.



Method	Output
<code>boolean nextBoolean();</code>	True/false τιμή
<code>int nextInt()</code>	Integer τιμή μεταξύ <code>Integer.MIN_VALUE</code> και <code>Integer.MAX_VALUE</code>
<code>long nextLong()</code>	Long τιμή μεταξύ <code>Long.MIN_VALUE</code> και <code>Long.MAX_VALUE</code>
<code>float nextFloat()</code>	Float number ≥ 0.0 και < 1.0
<code>double nextDouble()</code>	Double number ≥ 0.0 και < 1.0

Παραδείγματα

```
import java.util.Random;
public class RandomNum{
    public static void main(String[] args) {
        Random rndNum= new Random();
        int randomNumber= rndNum.nextInt();
        System.out.println("Random Number: " + randomNumber);
    }
}
```

Output

Random Number: 1660093261

Εκτυπώνει έναν τυχαίο
ακέραιο αριθμό κάθε φορά

```
import java.util.Random;
public class RandomNum{
    public static void main(String[] args) {
        Random num= new Random();
        System.out.println("Random Number 1: "+num.nextInt());
        System.out.println("Random Number 2: "+num.nextInt());
        System.out.println("Random Number 3: "+num.nextInt());
        System.out.println("Random Number 4: "+num.nextInt());
        System.out.println("Random Number 5: "+num.nextInt());
    }
}
```

Output

Random Number 1: 1814918663
Random Number 2: -944814285
Random Number 3: 767298538
Random Number 4: 762007235
Random Number 5: 1220127792

Εκτυπώνει μία ακολουθία
τυχαίων ακεραιών αριθμών κάθε
φορά

Παραδείγματα

```
import java.util.Random;  
public class RandomNum {  
    public static void main(String[] args) {  
        Random num = new Random();  
        double randomDouble = num.nextDouble();  
        System.out.println("Random Number: " + randomDouble);  
    }  
}
```



Random Number: 0.8502641005640065



Εκτυπώνει έναν τυχαίο
πραγματικό αριθμό κάθε φορά

Παραδείγματα

```
import java.util.Random;  
public class RandomNum{  
    public static void main(String[] args) {
```

```
        Random rand1 = new Random();  
        int randomnum= rand1.nextInt(20);  
        System.out.println("Random Number: " + randomnum);
```

Random Number: 17

Εκτυπώνει έναν τυχαίο ακέραιο
αριθμό από 0 έως 19

```
        Random rand2 = new Random();  
        randomnum = rand2.nextInt(40)+1;  
        System.out.println("Random Number: " + randomnum);
```

Random Number: 29

Εκτυπώνει έναν τυχαίο ακέραιο
αριθμό από 1 έως 40

```
        Random rand3 = new Random();  
        randomnum = rand3.nextInt(31)+5;  
        System.out.println("Random Number: " + randomnum);
```

Random Number: 24

Εκτυπώνει έναν τυχαίο ακέραιο
αριθμό από 5 έως 35

```
    }  
}
```

Εφαρμογή Λοταρίας



```
import java.util.*;
public class Lottery{
    public static void main(String[] args) {
        Scanner numberScanner= new Scanner(System.in);
        System.out.print("Enter a number between 1 and 10: ");
        int userNum= numberScanner.nextInt();
        Random rnd= new Random();
        int winningNum= rnd.nextInt(10) + 1;
        System.out.println("Your Number: "+ userNum);
        System.out.println("The winning number is:"+ winningNum);
    }
}
```

→ Τι πρέπει να αλλάξω για να αυξήσω τις πιθανότητες να κερδίσω;-)

Άσκηση 4.7: FlipCoin

- Ανοίξτε το FlipCoin.java.
- Παρατηρήστε τον τυχαίο αριθμό που παράγεται κάθε φορά:
 - Εάν <0.5 , καταγράψτε το αποτέλεσμα ως «κεφάλι»
 - αλλιώς καταγράψτε το αποτέλεσμα ως «γράμματα»
- Επαναλάβετε πολλές φορές.

Άσκηση 4.8: RockPaperScissor

- Ανοίξτε το RockPaperScissor.java και κάντε προσομοίωση του παιχνιδιού «Πέτρα-ψαλίδι-χαρτί», ως εξής:
 - Δημιουργείστε έναν τυχαίο ακέραιο αριθμό από 0 έως 3.
 - Συγκρίνετε τον παραγόμενο αριθμό με τους ακόλουθους:
 - Εάν αριθμός = 0 τότε «πέτρα»
 - Εάν αριθμός = 1 τότε «χαρτί»
 - Εάν αριθμός = 2 τότε «ψαλίδι»
 - Καταγράψτε το αποτέλεσμα και επαναλάβετε πολλές φορές

Γεννήτρια τυχαίων αριθμών

- Μερικές φορές χρειάζεται να δημιουργηθεί η ίδια ακολουθία τυχαίων αριθμών κάθε φορά, χρησιμοποιώντας μια αρχική σταθερή τιμή που ονομάζεται γεννήτρια.
- Όταν δημιουργείτε ένα αντικείμενο της κλάσης `Random`, περάστε έναν σταθερό ακέραιο για να καθορίσετε τη γεννήτρια:
 - `Random rndNumbers = new Random(20L);`
Γεννήτρια (*seed*)
- Με τη μέθοδο `setSeed()`, μπορείτε να αλλάξετε τη γεννήτρια.
- Ωστόσο, κάθε φορά που ορίζετε την ίδια γεννήτρια, επιστρέφεται η ίδια τυχαία ακολουθία.

Παράδειγμα

```
public static void main(String[] args) {  
    Random rand = new Random(20L);  
    System.out.println("Random Number 1: " + rand.nextInt(100));  
    System.out.println("Random Number 2: " + rand.nextInt(100));  
    System.out.println("Random Number 3: " + rand.nextInt(100));  
    System.out.println("Changing seed to change to sequence");  
    rand.setSeed(5L);  
    System.out.println("Random Number 4: " + rand.nextInt(100));  
    System.out.println("Random Number 5: " + rand.nextInt(100));  
    System.out.println("Random Number 6: " + rand.nextInt(100));  
    System.out.println("Setting seed 20 produce previous  
sequence");  
    rand.setSeed(20L);  
    System.out.println("Random Number 7: " + rand.nextInt(100));  
    System.out.println("Random Number 8: " + rand.nextInt(100));  
    System.out.println("Random Number 9: " + rand.nextInt(100));  
}
```

Random Number 1: 53
Random Number 2: 36
Random Number 3: 1
Changing seed to change to sequence
Random Number 4: 87
Random Number 5: 92
Random Number 6: 74
Setting seed 20 produce previous sequence
Random Number 7: 53
Random Number 8: 36
Random Number 9: 1

Η κλάση Math

π

java.lang.Math

- Η κλάση Math περιέχει μεθόδους για την εκτέλεση μαθηματικών συναρτήσεων και πράξεων π.χ. μέγιστο/ελάχιστο δύο τιμών, στρογγυλοποίηση, λογαριθμικές/τριγωνομετρικές πράξεις, τετραγωνική ρίζα, κ.α.
- Τεκμηρίωση της κλάσης Math: Περιλαμβάνεται στο πακέτο java.lang
- <http://docs.oracle.com/javase/8/docs/api/index.html>

Ενδεικτικές Μέθοδοι της κλάσης Math

Method Name	Description
<code>abs(value)</code>	απόλυτη τιμή
<code>ceil(value)</code>	στρογγυλοποίηση (προς τα πάνω)
<code>cos(value)</code>	συνημίτονο (σε rads)
<code>floor(value)</code>	στρογγυλοποίηση (προς τα κάτω)
<code>log(value)</code>	λογάριθμος
<code>log10(value)</code>	δεκαδικός λογάριθμος
<code>max(value1, value2)</code>	μεγαλύτερη τιμή από δύο τιμές
<code>min(value1, value2)</code>	μικρότερη τιμή από δύο τιμές
<code>pow(base, exponent)</code>	εκθετική (exponent) τιμή της βάσης (base)
<code>random()</code>	πραγματικός τυχαίος αριθμός μεταξύ 0 και 1
<code>round(value)</code>	πλησιέστερος ακέραιος αριθμός
<code>sin(value)</code>	ημίτονο (σε rads)
<code>sqrt(value)</code>	τετραγωνική ρίζα

Μέθοδοι της κλάσης Math

- Οι μέθοδοι της κλάσης Math είναι static
 - Μπορούν να χρησιμοποιηθούν μέσω του ονόματος της κλάσης.
 - Δεν χρειάζεται να δημιουργηθεί αντικείμενο του Math class, πρώτα, όπως κάναμε στην Random.
 - Σύνταξη: Math.methodName(parameters)

```
Math.sqrt(121.0);           // δεν χρειάζεται να δημιουργηθεί  
                             // αντικείμενο Math πρώτα
```

```
Random rndNum= new Random(); // πρέπει να δημιουργηθεί  
int randomNumber = rndNum.nextInt(); // αντικείμενο Random πρώτα
```

Μέθοδοι της κλάσης Math

- Υπολογίζουν κάποιο αριθμητικό αποτέλεσμα αλλά δεν το εκτυπώνουν:

```
public static void main(String[] args) {  
    Math.sqrt(121.0);  
}
```

//δεν εκτυπώνει κάτι

- ...συνεπώς:

```
public static void main(String[] args) {  
    System.out.println("Square root: "+ Math.sqrt(121.0));  
}
```

//εκτυπώνει 11.0

- ..ή

```
public static void main(String[] args) {  
    double sqrt = Math.sqrt(121.0);  
    System.out.println("Square root: "+sqrt);  
}
```

// αποθηκεύει και
//εκτυπώνει 11.0

Τι εκτυπώνει το παρακάτω;

```
public class Calculate{  
    public static void main(String[] args) {  
        double result = Math.min(3, 7) + Math.abs(-50);  
        System.out.println("Result is " + result);  
    }  
}
```

Τι εκτυπώνει το παρακάτω;

```
public class Calculate{  
    public static void main(String[] args) {  
        double result = Math.min(3, 7) + Math.abs(-50);  
        System.out.println("Result is " + result);  
    }  
}
```

Result is 53.0

Τι υπολογίζουν τα παρακάτω;

- `Math.abs (-1.23)`
- `Math.pow (3, 2)`
- `Math.sqrt (121.0)`
- `Math.sqrt (256.0)`
- `Math.abs (Math.min (-3, -5))`

Τι υπολογίζουν τα παρακάτω;

- `Math.abs (-1.23)` 1.23
- `Math.pow (3, 2)` 9.0
- `Math.sqrt (121.0)` 11.0
- `Math.sqrt (256.0)` 16.0
- `Math.abs (Math.min (-3, -5))` 5.0

Άσκηση 4.10: Math Methods

- Εξετάστε την τεκμηρίωση της κλάσης Math:
 - <http://docs.oracle.com/javase/8/docs/api/>
- Μπορείτε να εκτυπώσετε την τιμή του π ;
- Μπορείτε να υπολογίσετε την τετραγωνική ρίζα ενός αριθμού;

Άσκηση 4.11: Age Limit

- Θεωρείστε μια ακέραια μεταβλητή **age**.
- Χρησιμοποιήστε τις μεθόδους **Math.max** και **Math.min** για να απαντήσετε στις ακόλουθες ερωτήσεις:
 - Πως θα αντικαταστήσετε τις αρνητικές ηλικίες με 0;
 - Πως θα περιορίσετε τη μέγιστη ηλικία σε 40;

Σταθερές

- Η κλάση Math περιέχει δύο σταθερές:

Field	Description
Math.E	2.7182818...
Math.PI	3.1415926...

- Το PI σε ένα πρόγραμμα, χρησιμοποιείται ως εξής:
 - Math.PI

Υπολογισμός επιφάνειας κύκλου

```
import java.util.Scanner;  
public class AreaOfCircle {  
    public static void main(String args[]) {  
        Scanner sc = new Scanner(System.in);  
        System.out.print("Enter the radius: ");  
        double radius = sc.nextDouble();  
        double area = Math.PI * radius * radius;  
        System.out.println("The area of circle is: " + area);  
    }  
}
```

Σε αυτήν την ενότητα μιλήσαμε για:

1. Πως δημιουργείται/δομείται μία κλάση
2. Τις Βιβλιοθήκες Java και πως τις χρησιμοποιούμε
3. Περιγραφή των κλάσεων String, Random και Math και των μεθόδων τους

Βιβλιογραφία

- Harvey Deitel, Paul Deitel, «Java Προγραμματισμός, 8η έκδοση», Γκιούρδας, 2010.
- Γιώργος Λιακέας, «Εισαγωγή στη Java», Κλειδάριθμος, 2009.
- Χρήστος Κυτάγιας, Κώστας Δ. Κυτάγιας, Γιώργος Πρεζεράκος, Δημήτρης Κυτάγιας, «Αντικειμενοστραφής Προγραμματισμός με Java», Σύγχρονη Εκδοτική, 2013.
- Wanda Dann, Stephen Cooper, and Randy Pausch, «Learning to program with Alice» 3rd edition, Pearson Education, 2012.
- Michael Kölling, «Introduction to Programming with Greenfoot».
- «Object-Oriented Programming in Java with Games and Simulations», Pearson Education, August 2009.
- Οι σημειώσεις είναι βασισμένες και σε υλικό του καθηγητή του ΤΕΙ Πειραιά Γ. Πρεζεράκου.
- Oracle Academy/Java Foundations/Lesson4 - Java Methods and Library Classes
<http://academy.oracle.com>



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

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

