



ΑΤΕΙ ΘΕΣΣΑΛΟΝΙΚΗΣ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ
Προγραμματισμός Η/Υ-Ι



Παναγιώτης Σφέτσος
sfetsos@it.teithe.gr

Περισσότερα για τον έλεγχο προσπέλασης και τις μεθόδους Static

Πρώτη δήλωση στην υπογραφή (*signature*) μιας μεθόδου είναι ο προαιρετικός καθοριστής πρόσβασης (*optional access specifier*), που μπορεί να είναι **γενικός** (*public*), **ιδιωτικός** (*private*) ή **προστατευόμενος** (*protected*). Μπορούμε να καλέσουμε μία γενική μέθοδο από παντού, ενώ μία ιδιωτική μέθοδο μόνο από την κλάση που προσδιορίζεται. Μία προστατευμένη μέθοδος μπορεί να χρησιμοποιηθεί μόνο εντός του πακέτου που προσδιορίζεται. Όταν η μέθοδος δεν προσδιορίζεται ως γενική ή ιδιωτική, τότε θεωρείται ως προστατευόμενη.

Ιδιωτικό μέλος – καλή επιλογή, για την μη πρόσβαση από άλλη κλάση.

Κανένας προσδιοριστής – πρόσβαση μόνο μέσα από το ίδιο πακέτο κλάσεων (*package*) – κατάλογο (*directory*).

Προστατευόμενο μέλος – χρησιμοποιείται όταν θέλουμε να έχουμε πρόσβαση από κάποια υποκλάση.

Γενικό μέλος – όταν θέλουμε να προσπελαθεί και από άλλες κλάσεις, ειδικά στις μεθόδους.

Παράδειγμα

Στο παρακάτω παράδειγμα θα γράψουμε την γνωστή μέθοδο μετατροπής χιλ. σε μίλια με τον ιδιωτικό προσδιοριστή, ώστε η πρόσβαση να γίνεται μόνο από την ίδια κλάση.

```
class KmToMilesMethod {  
    private static final double MILES_PER_KILOMETER = 0.621;  
  
    public static void main(String[] args) {  
        String kmStr;
```

```

double km =320.05;
double mi;

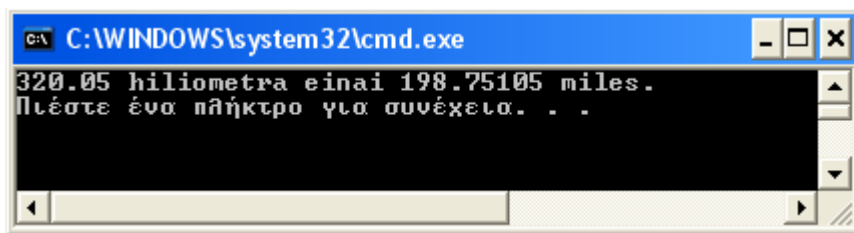
//... ypologismos
mi = convertKmToMi(km);

//... emfanisi apotelesmatos
System.out.println(km + " hiliometra einai " + mi + " miles.");
}

//metatropi se milia
private static double convertKmToMi(double kilometers) {
    double miles = kilometers * MILES_PER_KILOMETER;
    return miles;
}}

```

Το αποτέλεσμα:



Προσοχή!!

Η μέθοδος **convertKmToMi(double kilometers)** δεν μπορεί πλέον να προσπελαθεί από άλλη κλάση. Δηλαδή, αν τροποποιήσουμε τον κώδικα του παραδείγματος σύμφωνα με την παρακάτω παραλλαγή:

```

class KmToMilesMethod1 {

private static final double MILES_PER_KILOMETER = 0.621;

//metatropi se milia
private static double convertKmToMi(double kilometers) {
    double miles = kilometers * MILES_PER_KILOMETER;
    return miles; }}

```

```

class TestKmToMilesMethod1 {

    public static void main(String[] args) {
        String kmStr;
        double km =320.05;
        double mi;

        //... υπολογισμος
        mi = KmToMilesMethod1.convertKmToMi(km);

        //... emfanisi apotelesmatos
        System.out.println(km + " hiliometra einai " + mi + " miles.");
    }
}

```

Θα πάρουμε το παρακάτω αποτέλεσμα που σημαίνει ότι δεν υπάρχει πρόσβαση στην μέθοδο `convertKmToMi(double kilometers)`, από άλλη κλάση.

```

C:\ > Γραμμή εντολών
C:\JAVΑ_YLH\WORK\LESSON_11_EPANALIPSI>javac TestKmToMilesMethod1.java
TestKmToMilesMethod1.java:22: convertKmToMi(double) has private access in KmToMilesMethod1
    mi = KmToMilesMethod1.convertKmToMi(km);
                           ^
1 error
C:\JAVΑ_YLH\WORK\LESSON_11_EPANALIPSI>

```

Περισσότερα για τις Στατικές Μεθόδους ή μεθόδους κλάσεων

- Εξ' ορισμού οι μέθοδοι δεν είναι στατικές, δηλαδή δεν είναι μέθοδοι κλάσεων, αλλά μέθοδοι αντικειμένων. Πρέπει να οριστούν `static`.
- Όταν ορίζεται μία στατική μέθοδος σε μία κλάση, τότε όλα τα αντικείμενα της κλάσης μοιράζονται ένα αντίγραφο της στατικής μεθόδου.
- Αν η στατική μέθοδος βρίσκεται σε άλλη κλάση τότε καλείται με το όνομα της κλάσης. Αν βρίσκεται στην ίδια κλάση τότε καλείται απ' ευθείας.

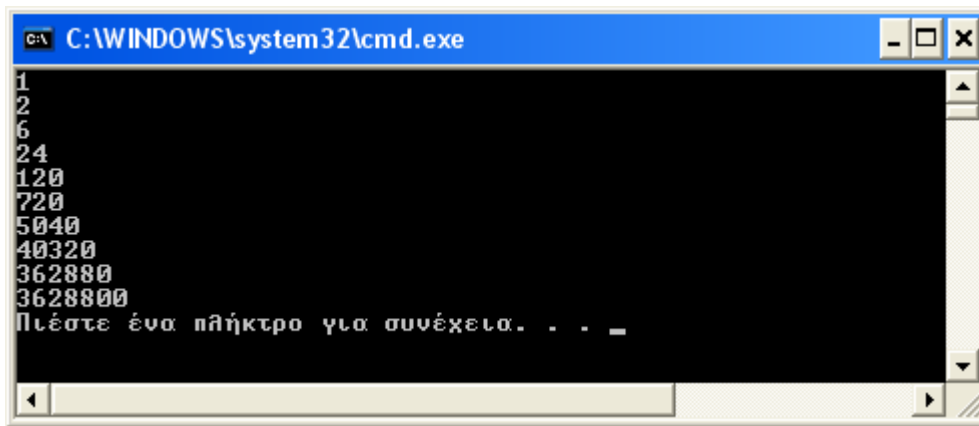
1° Παράδειγμα

Στο παρακάτω παράδειγμα θα φτιάξουμε μία στατική μέθοδο που θα υπολογίζει την παράγωγο ενός ακεραίου. Προσέξτε την κλήση της στατικής μεθόδου **factorial (int n)**.

```
class FactorialTest
{
static long factorial (int n)
{
    int i;
    long result=1;
    for (i=1; i <= n; i++)
    {
        result *= i;
    } return result; }

public static void main(String args[])
{
    int n;
    int i;
    long result;
for (i=1; i <=10; i++)
    {
        result = factorial(i);
        System.out.println(result);
    }}}
```

Το αποτέλεσμα:



```
C:\WINDOWS\system32\cmd.exe
1
2
6
24
120
720
5040
40320
362880
3628800
Πιέστε ένα πλήκτρο για συνέχεια. . . .
```

2° Παράδειγμα

Στο παράδειγμα αυτό θα λύσουμε την άσκηση της απλής μισθοδοσίας με δύο παραλλαγές. Στην πρώτη παραλλαγή ο ακαθάριστος και ο καθαρός μισθός θα υπολογίζονται σε μεθόδους αντικειμένων, ενώ στην δεύτερη παραλλαγή σε στατικές μεθόδους – μεθόδους κλάσεων. Προσέξτε τον διαφορετικό τρόπο κλήσεως των μεθόδων στις δύο παραλλαγές. Σας υπενθυμίζω ότι ο ακαθάριστος και ο καθαρός μισθός υπολογίζονται με τις σχέσεις:

AKM = Μέρες Εργασίας x Ημερομίσθιο + 0.2 x Ημερομίσθιο x Υπερωρίες
KM = AKM – (AKM X Συντ_Κρατήσεων)

A-Παραλλαγή (μέθοδοι αντικειμένων):

```
class Employee {
```

```
    String name;
```

```
    int meres;
```

```
    double hm;
```

```
    double yper;
```

```
    double synt;
```

```
    Employee(String name, int meres, double hm, double yper, double synt) {
```

```
        this.name=name;
```

```
        this.meres=meres;
```

```
        this.hm=hm;
```

```
        this.yper=yper;
```

```

    this.synt=synt;
}
public double ak_misth(int m,double h,double w)
{
    double ak_m=m*h+0.2*h*w;
    return ak_m;
}

public double kath_misth (double ak_m, double s)
{
    double k_m=ak_m-(ak_m*s);
    return k_m;
}}

class misthos2
{
public static void main(String args [])
{
    Employee Ypal=new Employee("Nikas Nikos", 25, 60.5, 10.5, 0.25);
    double akm=Ypal.ak_misth (25, 60.5, 10.5);
    double km=Ypal.kath_misth (akm, 0.25);
    System.out.println ("Onom/mo : " +Ypal.name);
    System.out.println ("Akatharistos Mishos= " + akm);
    System.out.println ("Katharos Misthos= " + km);
}}

```

Το αποτέλεσμα:

```

C:\WINDOWS\system32\cmd.exe
Onom/mo : Nikas Nikos
Akatharistos Mishos= 1639.55
Katharos Misthos= 1229.6625
Πιέστε ένα πλήκτρο για συνέχεια. . . _

```

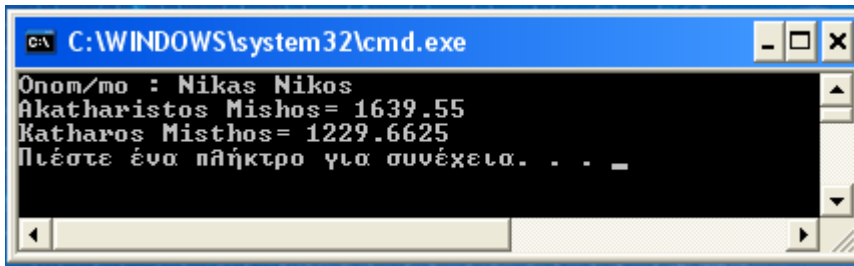
B-Παραλλαγή (στατικές μέθοδοι – μέθοδοι κλάσεων):

```
class Employee
{
public static double ak_misth(int m,double h,double w)
    {
        double ak_m=m*h+0.2*h*w;
        return ak_m;
    }

public static double kath_misth (double ak_m, double s)
    {
        double k_m=ak_m-(ak_m*s);
        return k_m;
    }
}

class TestMisthos
{
public static void main(String args [])
    {
        String name="Nikas Nikos";
        int Meres=25;
        double Hmer=60.5;
        double Yper=10.5;
        double Synt=0.25;
        double akm = Employee.ak_misth (Meres, Hmer, Yper);
        double km = Employee.kath_misth (akm, Synt);
        System.out.println ("Onom/mo : " + name);
        System.out.println ("Akatharistos Mishos= " + akm);
        System.out.println ("Katharos Misthos= " + km);
    }
}
```

Το αποτέλεσμα θα είναι το ίδιο με την προηγούμενη παραλλαγή:



```
C:\WINDOWS\system32\cmd.exe
Όνομ/νο : Nikas Nikos
Akatharistos Mishos= 1639.55
Katharos Misthos= 1229.6625
Πιέστε ένα πλήκτρο για συνέχεια. . . _
```

Προσοχή!!

Στην παραλλαγή αυτή δεν χρησιμοποιείται αντικείμενο. Έτσι η είσοδος των στοιχείων του υπάλληλου γίνεται με απλές μεταβλητές και όχι με αρχικοποίηση μέσω δομητή.