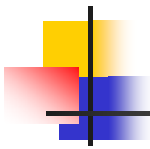


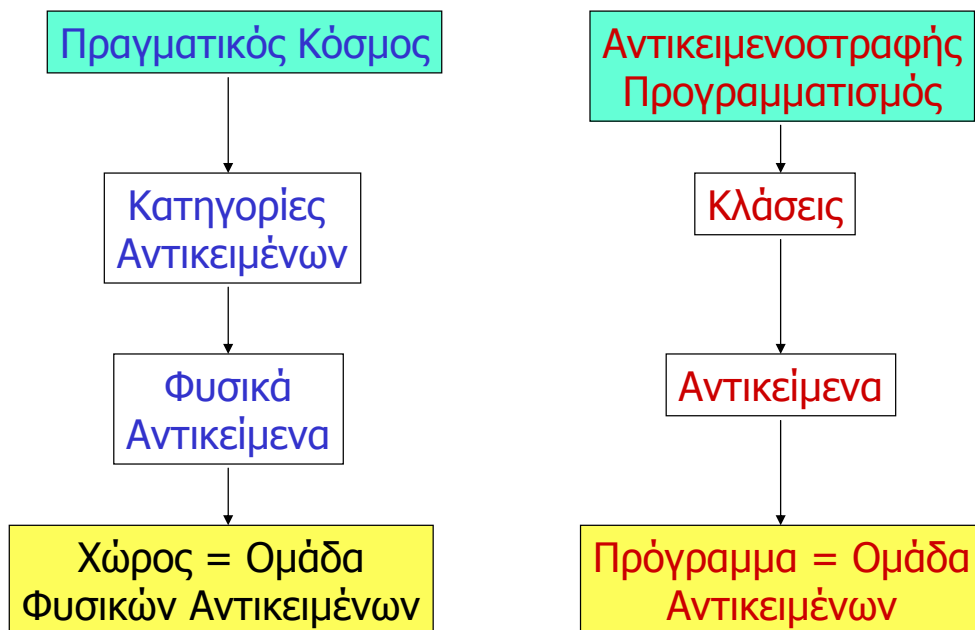
Κλάσεις

- Αντικειμενοστραφής Προγραμματισμός
- Κλάσεις – Αντικείμενα
- Ιεραρχία κλάσεων
- Κλάσεις. Ιδιότητες – Συμπεριφορά
- Ιδιότητες (Μεταβλητές)
- Συμπεριφορά (Μέθοδοι)
- Κληρονομικότητα
- Μέθοδοι επικάλυψης
- Η χρήση του this και του super

Κλάσεις



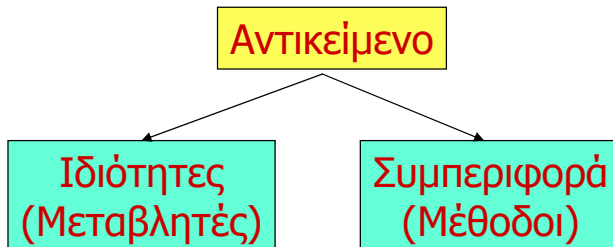
Αντικειμενοστραφής Προγραμματισμός



Κλάσεις

Κλάσεις – Αντικείμενα

- Στον αντικειμενοστραφή προγραμματισμό δημιουργούμε τις **κλάσεις**.
- Οι κλάσεις χρησιμοποιούνται ως **πρότυπα** για την δημιουργία των **αντικειμένων**.

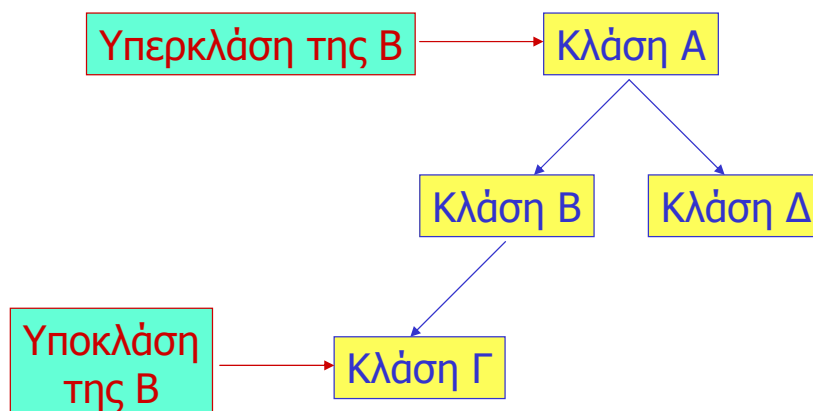


- **Ιδιότητες** : Περιγράφουν το αντικείμενο και δείχνουν σε τι διαφέρει σε σχέση με τα άλλα αντικείμενα.
- **Συμπεριφορά** : Είναι τα όσα κάνει ένα αντικείμενο.

Κλάσεις

Ιεραρχία κλάσεων

- Οι κλάσεις οργανώνονται σε ιεραρχίες κλάσεων με μορφή πυραμίδας.
- Με αυτόν τον τρόπο επιτυγχάνεται η **κληρονομικότητα (inheritance)** στον αντικειμενοστραφή προγραμματισμό.



Κλάσεις

Κλάσεις – Αντικείμενα

- Γενικό διάγραμμα κλάσης:

```
public class ΟνομαΚλασης {
```

```
int a=10; double b;  
public String name="Panos";  
private double x, y;
```

Μεταβλητές
(Ιδιότητες)

```
public method_1 {  
    .....  
}  
.....  
public method_n {  
    .....  
}
```

Μέθοδοι
(Συμπεριφορά)

```
}
```

Κλάσεις

Κλάσεις – Αντικείμενα

- Παράδειγμα δημιουργίας αντικειμένου με το **new**:

```
ΟνομαΚλασης ονομαΑντικειμένου = new ΟνομαΚλασης();
```

- Η πρόσβαση των κλάσεων καθορίζεται από τα παρακάτω:
 - **public** : Είναι προσβάσιμη από κάθε άλλη κλάση.
 - **final** : Δεν μπορούν να ορισθούν υποκλάσεις της.

Κλάσεις

Ιδιότητες (Μεταβλητές)

- Οι **ιδιότητες** ενός αντικειμένου αντιπροσωπεύουν τις **μεταβλητές** που χρειάζονται για να λειτουργήσει ένα αντικείμενο.
- Πρόσβαση μεταβλητών:
 - **public** : Είναι προσβάσιμη από κάθε άλλη κλάση.
 - **protected** : Είναι προσβάσιμη μόνο μέσα από τη δική της κλάση και από τις υποκλάσεις της.
 - **private** : Είναι προσβάσιμη μόνο μέσα από τη δική της κλάση
 - **final** : Είναι σταθερά η οποία αρχικοποιείται μόνο αρχικά και δεν αλλάζει τιμή.
 - **static** : Κοινή μεταβλητή για όλα τα αντικείμενα της κλάσης (Μεταβλητή κλάσης).
 - Έλλειψη δήλωσης : Χρησιμοποίηση μόνο μέσα από τη δική της κλάση και από κλάσεις στο ίδιο πακέτο.

Κλάσεις

Ιδιότητες (Μεταβλητές)

- Τρόπος πρόσβασης μεταβλητής ενός αντικειμένου.

```
public class TestClass{  
    public int a=10;  
    public String b = "test!";  
}
```

Απλή κλάση

```
TestClass x = new TestClass();
```

Δημιουργία Αντικειμένου x

x.a → Ο ακέραιος 10

x.b → Η συμβολοσειρά test

Κλάσεις

Συμπεριφορά (Μέθοδοι)

- Η **συμπεριφορά** περιγράφει τα διάφορα τμήματα μιας κλάσης που εκτελούν συγκεκριμένες εργασίες. Καθένα από αυτά τα τμήματα ονομάζεται **μέθοδος**.
- Πρόσβαση μεθόδων
 - **public** : Είναι προσβάσιμη από κάθε άλλη κλάση.
 - **protected** : Είναι προσβάσιμη μόνο μέσα από τη δική της κλάση και από τις υποκλάσεις της.
 - **private** : Είναι προσβάσιμη μόνο μέσα από τη δική της κλάση
 - **final** : Δεν μπορεί να επικαλυφθεί (override) από κάποια υποκλάση.
 - **static** : Ανήκει στην κλάση κατ' ευθείαν.
 - **native** : Το σώμα της εκτελείται σε άλλη γλώσσα προγραμματισμού.

Κλάσεις

Συμπεριφορά (Μέθοδοι)

- Τρόπος πρόσβασης μιας μεθόδου ενός αντικειμένου.

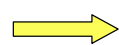
```
public class TestClass {  
    .....  
    public void printName(String name) {  
        System.out.println(name);  
    }  
    .....  
}
```

Απλή Κλάση

```
TestClass x = new TestClass();
```

Δημιουργία
Αντικειμένου x

```
x.printName("Panos");
```



Εκτύπωση του *Panos* μέσω της μεθόδου `printName()` του αντικειμένου `x`.

Κλάσεις



Δημιουργία τυπικής κλάσης

```
public class ΟνομαΚλασης {
```

Δήλωση Ιδιοτήτων (μεταβλητών)

Μέθοδοι Κατασκευής (Constructors).

Μέθοδοι πρόσβασης (Accessors).

Μέθοδοι ελέγχου ισότητας αντικειμένων (equals).

Άλλες μέθοδοι

Μέθοδος εκτύπωσης αντικειμένου (toString()).

```
}
```

Κλάσεις



Συμπεριφορά (Μέθοδοι)

- Βασικές κατηγορίες μεθόδων για την δημιουργία κλάσεων:
 - **Μέθοδοι Κατασκευής (Constructors).**
 - Η δημιουργία ενός αντικειμένου γίνεται με την χρήση της δήλωσης **new**.
Παράδειγμα: `TestClass x = new TestClass();`
 - Η δήλωση **new** καλεί μια ειδική μέθοδο της κλάσης του αντικειμένου η οποία ονομάζεται **μέθοδος κατασκευής** γιατί αναλαμβάνει την εργασία που απαιτείται για την δημιουργία του αντικειμένου.
 - Μία κλάση μπορεί να περιέχει και περισσότερες από μία **μεθόδους κατασκευής**. Όλες φέρουν το όνομα της κλάσης.
 - Εάν δεν περιλαμβάνονται μέθοδοι κατασκευής σε μία κλάση, αυτή θα κληρονομήσει μια μέθοδο κατασκευής χωρίς ορίσματα από την υπερκλάση της.

Κλάσεις



Συμπεριφορά (Μέθοδοι)

- **Μέθοδοι πρόσβασης (Accessors).**
 - Είναι απλές μέθοδοι οι οποίες επιστρέφουν τιμές των βασικών μεταβλητών ενός αντικειμένου.
 - Οι μέθοδοι πρόσβασης μπορούν να παραληφθούν, για παράδειγμα όταν οι μεταβλητές είναι public. Καλό όμως είναι να αποφεύγουμε τις public μεταβλητές και να χρησιμοποιούμε μεθόδους πρόσβασης.
- **Μέθοδοι ελέγχου ισότητας αντικειμένων (equals).**
 - Αποτελούν βασικές μεθόδους οι οποίες ελέγχουν εάν δύο αντικείμενα μιας κλάσης είναι ίσα μεταξύ τους.
 - Κατά την ανάπτυξή τους πρέπει να λαμβάνονται υπ' όψη όλες οι ιδιότητες και συμπεριφορές των αντικειμένων.

Κλάσεις



Συμπεριφορά (Μέθοδοι)

- **Μέθοδος εκτύπωσης αντικειμένου (toString()).**
 - Υπάρχει μια ειδική μέθοδος στη Java για την εκτύπωση ενός αντικειμένου

```
public String toString(){  
    .....  
}
```
 - Με αυτόν τον τρόπο είναι δυνατή η εκτύπωση ενός αντικειμένου μέσω της μεθόδου println().

Κλάσεις



Παράδειγμα 1 με κλάση

- Δημιουργήστε μια κλάση αντικειμένων Point η οποία να περιγράφει σημεία στο επίπεδο. Να περιλαμβάνει μια μέθοδο κατασκευής, τις μεθόδους πρόσβασης στις συντεταγμένες x και y του σημείου, μια μέθοδο ελέγχου ισότητας δύο αντικειμένων και μια μέθοδο εκτύπωσης ενός αντικειμένου. Στη συνέχεια γράψτε ένα πρόγραμμα στο οποίο να χρησιμοποιήσετε την κλάση που φτιάξατε και να δημιουργήσετε δύο σημεία στο επίπεδο. Με την βοήθεια των δύο αυτών σημείων να ελέγξετε την κλάση των αντικειμένων που δημιουργήσατε. (Άσκηση 5.1.1)

Κλάσεις



Παράδειγμα 1 με κλάση

```
public class Point
{
    // Sintetagmenes x kai y tou shmeiou
    private double x, y;

    // Methodos kataskevis (Constructor)
    public Point(double a, double b)
    {
        x=a;
        y=b;
    }

    // Methodos prosvasis sti sintetagmeni x (Accessor Method)
    public double x()
    {
        return x;
    }

    // Methodos prosvasis sti sintetagmeni y (Accessor Method)
    public double y()
    {
        return y;
    }

    // Methodos elegxou isotitas antikeimenon
    public boolean equals(Point p)
    {
        return (x==p.x && y==p.y);
    }

    // Methodos ektyposis antikeimenou
    public String toString()
    {
        return new String("(" + x + ", " + y + ")");
    }
}
```

Point.java (Java)--L36--A11

Κλάσεις

Παράδειγμα 1 με κλάση

```
class TestPoint
{
    public static void main(String[] arguments)
    {
        Point a = new Point(4,5);
        System.out.println("a.x = " + a.x() + " a.y = " + a.y());
        System.out.println("a = " + a);

        Point b = new Point(2,3);
        System.out.println("b = " + b);
        if(a.equals(b)) System.out.println("To a isoute me to b");
        else System.out.println("To a einai diaforetiko tou b");

        b = new Point(4,5);
        System.out.println("b = " + b);
        if(a.equals(b)) System.out.println("To a isoute me to b");
        else System.out.println("To a einai diaforetiko tou b");
    }
}
--:-- TestPoint.java (Java)--L19--All-----
```

Κλάσεις

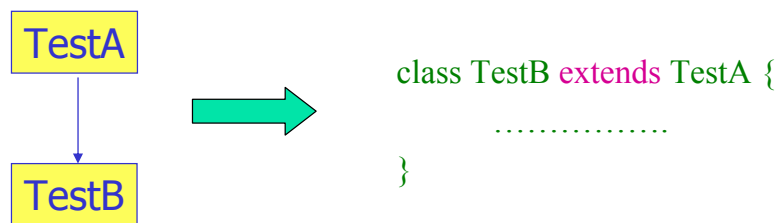
Παράδειγμα 1 με κλάση

```
[student1@pc244 kef5]$
[student1@pc244 kef5]$ javac Point.java
[student1@pc244 kef5]$ javac TestPoint.java
[student1@pc244 kef5]$ java TestPoint
a.x = 4.0 a.y = 5.0
a = (4.0, 5.0)
b = (2.0, 3.0)
To a einai diaforetiko tou b
b = (4.0, 5.0)
To a isoute me to b
[student1@pc244 kef5]$ █
```

Κλάσεις

Κληρονομικότητα

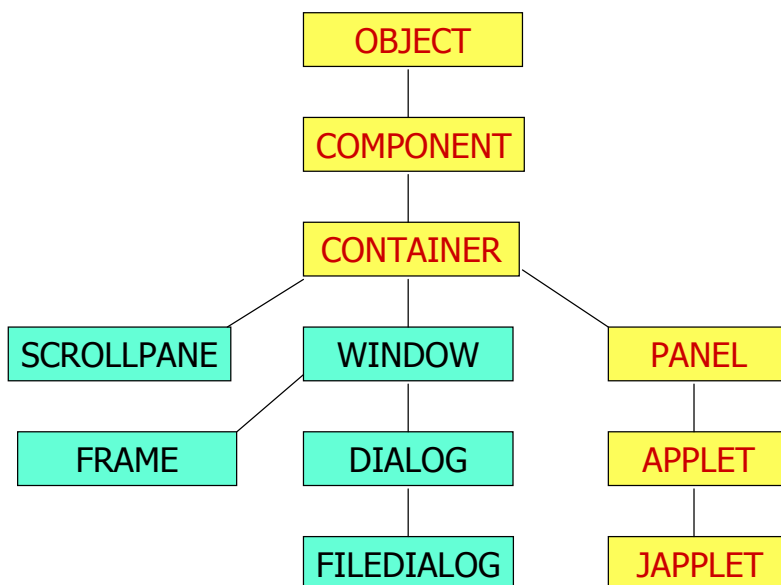
- **Κληρονομικότητα (inheritance)** είναι ο τρόπος με τον οποίο ένα αντικείμενο μπορεί να κληρονομήσει ιδιότητες και συμπεριφορά από άλλα αντικείμενα, που είναι όμοια με αυτό.
- Μέσω της κληρονομικότητας μπορούμε να δημιουργήσουμε μια νέα κλάση αντικειμένων προσδιορίζοντας μόνο τα σημεία εκείνα στα οποία αυτή διαφέρει από μια υπάρχουσα κλάση.



Κλάσεις

Κληρονομικότητα

- Παράδειγμα : Το γενεαλογικό δένδρο της κλάσης JAPPLET



Κλάσεις

Μέθοδοι επικάλυψης

- Όταν μια μέθοδος ορίζεται σε μια υποκλάση και στην υπερκλάση, χρησιμοποιείται η μέθοδος της υποκλάσης.
- Αυτό επιτρέπει σε μια υποκλάση να αλλάζει, να αντικαθιστά ή να καταργεί τελείως μέρος της συμπεριφοράς ή των ιδιοτήτων των υπερκλάσεων της.
- Η δημιουργία μιας νέας μεθόδου σε μια υποκλάση, έτσι ώστε να αλλάξει η συμπεριφορά που κληρονομήθηκε από μια υπερκλάση ονομάζεται **μέθοδος επικάλυψης (Overriding)**.

Κλάσεις

Η χρήση του this και του super

- Με τη λέξη **this** αναφερόμαστε σε μεταβλητές του τρέχοντος αντικειμένου μας.

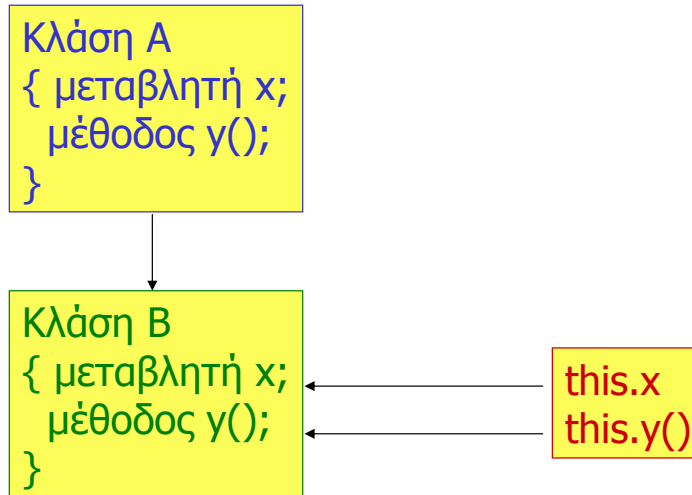
```
public class ΟνομαΚλασης {  
    private int a;  
  
    public μεθοδοςΑ( int a){  
        this.a = 15;  
        ....  
    }  
}
```

Κλάσεις



Η χρήση του this και του super

- Σε υποκλάσεις η λέξη **this** χρησιμοποιείται όταν θέλουμε να αναφερθούμε σε μεταβλητές και μεθόδους του τρέχοντος αντικειμένου μας.

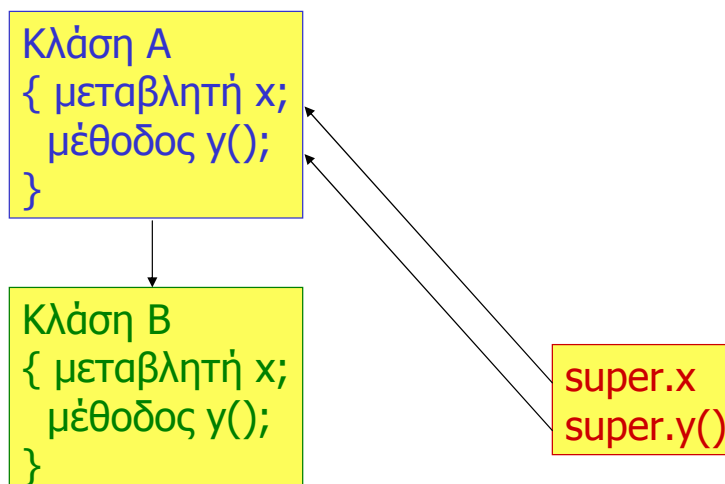


Κλάσεις



Η χρήση του this και του super

- Η λέξη **super** χρησιμοποιείται όταν θέλουμε να αναφερθούμε σε μεταβλητές και μεθόδους της αμέσως προηγούμενης υπερκλάσης του αντικειμένου μας.



Κλάσεις



Η χρήση του this και του super

- Η λέξη **super** χρησιμοποιείται στη μέθοδο κατασκευής μιας υποκλάσης. Με αυτόν τον τρόπο συνδέουμε την μέθοδο κατασκευής μιας υποκλάσης με αυτήν της υπερκλάσης της.

Για παράδειγμα ας υποθέσουμε τις ClassA και ClassB. Εάν ορίσουμε

```
public class ClassB extends ClassA {  
    public ClassB(int x, String a) {  
        super(x,a);  
    }  
    .....  
}
```

Η μέθοδος κατασκευής της υποκλάσης ClassB χρησιμοποιεί το super(x,a) για να καλέσει μια ανάλογη μέθοδο κατασκευής από την υπερκλάση της ClassA.

Κλάσεις



Παράδειγμα 2 με κλάση

- Δημιουργήστε μια κλάση αντικειμένων **Student** η οποία να περιγράφει φοιτητές. Ένας φοιτητής έχει το όνομα το επώνυμό και τον αριθμό μητρώου του. Η κλάση που θα δημιουργήσετε πρέπει να περιλαμβάνει μια μέθοδο κατασκευής, τις μεθόδους πρόσβασης στο όνομα, το επώνυμό και τον αριθμό μητρώου του φοιτητή και μια μέθοδο εκτύπωσης ενός αντικειμένου. Στη συνέχεια να δημιουργήσετε την κλάση **JavaStudent** η οποία να επεκτείνει την κλάση Student με τον βαθμό κάθε φοιτητή στο μάθημα της Java. Η νέα κλάση πρέπει να περιλαμβάνει μια μέθοδο κατασκευής, μια μέθοδο πρόσβασης στο βαθμό του φοιτητή και μια μέθοδο εκτύπωσης ενός αντικειμένου. Στη συνέχεια γράψτε ένα πρόγραμμα στο οποίο να δημιουργήσετε τέσσερα αντικείμενα της κλάσης JavaStudent. Εισάγετε κατάλληλα δεδομένα σε αυτά και εκτυπώστε τα. (Άσκηση: 5.1.4)

Κλάσεις

Παράδειγμα 2 με κλάση

```
public class Student
{
    protected String onoma;
    protected String eponimo;
    protected int AM; // Arithmos mitroou

    // Methodos kataskevis (Constructor)
    public Student(String onoma, String eponimo, int AM)
    {
        this.onoma=onoma;
        this.eponimo=eponimo;
        this.AM=AM;
    }

    public String onoma() // Methodos prosvasis sto onoma
    {
        return onoma;
    }

    public String eponimo() // Methodos prosvasis sto eponimo
    {
        return eponimo;
    }

    public int AM() // Methodos prosvasis ston AM
    {
        return AM;
    }

    public String toString() // Methodos ektyposis antikeimenou
    {
        String a = new String(onoma + " " + eponimo + " AM : " + AM);
        return a;
    }
}
--:-- Student.java (Java)--L35--A11-----
```

Κλάσεις

Παράδειγμα 2 με κλάση

```
public class JavaStudent extends Student
{
    protected double bathmos; // Bathmos sto mathima Java

    // Methodos kataskevis (Constructor)
    public JavaStudent(String onoma, String eponimo, int AM, double bathmos)
    {
        super(onoma,eponimo,AM);
        this.bathmos=bathmos;
    }

    public double bathmos() // Methodos prosvasis sto bathmo
    {
        return bathmos;
    }

    public String toString() // Methodos ektyposis antikeimenou
    {
        String a = new String(super.toString() + " Bathmos : " + bathmos);
        return a;
    }
}
--:-- JavaStudent.java (Java)--L22--A11-----
```

Κλάσεις

Παράδειγμα 2 με κλάση

```
class TestJavaStudent
{
    public static void main(String[] arguments)
    {
        JavaStudent a = new JavaStudent("Nikolaos", "Apostolou", 3145, 8.5);
        JavaStudent b = new JavaStudent("Ioannis", "Dalas", 3256, 5.0);
        JavaStudent c = new JavaStudent("Georgios", "Ladas", 3372, 7.0);
        JavaStudent d = new JavaStudent("Maria", "Nikolaou", 3484, 6.5);

        System.out.println(a);
        System.out.println(b);
        System.out.println(c);
        System.out.println(d);
    }
}
```

--:-- TestJavaStudent.java (Java)--L16--All-----

Κλάσεις

Παράδειγμα 2 με κλάση

```
[student1@pc244 kef5]$
[student1@pc244 kef5]$ javac Student.java
[student1@pc244 kef5]$ javac JavaStudent.java
[student1@pc244 kef5]$ javac TestJavaStudent.java
[student1@pc244 kef5]$ java TestJavaStudent
Nikolaos Apostolou AM : 3145 Bathmos : 8.5
Ioannis Dalas AM : 3256 Bathmos : 5.0
Georgios Ladas AM : 3372 Bathmos : 7.0
Maria Nikolaou AM : 3484 Bathmos : 6.5
[student1@pc244 kef5]$ █
```

Κλάσεις