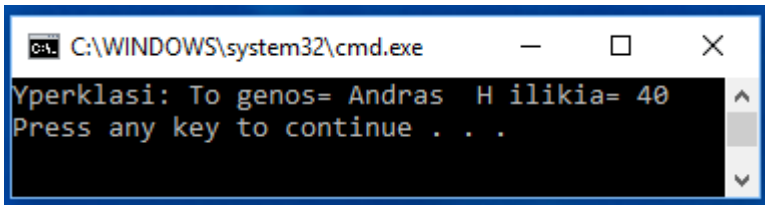
ΠΡΟΤΕΙΝΟΜΕΝΕΣ ΛΥΣΕΙΣ ΑΣΚΗΣΕΩΝ
ΚΛΗΡΟΝΟΜΙΚΟΤΗΤΑ

ΑΣΚΗΣΗ-1η

Κληρονομικότητα : Το αρχείο αποθηκεύεται με το όνομα **Subclass.java**

```
class Superclass {
 String gender;
 int age;
 Superclass(String g, int a) {this.gender=g;this.age = a; }
 public void getGenderAndAge(){
  System.out.println("Yperklasi: To genos= "+gender+"H ilikia= "+age);
}}
class Subclass extends Superclass {
   Subclass(String g1, int a1) {
      super(g1, a1); }
public static void main(String argd[]) {
      Subclass s = new Subclass("Andras", 40);
      s.getGenderAndAge();  } }
```

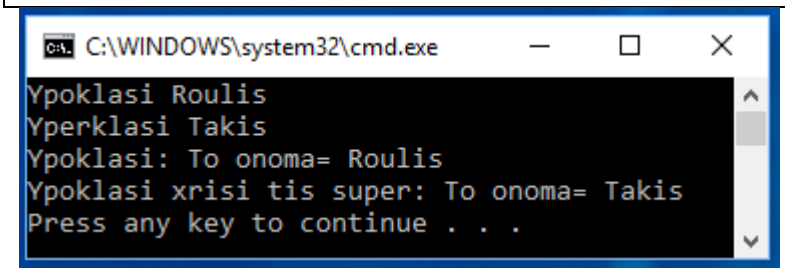


ΑΣΚΗΣΗ-2η

Κληρονομικότητα : Το αρχείο αποθηκεύεται με το όνομα **Subclass1.java**

```
class Superclass1 {
 String name = "Takis";
 public void display()
  {System.out.println("Yperklasi "+name);  }}
public class Subclass1 extends Superclass1 {
 String name = "Roulis";
public void display()  {System.out.println("Ypoklasi "+name);}
```

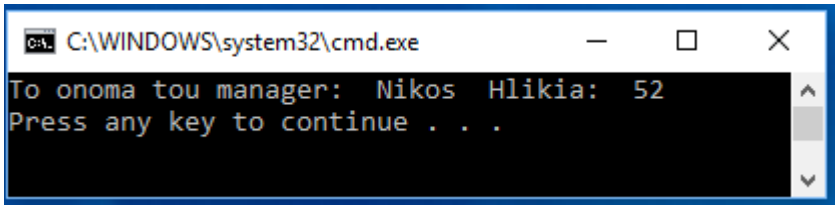
```
public void A_method() {
 Subclass1 s = new Subclass1();
 s.display();
 super.display();
 System.out.println("Ypoklasi: To onoma= "+ s.name);
  System.out.println("Ypoklasi xrisi tis super: To onoma= "+ super.name); }

public static void main(String args[])  {
     Subclass1 obj = new Subclass1();
     obj.A_method();  } }
```



ΑΣΚΗΣΗ-3η

Κληρονομικότητα : Χρήση χαρακτηριστικών και μεθόδων της υπερκλάσης. Προσπέλαση σε private μέλη της υπερκλάσης με setter()-getter().

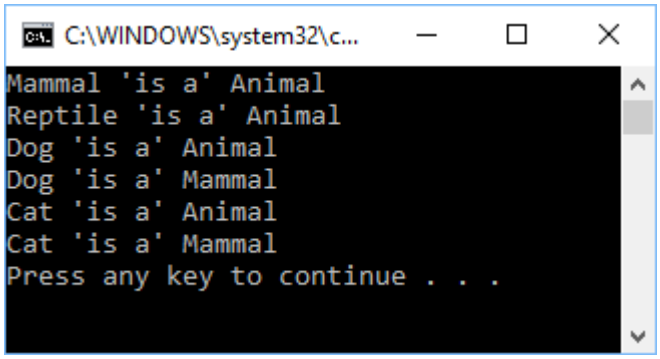
```
class Employee1 {
  private String name = "Kanena onoma";
  private int age;
  public void setName(String name) {this.name = name;}
  public String getName() {return name;}
  public void setAge(int age) {this.age=age;}
  public int getAge() {return age;}}
class Manager extends Employee1 { }

class TestEmployee1 {
 public static void main(String[] args) {
    Manager mgr = new Manager();
    mgr.setName("Nikos");   //to onoma tou Mgr, (stin yperklasi)
    String mgrName = mgr.getName();  //to onoma tou Mgr, (apo tin yperklasi)
    mgr.setAge(52);     //set Hlikia tou Mgr (stin yperklasi)
    int mgrHlikia=mgr.getAge();   //Hlikia tou Mgr, (apo tin yperklasi)
    System.out.println("To onoma tou manager:  " + mgrName+ "  Hlikia:  "+
                       mgrHlikia);}}
```



ΑΣΚΗΣΗ-4η

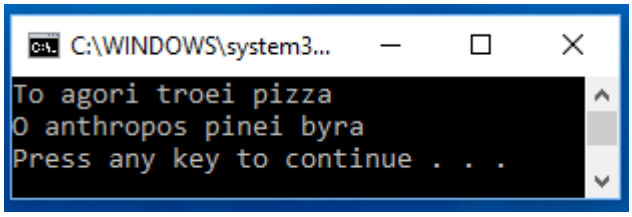
Κληρονομικότητα : Χρήση της **instanceof**

```
class Animal { }
class Mammal extends Animal { }
class Reptile extends Animal { }
class Cat extends Mammal {}
public class Dog extends Mammal {
  public static void main(String args[]) {
    Animal a = new Animal();
    Mammal m = new Mammal();
    Reptile r = new Reptile();
    Dog d = new Dog();
    Cat c = new Cat();
     if (m instanceof Animal)System.out.println("Mammal 'is a' Animal");
     if (r instanceof Animal)System.out.println("Reptile 'is a' Animal");
     if (d instanceof Animal)System.out.println("Dog 'is a' Animal");
     if (d instanceof Mammal)System.out.println("Dog 'is a' Mammal");
     if (c instanceof Animal)System.out.println("Cat 'is a' Animal");
     if (c instanceof Mammal)System.out.println("Cat 'is a' Mammal");
}}
```



ΑΣΚΗΣΗ-5η (Υπέρβαση μεθόδων)

```
class Human{
   public void eat() {System.out.println("O anthropos troei pizza");}
   public void drink() {System.out.println("O anthropos pinei byra");}
}

class Boy1 extends Human{
   public void eat(){System.out.println("To agori troei pizza");}  //ypervasi
   public void drink(){System.out.println("To agori pinei byra");} //ypervasi
 public void A_method() {
   Boy1 b = new Boy1();
   b.eat();              //ypervasi methodou
   super.drink();   // xrisi leitourgias tis yperklasis  }

public static void main( String args[]) {
      Boy1 obj = new Boy1();
      obj.A_method();
}}
```

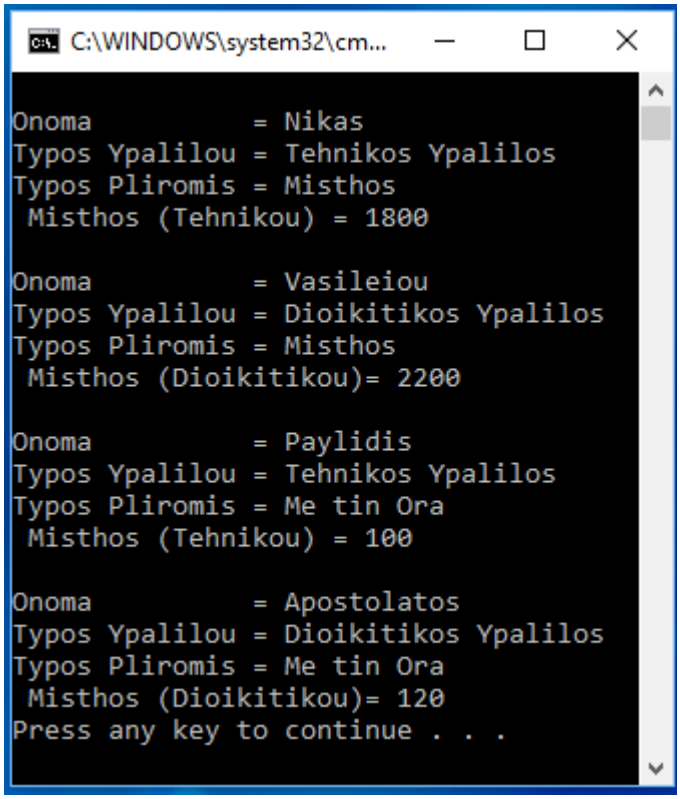


ΑΣΚΗΣΗ-6η (Κληρονομικότητα - Employee)

```
class Employee {
  protected String name;
  protected int b;  //bonus
  protected int h;  //hours
  protected int payType;     //0=salary, 1=byhour
  Employee(String s, int b_, int h_, int p) {name=s; b=b_; h=h_; payType=p;}
  public String getName() {return name;}
  public String getPayType() {
   String pType;
   if (payType==0) pType="Misthos";
    else pType="Me tin Ora";
      return pType; }
```

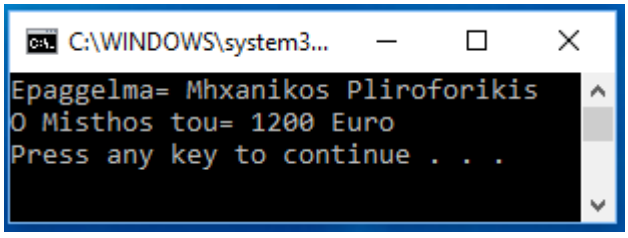
```
  public String getEmpType() {return "Employee";}
  public void calcSalary() {
      int s=1000;
            System.out.println(" Misthos Employee  = " + s); }}

class Administrator extends Employee {
  Administrator(String s, int b, int h, int p) {super(s,b,h,p);}
  public String getEmpType() {return "Dioikitikos Ypalilos";}
  public void calcSalary() {
    int s=0;
   if (payType==0) s=1200+b;  //vasikos=1200
      else s=(h*12);              //12 Euro per hour
          System.out.println(" Misthos (Dioikitikou) = " + s); }}

class Technical extends Employee {
  Technical(String s, int b, int h, int p) {super(s,b,h,p);}
  public String getEmpType() {return "Tehnikos Ypalilos";}
  public void calcSalary() {
          int s=0;
          if (payType==0) s=800+b;   //vasikos=800
            else s=(h*10);              //10 Euro per hour
              System.out.println(" Misthos (Tehnikou)= " + s); } }

/* Βοηθητική κλάση ορισμού πίνακα αντικειμένων, γέμισμα του πινακα
 * με τα 4 αντικείμενα setEmployee(), και εμφάνιση των στοιχείων
 * των υπαλλήλων και του μισθού των */
class Company {
  Employee emp[]= new Employee[4];

  public void setEmployee(Employee e, int a) {emp[a]=e;}
  public void printAll() {
      for (int i=0;i<4;i++) {
        System.out.println();
        System.out.println("Onoma          = "+ emp[i].getName());
        System.out.println("Typos Ypalilou = "+ emp[i].getEmpType());
        System.out.println("Typos Pliromis = "+ emp[i].getPayType());
        emp[i].calcSalary(); } } }
```

```
class TestEmployee {
  public static void main(String[] args) {
  int a;
    Employee e1= new Technical("Nikas",1000,10,0);
    Employee e2=new Administrator("Vasileiou",1000,20,0);
    Employee e3= new Technical("Paylidis",10,10,1);
    Employee e4=new Administrator("Apostolatos",10,10,1);
    Company c = new Company();

    //εισαγωγή των αντικειμένων στον πίνακα
    c.setEmployee(e1,0);
    c.setEmployee(e2,1);
    c.setEmployee(e3,2);
    c.setEmployee(e4,3);
    c.printAll();
} }
```



ΑΣΚΗΣΗ-7η (Σύνθεση)

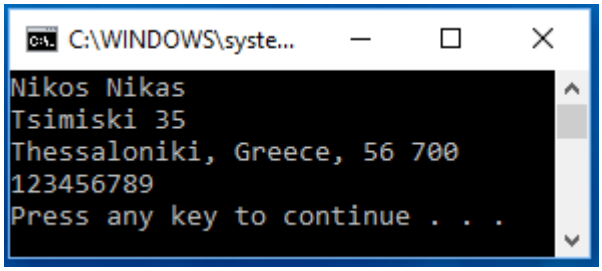
```
class Job {
  private String perigrafi;
  private int misthos;
  private int kodikos;
  public String getPerigrafi(){return perigrafi;}
  public void setPerigrafi(String per) {this.perigrafi = per;}
  public int getMisthos() {return misthos;}
  public void setMisthos(int misthos) {this.misthos = misthos;}
  public int getKodikos() {return kodikos; }
  public void setKodikos(int kodikos) {this.kodikos = kodikos;} }

class Person {
   //composition has-a relationship
   private Job job;
   public Person(){
      this.job=new Job();
      job.setMisthos(1200);
      job.setPerigrafi("Mhxanikos Pliroforikis");
      }
   public int getMisthos() {return job.getMisthos(); }
   public String getPerigrafi() {return job.getPerigrafi(); }
}
 class TestPerson {
   public static void main(String[] args) {
      Person person = new Person();
      String p = person.getPerigrafi();
      int misthos = person.getMisthos();
      System.out.println("Epaggelma= " + p);
      System.out.println("O Misthos tou= "+misthos + " Euro"); } }
```



ΑΣΚΗΣΗ-8η (Σύνθεση)

```
class Name {
   String firstname;
   String lastname;
   public Name(String newFirstname, String newLastname) {
     firstname = newFirstname;
     lastname = newLastname;}
   public String getFirstname() {return firstname;}
   public String getLastname() {return lastname;}
   public String getFirstLast() {return firstname + " " + lastname;} }

class Address {
 String street;
 String city;
 String state;
 String zip;
 public Address(String newStreet, String newCity, String newState,
     String newZip) {
  street = newStreet;
  city = newCity;
  state = newState;
  zip = newZip; }
 public String getStreet(){return street; }
 public String getCity(){return city;}
 public String getState(){return state;}
 public String getZip(){return zip;}
 public String getFullAddress() {
    return street + "\n" + city + ", " + state + ", " + zip; }}

class Employee {
 private Name myName;        //antikeimeno typou Name
 private Address myAddress;  //antikeimeno typou Address
 private String AFM;
 public Employee(Name n, Address a, String newAFM) {
   myName = n;
   myAddress = a;
   AFM = newAFM; }
```

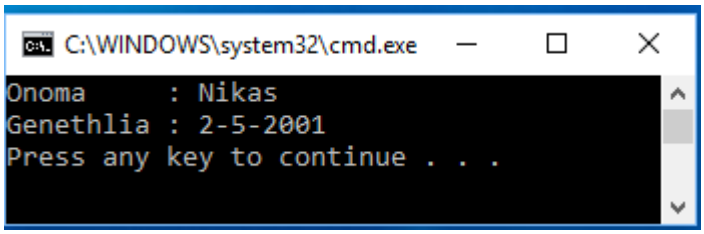
```
 public Name getName() {return myName; }
 public Address getAddress() {return myAddress;}
 public String getAFM() {return AFM; } }

class TestEmployeeComposition {
 public static void main(String[] args) {
  Name eponymia = new Name("Nikos", "Nikas");
  Address dieythinsi = new Address("Tsimiski 35", "Thessaloniki", "Greece",
                                   "56 700");
  String AFM = "123456789";
  Employee theEmployee = new Employee(eponymia, dieythinsi, AFM);
  System.out.println(theEmployee.getName().getFirstLast() + "\n" +
  theEmployee.getAddress().getFullAddress()+"\n"+theEmployee.getAFM());}}
```



ΑΣΚΗΣΗ-9η (Σύνθεση)

```
class Person {
  private double salary;
  private String name;
  private Birthday bday;    //αναφορά σε αντικείμενο της Birthday
  public Person(int d,int m,int y,String name){
   bday=new Birthday(d, m, y);  //αρχικοποίηση του αντικ. στον δομητή
   this.name=name;
  }
  public double getSalary() {return salary;}
  public String getName() {return name;}
  public Birthday getBday() {return bday;}   //λήψη της Birthday
 }
```

```
class Birthday{
  int day, month,year;
  public Birthday(int d,int m,int y){
    day=d;
    month=m;
    year=y; }
  public String toString(){return String.format("%s-%s-%s",
       day,month,year); } }

class CompositionTst1 {
  public static void main(String[] args) {
    Person person=new Person(2, 5, 2001, "Nikas");
    System.out.println("Onoma     : "+person.getName());
    System.out.println("Genethlia : "+person.getBday());
} }
```



ΑΣΚΗΣΗ-10η (Σύνθεση και κληρονομικότητα)

```
class Car {
  private String color;
  private int maxSpeed;
  public void carInfo(){
     System.out.println("Xroma autokinitou= "+color + " Megisti Tahytita= " + maxSpeed); }
  public void setColor(String color){this.color = color;}
  public void setMaxSpeed(int maxSpeed) {this.maxSpeed = maxSpeed;}
}

class Ferrari extends Car{
  /*H klasi Ferrari klironomei oles tis
  * methodous tis Car (ektos apo tis final kai static)*/
  public void FerrariDemo(){
     Engine FerrariEngine = new Engine();
     FerrariEngine.start(); } }
```

```
class Engine {
    public void start(){
        System.out.println("H mixani douleyei...");
    }
    public void stop(){
        System.out.println("H mixani stamatise..."); }
  }

public class InheritanceAndComposition {
    public static void main(String[] args) {
        Ferrari myFerrari = new Ferrari();
        myFerrari.setColor("Kokkini");
        myFerrari.setMaxSpeed(290);
        myFerrari.carInfo();
        myFerrari.FerrariDemo();
    } }
```