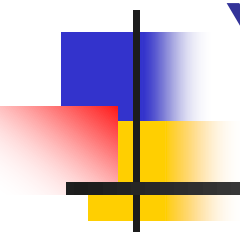
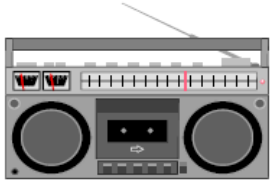
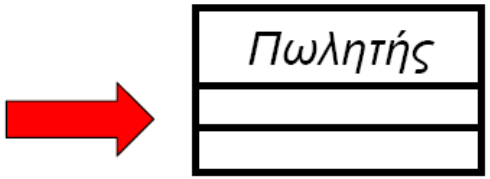
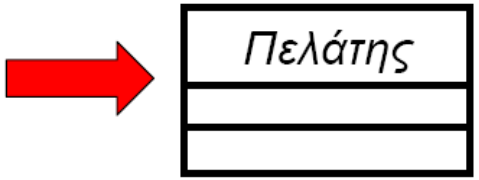


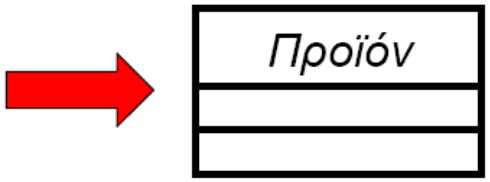
Εισαγωγή στην Επιστήμη των Υπολογιστών

Αντικείμενα



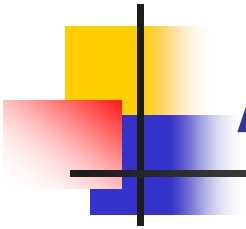






Πλεονεκτήματα αντικειμενοστραφούς σχεδίασης

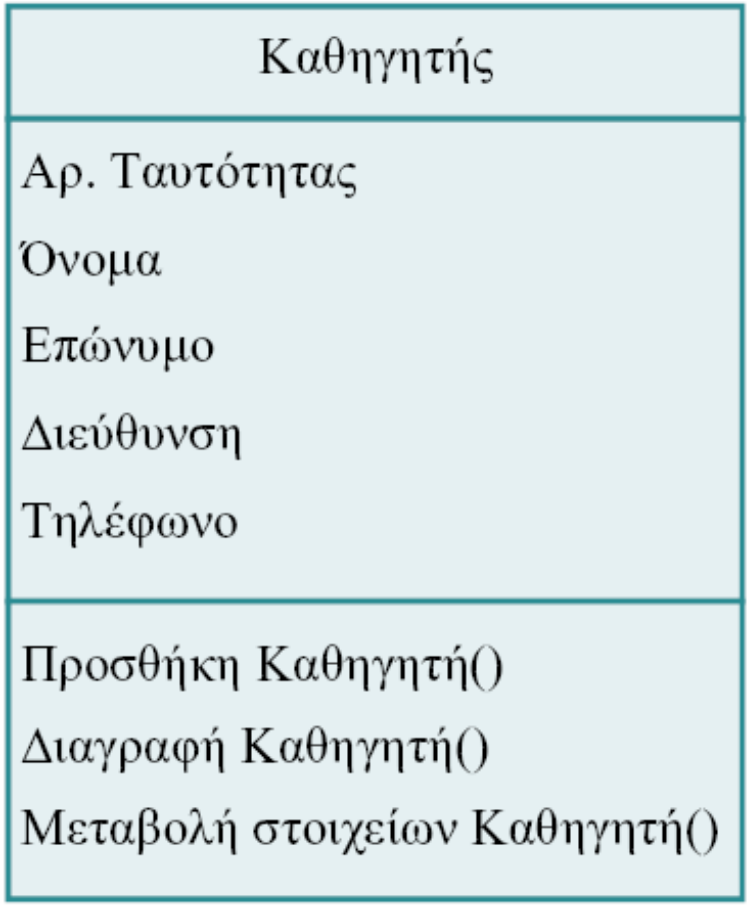
- Ευκολότερη συντήρηση. Τα αντικείμενα είναι κατανοητά ως αυτόνομες οντότητες.
- Τα αντικείμενα είναι κατάλληλα επαναχρησιμοποιήσιμα στοιχεία
- Για κάποια συστήματα, υπάρχει φανερή αντιστοίχιση μεταξύ οντοτήτων του πραγματικού κόσμου και αντικειμένων του συστήματος

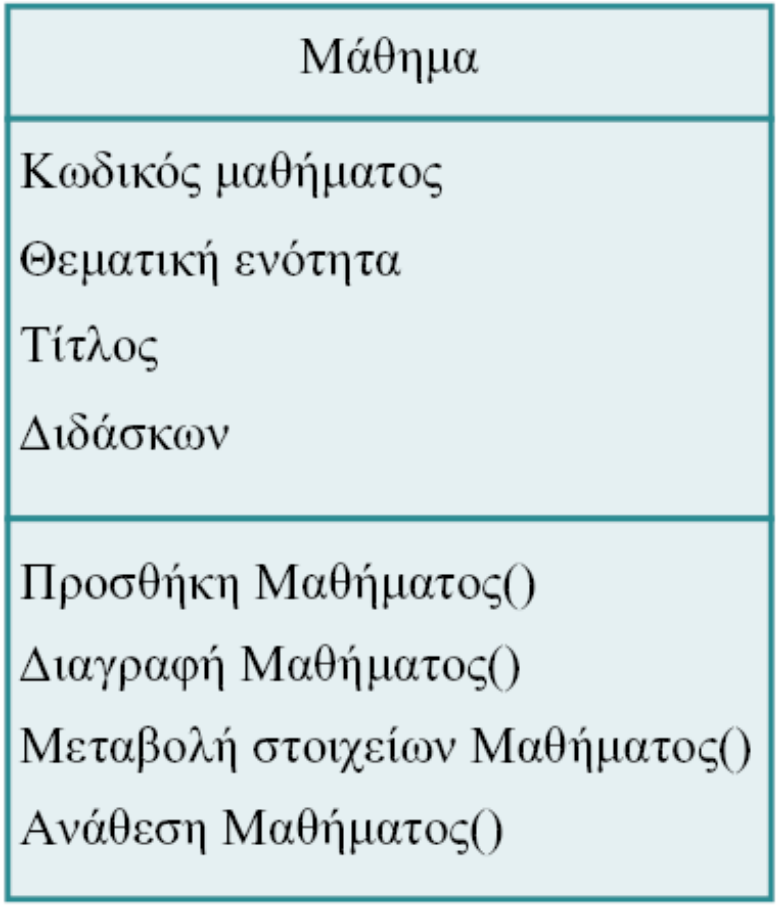


Αντικείμενα, Κλάσεις

- Τα αντικείμενα είναι οντότητες ενός συστήματος λογισμικού που αναπαριστούν στιγμές του πραγματικού κόσμου και οντότητες συστήματος
- Οι κλάσεις αντικειμένων είναι φόρμες αντικειμένων. Χρησιμοποιούνται για την δημιουργία αντικειμένων

Κλάσεις


Καθηγητής
Αρ. Ταυτότητας
Όνομα
Επώνυμο
Διεύθυνση
Τηλέφωνο
Προσθήκη Καθηγητή()
Διαγραφή Καθηγητή()
Μεταβολή στοιχείων Καθηγητή()

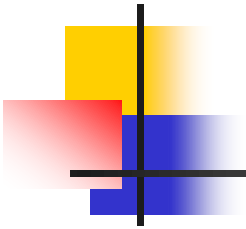

Μάθημα
Κωδικός μαθήματος
Θεματική ενότητα
Τίτλος
Διδάσκων
Προσθήκη Μαθήματος()
Διαγραφή Μαθήματος()
Μεταβολή στοιχείων Μαθήματος()
Ανάθεση Μαθήματος()

Αντικείμενα – στιγμιότυπα των κλάσεων

Καθηγητής 01
Αρ. Ταυτότητας: **Α123456** Όνομα: **Βασίλειος** Επώνυμο: **Βασιλείου** Διεύθυνση: **Αγ. Βασιλείου 1** Τηλέφωνο: **9876543**
Προσθήκη Καθηγητή() Διαγραφή Καθηγητή() Μεταβολή στοιχείων Καθηγητή()

Καθηγητής 02
Αρ. Ταυτότητας: **Β987654** Όνομα: **Γεώργιος** Επώνυμο: **Γεωργίου** Διεύθυνση: **Αγ. Γεωργίου 1** Τηλέφωνο: **7654321**
Προσθήκη Καθηγητή() Διαγραφή Καθηγητή() Μεταβολή στοιχείων Καθηγητή()

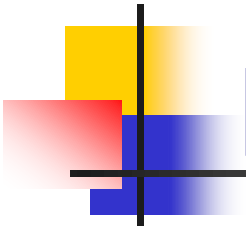
Μάθημα 01
Κωδικός μαθήματος: **ΠΛ-034** Θεματική ενότητα: **Πληροφορική** Τίτλος: **Προγραμματισμός Η/Υ** Διδάσκων: **Β987654**
Προσθήκη Μαθήματος() Διαγραφή Μαθήματος() Μεταβολή στοιχείων Μαθήματος() Ανάθεση Μαθήματος()



Παράδειγμα κλάσης σε python

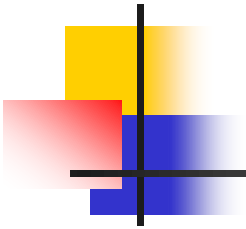
```
class Person(object):
  def __init__(self, name, age):
    self.name = name
    self.age = age
  def is_old(self):
    return self.age > 40
```

Αντικείμενα και κλήση μεθόδου



```
person = Person('G. H.', 50)
print person.is_old()
person2 = Person('G2. H2.', 30)
print person2.is_old()
```

```
True
False
```



Παράδειγμα κλάσης σε python

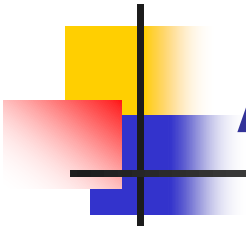
```
class BankAccount:
  def __init__(self):
    self.balance = 0
  def withdraw(self, amount):
    self.balance -= amount
    return self.balance
  def deposit(self, amount):
    self.balance += amount
    return self.balance
```

Αντικείμενα και κλήση μεθόδων

```
>>> a = BankAccount()
>>> b = BankAccount()
>>> a.deposit(100)
100
>>> b.deposit(50)
50
>>> b.withdraw(10)
40
 >>> a.withdraw(10)
90
```

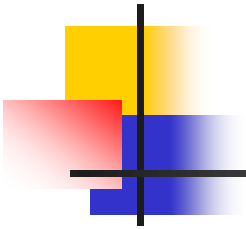
Παράδειγμα κλάσης σε python

```
class Employee:
  empCount = 0
  def __init__(self, name, salary):
    self.name = name
    self.salary = salary
    Employee.empCount += 1
  def displayCount(self):
    print "Total Employee %d" % Employee.empCount
  def displayEmployee(self):
    print "Name : ", self.name, ", Salary: ", self.salary
```



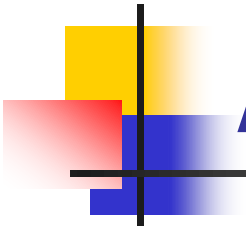
Δημιουργία αντικειμένων

```
emp1 = Employee("Zara", 2000)
emp2 = Employee("Manni", 5000)
```



Κλήση μεθόδου

```
emp1.displayEmployee()
emp2.displayEmployee()
print "Total Employee %d" %
Employee.empCount
```

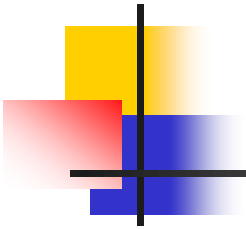


Αποτέλεσμα

Name : Zara , Salary: 2000

Name : Manni , Salary: 5000

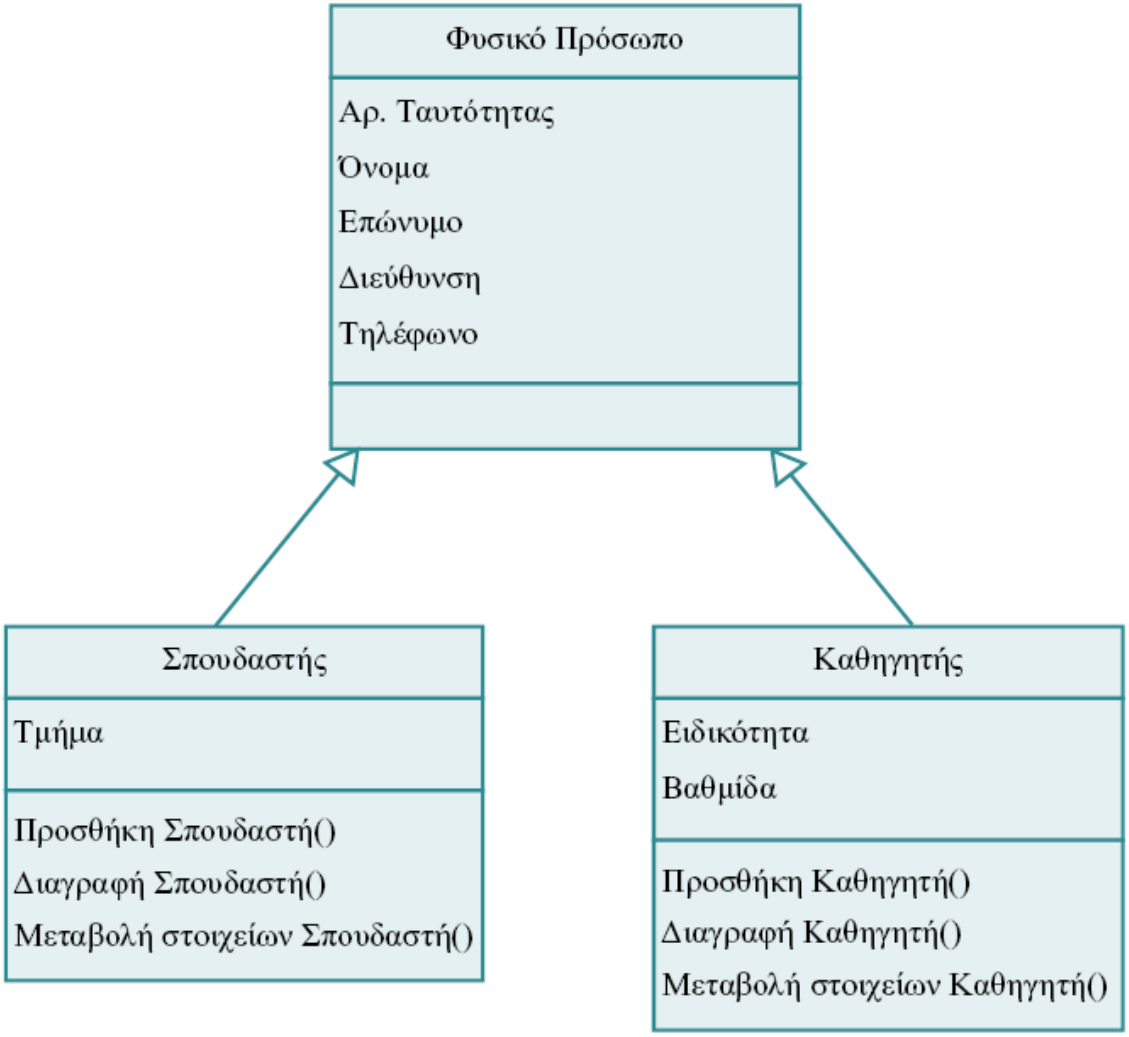
Total Employee 2



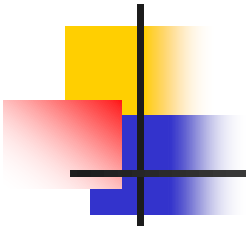
Κληρονομικότητα

- Τα αντικείμενα είναι μέλη κλάσεων τα οποία ορίζουν τύπους χαρακτηριστικών και λειτουργίες
- Οι κλάσεις μπορεί να τακτοποιηθούν σε ιεραρχία κλάσεων όπου μία κλάση προέρχεται από μία υπάρχουσα κλάση (υπερ - κλάση)
- Μία υποκλάση κληρονομεί τα χαρακτηριστικά από μία υπερ - κλάση και μπορεί να προσθέσει νέες μεθόδους ή χαρακτηριστικά

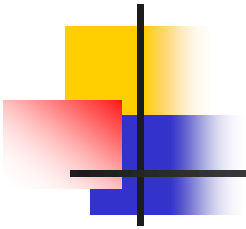
Κληρονομικότητα



Πλεονεκτήματα Κληρονομικότητας

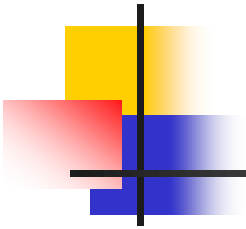


- Είναι αφαιρετικός μηχανισμός που χρησιμοποιείται για την ταξινόμηση οντοτήτων
- Είναι μηχανισμός επαναχρησιμοποίησης σε επίπεδο σχεδιασμού και προγραμματισμού



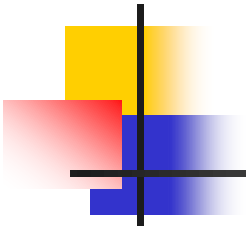
Παράδειγμα

```
class Parent: # υπερκλάση
  parentAttr = 100
  def __init__(self):
    print "Calling parent constructor"
  def parentMethod(self):
    print 'Calling parent method'
  def setAttr(self, attr):
    Parent.parentAttr = attr
  def getAttr(self):
    print "Parent attribute :", Parent.parentAttr
```



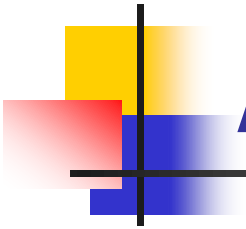
Παράδειγμα (2)

```
class Child(Parent): # υποκλάση
  def __init__(self):
    print "Calling child constructor"
  def childMethod(self):
    print 'Calling child method'α
```



Κλήση μεθόδων

```
c = Child() # instance of child
c.childMethod() # child calls its method
c.parentMethod() # calls parent's method
c.setAttr(200) # again call parent's method
c.getAttr() # again call parent's method
```



Αποτέλεσμα

Calling child constructor

Calling child method

Calling parent method

Parent attribute : 200