



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

ΑΝΟΙΚΤΑ ακαδημαϊκά
μαθήματα ΠΠ

Εισαγωγή στους Υπολογιστές

Ενότητα 4: Αντικειμενοστραφής προγραμματισμός
στην Python

Αβούρης Νικόλαος

Πολυτεχνική Σχολή

Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας
Υπολογιστών

Σκοποί ενότητας

- Σκοπός αυτής της ενότητας είναι η κατανόηση του αντικειμενοστραφούς τρόπου οργάνωσης ενός προγράμματος στην Python καθώς και η κατανόηση μέσω παραδειγμάτων εννοιών όπως κλάσεις, αντικείμενα, στιγμιότυπα, μέθοδοι.



Περιεχόμενα ενότητας

- Αντικειμενοστραφής προγραμματισμός στην Python
- Κλάσεις
- Στιγμιότυπα
- Μέθοδοι καταστροφής και εκτύπωσης αντικειμένων
- Παραδείγματα δημιουργίας κλάσεων



Αντικειμενοστραφής προγραμματισμός στην Python

Αντικειμενοστραφής προγραμματισμός

- Η μέθοδος προγραμματισμού που λέγεται αντικειμενοστραφής προγραμματισμός (**object-oriented programming**) έχει ως κεντρική έννοια την κλάση (**class**) μια αναπαράσταση μιας κατηγορίας αντικειμένων.
- Τα αντικείμενα είναι **στιγμιότυπα (instances)** αυτής της κλάσης
- Οι περισσότερες σύγχρονες γλώσσες όπως η Java και C++ υποστηρίζουν αυτή τη μέθοδο προγραμματισμού.



Βασικές έννοιες

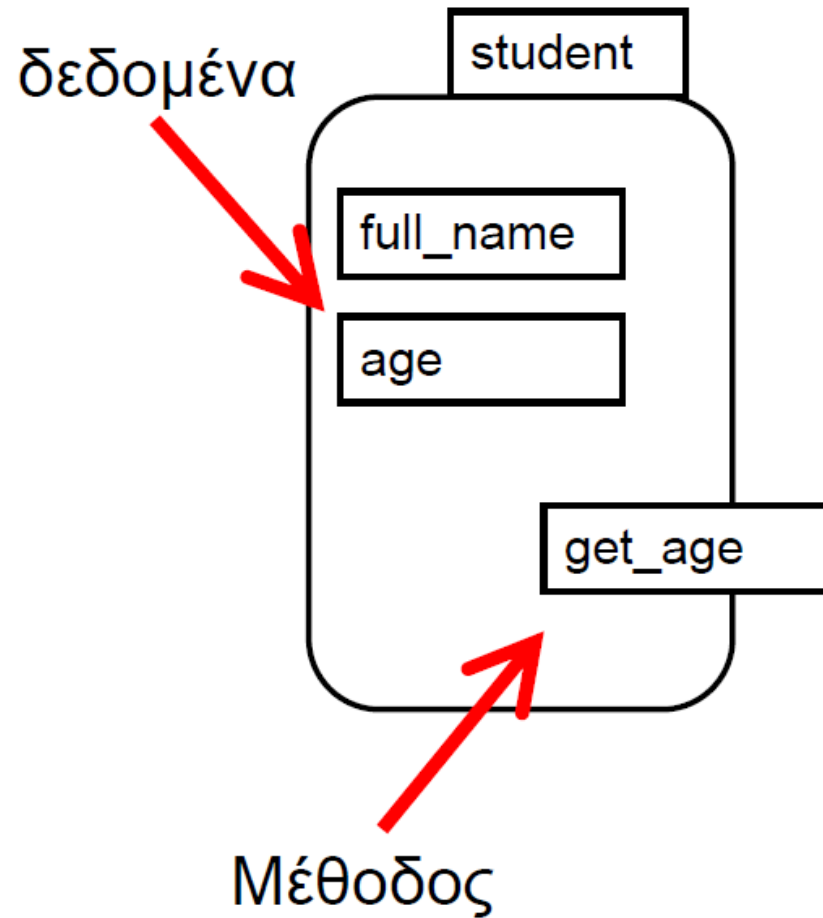
- **Κλάση (class)** : ένας νέος τύπος δεδομένων που ορίζει τη δομή μιας κατηγορίας αντικειμένων. Περιλαμβάνει δεδομένα και τη συμπεριφορά τους (μεθόδους)
- **Στιγμιότυπα (instances)** μιας κλάσης είναι αντικείμενα που δημιουργούνται σύμφωνα με τον ορισμό της κλάσης



Παράδειγμα



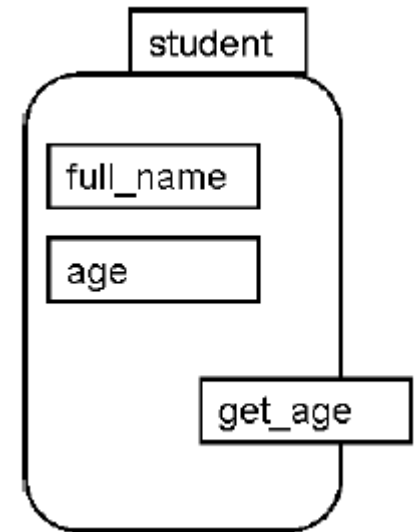
student



Δημιουργία κλάσης student

```
>>> class student:
    """This is a class of a person that studies"""
    def __init__(self,n,a):
        self.full_name = n
        self.age = a
    def get_age(self):
        return self.age
```

docstring



```
>>> A=student('Αδριαστος', 22)
```

```
>>> A.get_age()
```

```
22
```

```
>>> print student.__doc__
```

```
This is a class of a person that studie
```

```
>>> |
```



Παράδειγμα #2

```
class Employee:
    'Common base class for all employees'
    empCount=0 # class variable
    def __init__(self, name, salary):
        self.name =name# instance attribute
        self.salary=salary # instance attribute
        Employee.empCount+=1 # class attribute
    def displayCount(self):
        print "Total Employee %d"%Employee.empCount
    def displayEmployee(self):
        print "Name : ",self.name, ", Salary: " ,self.salary
```



Δημιουργία εργαζομένων

```
emp1 = Employee("Γ. Γεωργίου", 2000)
```

```
emp2 = Employee("Κ.Αντωνίου", 5000)
```

Ποιο το αποτέλεσμα των εντολών:

```
emp1.displayEmployee()
```

```
emp2.displayEmployee()
```

```
print "Total Employee %d" % Employee.empCount
```

Name : Γ. Γεωργίου,Salary: 2000

Name : Κ.Αντωνίου,Salary: 5000

Total Employee 2



Δημιουργία νέων ιδιοτήτων

`emp1.age = 7` # δημιουργία ιδιότητας 'age'.

`emp1.age = 8` # Τροποποίηση ιδιότητας 'age'.

`del emp1.age` # Διαγραφή ιδιότητας 'age'.



Διαχείριση ιδιοτήτων

- **getattr**(obj, name[, default]) : πρόσβαση σε ιδιότητα
- **hasattr**(obj,name) : έλεγχος ύπαρξης ιδιότητας
- **setattr**(obj,name,value) : τιμή ιδιότητας, αν δεν υπάρχει δημιουργία ιδιότητας
- **delattr**(obj, name) : διαγραφή ιδιότητας



Μέθοδοι καταστροφής και εκτύπωσης αντικειμένων

```
class Point:
    def __init__( self, x=0, y=0):
        self.x= x
        self.y= y
    def __del__(self):
        class_name= self.__class__.__name__
        print class_name, "destroyed"
    def __str__(self):
        return "x="+str(self.x)+ " y=" + str(self.y)
```



Μέθοδοι καταστροφής και εκτύπωσης αντικειμένων

```
P1=Point(100,200)
```

```
P2=Point(300,400)
```

```
print P1
```

```
x=100 y=200
```

```
del P1
```



Παράδειγμα Εργαστηριακής Άσκησης

- Θέλουμε να δημιουργήσουμε ένα παιχνίδι που παίζεται με χαρτιά
- Ορίζουμε βασικές οντότητες:
 - Το τραπουλόχαρτο card
 - την τράπουλα deck



Το αντικείμενο card











































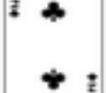
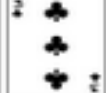





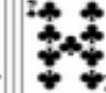
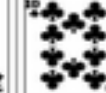



Ιδιότητες ενός τραπουλόχαρτου:

color='R' ή 'B'

value='A', '2', ... ,10, 'J', 'Q', 'K'

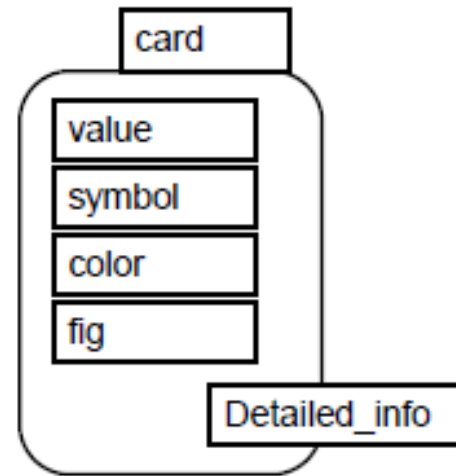
symbol='s', 'c', 'd', 'h'

Fig= True , False

Suit	Ace	2	3	4	5	6	7	8	9	10	Jack	Queen	King
Spades													
Hearts													
Diamonds													
Clubs													



Η κλάση card

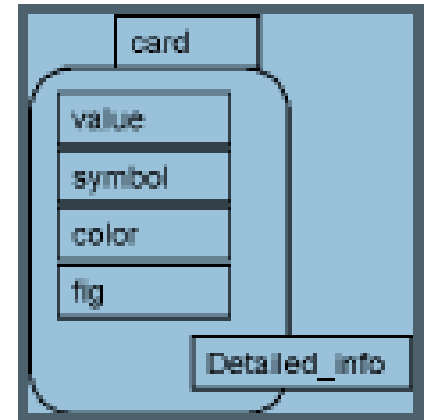


- Επίσης χρειάζεται μέθοδος για τον ορισμό ενός φύλλου (ορισμό στιγμιότυπου της κλάσης) μέθοδο: **`__init__(self, <param>)`**
- Και μέθοδο για την εκτύπωση περιεχομένου, μέθοδο: **`__str__(self)` informal string representation of the object**

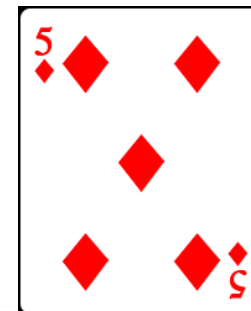


Δημιουργία κλάσης card

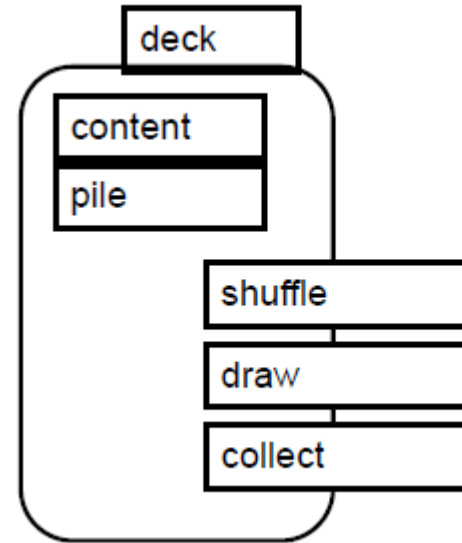
```
>>> class card():
def __init__(self, val, sym):
    self.value=val
    self.symbol=sym
    if self.symbol in "sc": self.color='B'
    else: self.color='R'
    if self.value in "JQK": self.fig=True
    else: self.fig=False
def __str__(self):
    return self.value+self.symbol
def detailed_info(self):
    print 'Αξία=',self.value, '
    print 'Χρώμα=',self.color, '
    print 'Σύμβολο=',self.symbol
    print 'Φιγούρα=',self.fig
```



```
>>> c1=card('5', 'd')
>>> c1.detailed_info()
Αξία= 5
Χρώμα= R
Σύμβολο= d
Φιγούρα= False
```



Η κλάση deck: Τράπουλα



- Η λίστα **content** περιλαμβάνει όλα τα φύλλα της τράπουλας που δεν έχουν ριχτεί, η λίστα **pile** τα χαρτιά που έχουν μοιραστεί
- Η μέθοδος **shuffle** ανακατεύει την τράπουλα, η **draw** τραβάει φύλλο, η **collect** μαζεύει τα φύλλα που έχουν πέσει κάτω



Η κλάση deck: μέθοδος κατασκευής στιγμιοτύπων

```
>>> class deck():
values="A23456789TJQK" # Όλες οι αξίες
symbols="shcd" # Όλα τα σύμβολα
content=[] # Χαρτιά που βρίσκονται στην τράπουλα
pile=[] # Χαρτιά που βγήκαν από την τράπουλα
def __init__(self):
self.content=[]
self.pile=[]
for s in self.symbols:
for v in self.values:
c=card(v,s)
self.content.append(c)
```



Η κλάση deck: μέθοδος εκτύπωσης τράπουλας

```
def __str__(self):  
    s=""  
    cntr=0  
    for i in self.content:  
        s=s+str(i)+" "  
        cntr=cntr+1  
        if cntr%13==0:  
            s=s+'\n'  
    if s[-1]<>'\n':  
        s=s+'\n'  
    s=s+str(len(self.content))+"-"+str(len(self.pile))  
    return s
```



Η κλάση deck: μέθοδοι shuffle, draw, collect

```
def shuffle(self):  
    random.shuffle(self.content)  
def draw(self):  
    if len(self.content)<1: return "empty"  
    c=self.content[0]  
    self.content=self.content[1:]  
    self.pile.append(c)  
    return c  
def collect(self):  
    self.content=self.content+self.pile  
    self.pile=[]
```



Ανακατεύουμε ...

```
>>> d=deck ()
>>> import random
>>> d.shuffle ()
>>> print d
8s 2d Th 7h 9c Kd Td 9h 9s 3h 6c 5s Jc
Qd 4h Qs 5h 8c Ah Kh 8h Ks 6d Qh 7d Jh
3d 4c 8d 2c Ad 2s 2h Ts 6s 5c 9d 3c Tc
6h 4d 4s 7c As Kc 3s 7s Js 5d Jd Ac Qc
52-0
|
```



Τραβάμε φύλλο ...

```
>>> h=d.draw()
```

```
>>> print h
```

```
8s
```

```
>>> print d
```

```
2d Th 7h 9c Kd Td 9h 9s 3h 6c 5s Jc Qd
```

```
4h Qs 5h 8c Ah Kh 8h Ks 6d Qh 7d Jh 3d
```

```
4c 8d 2c Ad 2s 2h Ts 6s 5c 9d 3c Tc 6h
```

```
4d 4s 7c As Kc 3s 7s Js 5d Jd Ac Qc
```

```
51-1
```

```
>>> |
```



Μοιράζουμε 10 ακόμη φύλλα ..

```
>>> for i in range(10): h=d.draw()
```

```
>>> print d
```

```
5s Jc Qd 4h Qs 5h 8c Ah Kh 8h Ks 6d Qh
```

```
7d Jh 3d 4c 8d 2c Ad 2s 2h Ts 6s 5c 9d
```

```
3c Tc 6h 4d 4s 7c As Kc 3s 7s Js 5d Jd
```

```
Ac Qc
```

```
41-11
```

```
>>> |
```



Κληρονομικότητα κλάσεων

```
class pack(deck):  
    def __init__(self,number_of_decks=2):  
        d=deck()  
        self.content=d.content*number_of_decks
```

Η κλάση pack
κληρονομεί και
επεκτείνει την κλάση
deck

```
>>> p=pack(3)  
>>> print p  
As 2s 3s 4s 5s 6s 7s 8s 9s Ts Js Qs Ks  
Ah 2h 3h 4h 5h 6h 7h 8h 9h Th Jh Qh Kh  
Ac 2c 3c 4c 5c 6c 7c 8c 9c Tc Jc Qc Kc  
Ad 2d 3d 4d 5d 6d 7d 8d 9d Td Jd Qd Kd  
As 2s 3s 4s 5s 6s 7s 8s 9s Ts Js Qs Ks  
Ah 2h 3h 4h 5h 6h 7h 8h 9h Th Jh Qh Kh  
Ac 2c 3c 4c 5c 6c 7c 8c 9c Tc Jc Qc Kc  
Ad 2d 3d 4d 5d 6d 7d 8d 9d Td Jd Qd Kd  
As 2s 3s 4s 5s 6s 7s 8s 9s Ts Js Qs Ks  
Ah 2h 3h 4h 5h 6h 7h 8h 9h Th Jh Oh Kh  
Ac 2c 3c 4c 5c 6c 7c 8c 9c Tc Jc Qc Kc  
Ad 2d 3d 4d 5d 6d 7d 8d 9d Td Jd Qd Kd  
156-0
```



Τέλος Ενότητας

Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στο πλαίσιο του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Αθηνών**» έχει χρηματοδοτήσει μόνο την αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Σημειώματα

Σημείωμα Ιστορικού Εκδόσεων Έργου

Το παρόν έργο αποτελεί την έκδοση **1.0**.

Έχουν προηγηθεί οι κάτωθι εκδόσεις:

- Έκδοση **1.0** διαθέσιμη [εδώ](#).



Σημείωμα Αναφοράς

Copyright Πανεπιστήμιο Πατρών, Αβούρης Νικόλαος, Παλιουράς Βασίλειος, Κουκιάς Μιχαήλ, Σγάρμπας Κυριάκος. «Εισαγωγή στους Υπολογιστές Ι, Αντικειμενοστραφής προγραμματισμός στην Python». Έκδοση: 1.0. Πάτρα 2014. Διαθέσιμο από τη δικτυακή διεύθυνση:

https://eclass.upatras.gr/modules/course_metadata/opencourses.php?fc=15



Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά, Μη Εμπορική Χρήση Παρόμοια Διανομή 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



[1] <http://creativecommons.org/licenses/by-nc-sa/4.0/>

Ως **Μη Εμπορική** ορίζεται η χρήση:

- που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
- που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
- που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο

Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

Διατήρηση Σημειωμάτων

Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:

- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.



Σημείωμα Χρήσης Έργων Τρίτων

Το Έργο αυτό κάνει χρήση των ακόλουθων έργων:

Εικόνες/Σχήματα/Διαγράμματα/Φωτογραφίες

Διαφάνειες 20, 21, 22, 23, 24, 25, 26: χρήση του Python Editor IDLE

Διαφάνειες 16, 17, 18, 19: <https://www.kardwell.com/jumbo-piatnik-playing-cards.htm>

