

ΕΠΑΛ ΤΗΝΟΥ ΤΟΜΕΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ-2

ΔΡΑΣΤΗΡΙΟΤΗΤΕΣ

Δραστηριότητα εμπέδωσης παραγράφου

Δίνεται η παρακάτω κλάση:

```
class Car:
    def __init__(self, make):
        self.make=make
        self.speed = 60
    def speed_up(self,speed):
        self.speed = speed
        print "I am driving at a speed", self.speed, "km/h"
    def turn(self):
        print "I am turning ... "
```

1. Ποιος είναι ο κατασκευαστής (constructor) της κλάσης;

Κατασκευαστής της κλάσης:

```
Def __init__(self,make):
```

2. Να καταγράψετε τις ιδιότητες της κλάσης, τις μεθόδους της, καθώς και τις ενέργειες που πραγματοποιούν.

Ιδιότητες: make , speed

Μέθοδοι: speed_up,turn

Ενέργειες:

- Η speed_up αναθέτει τιμή στην ιδιότητα speed με βάση την τιμή που περνάμε σαν παράμετρο κατά την κλίση της. Στην συνεχεία εμφανίζει ανάλογο μήνυμα μαζί με την τιμή της ιδιότητας.
- Η turn εμφανίζει το μήνυμα "I am turning ...".

3. Να προσθέσετε τις ιδιότητες color και year που αντιπροσωπεύουν το χρώμα και το έτος κυκλοφορίας του αυτοκινήτου αντίστοιχα και να αρχικοποιούνται στον κατασκευαστή.

Με την προσθήκη των ιδιοτήτων color και year ο κατασκευαστής θα είναι:

```
Def __init__(self,make,color,year):  
    self.make = make  
    Self.speed = 60  
    Self.color = color  
    self.year = year
```

4. Να αλλάξετε τη μέθοδο turn, έτσι ώστε να δέχεται ως παράμετρο μια συμβολοσειρά που ορίζει αν το αυτοκίνητο θα στρίψει αριστερά ή δεξιά.

```
Def turn(self,direction):  
    Print "I am turning",direction
```

5. Να δημιουργήσετε τα παρακάτω στιγμιότυπα της κλάσης:

I. Αντικείμενο με όνομα convertible και μάρκα "bmw", χρώμα "μαύρο" και έτος κυκλοφορίας "2013".

```
Convertible=Car("bmw","Mauro","2013")
```

II. Αντικείμενο με όνομα sedan και μάρκα "toyota", χρώμα "κόκκινο" και έτος κυκλοφορίας "2009".

```
Sedan=Car("Toyota","kokkino","2009")
```

- 6.Να καλέσετε την κατάλληλη μέθοδο, ώστε το αντικείμενο convertible να στρίψει δεξιά.

```
Convertible.turn("right")
```

- 7.Να καλέσετε την κατάλληλη μέθοδο, ώστε το αντικείμενο sedan να τρέχει με 90 χιλ/ώρα.

```
Sedan.speed_up(90)
```

Παράδειγμα 2

Ορίζουμε την παρακάτω κλάση:

```
class Person(object):
    def __init__(self, name):
        self._name = name
```

Στη συνέχεια ορίζουμε την υποκλάση `Student` ως εξής:

```
class Student(Person):
    def __init__(self, name, classatt):
        super(Student, self).__init__(name)
        self.classatt = classatt
```

- Ποιες είναι οι ιδιότητες της κλάσης `Student`;
Ιδιότητες: `name, classatt`
- Δημιουργήστε στιγμιότυπο της κλάσης `Student` με όνομα "Γιάννης" και τάξη "B".
Giannis=student("Giannis", "B")
- Πώς θα εμφανίσετε τις ιδιότητες του στιγμιότυπου που δημιουργήσατε στο προηγούμενο ερώτημα;
Print Giannis.name, Giannis.classatt
- Να ορίσετε υποκλάση `Teacher` της κλάσης `Person`, η οποία εκτός από το όνομα, θα έχει επιπλέον τις ιδιότητες κλάδος (`field`) και χρόνια υπηρεσίας (`years`).

```
Class Teacher(person):
    Def __init__(self, field, years):
        Super(Teacher, self).__init__(name):
        Self.field = field
        Self.years = years
```

- Δημιουργήστε τα παρακάτω στιγμιότυπα της κλάσης `Teacher`:
--Στιγμιότυπο με όνομα "Αναστασίου", κλάδο "ΠΕ02" και χρόνια υπηρεσίας "11".

```
Anastasiou=Teacher("Anastasiou","PE02",11)
```

--Στιγμιότυπο με όνομα "Παπαχρήστου", κλάδο "ΠΕ03" και χρόνια υπηρεσίας "16".

```
Papaxristou=Teacher("Papaxristou","PE03",16)
```

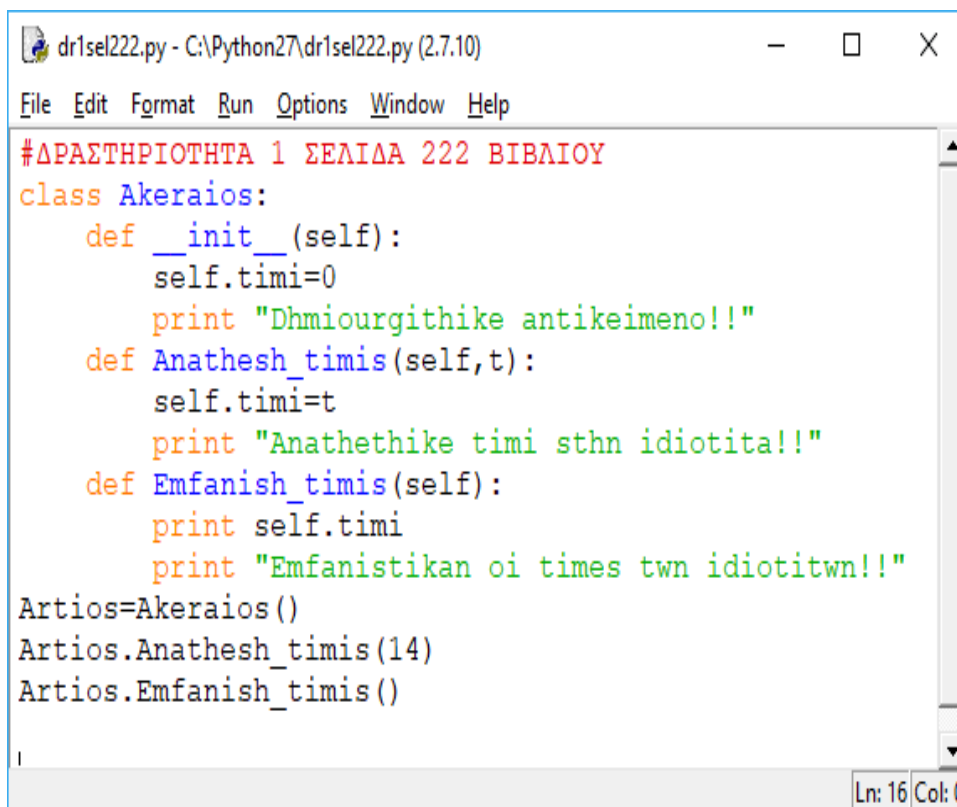
Δραστηριότητα 1

Να ορίσετε κλάση με όνομα *Akeraios*, η οποία θα έχει την ιδιότητα *timi* και τις παρακάτω μεθόδους:

I. **Anathese_timi(timi)**, η οποία θα αναθέτει τιμή στην ιδιότητα του αντικειμένου.

II. **Emfanise_timi()**, η οποία θα εμφανίζει την τιμή της ιδιότη-τας του αντικειμένου.

Στη συνέχεια να **δημιουργήσετε στιγμιότυπο** της κλάσης *Akeraios* με όνομα *Artios* και να **χρησιμοποιήσετε τις παραπάνω μεθόδους** για να δώσετε την τιμή 14 στην ιδιότητα του αντικειμένου και να την **εμφανίσετε**.



```
dr1sel222.py - C:\Python27\dr1sel222.py (2.7.10)
File Edit Format Run Options Window Help
#ΔΡΑΣΤΗΡΙΟΤΗΤΑ 1 ΣΕΛΙΔΑ 222 ΒΙΒΛΙΟΥ
class Akeraios:
    def __init__(self):
        self.timi=0
        print "Dhmiourgithike antikeimeno!!"
    def Anathesh_timis(self,t):
        self.timi=t
        print "Anathethike timi sthn idiotita!!"
    def Emfanish_timis(self):
        print self.timi
        print "Emfanistikan oi times tw n idiotitwn!!"
Artios=Akeraios()
Artios.Anathesh_timis(14)
Artios.Emfanish_timis()
|
Ln: 16 Col: 0
```

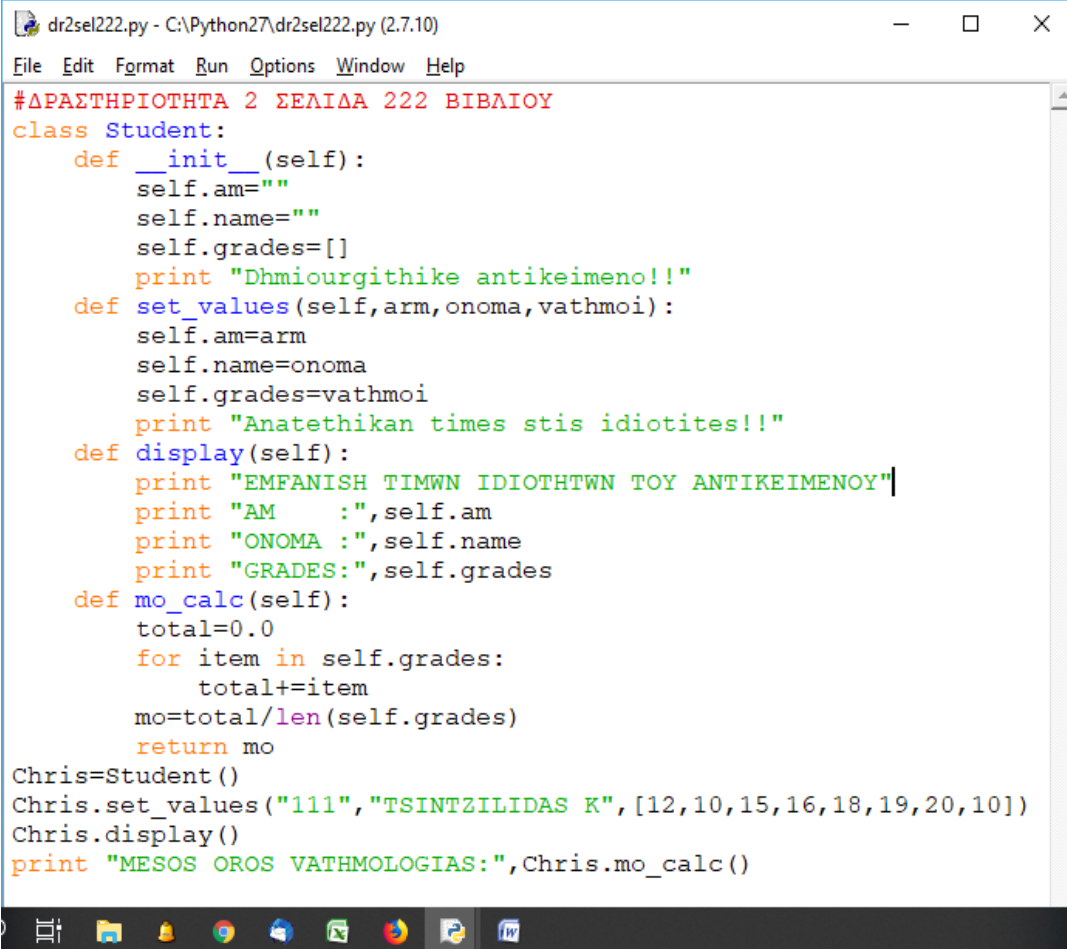
Δραστηριότητα 2

Να **ορίσετε** κλάση με όνομα Student η οποία θα έχει τρεις **ιδιότητες** για τον αριθμό μητρώου, το ονοματεπώνυμο και τους βαθμούς σε 8 μαθήματα (πίνακας).

Επίσης να **ορίσετε** τις παρακάτω **μεθόδους** της κλάσης:

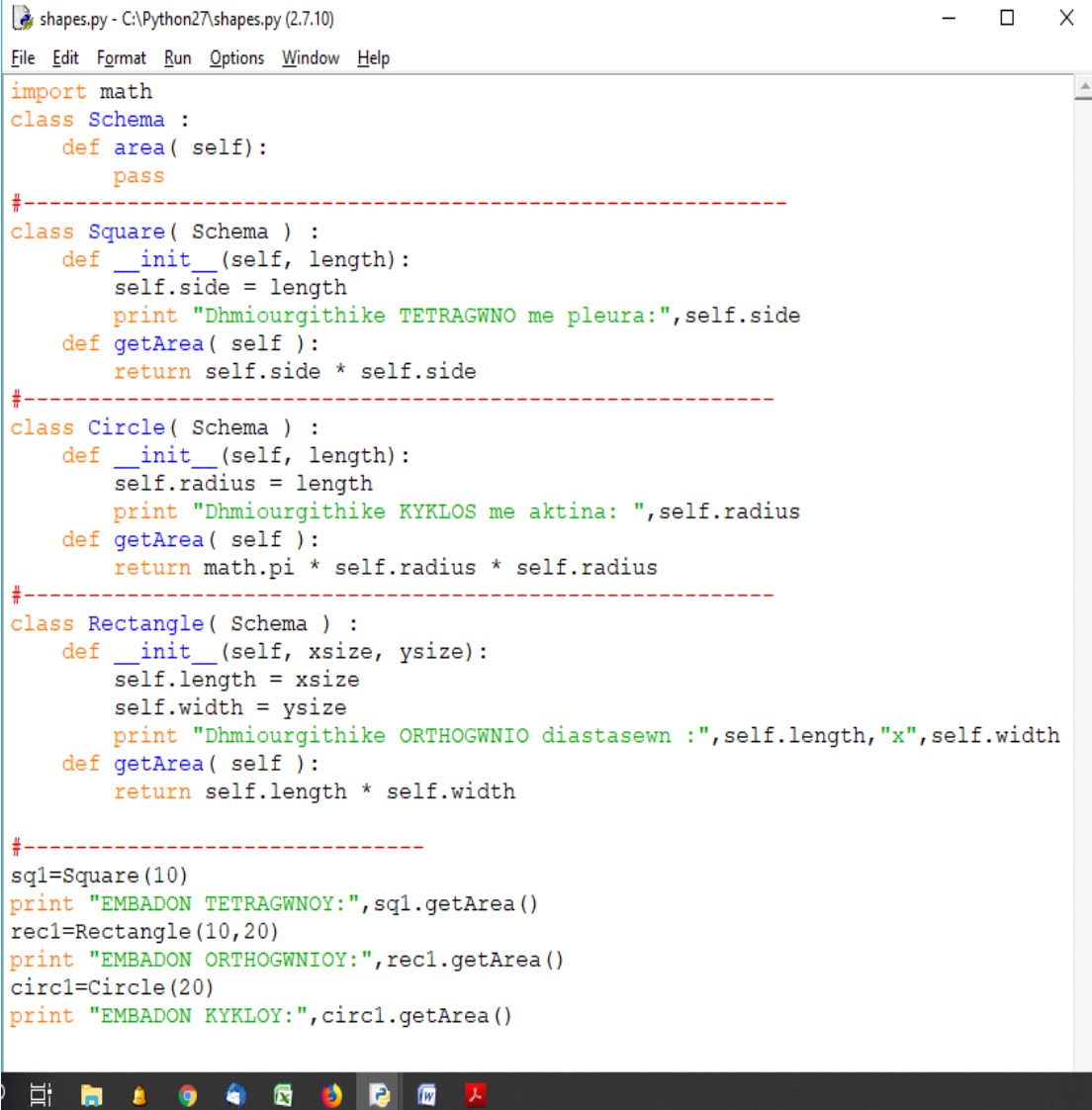
- i. Μια μέθοδο για την ανάθεση τιμών στις ιδιότητες ενός αντικειμένου.
- ii. Μια μέθοδο για την εμφάνιση των τιμών των ιδιοτήτων ενός αντικειμένου.
- iii. Μια μέθοδο για τον υπολογισμό και την επιστροφή του μέσου όρου βαθμολογίας στα 8 μαθήματα.

Στη συνέχεια να **ορίσετε** ένα **στιγμιότυπο** της κλάσης Student με όνομα Chris και να **χρησιμοποιήσετε** τις παραπάνω **μεθόδους** για να δώσετε τιμή στις ιδιότητες του αντικειμένου, να τις **εμφανίσετε** και να **υπολογίσετε** το μέσο όρο βαθμολογίας του.



```
dr2sel222.py - C:\Python27\dr2sel222.py (2.7.10)
File Edit Format Run Options Window Help
#ΔΡΑΣΤΗΡΙΟΤΗΤΑ 2 ΣΕΛΙΔΑ 222 ΒΙΒΛΙΟΥ
class Student:
    def __init__(self):
        self.am=""
        self.name=""
        self.grades=[]
        print "Dhmiourgithike antikeimeno!!"
    def set_values(self, arm, onoma, vathmoi):
        self.am=arm
        self.name=onoma
        self.grades=vathmoi
        print "Anatethikan times stis idiotites!!"
    def display(self):
        print "EMFANISH TIMWN IDIOTHYWN TOY ANTIKEIMENOU"
        print "AM      :",self.am
        print "ONOMA  :",self.name
        print "GRADES:",self.grades
    def mo_calc(self):
        total=0.0
        for item in self.grades:
            total+=item
        mo=total/len(self.grades)
        return mo
Chris=Student()
Chris.set_values("111", "TSINTZILIDAS K", [12,10,15,16,18,19,20,10])
Chris.display()
print "MESOS OROS VATHMOLOGIAS:",Chris.mo_calc()
```

Να σχεδιάσετε και να υλοποιήσετε μια ιεραρχία κλάσεων, όπου οι κλάσεις Τετράγωνο (Square), Κύκλος (Circle) και Ορθογώνιο (Rectangle) κληρονομούν από την κλάση Σχήμα (Schema). Να ορίσετε όλα τα δεδομένα και τις μεθόδους που θεωρείτε απαραίτητα για κάθε κλάση. Κάθε κλάση πρέπει να έχει και μια μέθοδο Εμβαδό (getArea) που θα υπολογίζει και θα επιστρέφει το εμβαδόν του αντίστοιχου σχήματος.



```
shapes.py - C:\Python27\shapes.py (2.7.10)
File Edit Format Run Options Window Help
import math
class Schema :
    def area( self):
        pass
#-----
class Square( Schema ) :
    def __init__(self, length):
        self.side = length
        print "Dhmiourgithike TETRAGWNO me pleura:",self.side
    def getArea( self ):
        return self.side * self.side
#-----
class Circle( Schema ) :
    def __init__(self, length):
        self.radius = length
        print "Dhmiourgithike KYKLOS me aktina: ",self.radius
    def getArea( self ):
        return math.pi * self.radius * self.radius
#-----
class Rectangle( Schema ) :
    def __init__(self, xsize, ysize):
        self.length = xsize
        self.width = ysize
        print "Dhmiourgithike ORTHOGWNIO diastasewn :",self.length,"x",self.width
    def getArea( self ):
        return self.length * self.width
#-----
sql=Square(10)
print "EMBADON TETRAGWNOY:",sql.getArea()
rec1=Rectangle(10,20)
print "EMBADON ORTHOGWNIOY:",rec1.getArea()
circl=Circle(20)
print "EMBADON KYKLOY:",circl.getArea()
```

Τετράδιο Εργασιών Δραστηριότητα 1 Σελίδα 108 (Υλοποίηση Στοίβας ως κλάση)

```
stackmeclss2.py - C:\Python27\stackmeclss2.py (2.7.10)
File Edit Format Run Options Window Help
#Υλοποίηση στοίβας με κλάση
#-----
class Stack():

    def __init__(self): # ορισμός του κατασκευαστή της στοίβας(η αντίστοιχη createstack
        self.lista = []
    #end_init
    def isEmpty(self): #μέθοδος για τον έλεγχο της στοίβας αν είναι κενή
        return self.lista == []
    #end_isEmpty
    def push(self, item): # μέθοδος για προσθήκη στοιχείου στη στοίβα
        return self.lista.append(item)
    #end_push
    def pop(self): # μέθοδος για εξαγωγή στοιχείου από την στοίβα
        return self.lista.pop()
    #end_pop
    def getElements(self): #μέθοδος για εμφάνιση της στοίβας.
        return self.lista
    #end_getElements
#-----
#main programm

stack = Stack() #δημιουργία αντικειμένου στοίβας
ar=input("Dwse stoixeio gia stoiva:")
while ar!=0:
    stack.push(ar) #εισαγωγή στοιχείου στη στοίβα
    print stack.getElements() #εμφάνιση στοίβας βηματικά
    ar=input("Dwse stoixeio gia stoiva:")#εισαγωγή στοιχείου στη στοίβα
print stack.getElements() #εμφάνιση τελικής στοίβας

while not stack.isEmpty():
    stack.pop() #εξαγωγή στοιχείου από την στοίβα
    print stack.getElements() #εμφάνιση στοίβας βηματικά
print stack.getElements() #εμφάνιση τελικής στοίβας
```


Δραστηριότητα 4.

Δίνεται το παρακάτω πρόγραμμα σε Python:

```
class Teacher:

    count = 0

    def __init__(self, name, subject):
        self.name = name
        self.subject = subject
        Teacher.count += 1

    def displayCount(self):
        print "All Teachers ", Teacher.count

    def displayTeacher(self):
        print "Name : ", self.name, ", Subject: ", self.subject

godel = Teacher("Turing", "Logic")
dijkstra = Employee("Dijkstra", "Computer Science")
godel.displayTeacher()
dijkstra.displayTeacher()
Teacher.displayCount()
```

Να καταγράψετε :

- 1) τον κατασκευαστή της κλάσης
κατασκευαστής: η συνάρτηση `__init__(self, name, subject)`
- 2) τις ιδιότητες της κλάσης
ΙΔΙΟΤΗΤΕΣ: `name, subject` (ιδιότητες αντικειμένων) ,
`count` (ιδιότητα όλης της κλάσης)
- 3) τις μεθόδους της κλάσης
ΜΕΘΟΔΟΙ: `displayCount(self)` και `displayTeacher(self)`
- 4) τα στιγμιότυπα κάθε κλάσης.
ΣΤΙΓΜΙΟΤΥΠΑ: `godel` της κλάσης `Teacher` και `dijkstra` της κλάσης `Employee`

Δραστηριότητα 5.

Δίνεται το παρακάτω πρόγραμμα:

```
class Customer:
    # Ένας πελάτης με καταθετικό λογαριασμό σε μια τράπεζα.
    # Οι πελάτες έχουν τις παρακάτω ιδιότητες:
    #   name: όνομα του πελάτη
    #   balance: το υπόλοιπο του λογαριασμού του πελάτη.

    def __init__( self, name, balance = 0.0 ):
        # επιστρέφει ένα αντικείμενο της κλάσης Customer
        # με όνομά name και αρχικό υπόλοιπο 0.0
        self.name = name
        self.balance = balance

    def withdraw(self, amount):
        # Επιστρέφει το νέο υπόλοιπο του λογαριασμού μετά την
        # ανάληψη του ποσού *amount*

        self.balance -= amount
        return self.balance

    def deposit(self, amount):
        # Επιστρέφει το νέο υπόλοιπο του λογαριασμού μετά την
        # κατάθεση του ποσού *amount*
        self.balance += amount
```

```
return self.balance
```

Να γράψετε εντολές που να κάνουν τα εξής:

- 1) Να δημιουργούν δύο αντικείμενα της κλάσης Customer, John και Mary.
- 2) Να γίνεται κατάθεση του ποσού 1400 ευρώ στο λογαριασμό του John.
- 3) Να γίνεται ανάληψη 300 ευρώ από το λογαριασμό του John.
- 4) Να γίνεται ανάληψη 400 ευρώ από το λογαριασμό της Mary.
- 5) Εμφανίζεται το υπόλοιπο των λογαριασμών

```
John = Customer("John")
Mary = Customer("Mary")
John.deposit(1400)
John.withdraw(300)
Mary.withdraw(400)
print John.balance
print Mary.balance
```