

Προετοιμασία

Στο προσωπικό σας φάκελο (ΟΧΙ ΣΤΗΝ ΕΠΙΦΑΝΕΙΑ ΕΡΓΑΣΙΑΣ παρακαλώ) φτιάξτε έναν φάκελο με το όνομα `pycl` (**python classes**)

Μέσα στον φάκελο `pycl` θα βάλουμε δύο αρχεία python:

Ένα που θα το ονομάσετε **example.py** (παράδειγμα)

Και ένα που θα το ονομάσετε με βάση το username (όνομα χρήστη) σας.

```
Αρχείο Επεξεργασία Προβολή Αναζήτηση
[haritak@flash ~]$ mkdir pycl
[haritak@flash ~]$ cd pycl
[haritak@flash pycl]$ nano example.py
[haritak@flash pycl]$ nano charitakis.py
[haritak@flash pycl]$ ls
example.py charitakis.py
[haritak@flash pycl]$
```

Κατόπιν αντιγράψτε τον παρακάτω κώδικα στο αρχείο **example.py**:

```
from charitakis import Human

a = Human()
b = Human()

a.xairetise()
b.apantise()
```

Στην θέση του `charitakis`, βάλτε το δικό σας username (όνομα χρήστη).

Απαντήστε τώρα τις ερωτήσεις:

Ποιά **αντικείμενα** βλέπετε στον παραπάνω κώδικα;

Ποιά είναι η **κλάση** των αντικειμένων;

Ποιά **μέθοδο** καλούμε σε κάθε αντικείμενο; (γράψτε το αντικείμενο και δίπλα την μέθοδο)

Ποιές **μεθόδους** πιστεύετε ότι έχει η **κλάση των αντικειμένων**;

Στην `pycl`, η δημιουργία ενός αντικειμένου γίνεται με πάρα πολύ απλό τρόπο:

```
a = Human()
```

Ο παραπάνω κώδικας δημιουργεί ένα αντικείμενο της κλάσης `Human` και του δίνει το όνομα `a`.

Αντιγράψτε από το `example.py` την γραμμή κώδικα που δημιουργεί **ένα δεύτερο αντικείμενο**:

Ποιό είναι το όνομα του δεύτερου αντικειμένου;
Ποιά είναι η κλάση του;

Στον δικό μας κώδικα, η κλάση του αντικειμένου `a` είναι `Human` και δημιουργούμε και ένα δεύτερο αντικείμενο με όνομα `b` που είναι επίσης της ίδιας κλάσης.

Δηλαδή και τα δύο αντικείμενα είναι της ίδιας κλάσης.

Υπάρχουν όμως κάποια ερωτήματα σχετικά με αυτή την άγνωστη κλάση `Human`.

Μπορείτε να τα απαντήσετε;

Που πιστεύετε ότι ορίζεται η κλάση <code>Human</code> ;
Ποιές πιστεύετε ότι είναι οι μέθοδοι που ορίζονται μέσα στην κλάση <code>Human</code> ;

Η γραμμή:

```
from charitakis import Human
```

δηλώνει στην `python` ότι θέλουμε από το `charitakis` να εισάγουμε το `Human`.

Το `charitakis` στην περίπτωση μας είναι το αρχείο `charitakis.py`.
Το `Human` στην περίπτωση μας είναι η κλάση `Human`.

Επομένως η κλάση `Human` βρίσκεται μέσα στο αρχείο με το δικό σας όνομα.

Ηρθε η ώρα να γράψουμε την δική μας κλάση!

Η κλάση `Human`



Μέσα στο **δικό σας αρχείο `py`** βάλτε τον παρακάτω κώδικα. _____

```
class Human:
    def __init__(self):
        pass

    def xairetise(self):
        print("Geia!")

    def apantise(self):
        print("Geia...")
```

Μερικές επεξηγήσεις. Βάλτε τικ σε όσες καταλαβαίνετε.

- `class` σημαίνει κλάση 
- `pass` σημαίνει μην κάνεις τίποτα, είναι μία ειδική εντολή για να μην μείνει το σώμα της μεθόδου `__init__()` άδειο 

Παρακάτω είναι τα δύο προγράμματα που μέχρι τώρα έχουμε φτιάξει.
Θυμηθείτε ότι είναι στον φάκελο pycl (python classes)
Επίσης θυμηθείτε ότι αντί για charitakis εχετε βάλει το δικό σας όνομα.

example.py	charitakis.py
<pre> from charitakis import Human a = Human() b = Human() a.xairetise() b.apantise() </pre>	<pre> class Human: def __init__(self): pass def xairetise(self): print("Geia!") def apantise(self): print("Geia...") </pre>

Μερικές ερωτήσεις επανάληψης

Τι κλάση είναι το αντικείμενο a;
Τι τύπου είναι το αντικείμενο b;
Ποιές μεθόδους έχει το αντικείμενο a;
Ποιές μεθόδους έχει το αντικείμενο b;
Για τα δύο αντικείμενα, a και b, που ορίζονται οι μεθόδοι τους;

Άσκηση 1



Προσθέστε στον κατασκευαστή (constructor) της κλάσης Human, την εντολή :

```
print("Dimiourgithika")
```



Κατόπιν **τρέξτε** το πρόγραμμά example.py.



Ποιά μέθοδο της κλάσης Human ονομάζουμε κατασκευαστή;
Πόσες print("Dimiourgithika") βάλατε ;
Πόσες φορές εμφανίστηκε το Dimiourgithika;
Γιατί εμφανίστηκε περισσότερες φορές;

def __init__(self): Ο **κατασκευαστής (constructor)** μιας κλάσης.

Εκτελείται κάθε φορά που δημιουργείται ένα αντικείμενο αυτής της κλάσης.



Ασκηση 2



Προσθέστε στην κλάση Human, μία καινούρια μέθοδο με το όνομα:

```
deikse_onoma ()
```

η οποία απλά να τυπώνει, “Με λένε Όμηρο”.

Αλλάξτε το example.py ώστε να καλείται η νέα μέθοδος τόσο για το αντικείμενο a όσο και για το αντικείμενο b.



Κατόπιν **τρέξτε** το πρόγραμμά example.py.

Στην οθόνη σας θα πρέπει να δείτε:

```
haritak@flash:/dropboxes/teemoir/Dropbox/Πρόχειρες Σημειώσεις -
Αρχείο Επεξεργασία Προβολή Αναζήτηση
[haritak@flash src]$ python example.py
Dimiourgithika!
Dimiourgithika!
Geia!
Geia...
Me lene Omiro
Me lene Omiro
[haritak@flash src]$
```

Σε ποιά **αρχείο** βάλατε τη νέα σας μέθοδο;

Μέσα σε **ποιά κλάση** βάλατε τη νέα σας μέθοδο;

Είναι η νέα μέθοδος **κοινή** για τα αντικείμενα a και b;

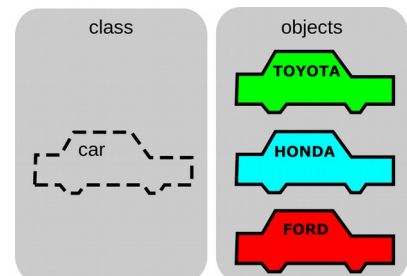
Με βάση την απάντησή σας στην προηγούμενη ερώτηση, μπορώ να κάνω την μέθοδο να εμφανίζει άλλο όνομα στο αντικείμενο a και άλλο όνομα στο αντικείμενο b; Τι πιστεύετε και γιατί;

Χαρακτηριστικά αντικειμένων

Ηρθε η ώρα κάθε **αντικείμενο** της κλάσης Human, να αποκτήσει **τα ιδιαίτερα, προσωπικά** χαρακτηριστικά του.

Κάθε αντικείμενο της κλάσης Human, πρέπει να έχει ένα name (όνομα).

Ετσι ώστε να μπορώ να δώσω το όνομα “Πελοπίδας” στο αντικείμενο a, και το όνομα “Παρμενίωνας” στο αντικείμενο b.



Οδηγίες προσθήκης χαρακτηριστικών σε κλάσεις.

<u>Βήμα</u>	<u>Περιγραφή</u>	<u>Παράδειγμα/επεξήγηση</u>
Πρώτο	Αποφασίζω πως θα ονομάσω το χαρακτηριστικό μου.	πχ θα το πω marka
Δεύτερο	Για να γίνει η προσθήκη του χαρακτηριστικού, σε κάθε αντικείμενο της κλάσης μου, θα πρέπει να την κάνω πάνω στο self .	το <code>self</code> είναι το αντικείμενο για το οποίο εκτελείται η μέθοδος της κλάσης μου. (αυτό πρέπει να το διαβάσετε μερικές φορές για να το καταλάβετε)
Τρίτο	Βάζω στο χαρακτηριστικό του <code>self</code> την αρχική τιμή.	<code>self.marka = "Pony"</code>
Τέταρτο	Για να γίνεται αυτό για κάθε αντικείμενο της κλάσης μου, πρέπει να το κάνω μέσα στον... Κ _ _ _ _ _ ή (συμπληρώστε τα κενά παρακαλώ) της κλάσης μου.	<pre>def __init__(self): self.marka = "Pony"</pre>

Συμπληρώστε τον παρακάτω πίνακα με βάση τις οδηγίες, ώστε στην κλάση `Human` να προσθέσουμε το χαρακτηριστικό `name`.

Άσκηση

<u>Βήμα</u>	<u>Περιγραφή</u>	<u>Συμπληρώστε αυτή την στήλη</u>
Πρώτο	Αποφασίζω πως θα ονομάσω το χαρακτηριστικό μου.	
Δεύτερο	Για να γίνει η προσθήκη του χαρακτηριστικού, σε κάθε αντικείμενο της κλάσης μου, θα πρέπει να την κάνω πάνω στο self .	
Τρίτο	Βάζω στο χαρακτηριστικό του <code>self</code> την αρχική τιμή.	
Τέταρτο	Για να γίνεται αυτό για κάθε αντικείμενο της κλάσης μου, πρέπει να το κάνω μέσα στον κατασκευαστή της κλάσης μου.	



Κάντε τις παραπάνω αλλαγές στην κλάση `Human`.

Στο τέλος ο κώδικάς σας θα πρέπει να είναι κάπως έτσι:



```
class Human:
    def __init__(self):
        print("Dimiourgithika!")
        self.name = "Πελοπίδας"

    def xairetise(self):
        print("Geia!")

    def apantise(self):
        print("Geia...")

    def deikse_onoma(self):
        print("Me lene Omiro")
```



Κατόπιν **τρέξτε** το πρόγραμμά example.py.



Στην οθόνη σας θα πρέπει να δείτε:

```
haritak@flash:/dropboxes/teemoir/Dropbox/Πρόχειρες Σημειώσεις -
Αρχείο Επεξεργασία Προβολή Αναζήτηση
[haritak@flash src]$ python example.py
Dimiourgithika!
Dimiourgithika!
Geia!
Geia...
Me lene Omiro
Me lene Omiro
[haritak@flash src]$ █
```

Επ; Γιατί γράφει ακόμα ότι τον λένε Omiro;



Διορθώστε την μέθοδο deikse_onoma, ώστε να δείχνει το self.name αντί του Omirou.



Κατόπιν **τρέξτε** το πρόγραμμά example.py.



Στην οθόνη σας θα πρέπει να δείτε:

```
haritak@flash:/dropboxes/teemoir/Dropbox/Πρόχειρες Σημειώσεις -
Αρχείο Επεξεργασία Προβολή Αναζήτηση
[haritak@flash src]$ python example.py
Dimiourgithika!
Dimiourgithika!
Geia!
Geia...
Me lene Πελοπίδας
Me lene Πελοπίδας
[haritak@flash src]$ █
```



Σε αυτό το σημείο, τι έχουμε πετύχει και τι δεν έχουμε πετύχει; Βάλτε **TIK** ανάλογα.

Στόχος	Το πετύχαμε	Δεν το πετύχαμε

Να προσθέσουμε ένα νέο χαρακτηριστικό στην κλάση Human.		
Να ονομάσουμε αυτό το νέο χαρακτηριστικό name.		
Όλα τα αντικείμενα της κλάσης Human να έχουν το χαρακτηριστικό name.		
Σε κάθε αντικείμενο να μπορώ να δώσω ότι όνομα θέλω.		

Μπορεί να προσθέσαμε το χαρακτηριστικό name, αλλά εξακολουθούν όλα τα αντικείμενα να έχουν το ίδιο όνομα.

Ιδανικά θα θέλαμε να μπορούμε να κάνουμε το εξής:

```
a = Human("Πελοπίδας")
b = Human("Παρμενίωνας")
```

Δηλαδή, κατά την κατασκευή των αντικειμένων, να γίνεται και η βάφτισή τους. Να τους δίνω τιμή για το όνομα.

Τι πιστεύετε ότι χρειάζεται να κάνουμε από τα παρακάτω; Απαντήστε με ΝΑΙ ή ΟΧΙ.

	Χρειάζεται;
1. Να προσθέσω μία νέα μέθοδο στην κλάση Human ώστε να δίνει ότι τιμή θέλω στο χαρακτηριστικό name.	
2. Να προσθέσω άλλο ένα χαρακτηριστικό στην κλάση Human	
3. Να δέχεται ο κατασκευαστής μία νέα παράμετρο και αυτή την παράμετρο να την περνάει ως τιμή στο χαρακτηριστικό name.	

Πέρασμα παραμέτρων από τον κατασκευαστή

Ας δοκιμάσουμε την 3η επιλογή :

3. Να δέχεται ο κατασκευαστής μία νέα παράμετρο και αυτή την παράμετρο να την περνάει ως τιμή στο χαρακτηριστικό name.	
---	--



Προσθέστε στον κατασκευαστή της κλάσης Human, μία νέα παράμετρο με το όνομα n:

```
def __init__(self, n):
```

Μετά, δώστε την τιμή n, στο χαρακτηριστικό self.name:

```
self.name = n
```



Κατόπιν **τρέξτε** το πρόγραμμα example.py.

Στην οθόνη σας θα πρέπει να δείτε:



```

haritak@flash:/dropboxes/teemoir/Dropbox/Πρόχειρες Σημειώσεις - Φύλλα εργασίας/Γ - Ειδικά θέματα/classes in p
Αρχείο Επεξεργασία Προβολή Αναζήτηση Τερματικό Βοήθεια
[haritak@flash src]$ python example.py
Traceback (most recent call last):
  File "example.py", line 4, in <module>
    a = Human()
TypeError: __init__() missing 1 required positional argument: 'n'
[haritak@flash src]$ █

```

Δηλαδή υπάρχει κάποιο πρόβλημα!

Τι πιστεύετε ότι σημαίνει το πρόβλημα:

TypeError: __init__() missing 1 required positional argument: 'n'

(μετάφραση: Λάθος Τύπου: __init__() λείπει 1 απαραίτητη παράμετρος θέσης: 'n')

Απο πού λείπει η παράμετρος;

Πράγματι, η `__init__()` χρειάζεται τώρα μία επιπλέον παράμετρο την `n`. Εμείς όμως στο `example.py`, δεν της την δίνουμε!



Εντοπίστε στο `example.py` που καλείται ο κατασκευαστής.

Κατόπιν, στο σημείο που καλείται, δώστε τιμή στο `n`.

Στην μία φορά που καλείται δώστε για τιμή το “Πελοπίδας”, και στην άλλη “Παρμενίωνας”



Κατόπιν **τρέξτε** το πρόγραμμά `example.py`.

Στην οθόνη σας θα πρέπει να δείτε:

```

haritak@flash:/dropboxes/teemoir/Dropbox/Πρόχειρες Σημειώσεις -
Αρχείο Επεξεργασία Προβολή Αναζήτηση
[haritak@flash src]$ python example.py
Dimiourgithika!
Dimiourgithika!
Geia!
Geia...
Me lene Πελοπίδας
Me lene Παρμενίωνας
[haritak@flash src]$ █

```

Σε αυτό το σημείο ο κώδικάς σας θα πρέπει να είναι κάπως έτσι:
(θυμηθείτε αντί `charitakis.py` θα είναι το δικό σας όνομα)

`example.py`

`charitakis.py`

Όνοματεπώνυμο: _____

<pre> from charitakis import Human a = Human("Πελοπίδας") b = Human("Παρμενίωνας") a.xairetise() b.apantise() a.deikse_onoma() b.deikse_onoma() </pre>	<pre> class Human: def __init__(self,n): print("Dimiourgithika!") self.name = n def xairetise(self): print("Geia!") def apantise(self): print("Geia...") def deikse_onoma(self): print("Me lene", self.name) </pre>
---	---

Και τέλος μερικές ερωτήσεις επανάληψης:

Με βάση τον παραπάνω κώδικα, γράψτε ποιές **μεθόδους** έχει το αντικείμενο a:

Με βάση τον παραπάνω κώδικα, γράψτε ποιές **μεθόδους** έχει το αντικείμενο b:

Με βάση τον παραπάνω κώδικα, γράψτε ποιά **χαρακτηριστικά** έχει το αντικείμενο a:

Με βάση τον παραπάνω κώδικα, γράψτε ποιά **χαρακτηριστικά** έχει το αντικείμενο b:

Έχει το αντικείμενο a τις ίδιες μεθόδους με το αντικείμενο b;

Έχει το αντικείμενο a τα ίδια χαρακτηριστικά με το αντικείμενο b;

Έχουν τα χαρακτηριστικά του αντικειμένου a τις ίδιες τιμές με αυτά του αντικειμένου b;

Τελικά τα αντικείμενα a και b σε ποιά σημεία διαφέρουν και σε ποιά σημεία είναι ίδια;