

Αντικειμενοστρεφής Προγραμματισμός–Python

Κ.Π. Γιαλούρης

Στόχοι του σημερινού μαθήματος

- Κατανόηση της έννοιας του Τμηματικού Προγραμματισμού στη Python
- Γνώση δήλωσης και κλήσης των συναρτήσεων οριζόμενων από τον χρήστη

Τμηματικός Προγραμματισμός

- ❑ **Ο Τμηματικός Προγραμματισμός έχει ως βασικό στοιχείο το** χωρισμό ενός προγράμματος σε ανεξάρτητες λογικές ενότητες - τμήματα (υποπρογράμματα).
- ❑ Τα τμήματα αυτά πρέπει να είναι όσο το δυνατόν μικρότερα, ώστε να διευκολύνεται ο καταμερισμός του προγράμματος σε μικρά τμήματα τα οποία είναι ευκολότερο να διορθωθούν.

Τμηματικός Προγραμματισμός

- ❑ Με τον Τμηματικό Προγραμματισμό επιταχύνεται η ανάπτυξη μιας εφαρμογής μια και τα διάφορα υποπρογράμματα μπορούν να αναπτύσσονται παράλληλα από δύο ή περισσότερους προγραμματιστές.
- ❑ Στην Python ο Τμηματικός Προγραμματισμός μπορεί να επιτευχθεί με χρήση συναρτήσεων.

Τμηματικός Προγραμματισμός

- Ο χωρισμός ενός προγράμματος σε συναρτήσεις στην Python, έχει ως βασική προϋπόθεση:
 - Την καλή ανάλυση του προβλήματος,
 - τη διάσπασή του σε μικρότερα προβλήματα τα οποία μπορούν να αντιμετωπισθούν από πλευράς προγραμματισμού διακριτά και αυτόνομα με στόχο κάθε μικρότερο πρόβλημα να επιλύεται με μία συνάρτηση.

Συναρτήσεις οριζόμενες από τον χρήστη

- ❑ Μία συνάρτηση οριζόμενη από τον χρήστη, χαρακτηρίζεται και ως υποπρόγραμμα, είναι ένα σύνολο εντολών που εκτελούν μία συγκεκριμένη λειτουργία και η κλήση τους στο κύριο πρόγραμμα επιτυγχάνεται μέσω του ονόματός τους.
- ❑ Οι συναρτήσεις οριζόμενες από τον χρήστη ομοιάζουν με ενσωματωμένες συναρτήσεις της Python.

Συναρτήσεις οριζόμενες από τον χρήστη

- ❑ Μία συνάρτηση καλείται από το κύριο πρόγραμμα ή από άλλη συνάρτηση με ευθύνη του προγραμματιστή.
- ❑ Για να μπορεί μια συνάρτηση να χρησιμοποιηθεί, πρέπει να έχει προηγουμένως δηλωθεί.
- ❑ Κάθε συνάρτηση καλείται με το όνομά της και εφόσον υπάρχουν παράμετροι ακολουθούν μέσα σε παρενθέσεις.

Η βασική σύνταξη μίας συνάρτησης

```
def ονομαΣυνάρτησης (ορισμα1, ορισμα2, ..., ορισμαn) :  
    εντολη-1  
    εντολη-2  
    .....  
    εντολη-ν  
return val1, val2, ..valn
```


Συναρτήσεις οριζόμενες από τον χρήστη

- Μία συνάρτηση
 - Καλείται με το όνομά της.
 - Μπορεί να δεχθεί ένα ή περισσότερα ορίσματα.
 - Μπορεί να μην επιστρέφει καμία τιμή στο κύριο πρόγραμμα.
 - Μπορεί να επιστρέφει στο κύριο πρόγραμμα μία περισσότερες τιμές. Όταν επιστρέφει μία ή περισσότερες τιμές η κλήση της συνάρτησης γίνεται μέσω εντολής ανάθεσης τιμής. Η επιστροφή των τιμών επιτυγχάνεται μέσω της εντολής `return`.

Συναρτήσεις οριζόμενες από τον χρήστη

- Σε μία συνάρτηση μπορεί να μεταβιβάζονται από το κύριο πρόγραμμα ή από μία άλλη συνάρτηση μία σειρά από μεταβλητές. Η μεταβίβαση των μεταβλητών αυτών επιτυγχάνεται μέσω των της λίστας των ορισμάτων/παραμέτρων που καθορίστηκαν κατά τη δημιουργία της συνάρτησης. Τα ορίσματα αυτά μπορούμε να τα χαρακτηρίσουμε ως μεταβλητές εισόδου στη συνάρτηση.

Συναρτήσεις οριζόμενες από τον χρήστη

- Στη συνάρτηση μπορούμε να ορίζουμε μεταβλητές όπως και στο κύριο πρόγραμμα, οι μεταβλητές αυτές δεν είναι ορατές στο πρόγραμμα που καλεί τη συνάρτηση. Οι μεταβλητές αυτές είναι τοπικές της συνάρτησης.
- Αντίθετα οι μεταβλητές που έχουν ορισθεί στο κύριο πρόγραμμα είναι ορατές στη συνάρτηση. Βέβαια δεν είναι σωστή τακτική να χρησιμοποιούμε μεταβλητές του κύριου προγράμματος μέσα στη συνάρτηση χωρίς να περάσουν ως ορίσματα της συνάρτησης επειδή η συνάρτηση θα είναι εξαρτημένη από το πρόγραμμα που την καλεί.

Πλεονεκτήματα συναρτήσεων

- ❑ Καταργούν την επανάληψη γραμμών κώδικα
- ❑ Βελτιώνουν την αναγνωσιμότητα του προγράμματος
- ❑ Απλοποιούν την ανάπτυξη και την συντήρηση του προγράμματος
- ❑ Αυξάνουν την ταχύτητα ανάπτυξης ενός προγράμματος

Η βασική σύνταξη μίας συνάρτησης

- ❑ Η σύνταξη της συνάρτησης αρχίζει με τη λέξη κλειδί `def`. Στη συνέχεια ακολουθεί το όνομα της συνάρτησης.
- ❑ Ως όνομα συνάρτησης είναι ένα οποιοδήποτε έγκυρο όνομα για την Python όπως αναφέρθηκε στον ορισμό των μεταβλητών.
- ❑ Μέσα σε παρενθέσεις αναγράφονται οι παράμετροι /ορίσματα της συνάρτησης. Τα ορίσματα αυτά θα χρησιμοποιηθούν για περάσουν τιμές στη συνάρτηση και θεωρούνται τοπικές μεταβλητές της συνάρτησης.
- ❑ Η σύνταξη της συνάρτησης κλείνει με τη εντολή `return` που μπορεί να ακολουθείται από μεταβλητές που επιστρέφουν τιμές στο κύριο πρόγραμμα.

Συνάρτηση χωρίς επιστροφή τιμών

```
def ονομασυνάρτησης (ορίσματα) :
```

```
    Εντολη-1
```

```
    Εντολη-2
```

```
    .....
```

```
    Εντολη-ν
```

Συνάρτηση με επιστροφή μίας τιμής

```
def ονομαΣυνάρτησης (ορίσματα) :  
    εντολή-1  
    εντολή-2  
    .....  
    εντολή-ν  
return val1
```

Συνάρτηση με επιστροφή πολλών τιμών

```
def ονομαΣυνάρτησης (ορίσματα) :  
    εντολή-1  
    εντολη-2  
    .....  
    εντολή-ν  
return val1, val2, ..., valk
```


Παράδειγμα-1

- Να γραφεί πρόγραμμα το οποίο να διαβάζει ένα ακέραιο αριθμό n και να υπολογίζει το παραγοντικό του αριθμού αυτού $n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot n$

Λύση

```
def parag(k):  
    par=1  
    if k==0:  
        par=1  
    else:  
        for m in range(1,k+1):  
            par=par*m  
    return par
```

```
n=int(input("δώσε το n:"))  
n=abs(n)  
paragontiko=parag(n)  
print(paragontiko)
```

Παράδειγμα-2

- Να γραφεί πρόγραμμα στο οποίο δίνεται το **κόστος κατασκευής** ενός προϊόντος και υπολογίζει μέσω συνάρτησης.
 - Τον ΦΠΑ 24% επί του κόστους κατασκευής
 - Εισφορά ασφάλισης 27% επί του κόστους κατασκευής
 - Φορός 35% επί του κόστους κατασκευής
 - Κέρδος 30% επί της συνολικού κόστους (κόστος κατασκευής + φόροι κλπ)
 - Την τελική αξία πώλησης του προϊόντος

Λύση –δήλωση συνάρτησης

```
def ypologismos(axia_katask):  
    pos_fpa=0.24  
    pos_forou=0.35  
    pos_asf=0.27  
    pos_kerd=0.35  
    poso_fpa = axia_katask * pos_fpa  
    foros= axia_katask * pos_forou  
    asfal=axia_katask*pos_asf  
    syn_axia= axia_katask + poso_fpa + asfal+foros  
    kerd=pos_kerd*syn_axia  
    teliki=syn_axia+kerd  
return syn_axia,teliki,poso_fpa,foros,asfal,kerd
```

Λύση – κύριο πρόγραμμα

```
cost=input("κόστος κατασκευής ")
cost=float(cost)

s_cost,axia_pol, p_fpa,
p_forou,asfalisi,kerdos=ypologismos(cost)

print("Συνολικό Κοστος:{} ".format(s_cost))
print("ΦΠΑ:{} ".format(p_fpa))
print("Ασφ.Εισφ:{} ".format(asfalisi))
print("Αξία πώλησης:{} ".format(axia_pol))
print("Κέρδος:{} ".format(kerdos))
```

Αναδρομικές συναρτήσεις

- Αναδρομική συνάρτηση (**recursive function**) είναι μία συνάρτηση η οποία καλεί τον εαυτό της.
- Δηλαδή η συνάρτηση αυτή ορίζεται κάνοντας χρήση την ίδια την συνάρτηση.
 - Πχ
 - Ορισμός δύναμης :
 - $x^n = x^{n-1} \cdot x$, με αρχική τιμή $x^0 = 1$

Αναδρομική συνάρτηση υπολογισμού δύναμης

```
def dynam_i (k, n) :  
    if n==0 :  
        dyn=1  
    else :  
        dyn=k*dynam_i (k, n-1)  
    return dyn
```

Προσοχή

- Κάθε φορά που γίνεται κλήση μίας αναδρομικής συνάρτησης δεσμεύεται νέος χώρος στην μνήμη.
- Αυτό έχει ως αποτέλεσμα να υπάρχει περίπτωση να μην υπάρχει διαθέσιμη μνήμη όταν υπάρχει μεγάλος αριθμός αναδρομικών κλήσεων της συνάρτησης.