

## Σημειωματάριο Δευτέρας 20 Νοε. 2017

### Ένα πρόγραμμα "τηλεφωνικός κατάλογος"

Σήμερα φτιάξαμε μια πρώτη μορφή ενός σχετικά μεγάλου προγράμματος που σκοπό έχει να κρατάει και να διαχειρίζεται τηλέφωνα και άλλη πληροφορία για κάποια άτομα. Μας δίνει την ευκαιρία το πρόγραμμα αυτό να χρησιμοποιήσουμε λεξικά και να χρησιμοποιήσουμε και γράψιμο/διάβασμα σε αρχεία.

Η βασική πληροφορία του προγράμματος είναι ένα λεξικό με το όνομα `Person` στο οποίο τα κλειδιά είναι τα μικρά ονόματα των ατόμων (ή άλλη σύντομη περιγραφή τους) και οι τιμές του κάθε κλειδιού είναι μια λίστα με όλη την πληροφορία που κρατάμε για το άτομο. Σε πρώτη φάση η λίστα αυτή έχει δύο στοιχεία. Το πρώτο είναι το τηλέφωνο του ατόμου (το οποίο το κρατάμε ως `string` και όχι ως αριθμό) και το δεύτερο είναι το επώνυμο του ατόμου.

Η δομή του προγράμματος όπως τη διαμορφώσαμε σιγά-σιγά κατά τη διάρκεια του δώρου είναι ότι κατ' αρχήν υπάρχει μια συνάρτηση `main` η οποία καλεί συνεχώς μια συνάρτηση `menu`. Η συνάρτηση `menu` διαχειρίζεται την αλληλεπίδραση με το χρήστη σε ψηλό επίπεδο και ανάλογα με τις επιθυμίες του χρήστη καλεί άλλες συναρτήσεις που διεκπεραιώνουν αυτές τις δουλειές. Όλες αυτές οι συναρτήσεις αναφέρονται ρητά μέσα στη συνάρτηση `menu`.

Διαβάστε το πρόγραμμα με προσοχή διαβάζοντας τα σχόλια που υπάρχουν εκεί μέσα για να καταλάβετε πώς λειτουργεί.

Για να το τρέξετε θα πρέπει να έχετε και το αρχείο

<http://fourier.math.uoc.gr/~mk/prog1718/files/mydata.txt> (<http://fourier.math.uoc.gr/~mk/prog1718/files/mydata.txt>)

στον ίδιο κατάλογο με το πρόγραμμα αυτό.

```
In [ ]: # Η επόμενη γραμμή αφορά το καθάρισμα της οθόνης και δουλεύει μόνο στο περιβάλλον προγραμματισμού που
# χρησιμοποιώ στο μάθημα. Σε άλλα περιβάλλοντα θα έχει άλλη μορφή (δείτε π.χ. το πρόγραμμα κρεμάλας πο
# είχαμε γράψει πριν από μερικές εβδομάδες).
from IPython.display import clear_output

filename="mydata.txt" # Σε αυτό το αρχείο αποθηκεύουμε το λεξικό με τη συνάρτηση save() και από αυτό το α
ρχείο
                # φορτώνουμε το λεξικό με τη συνάρτηση load()

# Στη μεταβλητή Person (τύπου λεξικού) θα βρίσκεται αποθηκευμένη όλη η πληροφορία στο πρόγραμμά μας.
# Με άλλα λόγια αυτή είναι η "βάση δεδομένων" μας.
# Αρχικά είναι άδειο το λεξικό.
```

```

Person={}

# Αυτή είναι η κύρια συνάρτηση και η μόνη συνάρτηση που καλείται από το κυρίως πρόγραμμα.
def main():
    load() # Φορτώνουμε τα δεδομένα που είναι αποθηκευμένα στο αρχείο και γεμίζουμε τη μεταβλητή Person
    while True: # Τρέχουμε συνεχώς στη συνάρτηση menu εκτός αν αυτή επιστρέψει 1 που είναι σημάδι ότι
        ret = menu() # ο χρήστης ζήτησε τερματισμό του προγράμματος
        if ret == 1:
            break
        input("Πάτα enter για να συνεχίσεις: ") # Αν αφαιρεθεί αυτή η γραμμή τότε η οθόνη καθαρίζει πολύ
            # γρήγορα και ο χρήστης δεν προλαβαίνει να δει τι τυπώθηκε από αυτά που ζήτησε.

# Η συνάρτηση menu διαχειρίζεται την αλληλεπίδραση με το χρήστη σε ψηλό επίπεδο.
# Ρωτάει το χρήστη τι θέλει να κάνει και καλεί τις κατάλληλες συναρτήσεις.
def menu():
    # Επιστρέφει 0 αν θέλουμε να συνεχίσει να καλείται
    # και 1 αν θέλουμε να σταματήσει να καλείται
    clear_output() # καθαρίζουμε την οθόνη
    print("-----")
    print("Επιλέξτε")
    print(" s: Δείξτε όλο τον τηλεφωνικό κατάλογο")
    print(" ?: Αναζήτηση ατόμου")
    print(" i: Προσθήκη ατόμου")
    print(" e: Τροποποίηση ατόμου")
    print(" d: Σώστε τον κατάλογο στο δίσκο")
    print(" l: Φορτώστε τον κατάλογο από το δίσκο")
    print(" q: Κλείσιμο του προγράμματος")
    print("-----")
    # Η επιλογή του χρήστη υποδεικνύεται με ένα γράμμα, μοναδικό ανά εντολή.
    choice = input(" Η επιλογή σας: ") # Ζητάμε από το χρήστη να επιλέξει λειτουργία.
    if choice == "s":
        show() # Δείχνουμε όλες τις εγγραφές στη βάση δεδομένων μας
        return 0
    if choice == "?":
        search() # Αναζητούμε μια εγγραφή στη βάση
        return 0
    if choice == "i":
        insert() # Προσθήκη μιας εγγραφής
        return 0
    if choice == "e":
        edit() # Τροποποίηση μιας εγγραφής
        return 0

```

```

if choice == "d":
    save() # Αποθήκευση του λεξικού στο αρχείο
    return 0
if choice == "l":
    load() # Φόρτωμα του λεξικού από το αρχείο
    return 0
if choice == "q":
    return 1 # Τέλειωμα του προγράμματος. Μόνο τότε επιστρέφει 1.

# Η συνάρτηση αυτή, που δεν καλείται από την menu, αλλά από άλλες συναρτήσεις
# δείχνει απλά μια εγγραφή της βάσης στο χρήστη
def showperson(name, lastname, phone):
    print("*** {}: Επώνυμο: {} Αριθμ. Τηλ. {}".format(name, lastname, phone))

# Δείχνει όλες τις εγγραφές στο χρήστη (αλφαβητική σειρά κλειδιού)
def show():
    for p in sorted(Person.keys()):
        lastname = Person[p][1]
        phone = Person[p][0]
        showperson(p, lastname, phone)

# Αναζήτηση κλειδιού στη βάση
def search():
    k = input("Δώστε το μικρό όνομα του ατόμου: ").strip() # Ο ρόλος του strip είναι να απαλλαγούμε από
    if k not in Person.keys(): # κενούς χαρακτήρες στο τέλος της απάντησης του χρήστ
η
        print("Δεν υπάρχει τέτοιο όνομα")
        return
    L = Person[k] # Το βρήκαμε και φέρνουμε τα στοιχεία του από τη βάση.
    lastname = L[1]; phone = L[0]
    showperson(k, lastname, phone) # Τα δείχνουμε

# Εισαγωγή νέου ατόμου στη βάση
def insert():
    name = input("Δώστε το μικρό όνομα για καταχώρηση: ").strip()
    if name in Person.keys():
        print("Υπάρχει ήδη. Sorry.") # Το όνομα που ζήτησε να εισαχθεί υπάρχει ήδη
        return
    lastname = input("Δώστε το επώνυμο: ") # Νέο όνομα. Ζητάμε από το χρήστη τα στοιχεία του.
    phone = input("Δώστε το τηλέφωνο: ")
    Person[name] = [ phone, lastname ] # Νέα εγγραφή στη βάση.

```

```

# Τροποποίηση στοιχείων ατόμου
def edit():
    name = input("Δώστε το μικρό όνομα για τροποποίηση: ").strip()
    if name not in Person.keys():
        print("Δεν υπάρχει. Sorry.")
        return
    lastname = Person[name][1]
    phone = Person[name][0]
    print("Τα στοιχεία του {} είναι: {}".format(name)) # Δείχνουμε τα υπάρχοντα στοιχεία στο χρήστη
    print("  Επώνυμο: {}".format(lastname))
    print("  Τηλέφωνο: {}".format(phone))
    newlastname = input("Δώστε νέο επώνυμο (enter για ίδιο): ").strip() # Ζητάμε τα νέα στοιχεία
    newphone = input("Δώστε νέο τηλέφωνο (enter για ίδιο): ").strip()
    if newlastname=="": # Αν ο χρήστης πάτησε απλά enter τότε διατηρούμε τα παλιά στοιχεία.
        newlastname = lastname
    if newphone == "": # Αν ο χρήστης πάτησε απλά enter τότε διατηρούμε τα παλιά στοιχεία.
        newphone = phone
    Person[name] = [ newphone, newlastname] # Τροποποιούμε την τιμή του κλειδιού name στο λεξικό.

# Αποθήκευση του λεξικού στο αρχείο με όνομα filename
def save():
    f = open(filename, "w") # Ανοίγουμε το αρχείο για γράψιμο
    N = len(Person)
    print(N, file=f) # Στην πρώτη γραμμή γράφουμε το πλήθος των εγγραφών (για πιο εύκολο διάβασμα)
    for p in Person.keys(): # Για κάθε εγγραφή στη βάση μας γράφουμε τρεις διαδοχικές γραμμές στο αρχείο
        print(p, file=f) # Στην πρώτη το όνομα
        print(Person[p][1], file=f) # Στη δεύτερη το επώνυμο
        print(Person[p][0], file=f) # Στην τρίτη το τηλέφωνο
    f.close() # Κλείνουμε το αρχείο

# Γεμίζουμε το λεξικό μας από το αρχείο (που είναι γραμμένο όπως το γράφει η συνάρτηση load).
def load():
    global Person # Εδώ πρέπει να δηλωθεί το Person ως global μεταβλητή αλλιώς η συνάρτηση load τη θεωρεί τοπική.
    # Ο λόγος είναι ότι η πρώτη πρόσβαση που γίνεται στην Person είναι εγγραφή και όχι ανάγνωση.
    # Δε θα έβλαπτε γενικά να δηλώναμε την Person ως global μεταβλητή σε κάθε συνάρτηση σε αυτό το πρόγραμμα.
    f = open(filename, "r") # Ανοίγουμε το αρχείο για διάβασμα.
    # ***** Ως έχει θα αποτύχει και θα παρουσιάσει σφάλμα αν το αρχείο με όνομα filename δεν υπάρχει
    # ***** ή, για κάποιο άλλο λόγο, δε μπορεί να ανοιχθεί..
    L = [ s.strip() for s in f.readlines() ] # Διαβάζουμε όλες τις γραμμές του αρχείου στη λίστα L.

```