



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

Εισαγωγή στους Υπολογιστές

Εργαστήριο 10

Καθηγητές: Αβούρης Νικόλαος, Παλιουράς Βασίλης, Κουκιάς Μιχαήλ, Σγάρμπας Κυριάκος

Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών

ΑΝΟΙΚΤΑ ακαδημαϊκά **ΠΠ**
μαθήματα

Εργαστήριο 10: Άσκηση Προγραμματισμού Python

10.1 Γραφικές Διεπαφές με τη χρήση του Tkinter

Το module Tkinter μας διευκολύνει στην κατασκευή και στη διαχείριση γραφικών διεπαφών. Σύμφωνα με το προγραμματιστικό μοντέλο που βασίζεται στο Tkinter, τα παράθυρα είναι αντικείμενα πάνω στα οποία τοποθετούμε άλλα αντικείμενα, όπως για παράδειγμα κουμπιά, πεδία εισόδου, κείμενα, εικόνες. Τα αντικείμενα αυτά είναι τα widgets. Στο πρόγραμμα μας η δημιουργία και η διαχείριση των widgets γίνεται δημιουργώντας κατάλληλα στιγμιότυπα κλάσεων, τις οποίες παρέχει το Tkinter.

10.2 Δημιουργία βασικού παραθύρου

Να εκτελέσετε τον κάτωθι κώδικα

```
from Tkinter import *
root = Tk()
root.wm_geometry("400x300+20+40")
Label(root, text = "Hello world!").pack()
Button(root, text="Done", command=root.destroy).pack()
root.update()
root.mainloop()
```

Το root είναι στιγμιότυπο της κλάσης Tk. Στη συνέχεια του προγράμματος, η λέξη root χρησιμοποιείται ως αναφορά στο παράθυρο, το οποίο θα διαχειριστούμε. Με τη μέθοδο wm_geometry δηλώνουμε τις διαστάσεις του παραθύρου και τις συντεταγμένες της πάνω αριστερής γωνίας. Το σημείο (0,0) είναι η άνω αριστερά γωνία της οθόνης. Το αλφαριθμητικό "400x300+20+40" δηλώνει ότι οι διαστάσεις του παραθύρου είναι 400x300 και οι συντεταγμένες της άνω αριστερά γωνίας είναι (20,40). Ο καθορισμός της θέσης και των διαστάσεων του παραθύρου υλοποιείται με την

```
root.wm_geometry("400x300+20+40")
```

Στη συνέχεια προσθέτουμε ένα κείμενο στο παράθυρο χρησιμοποιώντας την κλάση Label με την έκφραση

```
Label(root, text = "Hello world!").pack()
```

Η έκφραση αυτή είναι ένας συνοπτικός τρόπος για να περιγράψουμε τη δημιουργία ενός στιγμιότυπου της Label και την εκτέλεση της μεθόδου pack από αυτό. Αναλυτικότερα θα μπορούσαμε να δώσουμε όνομα στο στιγμιότυπο ως εξής:

```
mylabel = Label(root, text = "Hello world!")
mylabel.pack()
```

Το αποτέλεσμα είναι το ίδιο. Δεν ακολουθήσαμε την ανωτέρω δεύτερη σύνταξη, γιατί στην περίπτωση αυτή, δεν θα χρησιμοποιήσουμε στη συνέχεια το όνομα mylabel, και άρα το αντίστοιχο widget. Σημειώστε ότι το πρώτο όρισμα που χρησιμοποιούμε στη δημιουργία

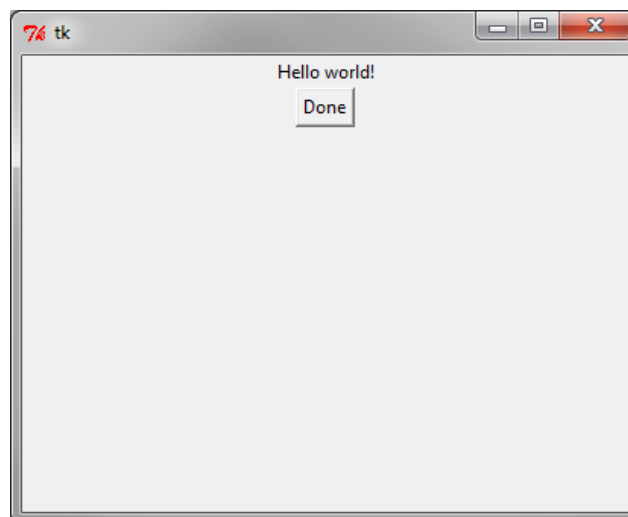
ενός Label widget, είναι το όνομα του παραθύρου στο οποίο θα προστεθεί το widget, ενώ "Hello world!" είναι το κείμενο που θα εμφανιστεί. Με τη μέθοδο pack, τοποθετούμε τα διάφορα widgets πάνω στα οποία εφαρμόζεται σε συγκεκριμένες θέσεις στο παράθυρο1. Εδώ απλώς τοποθετούνται το ένα κάτω από το άλλο. Με κατάλληλη χρήση της pack μπορούμε να ελέγξουμε πλήρως την εμφάνιση του παραθύρου.

Στη συνέχεια δημιουργούμε ένα κουμπί με την πρόταση
`Button(root, text="Done", command=root.destroy).pack()`

Η πρόταση αυτή δημιουργεί και τοποθετεί ένα κουμπί στο παράθυρο, πάνω στο οποίο αναγράφεται "Done". Μπορούμε να πατήσουμε το κουμπί με το ποντίκι. Τότε θα εκτελεστεί η μέθοδος destroy του παραθύρου root. Ως αποτέλεσμα της ενέργειας αυτής, το παράθυρο θα κλείσει και το πρόγραμμα θα τερματιστεί.

Με την πρόταση
`root.update()`
τα διάφορα widgets εμφανίζονται στο παράθυρο.

Το αποτέλεσμα της εκτέλεσης του κώδικα είναι το κάτωθι.



Το πρόγραμμα όταν δεν πιέζουμε το κουμπί, είναι σε κατάσταση αναμονής κάποιας ενέργειας από το χρήστη. Παρακολουθούνται οι συσκευές εισόδου, όπως το ποντίκι ή το πληκτρολόγιο, ενώ το παράθυρο παραμένει ενεργό. Η συμπεριφορά αυτή επιτυγχάνεται γράφοντας την πρόταση

```
root.mainloop()
```

Μπορείτε να εξηγήσετε τα ακόλουθα:

- Τι συμβαίνει όταν πατάμε το κουμπί;
- Σε ποιο σημείο του κώδικα ορίζεται η συμπεριφορά αυτή;

Άσκηση 1: Να τροποποιήσετε τον κώδικα έτσι ώστε να προστίθεται ένα δεύτερο κουμπί. Όταν πατάμε το νέο κουμπί, θα πρέπει να εκτυπώνεται ένα σχετικό μήνυμα. Για να γίνει αυτό θα πρέπει να γράψετε μια συνάρτηση που να υλοποιεί την επιθυμητή λειτουργία και να τη συνδέσετε με το κουμπί, μέσω της παραμέτρου command ενός widget Button.

10.3 Ανάγνωση δεδομένων από παράθυρο

Στη συνέχεια θα γράψουμε κώδικα που μας επιτρέπει να χειριστούμε και άλλα παράθυρα μέσω του βασικού παράθυρου, με σκοπό να επιτύχουμε λειτουργίες όπως ανάγνωση δεδομένων, ενημέρωση των μηνυμάτων που εμφανίζονται στα παράθυρα, κ.ο.κ.

Βήμα 2

Στη συνέχεια θα τροποποιήσουμε τον κώδικα του πρώτου μέρους. Καλύτερα να δημιουργήσετε ένα νέο αρχείο πηγαίου κώδικα. Να προσθέσετε στον κώδικά σας την ακόλουθη κλάση

```
class ADialog:
def __init__(self, parent):
top = self.top = Toplevel(parent)
Label(top, text="Value").pack()
self.e = Entry(top)
self.e.pack(padx=15)
b = Button(top, text="OK", command=self.ok)
b.pack(pady=5)
def ok(self):
self.x = self.e.get()
print "value is", self.x
self.top.destroy()
```

Όταν δημιουργηθεί ένα στιγμίοτυπο της κλάσης ADialog, η μέθοδος __init__ θα δημιουργήσει ένα νέο παράθυρο. Εκτός από τα γνωστά widgets Label και Button, χρησιμοποιούνται δύο νέα widgets:

- Toplevel δημιουργεί ένα καινούριο παράθυρο
- Entry δημιουργεί ένα widget που παρέχει ένα πλαίσιο στο οποίο ο χρήστης μπορεί να εισάγει δεδομένα. Με τη μέθοδο get η οποία εκτελείται από στιγμίοτυπα της κλάσης Entry, το πρόγραμμά μας διαβάζει τα περιεχόμενα του πλαισίου. Στον ανωτέρω κώδικα, αυτό γίνεται όταν εκτελείται η μέθοδος OK, δηλαδή όταν ο χρήστης πατήσει το κουμπί OK.

Να τροποποιήσετε το κυρίως πρόγραμμά σας ώστε να δημιουργεί ένα στιγμίοτυπο της κλάσης ADialog:

```
root = Tk()
root.wm_geometry("400x300+20+40")

message=StringVar()
message.set("Complete the form")
Label(root, textvariable=message).pack(padx=30)
root.update()

dial = ADialog(root)
root.wait_window(dial.top)

message.set("Form completed")

Label(root, text="Got value "+????????).pack()

Button(root, text="Done", command=root.destroy).pack()
root.update()
root.mainloop()
```

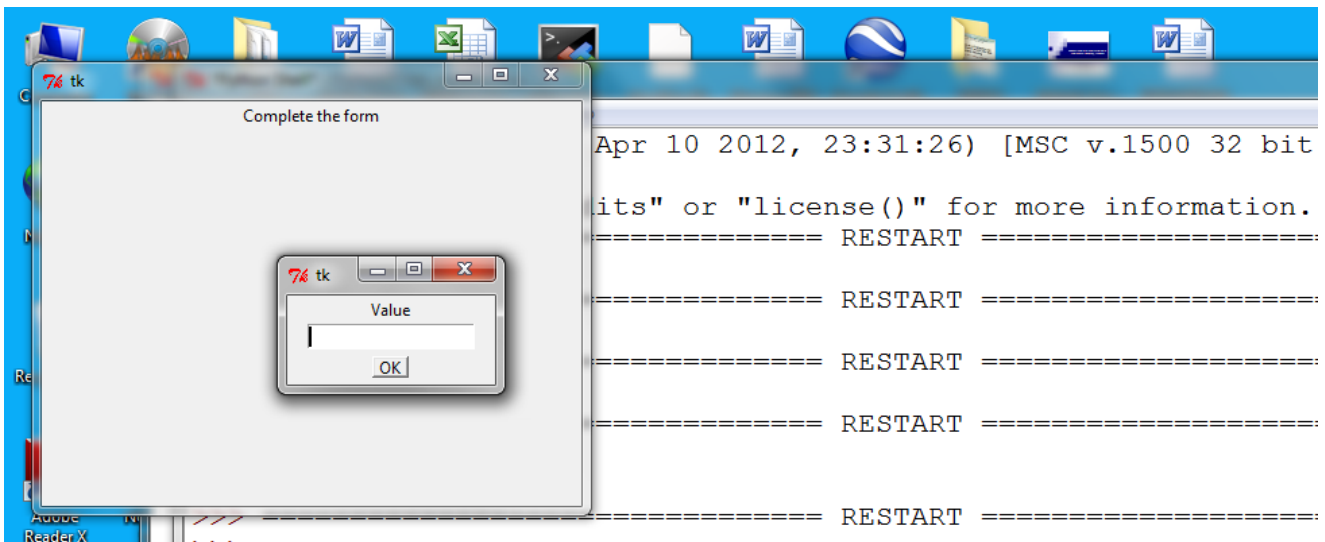
Ο ανωτέρω κώδικας χρησιμοποιεί την κλάση `StringVar2` και δημιουργεί ένα στιγμιότυπό της, το `message`. Τα στιγμιότυπα της κλάσης αυτής είναι συγχρονισμένα με το αντίστοιχο κείμενο μηνύματος που εμφανίζεται στο βασικό παράθυρο. Κάθε φορά που αλλάζουν τιμή, αλλάζει και το μήνυμα στο παράθυρο. Σημειώστε ότι η συμπεριφορά αυτή είναι διαφορετική από εκείνη ενός απλού αλφαριθμητικού. Για να επιτευχθεί, προσέξτε τη χρήση της παραμέτρου `textvariable` (αντί `text`) στο widget `Label` και τη χρήση της μεθόδου `set` στο στιγμιότυπο `message` της κλάσης `StringVar`

Η πρόταση:

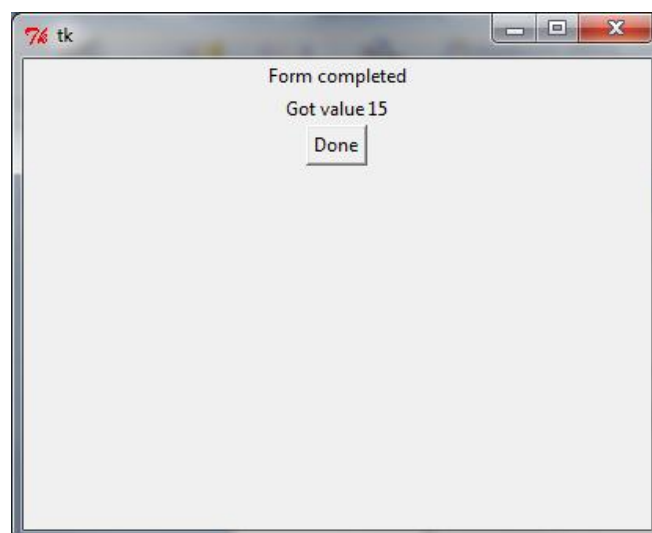
```
root.wait_window(dial.top)
```

χρησιμοποιεί τη μέθοδο `wait_window` για να εξασφαλίσει ότι πρώτα θα ολοκληρωθούν οι ενέργειες που σχετίζονται με το παράθυρο διαλόγου, αυτό θα κλείσει και στη συνέχεια θα επιστραφεί ο έλεγχος στο `root`.

Να αντικαταστήσετε την έκφραση `?????` ώστε να εμφανίζεται η τιμή την οποία συμπληρώσαμε στο παράθυρο.

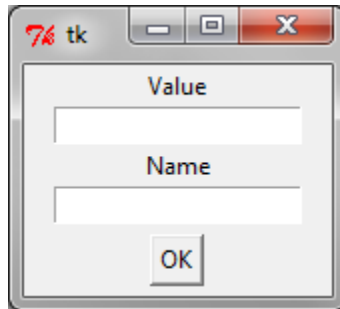


Μετά τη συμπλήρωση της φόρμας, αφού προσθέσουμε την κατάλληλη έκφραση, παίρνουμε την ακόλουθη εικόνα:



Βήμα 3

Στη συνέχεια να τροποποιήσετε την κλάση ADialog για να διαβάζει και μια δεύτερη μεταβλητή και να εμφανίζει τα αποτελέσματα στο root. Θα πρέπει να πάρετε την ακόλουθη συμπεριφορά.



Βήμα 4

Χρησιμοποιώντας τη γραφική διεπαφή, να γράψετε κώδικα ρυθμιστή ο οποίος να δημιουργεί ένα τυχαίο πίνακα διαστάσεων που διαβάζονται από το παράθυρο και στη συνέχεια να τον εμφανίζει στο βασικό παράθυρο.

Βήμα 5

Να τροποποιήσετε τον ανωτέρω κώδικα, ώστε να δημιουργεί δύο τυχαίους πίνακες και να υπολογίζει το άθροισμα ή τη διαφορά τους με βάση το πάτημα κατάλληλου κουμπιού. Δείτε τι άλλες επιλογές και widgets παρέχει το module Tkinter.

Παραδοτέα:

Οι κώδικες που αναπτύξατε στα βήματα 1 έως και 5.

Σημειώματα

Σημείωμα Ιστορικού Εκδόσεων Έργου

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

- Έκδοση **1.0** διαθέσιμη [εδώ](#).

Σημείωμα Αναφοράς

Copyright Πανεπιστήμιο Πατρών, Αβούρης Νικόλαος, Παλιουράς Βασίλειος, Κουκιάς Μιχαήλ, Σγάρμπας Κυριάκος. «Εισαγωγή στους Υπολογιστές Ι, Κοινωνική Διάσταση». 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

Εικόνες: Προέρχονται από Python IDLE.

Πίνακες

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 τη 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.

