Εφαρμογές Η/Υ

8ο Μάθημα

Γραφικό Περιβάλλον Διεπαφής (Graphical User Interface - GUI)

Διδάσκοντες

Νικόλαος Λαγαρός, Επίκουρος Καθηγητής, Εργαστήριο Στατικής & Αντισεισμικών Ερευνών

☎ 210 772 2625, ✉ `nlagaros@central.ntua.gr`

Αθανάσιος Στάμος, ΕΔΙΠ, Τομέας Δομοστατικής

☎ 210 772 3665, ✉ `stamthan@central.ntua.gr`

Χριστόδουλος Φραγκουδάκης, ΕΔΙΠ, Κέντρο Ηλεκτρονικών Υπολογιστών

☎ 210 772 2434, ✉ `chfrag@central.ntua.gr`

Γραφικό Περιβάλλον Διεπαφής (GUI)

Διαχωρισμός GUI από υπόλοιπο πρόγραμμα (υπολογισμούς κλπ)

- Συνιστάται θερμώς η λογική του προγράμματος (υπολογισμοί κλπ) να είναι σε άλλες κλάσεις/ενότητες/συναρτήσεις από το GUI.
- Το GUI μπορεί να αλλάξει (πχ να γίνει PyQt, ή text), οι υπολογισμοί όχι.
- Στο παράδειγμα η μέθοδος ypolPar():
  - Ελέγχει τις τιμές όλων των widgets με χρήση for.
  - Καλεί τη συνάρτηση ypolTaxPar() για τους υπολογισμούς.
  - Κάνει τα widgets αποτελεσμάτων προσωρινά με state=tk.NORMAL.
  - Γράφει την υπολογισμένη παροχή και μέση ταχύτητα.
  - Κάνει τα widgets αποτελεσμάτων με state αυτό που είχαν προηγουμένως.

Γραφικό Περιβάλλον Διεπαφής (GUI)

Διαχωρισμός GUI από υπόλοιπο πρόγραμμα (υπολογισμούς κλπ)

```
    def ypolPar(self):
        "Compute discharge."
        print("Computation.....")
        vals = []
        for wid in self.entManning, self.entWidth, self.entHeight, self.entSlope:
            t = wid.get().replace(",", ".")
            try: temp = float(t)
            except ValueError: temp = None
            if temp is None or temp <= 0:
                tk.messagebox.showinfo("Error", "A positive number was expected",
                    parent=self.root, icon=tk.messagebox.ERROR)
                wid.focus_set()
                return
            vals.append(temp)

        v, q = ypolTaxPar(*vals)
        self.entVel.config(state=tk.NORMAL)
        self.entVel.delete(0, tk.END)
        self.entVel.insert(0, "{:.3f}".format(v))
        self.entVel.config(state="readonly")
```

Γραφικό Περιβάλλον Διεπαφής (GUI)

Διαχωρισμός GUI από υπόλοιπο πρόγραμμα (υπολογισμούς κλπ)

```
        self.entDis.config(state=tk.NORMAL)
        self.entDis.delete(0, tk.END)
        self.entDis.insert(0, "{:.3f}".format(q))
        self.entDis.config(state="readonly")


def ypolTaxPar(n, b, h, s):
    "Υπολογισμός παροχής ανοικτού αγωγού: n=manning, b=width(m), h=height(m), s=sl
    per = b + 2*h
    brexper = b + 2*h
    brexepif = b * h
    rh = brexepif / brexper
    v = rh ** (2.0/3.0) * sqrt(s/100.0) / n
    q = brexepif * v
    return v, q
```

Γραφικό Περιβάλλον Διεπαφής (GUI)

Widgets αναζήτησης αρχείου αποθήκευσης

- Για την αναζήτηση αρχείου αποθήκευσης, το tkinter έχει στο υπο-πακέτο filedialog τη συνάρτηση:

```
fw = tk.filedialog.asksaveasfile(parent, defaultextension,
    initialdir, initialfile, title)
```

- Επιστρέφει το ανοιγμένο αρχείο fw ή None αν ο χρήστης ακύρωσε την αναζήτηση.
- Στο παράδειγμα:
  - Δημιουργείται μέθοδος getVals() που επιστρέφει τις τιμές όλων των widgets δεδομένων ως λίστα, ή None αν υπάρχει λάθος.
  - Δημιουργείται μέθοδος saveAll() που ανοίγει αρχείο και γράφει τα δεδομένα και αποτελέσματα.
  - Δημιουργείται νέα εντολή στο μενού που καλεί τη μέθοδο saveAll().
  - Η μέθοδος ypolPar() χρησιμοποιεί τη μέθοδο getVals() για να διαβάζει τα δεδομένα.

Γραφικό Περιβάλλον Διεπαφής (GUI)

Widgets αναζήτησης αρχείου αποθήκευσης

```
    def setMenus(self):
        ...
        menu.add_command(label="Save", command=self.saveAll, foreground="blue") #N
        menu.add_separator()
        ...

    def getVals(self):
        "Validate and return the values of the widgets."
        vals = []
        for wid in self.entManning, self.entWidth, self.entHeight, self.entSlope:
            t = wid.get().replace(",", ".")
            try: temp = float(t)
            except ValueError: temp = None
            if temp is None or temp <= 0:
                tk.messagebox.showinfo("Error", "A positive number was expected",
                    parent=self.root, icon=tk.messagebox.ERROR)
                wid.focus_set()
                return None
            vals.append(temp)
        return vals
```

Γραφικό Περιβάλλον Διεπαφής (GUI)

Widgets αναζήτησης αρχείου αποθήκευσης

```
    def ypolPar(self):
        "Compute discharge."
        vals = self.getVals()
        if vals is None: return #Errors were found
        ...

    def saveAll(self):
        "Write all the data and results to a file."
        vals = self.getVals()
        if vals is None: return #Errors were found
        vals.append(self.entVel.get())
        vals.append(self.entDis.get())
        fw = tk.filedialog.asksaveasfile(parent=self.root, defaultextension=".txt"
            title="Open file for save")
        if fw is None: return   #User cancelled save
        for temp in vals:
            fw.write("{}\n".format(temp))
        fw.close()
```

Γραφικό Περιβάλλον Διεπαφής (GUI)

Widgets αναζήτησης αρχείου ανάγνωσης

- Για την αναζήτηση αρχείου ανάγνωσης, το tkinter έχει στο υπο-πακέτο filedialog τη συνάρτηση:

```
fr = tk.filedialog.askopenfile(parent, defaultextension,
    initialdir, initialfile, title)
```

- Επιστρέφει το ανοιγμένο αρχείο fr ή None αν ο χρήστης ακύρωσε την αναζήτηση.
- Στο παράδειγμα:
  - Δημιουργείται μέθοδος readData() που ανοίγει αρχείο, διαβάζει τα δεδομένα, τα τοποθετεί στα widgets και κάνει τον υπολογισμό.
  - Δημιουργείται νέα εντολή στο μενού που καλεί τη μέθοδο readData().

```
    def setMenus(self):
        ...
        menu.add_command(label="Open", command=self.readData, foreground="blue")
        menu.add_command(label="Save", command=self.saveAll, foreground="blue")
        menu.add_separator()
        ...
```

Γραφικό Περιβάλλον Διεπαφής (GUI)

Widgets αναζήτησης αρχείου ανάγνωσης

```
    def readData(self):
        "Read the data from a file."
        fr = tk.filedialog.askopenfile(parent=self.root, defaultextension=".txt",
            title="Open file")
        if fr is None: return   #User cancelled save
        dlines = fr.readlines()
        for i, wid in enumerate((self.entManning, self.entWidth, self.entHeight,
            self.entSlope)):
            temp = dlines[i].strip().replace(",", ".") if i < len(dlines) else ""
            wid.delete(0, tk.END)
            wid.insert(0, temp)
        fr.close()
        self.ypolPar()
```

Γραφικό Περιβάλλον Διεπαφής (GUI)

Σχεδίαση γραμμών, κύκλων κλπ - Canvas

- Το widget Canvas ("καμβάς") επιτρέπει αυθαίρετη σχεδίαση (όπως πχ στο AutoCAD):

```
dc = tk.Canvas(master, height, width)
```

- Τα width και height είναι σε pixels και μπορεί να παραληφθούν.
- Έχει βασικές εντολές σχεδίασης ως μεθόδους:
  - create_line(coords): Σχεδίαση τεθαλσμένης γραμμής: coords είναι λίστα/πλειάδα από λίστες/πλειάδες (x, y).
  - create_rectangle(bbox): Σχεδίαση ορθογωνίου: bbox είναι λίστα 2 πλειάδων: x, y του κάτω αριστερά και πάνω δεξιά σημείου.
  - create_polygon(coords)): Σχεδίαση (κλειστού) πολυγώνου.
  - create_oval(bbox): Σχεδίαση κύκλου/έλλειψης: bbox είναι οι συντεταγμένες του περιγεγραμμένου ορθογωνίου.
  - create_arc(bbox): Σχεδίαση κυκλικού/έλλειπτικού τόξου: bbox είναι οι συντεταγμένες του περιγεγραμμένου ορθογωνίου.
  - create_image(position): Σχεδίαση εικόνας raster: position είναι πλειάδα x,y του πάνω αριστερά σημείου της εικόνας στο καμβά.
  - create_text(position): Σχεδίαση κειμένου: position είναι η θέση του κειμένου στο καμβά.

Γραφικό Περιβάλλον Διεπαφής (GUI)

Σχεδίαση γραμμών, κύκλων κλπ - Canvas

Οι μέθοδοι σχεδίασης έχουν και τα ορίσματα:

- fill: το χρώμα, πχ "white".
- width: το πάχος της γραμμής σε pixels.

Ο άξονας x έχει φορά προς τα δεξιά και ο άξονας y προς τα κάτω. Η αρχή αξόνων είναι η πάνω αριστερή γωνία του καμβά.

Ο καμβάς έχει επίσης τις μεθόδους:

- winfo_width(): Επιστέφει το πλάτος του καμβά σε pixels
- winfo_height(): Επιστέφει το ύψος του καμβά σε pixels
- update_idletasks(): Δίνει χρόνο στο καμβά να σταθεροποιηθεί μετά από αλλαγή μεγέθους του κύριου παραθύρου από το χρήστη

Οι μέθοδοι αυτοί υπάρχουν και στα άλλα wdigets του tkinter.

Γραφικό Περιβάλλον Διεπαφής (GUI)

Σχεδίαση γραμμών, κύκλων κλπ - Canvas

Για τη σχεδίαση του αγωγού:

- Θεωρούμε τον αγωγό κατά 10% μεγαλύτερο από τη στάθμη του νερού σε ύψος.
- Αφήνουμε περιθώριο δεξιά και αριστερά ίσο με 5% του πλάτους του καμβά.
- Αφήνουμε περιθώριο κάτω και πάνω ίσο με 5% του ύψους του καμβά.
- Οι ενεργές διαστάσεις του καμβά είναι οι διαστάσεις χωρίς τα περιθώρια.
- Η κλίμακα κατά x είναι ενεργό πλάτος καμβά δια πλάτος αγωγού.
- Η κλίμακα κατά y είναι ενεργό ύψος καμβά δια πλάτος αγωγού.
- Από τις 2 κλίμακες λαμβάνουμε τη μικρότερη για να μην έχουμε παραμόρφωση.
- Για μετατροπή συντεταγμένης x από m σε pixels πολλαπλασιάζουμε επί την κλίμακα και προσθέτουμε το περιθώριο x.
- Για μετατροπή συντεταγμένης y από m σε pixels πολλαπλασιάζουμε επί την κλίμακα και προσθέτουμε το περιθώριο y. Στη συνέχεια αυτό το αφαιρούμε από το ύψος του καμβά, επειδή ο άξονας y έχει φορά προς τα κάτω και μηδέν στην πάνω αριστερη γωνία.
- Η σχεδίαση γίνεται κάθε φορά που γίνεται υπολογισμός.

Γραφικό Περιβάλλον Διεπαφής (GUI)

Σχεδίαση γραμμών, κύκλων κλπ - Canvas

```
    def __init__(self):
        ...
        temp = tk.Label(self.root, text="Σχολή Πολιτικών Μηχανικών, ΕΜΠ", backgrou
        temp.grid(row=ir, column=0, columnspan=2, sticky="w")

        ir += 1
        temp = tk.Label(self.root, text="Manning:")
        temp.grid(row=ir, column=0, sticky="e")
        self.canChan = tk.Canvas(self.root, height=200, background="lightyellow")
        self.canChan.grid(row=ir, column=0, columnspan=2, sticky="we")
        ...

    def ypolPar(self):
        ...
        self.entDis.insert(0, "{:.3f}".format(q))
        self.entDis.config(state="readonly")
        self.draw(*vals)
```

Γραφικό Περιβάλλον Διεπαφής (GUI)

Σχεδίαση γραμμών, κύκλων κλπ - Canvas

```
    def draw(self, n, b, h, s):
        "Draw the open channel."
        dc = self.canChan
        dc.update_idletasks()            # _idletasks breaks WinDoze 98 support
        dcw = dc.winfo_width()             # Pixels
        dch = dc.winfo_height()            # Pixels
        if dcw < 2 or dch < 2: return    #tkinter reported wrong width or height
        perx = dcw*0.05           #Margin x
        pery = dch*0.05           #Margin y
        dcwef = dcw - 2*perx      #Effective width is now smaller
        dchef = dch - 2*pery      #Effective height is now smaller
        dc.delete(tk.ALL)                 # Clear window

        hchan = h*1.1     #Height of channel is bigger than water level
        bchan = b
```

Γραφικό Περιβάλλον Διεπαφής (GUI)

Σχεδίαση γραμμών, κύκλων κλπ - Canvas

```
        akl = dcwef/bchan
        akly= dchef/hchan
        if akly < akl: akl = akly
        def topix(x, y): return perx+x*akl, dch-(pery+y*akl)

        cs = ( (0, 0), (b, h) )    #Draw water surface
        cs = [topix(x, y) for x, y in cs]
        dc.create_rectangle(cs, fill="cyan", outline=None, stipple="gray25")

        cs = ( (0, h), (b, h) )    #Draw water level
        cs = [topix(x, y) for x, y in cs]
        dc.create_line(cs, fill="blue", width=3)

        cs = ( (0, hchan), (0, 0), (bchan, 0), (bchan, hchan) )   #Draw channel
        cs = [topix(x, y) for x, y in cs]
        dc.create_line(cs, fill="brown", width=3)
```

Γραφικό Περιβάλλον Διεπαφής (GUI)

Οδηγός γεωμετρίας - αύξηση ύψους widget

Για μεγαλύτερα σχήματα ο καμβάς πρέπει να καταλαμβάνει όλο το διαθέσιμο υψος παραθύρου:

```
    def __init__(self):
        ...
        self.canChan = tk.Canvas(self.root, height=200, background="lightyellow")
        ...
        self.root.columnconfigure(1, weight=1)
        self.root.rowconfigure(1, weight=1)
```

Όταν μεγαλώνει τα παράθυρο, μεγαλώνει και το ύψος του καμβά, αλλά όχι και το ύψος του σχεδιασμένου αγωγού. Αυτό απαιτεί επαναϋπολογισμό της ροής (και συνεπώς επανασχεδίαση).

Γραφικό Περιβάλλον Διεπαφής (GUI)

Μενού - Menubutton

- Ο συντελεστής Manning λαμβάνει διάφορες τιμές για διάφορα υλικά επένδυσης των αγωγών.
- Το widget Menubutton επιτρέπει να δώσουμε μενού με επιλογές με διάφορα υλικά.
- Κάθε φορά που ο χρήστης πατάει την επιλογή, η ενδεικτική τιμή για αυτό το υλικό συμπληρώνεται αυτόματα στο Entry του Manning.
- Καλείται μία συνάρτηση για κάθε υλικό.
- Δημιουργούμε τη μέθοδο setManning() για να απλουστεύσουμε τον κώδικα.

```
    def setManning(self, n):
        "Set a value to the Manning widget."
        self.entManning.delete(0, tk.END)
        self.entManning.insert(0, str(n))

    def n0017(self): self.setManning(0.017)
    def n0012(self): self.setManning(0.012)
    def n0014(self): self.setManning(0.014)
    def n0016(self): self.setManning(0.016)
```

Γραφικό Περιβάλλον Διεπαφής (GUI)

Μενού - Menubutton

```
    def __init__(self):
        ...
        self.entManning = tk.Entry(self.root, width=30)
        self.entManning.grid(row=ir, column=1, sticky="we")

        temp = tk.Menubutton(self.root, text="Suggest", relief=tk.RAISED,
            bg="lightcyan", activebackground="cyan")
        temp.grid(row=ir, column=2, sticky="w")
        menu = tk.Menu(temp, tearoff=False,
            font=tk.font.Font(name="TkFixedFont", exists=True))
        menu.add_command(label="Compressed air formed: 0.014-0.021",
            command=self.n0017)
        menu.add_command(label="Very smooth          : 0.010-0.013",
            command=self.n0012)
        menu.add_command(label="Steel form           : 0.013-0.015",
            command=self.n0014)
        menu.add_command(label="Wooden form          : 0.015-0.018",
            command=self.n0016)
        temp["menu"] = menu
        ...
```

Γραφικό Περιβάλλον Διεπαφής (GUI)

Widget Text: εισαγωγή κειμένου με πολλές σειρές

- Το widget Text λειτουργεί ως το Entry αλλά με πολλές σειρές:

```
txt = tk.Text(master, width, height)
```

- Έχει τις μεθόδους:
  - get("l1.c1", "l2.c2"): επιστρέφει το κείμενο που βρίσκεται μεταξύ του χαρακτήρα c1 της γραμμής l1 και του χαρακτήρα c2 της γραμμής l2.
  - get("l1.c1", tk.END): επιστρέφει το κείμενο που βρίσκεται από το χαρακτήρα c1 της γραμμής l1 μέχρι το τέλος. Λόγω σφάλματος (bug) επιστρέφει ένα χαρακτήρα παραπάνω (τον χαρακτήρα '\n').
  - insert("l1.c1", t): εισάγει το κείμενο t1 στο χαρακτήρα c1 της γραμμής l1, μετατοπίζοντας προς τα δεξιά και κάτω το υφιστάμενο κείμενο.
  - delete("l1.c1", "l2.c2"): διαγράφει το υφιστάμενο κείμενο που βρίσκεται μεταξύ του χαρακτήρα c1 της γραμμής l1 και του χαρακτήρα c2 της γραμμής l2.
  - delete("0.0", tk.END): διαγράφει όλο το υφιστάμενο κείμενο.

Παρακάτω δημιουργούμε παράθυρο με widget Text με μενού και δυνατότητα αποθήκευσης σε αρχείο και ανάγνωσης από αρχείο.

Γραφικό Περιβάλλον Διεπαφής (GUI)

Widget Text: εισαγωγή κειμένου με πολλές σειρές

```
import tkinter as tk
import tkinter.font, tkinter.filedialog


class TextWin:
    "A tkinter window for displaying text."
    def __init__(self):
        "Make the window."
        self.root = tk.Tk()
        self.setFonts()
        self.setMenus()
        self.root.title("Widget για Κείμενο")
        ir = 0
        temp = tk.Label(self.root, text="Σχολή Πολιτικών Μηχανικών, ΕΜΠ",
            background="yellow")
        temp.grid(row=ir, column=0, sticky="w")

        ir += 1
        self.txtCivil = tk.Text(self.root, width=80, height=10, wrap=tk.NONE,
            bg="orange", fg="blue")   #Width, height in characters
        self.txtCivil.grid(row=ir, column=0, sticky="wesn")

        self.root.columnconfigure(0, weight=1)
        self.root.rowconfigure(1, weight=1)
```

Γραφικό Περιβάλλον Διεπαφής (GUI)

Widget Text: εισαγωγή κειμένου με πολλές σειρές

```
    def setFonts(self):
        "Set the fonts a little bigger."
        for t in "TkDefaultFont", "TkFixedFont", "TkMenuFont", "TkCaptionFont", "T
            font1 = tk.font.Font(name=t, exists=True)
            font1.config(size=14)   # Negative size means size in pixels
        font1.config(family="Arial")

    def setMenus(self):
        "Create the menu system."
        menuBar = tk.Menu(self.root, activebackground="green")    #Σύστημα μενού π
        menu = tk.Menu(menuBar, activebackground="green", tearoff=False)    #Νέο d
        menu.add_command(label="Open", command=self.readData, foreground="blue")
        menu.add_command(label="Save", command=self.saveAll, foreground="blue")
        menu.add_separator()
        menu.add_command(label="Exit", command=self.telos, foreground="blue")
        menuBar.add_cascade(label="File", menu=menu, foreground="blue")     #Καταχ
        self.root.config(menu=menuBar)                                    #Ορισμός του συσ

    def telos(self):
        "Delete all references to widgets and the detsroy the root window."
        del self.txtCivil
        self.root.destroy()
```

Γραφικό Περιβάλλον Διεπαφής (GUI)

Widget Text: εισαγωγή κειμένου με πολλές σειρές

```
    def saveAll(self):
        "Write all the data and results to a file."
        t = self.txtCivil.get("0.0", tk.END)[:-1]  #Avoid bug
        fw = tk.filedialog.asksaveasfile(parent=self.root, defaultextension=".txt"
            title="Open file for save")
        if fw is None: return   #User cancelled save
        fw.write(t)
        fw.close()

    def readData(self):
        "Read the dfata from a file."
        fr = tk.filedialog.askopenfile(parent=self.root, defaultextension=".txt",
            title="Open file")
        if fr is None: return   #User cancelled save
        t = fr.read()
        self.txtCivil.delete("0.0", tk.END)
        self.txtCivil.insert("0.0", t)

    def mainloop(self):
        "Start the execution of the GUI."
        self.root.mainloop()

win = TextWin()
win.mainloop()
```

Γραφικό Περιβάλλον Διεπαφής (GUI)

Widget Scrollbar: προσπέλαση άλλων widgets με το ποντίκι

- Το widget Scrollbar συνδέεται με ένα άλλο widget όπως το Text, και επιτρέπει την προσπέλαση του άλλου widget με το ποντίκι:

```
scr = tk.Scrollbar(self.root, orient)
```

- orient: παίρνει την τιμή tk.HORIZONTAL ή tk.VERTICAL.
- Η σύνδεση με το άλλο widget (Text) γίνεται:
  - Στο μέλος "command" του Scrollbar τίθεται ως τιμή η μέθοδος xview() ή yview() του Text:

```
scr['command'] = txt.xview
```

  - Στο μέλος "xscrollcommand" ή "yscrollcommand" του Text τίθεται ως τιμή η μέθοδος set() του Scrollbar:

```
txt['xscrollcommand'] = scr.set
```

- Το Scrollbar είναι widget και καταλαμβάνει ένα τετραγωνίδιο στο νοητό κάναβο, με τη μέθοδο grid().

Γραφικό Περιβάλλον Διεπαφής (GUI)

Widget Scrollbar: προσπέλαση άλλων widgets με το ποντίκι

```
    def __init__(self):
        ...
        temp = tk.Label(self.root, text="Σχολή Πολιτικών Μηχανικών, ΕΜΠ", backgrou
        temp.grid(row=ir, column=0, columnspan=2, sticky="w")

        ir += 1
        self.txtCivil = tk.Text(self.root, width=80, height=10, wrap=tk.NONE,
            bg="orange", fg="blue")   #Width, height in characters
        self.txtCivil.grid(row=ir, column=0, sticky="wesn")

        temp = tk.Scrollbar(self.root, orient=tk.VERTICAL)
        temp.grid(row=ir, column=1, sticky="sn")
        self.txtCivil['yscrollcommand'] = temp.set
        temp['command'] = self.txtCivil.yview

        ir += 1
        temp = tk.Scrollbar(self.root, orient=tk.HORIZONTAL)
        temp.grid(row=ir, column=0, sticky="we")
        self.txtCivil['xscrollcommand'] = temp.set
        temp['command'] = self.txtCivil.xview

        self.root.columnconfigure(0, weight=1)
        self.root.rowconfigure(1, weight=1)
```

Γραφικό Περιβάλλον Διεπαφής (GUI)

Widget Frame: δοχείο (container) άλλων με ανεξάρτητες γραμμές και στήλες

- Το widget Frame δρα ως ένα υπο-παράθυρο:

```
fra = tk.Frame(master, bd, relief)
```

- Καταλαμβάνει ένα τετραγωνίδιο στον νοητό κάναβο του παραθύρου, με τη μέθοδο grid().
- Χωρίζεται σε γραμμές και στήλες (έχει δικό του νοητό κάναβο).
- bd: το πάχος του συνόρου (περιγεγραμμένης γραμμής) σε pixels. Αν είναι 0 δεν σχεδιάζεται σύνορο.
- relief: Το είδος της περιγεγραμμένης γραμμής: tk.RAISED, tk.SUNKEN, tk.FLAT, tk.RIDGE, tk.SOLID, και tk.GROOVE.
- Στη μέθοδο grid() σχεδόν πάντα βάζουμε sticky="wesn".
- Το Frame έχει μεθοδους rowconfigure() και columnconfigure() που λειτουργούν όπως και στο κύριο παράθυρο.

Τα μέλη bd και relief υπάρχουν και σε άλλα widgets.

Ως παράδειγμα δημιουργούμε δύο κομβία κάτω από το widget Text, το ένα όσο αριστερά γίνεται και το άλλο όσο δεξιά γίνεται.

Γραφικό Περιβάλλον Διεπαφής (GUI)

Widget Frame: δοχείο (container) άλλων με ανεξάρτητες γραμμές και στήλες

```
    def __init__(self):
        ...
        temp['command'] = self.txtCivil.xview

        ir += 1
        fra = tk.Frame(self.root, bd=2, relief=tk.RIDGE)
        fra.grid(row=ir, column=0, columnspan=2, sticky="wesn")
        temp = tk.Button(fra, text="Ακύρωση")
        temp.grid(row=0, column=0, sticky="w")
        temp = tk.Button(fra, text="Εντάξει")
        temp.grid(row=0, column=1, sticky="e")
        fra.columnconfigure(0, weight=1)
        fra.columnconfigure(1, weight=1)
        fra.rowconfigure(0, weight=1)

        self.root.columnconfigure(0, weight=1)
        self.root.rowconfigure(1, weight=1)
```

Γραφικό Περιβάλλον Διεπαφής (GUI)

Δημιουργία νέων κλάσεων widgets

- Νέα κλάση widget απλοποιεί τον κώδικα:
  - Συγκεκριμένο σύνολο από widgets που χρησιμοποιείται συχνά.
  - Συναρτήσεις που χρησιμοποιούνται συχνά σε σχέση με συγκεκριμένο widget.
  - Αλλαγή ή κατάργηση μεθόδων.
- Τυπικά εξειδικεύουν την κλάση Frame που έχει τις λιγότερες μεθόδους και μέλη.
- Μπορούν να εξειδικεύουν και άλλες κλάσεις για αλλαγή/προσθήκη σχετικά λίγων μεθόδων.
- Όταν αντικαθίσταται μία μέθοδος method1() της βασικής κλάσης, η αρχική method1() μπορεί να κληθεί με τη βοήθεια της συνάρτησης super() αν χρειάζεται:

```
super().method1()
```

- Ειδικότερα η μέθοδος __init__() της βασικής κλάσης πρέπει οπωσδήποτε να κληθεί.

Ως παράδειγμα ένα Widget Text και δύο Srollbars θα αποτελέσουν νέα κλάση.

Γραφικό Περιβάλλον Διεπαφής (GUI)

Δημιουργία νέων κλάσεων widgets

```
class SText(tk.Frame):
    "A tkinter Text widget with scrollbars."

    def __init__(self, parent, **kw):
        "Create the text and the scrollbars widgets."
        super().__init__(parent)
        ir = 0
        self.txt = tk.Text(self, **kw)
        self.txt.grid(row=ir, column=0, sticky="wesn")

        temp = tk.Scrollbar(self, name="vbar", orient=tk.VERTICAL)
        temp.grid(row=ir, column=1, sticky="sn")
        self.txt['yscrollcommand'] = temp.set
        temp['command'] = self.txt.yview

        ir += 1
        temp = tk.Scrollbar(self, name='hbar', orient=tk.HORIZONTAL)
        temp.grid(row=ir, column=0, sticky="we")
        self.txt['xscrollcommand'] = temp.set
        temp['command'] = self.txt.xview

        self.columnconfigure(0, weight=1)
        self.rowconfigure(0, weight=1)
```

Γραφικό Περιβάλλον Διεπαφής (GUI)

Δημιουργία νέων κλάσεων widgets

- Η μεταβλητή στιγμιότυπου (μέλος) txt πρέπει να διαγραφεί όταν το στιγμιότυπο διαγράφεται.
- Οι μέθοδοι get(), insert() και delete() δεν υπάρχουν στην κλάση Frame και πρέπει να οριστούν έτσι ώστε να καλούν τις αντίστοιχες μεθόδους του Text.
- Το όρισμα *args συγκεντρώνει όλα τα ορίσματα της μεθόδου ως μία λίστα.
- Το όρισμα **kw συγκεντρώνει όλα τα ονοματισμένα ορίσματα της μεθόδου ως ένα λεξικό.

```
    def destroy(self):
        "Delete references to widgets."
        del self.txt
        super().destroy()

    def get(self, *args, **kw): return self.txt.get(*args, **kw)
    def delete(self, *args, **kw): self.txt.delete(*args, **kw)
    def insert(self, *args, **kw): self.txt.insert(*args, **kw)
```

Γραφικό Περιβάλλον Διεπαφής (GUI)

Δημιουργία νέων κλάσεων widgets

Το πρόγραμμα τώρα χρησιμοποιεί το νέο widget:

```
class TextWin:
    "A tkinter window for displaying text."
    def __init__(self):
        "Make the window."
        self.root = tk.Tk()
        self.setFonts()
        self.setMenus()
        self.root.title("Widget για Κείμενο")
        ir = 0
        temp = tk.Label(self.root, text="Σχολή Πολιτικών Μηχανικών, ΕΜΠ", backgrou
        temp.grid(row=ir, column=0,sticky="w")

        ir += 1
        self.txtCivil = SText(self.root, width=80, height=10, wrap=tk.NONE, bg="or
        self.txtCivil.grid(row=ir, column=0, sticky="wesn")

        ir += 1
        fra = tk.Frame(self.root, bd=2, relief=tk.RIDGE)
        fra.grid(row=ir, column=0, columnspan=2, sticky="wesn")
        ...
```

Γραφικό Περιβάλλον Διεπαφής (GUI)

Περιθώριο μεταξύ των widgets

Η μέθοδος grid() έχει τα ορίσματα padx (pady) που ορίζουν το κενό σε pixels που θα αφαιθεί αριστερά και δεξιά (κάτω και πάνω) απο το widget.

Στο παράδειγμα αφήνουμε 20 pixels πάνω και κάτω από το SText:

```
class TextWin:
    "A tkinter window for displaying text."
    def __init__(self):
        ...
        self.txtCivil.grid(row=ir, column=0, sticky="wesn", pady=20)   #pad in pix
        ...
```

Γραφικό Περιβάλλον Διεπαφής (GUI)

Πολλαπλά παράθυρα

- Το κύριο παράθυρο είναι το tk.Tk() και μπορεί να υπάρχει μόνο ένα.
- Μπορούν να υπάρχουν πολλά δευτερεύοντα παράθυρα tk.Toplevel() με γονικό παράθυρο το tk.Tk() ή άλλο υφιστάμενο Toplevel().
- Όταν διαγράφεται ένα παράθυρο (με τη μέθοδο destroy()) διαγράφονται όλα τα θυγατρικά του παράθυρα.
- Όταν διαγράφεται το κύριο παράθυρο, διαγράφονται όλα τα παράθυρα.

Στο παράδειγμα:

- Φτιάχνουμε νέα κλάση TextWinOther που διαφέρει μόνο στη δημιουργία του Toplevel αντί για Tk:
  - Χωρίζουμε την __init__() του TextWin σε init1() που έχει τον κοινό κώδικα και __init__() που έχει τον κώδικα που αλλάζει.
  - Στην κλάση TextWinOther αντικαθιστούμε την __init__().
- Φτιάχνουμε νέο πτυσσόμενο μενού για τη δημιουργία νέου παραθύρου.

Γραφικό Περιβάλλον Διεπαφής (GUI)

Πολλαπλά παράθυρα

```
class TextWin:
    "A tkinter window for displaying text."
    def __init__(self):
        "Make the window."
        self.root = tk.Tk()
        self.root.title("Widget για Κείμενο - Κύριο")
        self.init1()

    def init1(self):
        "Create the widgets."
        self.setFonts()
        ...

class TextWinOther(TextWin):
    "Another window for displaying text."
    def __init__(self, parent):
        "Make the window."
        self.root = tk.Toplevel(parent)
        self.root.title("Widget για Κείμενο")
        self.init1()
```

Γραφικό Περιβάλλον Διεπαφής (GUI)

Πολλαπλά παράθυρα

```
    def setMenus(self):
        "Create the menu system."
        menuBar = tk.Menu(self.root, activebackground="green")    #Σύστημα μενού π

        menu = tk.Menu(menuBar, activebackground="green", tearoff=False)    #Νέο d
        menu.add_command(label="Open", command=self.readData, foreground="blue")
        menu.add_command(label="Save", command=self.saveAll, foreground="blue")
        menu.add_separator()
        menu.add_command(label="Exit", command=self.telos, foreground="blue")
        menuBar.add_cascade(label="File", menu=menu, foreground="blue")     #Καταχ

        menu = tk.Menu(menuBar, activebackground="green", tearoff=False)    #Νέο d
        menu.add_command(label="New window", command=self.newWin, foreground="blue
        menuBar.add_cascade(label="Window", menu=menu, foreground="blue")    #Κατα

        self.root.config(menu=menuBar)                                #Ορισμός του συσ

    def newWin(self):
        "Make a new text window."
        temp = TextWinOther(self.root)
```

Ερωτήσεις;