**Практическая работа №2**

**Модуль tkinter. Размещение виджетов.**

**Создание приложения «Произведение двух чисел»**

**Цель:** рассмотрение различных способов размещения виджетов в окне. Создание приложения «Произведение двух чисел», используя методы pack() и grid().

В любом приложении виджеты не разбросаны по окну как попало, а хорошо организованы, интерфейс продуман до мелочей и обычно подчинен определенным стандартам.

1. **Расположение элементов друг за другом с помощью наиболее простого менеджера геометрии tkinter – метода pack():**

При использовании этого упаковщика с помощью свойства side можно указать к какой стороне родительского виджета он должен примыкать. Как правило этот упаковщик используют для размещения виджетов друг за другом (слева направо или сверху вниз).

Для создания приложения «Произведение двух чисел» понадобятся две метки, три текстовых поля и одна кнопка.

from tkinter import \*

root=Tk()

root.title('Произведение двух чисел')

Label(root,text = 'Первое число').pack()

EntryA = Entry(root, width = 10, font = 'Arial 16', justify = "center") # создаем текстовое поле ввода

EntryA.pack() # размещаем в главно окне методом pack()

Label(root,text = 'Второе число').pack()

EntryB=Entry(root, width = 10, font = 'Arial 16', justify = "center")

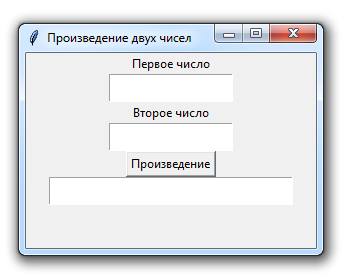
EntryB.pack()

but = Button(root, text = 'Произведение')

but.pack() # размещаем кнопку в главном окне

EntryC = Entry(root, width = 20, font = 'Arial 16', justify = "center")

EntryC.pack()

root.mainloop()

По умолчанию свойства side метод pack() располагает объекты сверху вниз в порядке, объявленном в программе. У свойств side могут быть аргументы ”left”/”right”/”top”/”bottom”

Напишем функцию, обрабатывающую событие нажатия на кнопку «Произведение»:

def proizv():

a = EntryA.get() # берем текст из первого поля

a = int(a) # преобразуем в число целого типа

b = EntryB.get()

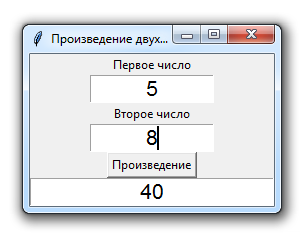
b = int(b)

result = str(a \* b) # результат переведем в строку для дальнейшего вывода

EntryC.delete(0, END) # очищаем текстовое поле полностью

EntryC.insert(0, result) # вставляем результат в начало

Теперь необходимо связать эту функцию с кнопкой. В предыдущей практической работе это делалось с помощью метода bind. Также это можно сделать следующим образом:

but = Button(root, text = 'Произведение', command = proizv)

1. **Размещение виджетов в окне с помощью метода grid().**

При размещении виджетов методом grid() родительский контейнер (обычно это окно) условно разделяется на ячейки подобно таблице. Адрес каждой ячейки состоит из номера строки и номера столбца. Нумерация начинается с нуля. Ячейки можно объединять как по вертикали, так и по горизонтали.

Никаких предварительных команд по разбиению родительского виджета на ячейки не выполняется. Tkinter делает это сам, исходя из указанных позиций виджетов.

Размещение виджета в той или иной ячейке задается через аргументы row и column, которым присваиваются соответственно номера строки и столбца. Чтобы объединить ячейки по горизонтали, используется атрибут columnspan, которому присваивается количество объединяемых ячеек. Опция rowspan объединяет ячейки по вертикали.

У метода grid() имеются атрибуты для задания внешних и внутренних отступов (padx, pady, ipadx, ipady).

Кроме этого есть атрибут sticky (липкий), который принимает значения направлений сторон света (N, S, W, E, NW, NE, SW, SE). Если, например, указать NW, то виджет прибьет к верхнему левому углу ячейки. Виджеты можно растягивать на весь объем ячейки (sticky=N+S+W+E) или только по одной из осей (N+S или W+E). Эффект от "липучки" заметен, только если виджет меньше ячейки.

Создайте новый файл и скопируйте в него созданный код. Внесите изменения в следующие строки:

# первая метка в строке 0, столбце 0 (0 по умолчанию)

Label(root, text = 'Первое число').grid(row = 0, sticky = W)

# вторая метка в строке 1

Label(root, text = 'Второе число').grid(row = 1, sticky = W)

# размещаем первые два поля справа от меток, второй столбец (отсчет от нуля)

EntryA.grid(row = 0, column = 1, sticky = E)

EntryB.grid(row = 1, column = 1, sticky = E)

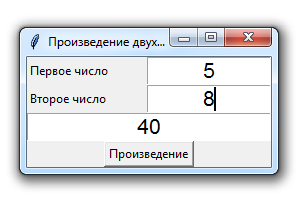
# третье текстовое поле ввода занимает всю ширину строки 2

EntryC.grid(row = 2, columnspan = 2)

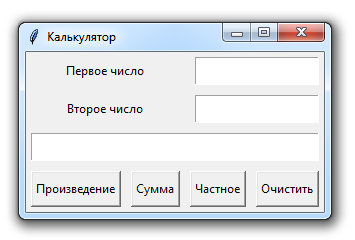
# размещаем кнопку в строке 3 во всю ширину

but.grid(row = 3, columnspan = 2)

Приложение станет примерно таким:



**Задание.**

Добавьте в приложение кнопки «Сумма», «Частное», «Очистить», нажатие на которые выполняет сложение, деление введенных чисел, а также очистку всех текстовых полей соответственно. Переименуйте приложение.

В работу кнопки «Частное» добавьте обработку случая введения в поле «Второе число» значения «0». Для этого необходимо добавить диалоговое окно messagebox с одной кнопкой для сообщения об ошибке showerror():

* В начало программы нужно добавить следующую команду:

from tkinter import messagebox as mb

* В работу кнопки «Частное» добавить следующую команду, вызывающую диалоговое окно «Ошибка» с сообщением «На 0 делить нельзя»:

mb.showerror('Ошибка', 'На 0 делить нельзя') 