

## Практическое задание № 17

**Тема:** Составление программ с использованием GUI Tkinter в IDE PyCharm Community, изучение возможностей модуля OS.

**Цели:** закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ с использованием GUI Tkinter в IDE PyCharm Community, изучить возможности модуля OS.

### Постановка задачи №1:

В соответствии с номером варианта перейти по ссылке на прототип. Реализовать его в IDE PyCharm Community с применением пакета tk. Получить интерфейс максимально приближенный к оригиналу (см. таблицу 1).

### Текст программы:

*# В соответствии с вариантом перейти по ссылке на прототип. Реализовать его с применением пакета tk*

```
from tkinter import *

root = Tk()
root.title("Register")
root.geometry("500x600")
root.configure(bg="gray")

title_label = Label(root, text="Elegant login and Register forms", background="gray",
fg="black", font=("Times New Roman", 16, 'italic'))
title_label.pack(pady=(20, 20))

main_frame = Frame(root, bg="black", pady=50)
main_frame.place(relx=0.5, rely=0.5, anchor="center")

label_main = Label(main_frame, text="Register", background="black", font='Arial 20',
bd=2, pady=25, fg='gray').grid(row=0, column=0, columnspan=2)

entry_name = Entry(main_frame, bg="black", fg="gray")
entry_name.insert(0, "First Name")
entry_name.grid(row=1, column=0, columnspan=2, pady=5, padx=20, sticky="ew")

entry_surname = Entry(main_frame, bg="black", fg="gray")
entry_surname.insert(0, "Last Name")
entry_surname.grid(row=2, column=0, columnspan=2, pady=5, padx=20, sticky="ew")

entry_email = Entry(main_frame, bg="black", fg="gray")
entry_email.insert(0, "Email Address")
entry_email.grid(row=3, column=0, columnspan=2, pady=5, padx=20, sticky="ew")

entry_username = Entry(main_frame, bg="black", fg="gray")
entry_username.insert(0, "User Name")
entry_username.grid(row=4, column=0, columnspan=2, pady=5, padx=20, sticky="ew")

entry_password = Entry(main_frame, bg="black", fg="gray", show='•')
entry_password.insert(0, "Password")
entry_password.grid(row=5, column=0, columnspan=2, pady=5, padx=20, sticky="ew")

entry_repeat_password = Entry(main_frame, bg="black", fg="gray", show='•')
entry_repeat_password.insert(0, "Password")
entry_repeat_password.grid(row=6, column=0, columnspan=2, pady=5, padx=20,
sticky="ew")
```

```

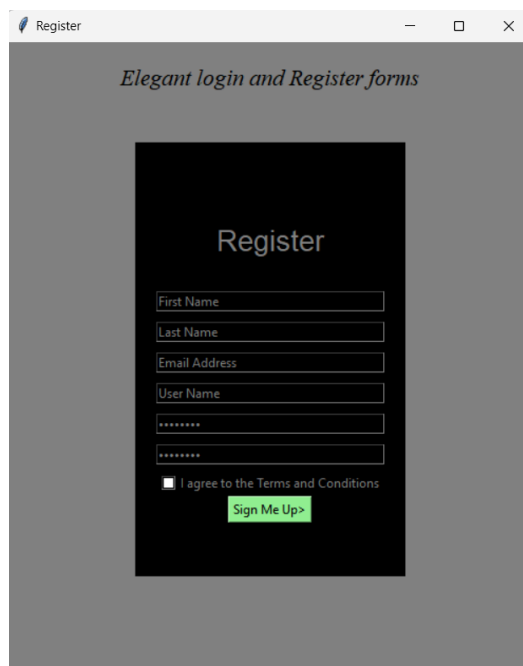
checkboxbutton = Checkbutton(main_frame, text="I agree to the Terms and
Conditions", bg="black", fg="grey",)
checkboxbutton.grid(row=7, column=0, columnspan=2, padx=20)

button = Button(main_frame, text="Sign Me Up>", bg="lightgreen", fg="black")
button.grid(row=8, column=0, columnspan=2)

root.mainloop()

```

### Протокол работы программы:



### Постановка задачи №2:

Разработать программу с применением пакета tk, взяв в качестве условия одну любую задачу из ПЗ №№ 2 – 9.

### Текст программы:

*# Задача (ПЗ №2): Скорость первого автомобиля V1 км/ч, а второго – V2 км/ч, расстояние между ними S км.  
# Определить расстояние между автомобилями через T часов, если они движутся навстречу друг другу.*

```

from tkinter import *
from tkinter import ttk

def calculate():
    try:
        V1 = int(entry_V1.get())
        V2 = int(entry_V2.get())
        S = int(entry_S.get())
        T = int(entry_T.get())
    except:
        result_label.config(text='Ошибка: Числа не введены')

    return

S1 = (V1 + V2) * T

```

```

S2 = abs(S - S1)

result_label.config(
    text=f'Расстояние между автомобилями равно: {int(S2)} км'
)

root = Tk()
root.title('Задача из ПЗ №2')

label = ttk.Label(root, text='Скорость первого автомобиля V1 км/ч, а второго - V2 км/ч, расстояние между ними S км.\n'
                             'Определить расстояние между автомобилями через T часов, если они движутся навстречу друг другу.',padding=5)

label.grid(column = 0, columnspan= 2, row = 0)

label_V1 = ttk.Label(root, text='Скорость первого автомобиля',padding=5)
label_V1.grid(column = 0, row = 1)
entry_V1 = ttk.Entry(root)
entry_V1.grid(column = 1, row = 1)

label_V2 = ttk.Label(root, text='Скорость второго автомобиля',padding=5)
label_V2.grid(column = 0, row = 2)
entry_V2 = ttk.Entry(root)
entry_V2.grid(column = 1, row = 2)

label_S = ttk.Label(root, text='Расстояние между автомобилями',padding=5)
label_S.grid(column = 0, row = 3)
entry_S = ttk.Entry(root)
entry_S.grid(column = 1, row = 3)

label_T = ttk.Label(root, text='Время',padding=5)
label_T.grid(column = 0, row = 4)
entry_T = ttk.Entry(root)
entry_T.grid(column = 1, row = 4)

button = ttk.Button(root, text='Рассчитать расстояние', command= calculate)
button.grid(column = 0, columnspan = 2, row = 5)

result_label = ttk.Label(root,padding=5)
result_label.grid(column = 0, columnspan = 2, row = 7)

root.mainloop()

```

### Протокол работы программы:

Задача из ПЗ №2

Скорость первого автомобиля V1 км/ч, а второго - V2 км/ч, расстояние между ними S км.  
Определить расстояние между автомобилями через T часов, если они движутся навстречу друг другу.

Скорость первого автомобиля	60
Скорость второго автомобиля	75
Расстояние между автомобилями	200
Время	2

**Рассчитать расстояние**

Расстояние между автомобилями равно: 70 км

### Постановка задачи №3:

Задание предполагает, что у студента есть проект с практическими работами (№№ 2-13), оформленный согласно требованиям. Все задания выполняются с использованием модуля OS:

- | перейдите в каталог PZ11. Выведите список всех файлов в этом каталоге. Имена вложенных подкаталогов выводить не нужно.
- | перейти в корень проекта, создать папку с именем test. В ней создать еще одну папку test1. В папку test переместить два файла из ПЗ6, а в папку test1 - один файл из ПЗ7. Файл из ПЗ7 переименовать в test.txt. Вывести в консоль информацию о размере файлов в папке test.
- | перейти в папку с PZ11, найти там файл с самым коротким именем, имя вывести в консоль. Использовать функцию `basename()` (`os.path.basename()`).
- | перейти в любую папку где есть отчет в формате .pdf и «запустите» файл в привязанной к нему программе. Использовать функцию `os.startfile()`.
- | удалить файл test.txt

### Текст программы:

```
#Задание предполагает, что у студента есть проект с практическими работами (№№ 2-13),
#оформленный согласно требованиям. Все задания выполняются с использованием модуля
#OS:
#| перейдите в каталог PZ11. Выведите список всех файлов в этом каталоге. Имена
#вложенных подкаталогов выводить не нужно.
#| перейти в корень проекта, создать папку с именем test. В ней создать еще одну папку
#test1. В папку test переместить два файла из ПЗ6, а в папку test1 - один файл из
#ПЗ7.
#Файл из ПЗ7 переименовать в test.txt. Вывести в консоль информацию о размере
#файлов в папке test.
#| перейти в папку с PZ11, найти там файл с самым коротким именем, имя вывести в
#консоль. Использовать функцию basename() (os.path.basename()).
#| перейти в любую папку где есть отчет в формате .pdf и «запустите» файл в
#привязанной к нему программе. Использовать функцию os.startfile().
#| удалить файл test.txt

import os

print('Файлы каталога PZ_11: ', os.listdir('../PZ_11'))

os.chdir('..')
os.makedirs('test/test1')
os.replace('../PZ_6/PZ_6_1.py', '../test/PZ_6_1.py')
os.replace('../PZ_6/PZ_6_2.py', '../test/PZ_6_2.py')
os.replace('../PZ_7/PZ_7_1.py', '../test/test1/PZ_7_1.py')
os.rename('../test/test1/PZ_7_1.py', '../test/test1/test.txt')
print(f"Информация о размере: {os.stat('../test')}")

print(f"Название файла: {os.path.basename('../PZ_11/data.txt')}")

os.chdir('../Reports')
os.startfile('Report PZ_2.pdf', 'open')

os.remove('../test/test1/test.txt')
```

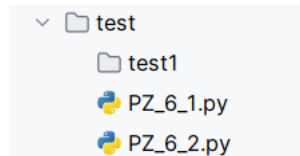
## Протокол работы программы:

Файлы каталога PZ\_11: ['data.txt', 'data\_2.txt', 'PZ\_11\_1.py', 'PZ\_11\_2.py', 'text18-1.txt', 'text18-2.txt']

Информация о размере: os.stat\_result(st\_mode=16895, st\_ino=50102545854497397, st\_dev=1477050819, st\_nlink=1, st\_uid=0, st\_gid=0, st\_size=0, st\_atime=1717794627, st\_mtime=1717792741, st\_ctime=1717792705)

Название файла: data.txt

Process finished with exit code 0



**Вывод:** в процессе выполнения практического занятия закрепил усвоенные знания, понятия, алгоритмы, основные принципы составления программ с использованием GUI Tkinter в IDE PyCharm Community, а также изучил возможности модуля OS.

Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовый программный код выложен на GitHub.



