

Практическое занятие № 17

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

Цель: 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.

Задание 1:

Постановка задачи: 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:

```
import tkinter as tk
from tkinter import ttk

def on_entry_click(event):
    if about_text.get("1.0", "end-1c") == 'Краткая
информация о ваших увлечениях':
        about_text.delete("1.0", tk.END)
        about_text.config(fg='black')

def on_focusout(event):
    if about_text.get("1.0", "end-1c") == '':
        about_text.insert("1.0", 'Краткая информация о
ваших увлечениях')
        about_text.config(fg='grey')

def confirm():
    name = name_entry.get()
    password = password_entry.get()
    age = age_entry.get()
    gender = gender_var.get()
```

```

        interests = ", ".join([interests_var[i].get() for i
in range(3)])
        country = country_entry.get()
        city = city_entry.get()
        about = about_text.get("1.0", "end-1c")

        confirmation_label.config(text=f"Имя:
{name}\nПароль: {password}\nВозраст: {age}\nПол:
{gender}\nУвлечения: {interests}\nСтрана:
{country}\nГород: {city}\nО себе: {about}", fg="white")

def clear_form():
    name_entry.delete(0, tk.END)
    password_entry.delete(0, tk.END)
    age_entry.delete(0, tk.END)
    about_text.delete("1.0", tk.END)
    about_text.insert("1.0", "Краткая информация о ваших
увлечениях")
    about_text.config(fg='grey')
    for i in range(len(interests_var)):
        interests_var[i].set(0)
    country_var.set(countries[0])
    city_var.set(cities[0])

root = tk.Tk()
root.title("Форма регистрации пользователя")
root.geometry("750x580")

style = ttk.Style()
style.configure('TFrame', background='white')

main_frame = ttk.Frame(root, padding=10, style='TFrame')
main_frame.grid(row=0, column=0, padx=10, pady=10,
sticky='nsew')

title_label = ttk.Label(main_frame, text="Форма
регистрации пользователя", font=('Arial', 16, 'bold'))
title_label.grid(row=0, column=0, columnspan=2, pady=10)

form_frame = ttk.Frame(main_frame, relief='groove',
borderwidth=2, style='TFrame')
form_frame.grid(row=1, column=0, columnspan=1, padx=10,
pady=10, sticky='nsew')

style = ttk.Style()
style.configure('TLabel', font=('Arial', 12),

```

```
background='white')
style.configure('TEntry', font=('Arial', 12),
fieldbackground='grey')
style.configure('TButton', font=('Arial', 12),
background='lightblue', foreground='black')
style.configure('TRadiobutton', font=('Arial', 12),
background='white')
style.configure('TCheckbutton', font=('Arial', 12),
background='white')

name_label = ttk.Label(form_frame, text="Ваше имя:",
background='white')
name_label.grid(row=0, column=0, padx=10, pady=5,
sticky="w")

name_entry = ttk.Entry(form_frame, width=60,
style='TEntry')
name_entry.grid(row=0, column=1, columnspan=2, padx=10,
pady=5, sticky="w")

password_label = ttk.Label(form_frame, text="Пароль:",
background='white')
password_label.grid(row=1, column=0, columnspan=2,
padx=10, pady=5, sticky="w")

password_entry = ttk.Entry(form_frame, show="*",
width=60, style='TEntry')
password_entry.grid(row=1, column=1, columnspan=2,
padx=10, pady=5, sticky="w")

age_label = ttk.Label(form_frame, text="Возраст:",
background='white')
age_label.grid(row=2, column=0, columnspan=2, padx=10,
pady=5, sticky="w")

age_entry = ttk.Entry(form_frame, width=60,
style='TEntry')
age_entry.grid(row=2, column=1, columnspan=2, padx=10,
pady=5, sticky="w")

gender_label = ttk.Label(form_frame, text="Пол:",
background='white')
gender_label.grid(row=3, column=0, sticky="w", padx=10,
pady=5)
gender_var = tk.StringVar()
gender_var.set("Мужской")
male_radio = ttk.Radiobutton(form_frame, text="Мужской",
variable=gender_var, value="Мужской",
```

```

style='TRadiobutton')
female_radio = ttk.Radiobutton(form_frame,
text="Женский", variable=gender_var, value="Женский",
style='TRadiobutton')
male_radio.grid(row=3, column=1, padx=10, pady=5,
sticky="w")
female_radio.grid(row=3, column=2, padx=10, pady=5,
sticky="w")

interests_label = ttk.Label(form_frame, text="Ваши
увлечения:", background='white')
interests_label.grid(row=4, column=0, sticky="w",
padx=10, pady=5)
interests_var = [tk.IntVar() for _ in range(3)]
interests = ["Музыка", "Видео", "Рисование"]
[ttk.Checkbutton(form_frame, text=interests[i],
variable=interests_var[i],
style='TCheckbutton').grid(row=4, column=i+1, padx=10,
pady=5, sticky="w") for i in range(len(interests))]

country_label = ttk.Label(form_frame, text="Ваша
страна:", background='white')
country_label.grid(row=7, column=0, sticky="w", padx=10,
pady=5)
countries = ["Россия", "США", "Германия"]
country_var = tk.StringVar()
country_var.set(countries[0])
country_entry = ttk.Combobox(form_frame,
values=countries, textvariable=country_var,
style='TCombobox')
country_entry.grid(row=7, column=1, padx=10, pady=5,
sticky="ew")

city_label = ttk.Label(form_frame, text="Ваш город:",
background='white')
city_label.grid(row=8, column=0, sticky="w", padx=10,
pady=5)
cities = ["Москва", "Нью-Йорк", "Берлин"] # Пример
списка городов
city_var = tk.StringVar()
city_var.set(cities[0])
city_entry = ttk.Combobox(form_frame, values=cities,
textvariable=city_var, style='TCombobox')
city_entry.grid(row=8, column=1, padx=10, pady=5,
sticky="ew")

about_label = ttk.Label(form_frame, text="Кратко о
себе:", background='white')

```

```
about_label.grid(row=9, column=0, sticky="w", padx=10,
pady=5)
about_text = tk.Text(form_frame, font=('Arial', 12),
width=40, height=5)
about_text.insert("1.0", "Краткая информация о ваших
увлечениях")
about_text.config(fg='grey')
about_text.bind('<FocusIn>', on_entry_click)
about_text.bind('<FocusOut>', on_focusout)
about_text.grid(row=9, column=1, columnspan=2, padx=10,
pady=5, sticky="nsew")
```

```
calculation_label = ttk.Label(form_frame, text="Решите
пример, запишите результат в поле ниже:",
background='white')
calculation_label.grid(row=11, column=0, columnspan=2,
padx=10, pady=5, sticky="w")
```

```
calculation_entry = ttk.Entry(form_frame, width=40,
style='TEntry')
calculation_entry.grid(row=12, column=1, columnspan=2,
padx=10, pady=5, sticky="ew")
```

```
confirm_button = ttk.Button(form_frame, text="Отменить
ввод", command=clear_form, style='TButton')
confirm_button.grid(row=13, column=1, padx=10, pady=10,
sticky="e")
```

```
cancel_button = ttk.Button(form_frame, text="Данные
подтверждаю ", command=root.quit, style='TButton')
cancel_button.grid(row=13, column=2, padx=10, pady=10,
sticky="w")
confirmation_label = ttk.Label(form_frame, text="",
font=('Arial', 12), background='white')
confirmation_label.grid(row=14, column=0, columnspan=2,
sticky='nsew')
```

```
root.mainloop()
```

Протокол работы программы:

Форма регистрации пользователя

Форма регистрации пользователя

Ваше имя:

Пароль:

Возраст:

Пол: ☒ Мужской ☐ Женский

Ваши увлечения: ☐ Музыка ☐ Видео ☐ Рисование

Ваша страна:

Ваш город:

Кратко о себе:

Решите пример, запишите результат в поле ниже:

Задание 2:

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

Текст программы:

```
import tkinter as tk
from tkinter import ttk

def display_letters():
    try:
        n = int(entry.get())
        if 1 < n < 26:
            letters = [chr(i + 65) for i in range(n)]
            result_label.config(text="Первые {} буквы алфавита: {}".format(n, ".join(letters)"))
        else:
            result_label.config(text="Число N должно быть в диапазоне (1, 26)")
    except ValueError:
        result_label.config(text="Введите целое число")
```

```

root = tk.Tk()
root.title("Вывод первых N букв алфавита")
root.geometry("700x200") # Установка размера окна

style = ttk.Style()
style.configure('TLabel', font=('Arial', 12))
style.configure('TButton', font=('Arial', 12))

label = ttk.Label(root, text="Введите число N (1 < N < 26):")
label.pack(pady=10)

entry = ttk.Entry(root, width=5, font=('Arial', 12))
entry.pack(pady=5)

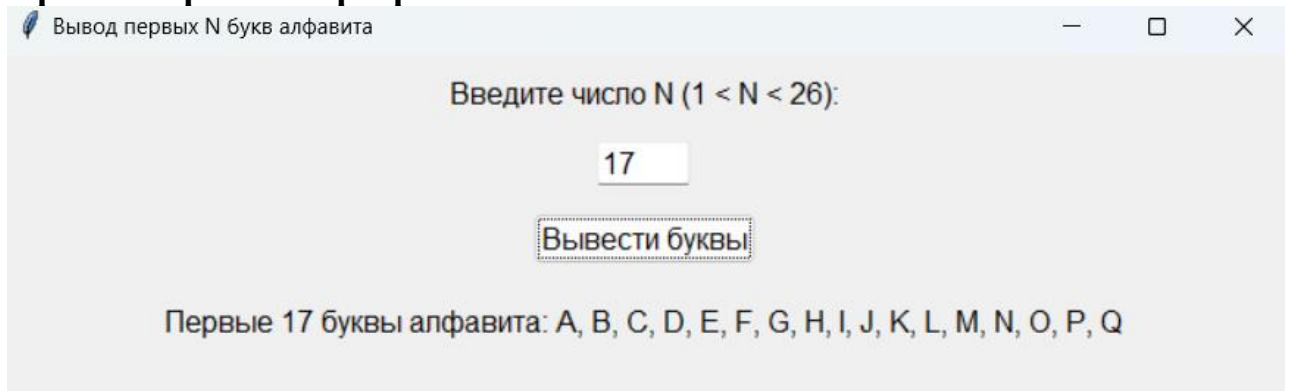
button = ttk.Button(root, text="Вывести буквы",
command=display_letters, style='TButton')
button.pack(pady=10)

result_label = ttk.Label(root, text="", style='TLabel')
result_label.pack(pady=10)

root.mainloop()

```

Протокол работы программы:



Задание 3:

Постановка задачи 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:

```
import os
```

```
folder_path = r"C:\Users\Yana\Documents\vedenova is-22\PZ 11"
```

```
os.chdir(folder_path)
```

```
files = os.listdir()
```

```
print("\nСписок всех файлов в каталоге PZ11:")
```

```
for file in files:
```

```
    if os.path.isfile(os.path.join(folder_path, file)):
        print(file)
```

```
# 2
```

```
os.chdir("../")
```

```
os.makedirs("test/test1")
```

```
os.replace(r"C:\Users\Yana\Documents\vedenova is-22\PZ6\PZ6_1.py", r"C:\Users\Yana\Documents\vedenova is-22\test\PZ6_1.py")
```

```
os.replace(r"C:\Users\Yana\Documents\vedenova is-22\PZ6\PZ6_2.py", r"C:\Users\Yana\Documents\vedenova is-22\test\PZ6_2.py")
```

```
os.replace(r"C:\Users\Yana\Documents\vedenova is-22\PZ7\PZ7_1.py", r"C:\Users\Yana\Documents\vedenova is-22\test\test1\test.txt")
```

```
folder_path = "C:/Users/Yana/Documents/vedenova is-22/test"
```

```
files = os.listdir(folder_path)
```

```
print('\nРазмеры файлов в папке test:')
```

```
for file in files:
```

```
    file_path = os.path.join(folder_path, file)
```

```
    if os.path.isfile(file_path):
```

```
        file_size = os.path.getsize(file_path)
```

```
        print(f"File: {file} | Size: {file_size} bytes")
```


3

```
os.chdir("C:/Users/Yana/Documents/vedenova is-22/PZ 11")
files = os.listdir()
shortest_file = min(files, key=len)
shortest_file_name = os.path.basename(shortest_file)
print(f"\n Файл с самым коротким именем в PZ11:
{shortest_file_name}")
```

4

```
folder_path = "C:/Users/Yana/Documents/vedenova is-
22/reports"
os.chdir(folder_path)
os.startfile("C:/Users/Yana/Documents/vedenova is-
22/reports/PZ_2.pdf")
```

5

```
file_path = "C:/Users/Yana/Documents/vedenova is-
22/test/test1/test.txt"
if os.path.exists(file_path):
    os.remove(file_path)
    print("File deleted successfully.")
else:
    print("File not found.")
```

Протокол работы программы:

Список всех файлов в каталоге PZ11:

```
file.txt
поem.txt
PZ11_1.py
PZ11_2.py
sequence.txt
text18-4.txt
__init__.py
```

Размеры файлов в папке test:

```
File: PZ6_1.py | Size: 683 bytes
File: PZ6_2.py | Size: 828 bytes
```

Файл с самым коротким именем в PZ11: test
File deleted successfully.

Process finished with exit code 0

Вывод: в процессе выполнения практического занятия выработала навыки составления программ с использованием GUI Tkinter в IDE PyCharm Community, изучила возможности модуля OS. Готовые программные коды выложены на GitHub.