

Практическое задание № 17

Тема: составление программ с использованием GUI Tkinter и модуля OS.

Цель: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ с использованием GUI Tkinter и модуля OS в IDE PyCharm Community

1. часть

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.

2. часть

Разработать программу с применением пакета tk, взяв в качестве условия одну любую задачу из ПЗ NoNo 2 – 9.

3. часть

Задание 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).

Задание 2. Разработать программу с применением пакета tk, взяв в качестве условия одну любую задачу из ПЗ NoNo 2 – 9.

Задание 3.

Задание предполагает, что у студента есть проект с практическими работами (NoNo 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 1.

```
#Разработать программу с применением пакета tk
#Используя словарь посчитать количество уникальных слов в заданном
#предложении «Изучаем язык Питон». Вывести на экран каждую пару
#«ключ:значение».

import tkinter as tk

from tkinter import ttk

def create_form():

    root = tk.Tk()

    root.title("Testform")

    frame = ttk.Frame(root, padding="10")

    frame.grid(row=0, column=0, sticky=(tk.W, tk.E, tk.N, tk.S))

    ttk.Label(frame, text="Name").grid(row=0, column=0,
sticky=tk.W, pady=2)

    name_entry = ttk.Entry(frame)

    name_entry.grid(row=0, column=1, sticky=(tk.W, tk.E), pady=2)

    ttk.Label(frame, text="Password").grid(row=1, column=0,
sticky=tk.W, pady=2)

    password_entry = ttk.Entry(frame, show="*")
```

```
password_entry.grid(row=1, column=1, sticky=(tk.W, tk.E),
pady=2)

ttk.Label(frame, text="Gender").grid(row=2, column=0,
sticky=tk.W, pady=2)

gender_frame = ttk.Frame(frame)

gender_frame.grid(row=2, column=1, sticky=(tk.W, tk.E))

gender_var = tk.StringVar()

ttk.Radiobutton(gender_frame, text="Male", variable=gender_var,
value="Male").grid(row=0, column=0, sticky=tk.W)

ttk.Radiobutton(gender_frame, text="Female",
variable=gender_var, value="Female").grid(row=0, column=1,
sticky=tk.W)

ttk.Label(frame, text="Continent").grid(row=3, column=0,
sticky=tk.W, pady=2)

continent_combo = ttk.Combobox(frame, values=["Please
select...", "Africa", "Asia", "Europe", "North America", "South
America", "Australia", "Antarctica"])

continent_combo.current(0)

continent_combo.grid(row=3, column=1, sticky=(tk.W, tk.E),
pady=2)

ttk.Label(frame, text="Meals").grid(row=4, column=0,
sticky=tk.W, pady=2)

meals_frame = ttk.Frame(frame)

meals_frame.grid(row=4, column=1, sticky=(tk.W, tk.E))

breakfast_var = tk.BooleanVar()
```

```
lunch_var = tk.BooleanVar()

dinner_var = tk.BooleanVar()

ttk.Checkbutton(meals_frame, text="breakfast",
variable=breakfast_var).grid(row=0, column=0, sticky=tk.W)

ttk.Checkbutton(meals_frame, text="lunch",
variable=lunch_var).grid(row=0, column=1, sticky=tk.W)

ttk.Checkbutton(meals_frame, text="dinner",
variable=dinner_var).grid(row=0, column=2, sticky=tk.W)


ttk.Label(frame, text="Remark").grid(row=5, column=0,
sticky=tk.W, pady=2)

remark_text = tk.Text(frame, height=5, width=40)

remark_text.grid(row=5, column=1, sticky=(tk.W, tk.E), pady=2)


button_frame = ttk.Frame(frame)

button_frame.grid(row=6, column=1, sticky=(tk.E, tk.W))

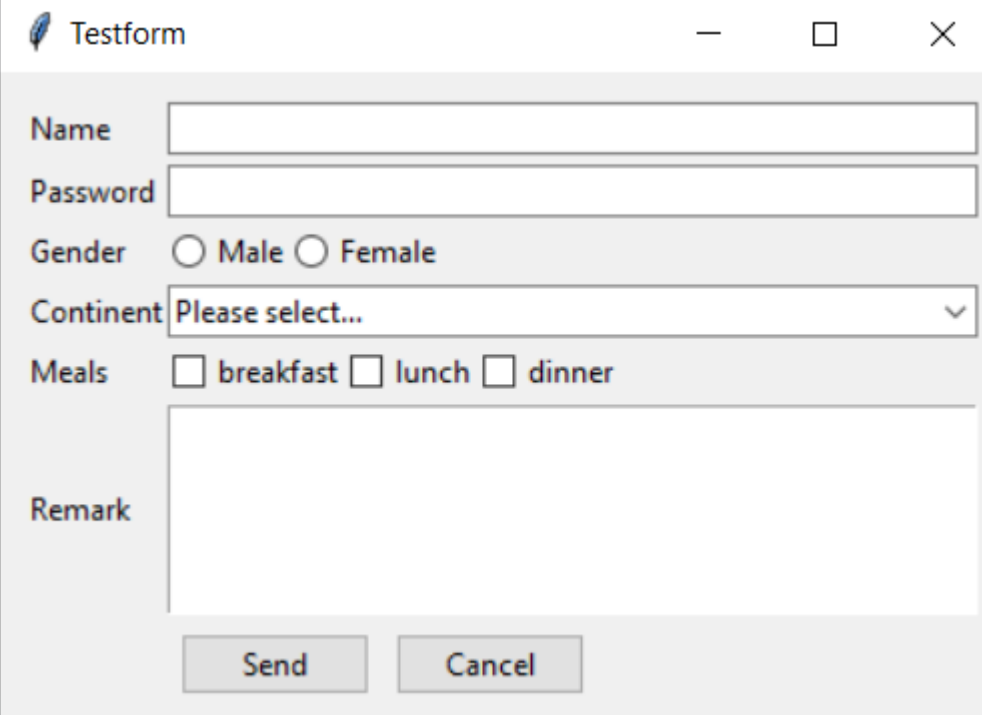
ttk.Button(button_frame, text="Send").grid(row=0, column=0,
padx=5, pady=5, sticky=tk.E)

ttk.Button(button_frame, text="Cancel").grid(row=0, column=1,
padx=5, pady=5, sticky=tk.E)


root.mainloop()

create_form()
```

Протокол работы:



The image shows a Tkinter window titled "Testform" with a standard macOS-style title bar (minimize, maximize, close buttons). The window contains the following elements:

- Name:** A single-line text entry field.
- Password:** A single-line text entry field.
- Gender:** Two radio buttons labeled "Male" and "Female".
- Continent:** A dropdown menu currently showing "Please select..." with a downward arrow.
- Meals:** Three checkboxes labeled "breakfast", "lunch", and "dinner".
- Remark:** A multi-line text area.
- Buttons:** Two buttons at the bottom labeled "Send" and "Cancel".

Текст программы 2.

```
#Разработать программу с применением пакета tk
#Используя словарь посчитать количество уникальных слов в заданном
#предложении «Изучаем язык Питон». Вывести на экран каждую пару
#«ключ:значение».
import tkinter as tk

# Создать окно
window = tk.Tk()
```

```
window.title("Количество уникальных слов")

# Создать словарь для подсчёта слов
word_counts = {}

# Получить предложение из текстового поля
def get_sentence():
    sentence = entry.get()
    return sentence

# Подсчитать количество уникальных слов
def count_words():
    sentence = get_sentence()
    words = sentence.split()
    for word in words:
        if word not in word_counts:
            word_counts[word] = 0
        word_counts[word] += 1

# Вывести пары "ключ:значение" на экран
def display_results():
    count_words()
    for word, count in word_counts.items():
        result_listbox.insert(tk.END, f"{word}: {count}")

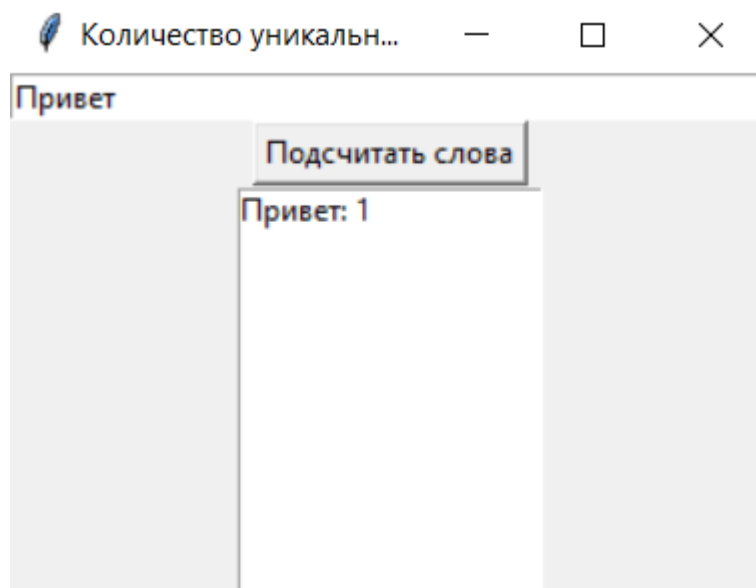
# Создать текстовое поле для ввода предложения
entry = tk.Entry(window, width=50)
entry.pack()

# Создать кнопку для подсчёта слов
count_button = tk.Button(window, text="Подсчитать слова",
                           command=display_results)
count_button.pack()

# Создать список для вывода результатов
result_listbox = tk.Listbox(window)
result_listbox.pack()

# Запустить окно
window.mainloop()
```

Протокол работы.



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