

Студент группы ИС-25 Несповитый А.Е.

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.

Постановка задачи

Блок заданий 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 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).
"""

import tkinter as tk
from tkinter import ttk

def create_form():
    root = tk.Tk()
    root.title("Форма регистрации пользователя")

    frame = ttk.Frame(root, padding="10")
    frame.grid(row=0, column=0, sticky=(tk.W, tk.E, tk.N, tk.S))

    title = ttk.Label(frame, text="Форма регистрации пользователя",
font=("Helvetica", 16))
    title.grid(row=0, column=0, columnspan=3, pady=10)

    ttk.Label(frame, text="Ваше имя:").grid(row=1, column=0,
sticky=tk.W, padx=5, pady=5)
    name_entry = ttk.Entry(frame)
    name_entry.grid(row=1, column=1, columnspan=2, sticky=(tk.W,
tk.E), pady=5)

    ttk.Label(frame, text="Пароль:").grid(row=2, column=0,
sticky=tk.W, padx=5, pady=5)
    password_entry = ttk.Entry(frame, show="*")
```

```

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

ttk.Label(frame, text="Возраст:").grid(row=3, column=0,
sticky=tk.W, padx=5, pady=5)
age_entry = ttk.Entry(frame)
age_entry.grid(row=3, column=1, columnspan=2, sticky=(tk.W,
tk.E), pady=5)

ttk.Label(frame, text="Пол:").grid(row=4, column=0, sticky=tk.W,
padx=5, pady=5)
gender_var = tk.StringVar()
ttk.Radiobutton(frame, text="Мужской", variable=gender_var,
value="Мужской").grid(row=4, column=1, pady=5, sticky=tk.W)
ttk.Radiobutton(frame, text="Женский", variable=gender_var,
value="Женский").grid(row=4, column=2, pady=5, sticky=tk.W)

ttk.Label(frame, text="Ваши увлечения:").grid(row=5, column=0,
sticky=tk.W, padx=5, pady=5)
hobbies_music = ttk.Checkbutton(frame, text="Музыка")
hobbies_video = ttk.Checkbutton(frame, text="Видео")
hobbies_drawing = ttk.Checkbutton(frame, text="Рисование")
hobbies_music.grid(row=5, column=1, sticky=tk.W, pady=5)
hobbies_video.grid(row=5, column=2, sticky=tk.W, pady=5)
hobbies_drawing.grid(row=5, column=3, sticky=tk.W, pady=5)

ttk.Label(frame, text="Ваша страна:").grid(row=6, column=0,
sticky=tk.W, padx=5, pady=5)
country_combobox = ttk.Combobox(frame)
country_combobox['values'] = ("Россия", "США", "Китай", "Япония",
"Германия")
country_combobox.grid(row=6, column=1, columnspan=2,
sticky=(tk.W, tk.E), pady=5)

ttk.Label(frame, text="Ваш город:").grid(row=7, column=0,
sticky=tk.W, padx=5, pady=5)
city_combobox = ttk.Combobox(frame)
city_combobox['values'] = ("Москва", "Нью-Йорк", "Пекин",
"Токио", "Берлин")
city_combobox.grid(row=7, column=1, columnspan=2, sticky=(tk.W,
tk.E), pady=5)

ttk.Label(frame, text="Кратко о себе:").grid(row=8, column=0,
sticky=tk.W, padx=5, pady=5)
short_info_text = tk.Text(frame, height=5, width=30)
short_info_text.grid(row=8, column=1, columnspan=2, pady=5,
sticky=(tk.W, tk.E))

```

```

    ttk.Label(frame, text="Решите пример, запишите результат в поле
ниже:").grid(row=9, column=0, columnspan=3, sticky=tk.W, padx=5,
pady=5)
    result_entry = ttk.Entry(frame)
    result_entry.grid(row=10, column=0, columnspan=3, sticky=(tk.W,
tk.E), pady=5)

    cancel_button = ttk.Button(frame, text="Отменить ввод")
    confirm_button = ttk.Button(frame, text="Данные подтверждаю")
    cancel_button.grid(row=11, column=0, pady=10)
    confirm_button.grid(row=11, column=2, pady=10)

    for widget in frame.winfo_children():
        widget.grid_configure(padx=5, pady=5)

    root.mainloop()

if __name__ == "__main__":
    create_form()

```

Протокол программы:

The screenshot shows a graphical user interface for a user registration form. The window has a title bar with the text "Форма регистрации пользователя" and standard minimize, maximize, and close buttons. The form itself has a light gray background and a title "Форма регистрации пользователя" at the top. It contains several input fields and buttons:

- Ваше имя:** A text input field.
- Пароль:** A text input field.
- Возраст:** A text input field.
- Пол:** Two radio buttons labeled "Мужской" and "Женский".
- Ваши увлечения:** Three checkboxes labeled "Музыка", "Видео", and "Рисование".
- Ваша страна:** A dropdown menu.
- Ваш город:** A dropdown menu.
- Кратко о себе:** A large text area for a short bio.
- Решите пример, запишите результат в поле ниже:** A text input field for a math problem result.
- Buttons:** Two buttons at the bottom, "Отменить ввод" (Cancel input) and "Данные подтверждаю" (Confirm data).

Блок заданий 2

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

Текст программы:

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

def validate_number(number_str):
    """
    Функция проверки, является ли введенное значение трехзначным
    числом.
    """
    if len(number_str.replace('-', '')) == 3 and
number_str.replace('-', '').isdigit():
        return True
    else:
        return False

def get_first_digit(number):
    """
    Функция получения первой цифры из трехзначного числа.
    """
    return abs(int(number)) // 100

def show_result(number):
    """
    Функция отображения результата в окне Tkinter.
    """
    first_digit = get_first_digit(number)
    result_label.config(text=f"Первая цифра: {first_digit}")

def main():
    """
    Основная функция программы.
    """
    number_str = number_entry.get()
    if validate_number(number_str):
        show_result(number_str)
        error_label.config(text='')
    else:
```

```

        result_label.config(text='')
        error_label.config(text="Введите трёхзначное число!")

# Создание окна Tkinter
root = tk.Tk()
root.title("Первая цифра трёхзначного числа")

# Создание элементов интерфейса
number_label = tk.Label(root, text="Введите число:")
number_entry = tk.Entry(root)
result_label = tk.Label(root, text="")
error_label = tk.Label(root, text="", fg="red")
submit_button = tk.Button(root, text="Показать результат",
command=main)

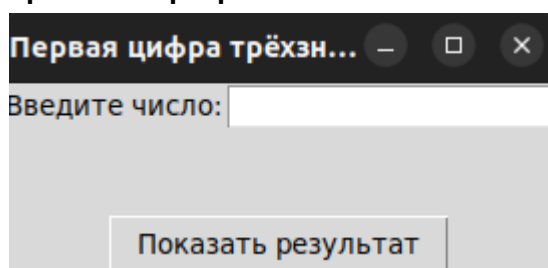
# Размещение элементов на окне
number_label.grid(row=0, column=0, sticky="w")
number_entry.grid(row=0, column=1)
result_label.grid(row=1, column=0, columnspan=2)
error_label.grid(row=2, column=0, columnspan=2)
submit_button.grid(row=3, column=0, columnspan=2)

# Запуск цикла обработки событий
root.mainloop()

if __name__ == "__main__":
    main()

```

Протокол программы:



Блок заданий 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:

```
import os

pz6 = '/home/alexey/PycharmProjects/Proj_1sem_Nespovitiyy/PZ_6'
pz7 = '/home/alexey/PycharmProjects/Proj_1sem_Nespovitiyy/PZ_7'
pz11 = '/home/alexey/PycharmProjects/Proj_1sem_Nespovitiyy/PZ_11'
report =
'/home/alexey/PycharmProjects/Proj_1sem_Nespovitiyy/Reports/'
p7_name = ''

# 1. Переход в каталог PZ11 и вывод списка файлов
os.chdir(pz11)
files = os.listdir()
for file in files:
    if not os.path.isdir(file):
        print(file)

# 2. Создание папок, перемещение файлов и вывод информации о
размере
os.chdir("../") # Переход в корень проекта
os.makedirs("test/test1", exist_ok=True) # Создание папок
for c, file in enumerate(os.listdir(pz6), start=1): # Перемещение
двух файлов из ПЗ6
    if c <= 2:
        os.rename(pz6 + "/" + file, "test/" + file)
    else:
        break
for c, file in enumerate(os.listdir(pz7), start=1): # Перемещение
одного файла из ПЗ7
    if c <= 1:
        os.rename(pz7 + "/" + file, "test/test1/test.txt")
    else:
        break
file_sizes = [file + ': ' + str(os.path.getsize("test/" + file)) +
' byte' for file in os.listdir("test")]
print("Размеры файлов в папке test:", file_sizes)

# 3. Поиск файла с самым коротким именем в PZ11
# shortest_filename = [[int(len(str(i))), str(i)] for i in
os.listdir(pz11)]
```

```

shortest_filename_2 = min([len(os.path.basename(i)),
os.path.basename(i)] for i in os.listdir(pz11)], key=lambda x:
x[0])
print("Файл с самым коротким именем:", shortest_filename_2)

# 4. Запуск PDF-файла
# os.startfile(report + pdf_file)
# os.system("evince
/home/alexey/PycharmProjects/Proj_1sem_Nespovitiyy/Reports/PZ_3_22
_1.pdf")

# 5. Удаление файла test.txt
with
open('/home/alexey/PycharmProjects/Proj_1sem_Nespovitiyy/test/test
1/test.txt', 'r') as file:
    print(file.read())
os.remove("/home/alexey/PycharmProjects/Proj_1sem_Nespovitiyy/test
/test1/test.txt")

```

Протокол программы:

```

/home/alexey/PycharmProjects/Proj_1sem_Nespovitiyy/venv/bin/python
/home/alexey/PycharmProjects/Proj_1sem_Nespovitiyy/PZ_17/PZ_17_22_3.py
ready_minus.txt
plus.txt
text18-22.txt
__init__.py
minus.txt
new_text.txt
PZ_11_22_2.py
PZ_11_22_1.py
ready_plus.txt
Размеры файлов в папке test: ['PZ_6_22_1.py: 645 byte', 'test1: 4096 byte', '__init__.py: 0
byte']
Файл с самым коротким именем: [8, 'plus.txt']
def find_longest_word_length():
    try:
        sentence = str(input("Введите строку из русских слов: "))
        words = sentence.split() # Разделение строки на слова
        longest_word_length = 0

        for word in words:
            word_length = len(word)
            if word_length > longest_word_length:
                longest_word_length = word_length

        length = longest_word_length
        # Вывод результата

```

```
print("Длина самого длинного слова:", length)
except:
print('ОШИБКА! ТРЕБУЕТСЯ ВЕСТИ ДАННЫЕ ПОВТОРНО!')
find_longest_word_length()
```

```
if __name__ == '__main__':
    find_longest_word_length()
```

Process finished with exit code 0

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