

Практическое занятие №17

Тема: Составление программ с использованием GUI Tkinter в IDE PyCharm, 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, изучить возможности модуля OS.

Постановка задачи:

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 № 2 – 9.

Задание 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.

Тип алгоритма: Циклический, Линейный

Текст программы:

17.1:

```
import tkinter as tk
from tkinter import ttk

# Функция, вызываемая при нажатии кнопки "Отправить информацию"
def submit_form():
    name = entry_name.get().strip()
    email = entry_email.get().strip()
    age = entry_age.get().strip()

    # Проверка обязательных полей
    if not name or not email or not age:
        print("Заполните обязательные поля!")
        return

    phone = entry_phone.get()
```

```

gender = gender_var.get()
qualities = entry_qualities.get("1.0", tk.END).strip()
favorite_animals = [animal for animal, var in animal_vars.items() if var.get()]

# Вывод введенной информации в консоль
print(f"Имя: {name}")
print(f"Телефон: {phone}")
print(f"Email: {email}")
print(f"Возраст: {age}")
print(f"Пол: {gender}")
print(f"Личные качества: {qualities}")
print(f"Любимые животные: {favorite_animals}")

# Инициализация главного окна
root = tk.Tk()
root.title("Форма заявки на работу в зоопарке")
root.geometry("450x410") # Задание фиксированного размера окна

# Контактная информация
frame_contact_info = tk.LabelFrame(root, text="Контактная информация", width=380)
frame_contact_info.grid(row=0, column=0, padx=10, pady=10, sticky="ew")

tk.Label(frame_contact_info, text="Имя*").grid(row=0, column=0, sticky="w")
entry_name = tk.Entry(frame_contact_info, width=30)
entry_name.grid(row=0, column=1)

tk.Label(frame_contact_info, text="Телефон").grid(row=1, column=0, sticky="w")
entry_phone = tk.Entry(frame_contact_info, width=30)
entry_phone.grid(row=1, column=1)

tk.Label(frame_contact_info, text="Email*").grid(row=2, column=0, sticky="w")
entry_email = tk.Entry(frame_contact_info, width=30)
entry_email.grid(row=2, column=1)

# Персональная информация
frame_personal_info = tk.LabelFrame(root, text="Персональная информация", width=380)
frame_personal_info.grid(row=1, column=0, padx=10, pady=10, sticky="ew")

tk.Label(frame_personal_info, text="Возраст*").grid(row=0, column=0, sticky="w")
entry_age = tk.Entry(frame_personal_info, width=30)
entry_age.grid(row=0, column=1)

tk.Label(frame_personal_info, text="Пол").grid(row=1, column=0, sticky="w")
gender_var = tk.StringVar(value="Женщина")
gender_dropdown = ttk.Combobox(frame_personal_info, textvariable=gender_var, width=27)
gender_dropdown['values'] = ("Женщина", "Мужчина")
gender_dropdown.grid(row=1, column=1)

tk.Label(frame_personal_info, text="Перечислите личные качества").grid(row=2, column=0,
sticky="nw")
entry_qualities = tk.Text(frame_personal_info, width=30, height=5)
entry_qualities.grid(row=2, column=1)

# Любимые животные
frame_favorite_animals = tk.LabelFrame(root, text="Выберите ваших любимых животных",
width=380)
frame_favorite_animals.grid(row=2, column=0, padx=10, pady=10, sticky="ew")

# Создание флажков для выбора любимых животных
animal_vars = {
    "Зебра": tk.BooleanVar(),
    "Слон": tk.BooleanVar(),
    "Кошка": tk.BooleanVar(),
    "Антилопа": tk.BooleanVar(),
    "Анаконда": tk.BooleanVar(),
    "Голубь": tk.BooleanVar(),
    "Человек": tk.BooleanVar(),
    "Краб": tk.BooleanVar()
}

```

```

}

for i, (animal, var) in enumerate(animal_vars.items()):
    tk.Checkbutton(frame_favorite_animals, text=animal, variable=var).grid(row=i // 4,
column=i % 4, sticky="w")

# Кнопка отправки информации
submit_button = tk.Button(root, text="Отправить информацию", command=submit_form)
submit_button.grid(row=3, column=0, pady=10)

# Настройка столбцов для одинаковой ширины
root.grid_columnconfigure(0, weight=1)
frame_contact_info.grid_columnconfigure(0, weight=1)
frame_contact_info.grid_columnconfigure(1, weight=1)
frame_personal_info.grid_columnconfigure(0, weight=1)
frame_personal_info.grid_columnconfigure(1, weight=1)
frame_favorite_animals.grid_columnconfigure(0, weight=1)
frame_favorite_animals.grid_columnconfigure(1, weight=1)
frame_favorite_animals.grid_columnconfigure(2, weight=1)
frame_favorite_animals.grid_columnconfigure(3, weight=1)

# Запуск приложения
root.mainloop()

```

17.2:

```

"""

```

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

1) перейдите в каталог PZ_11. Выведите список всех файлов в этом каталоге. Имена вложенных подкаталогов выводить не нужно.

2) Перейти в корень проекта, создать папку с именем test. В ней создать еще одну папку test1. В папку test переместить два файла из PZ_6, а в папку test1 - один файл из PZ_7. Файл из PZ_7 переименовать в test.txt. Вывести в консоль информацию о размере файлов в папке test.

3) Перейти в папку с PZ_11, найти там файл с самым коротким именем, имя вывести в консоль. Использовать функцию `basename()` (`os.path.basename()`).

4) Перейти в любую папку где есть отчет в формате .pdf и «запустить» файл в привязанной к нему программе. Использовать функцию `startfile()`.

5) Удалить файл test.txt.

```

"""

```

```

import os

```

```

# 1. Перейти в каталог PZ_11 и вывести список всех
# файлов в этом каталоге.

```

```

os.chdir("PZ_11")
files_in_pz11 = [f for f in os.listdir() if os.path.isfile(f)]
print("Список файлов в PZ_11:", files_in_pz11)

```

```

"""

```

```

2. Перейти в корень проекта, создать папку test и в
ней папку test1. Переместить два файла из PZ_6 в test
и один файл из PZ_7 в test1. Переименовать файл из
PZ_7 в test.txt. Вывести размер файлов в папке test.
"""

```

```

# Переход в корень проекта
os.chdir("..")
# Создание папок test и test1
os.makedirs("test/test1", exist_ok=True)

# Перемещение файлов из PZ_6 в test
pz6_files = os.listdir("PZ_6")[:2]
for file in pz6_files:
    os.rename(f"PZ_6/{file}", f"test/{file}")

# Перемещение и переименование файла из PZ_7 в test1
pz7_file = os.listdir("PZ_7")[0]
os.rename(f"PZ_7/{pz7_file}", "test/test1/test.txt")

# Вывод информации о размере файлов в папке test
test_files = [f for f in os.listdir("test")
               if os.path.isfile(os.path.join("test", f))]
for file in test_files:
    print(f"Файл: {file}, Размер: "
          f"{os.path.getsize(os.path.join("test", file))} байт")

# 3. Перейти в папку PZ_11 и найти файл с самым
# коротким именем, имя вывести в консоль.
os.chdir("PZ_11")
shortest_file = min(files_in_pz11, key=len)
print("Файл с самым коротким именем в PZ_11:",
      os.path.basename(shortest_file))

"""
4. Перейти в любую папку, где есть отчет в формате .pdf
и «запустить» файл в привязанной к нему программе.
(Для примера, предположим, что PDF файл находится в
папке PZ_8)
"""
os.chdir("../PZ_8") # Переход в папку PZ_8
pdf_files = [f for f in os.listdir() if f.endswith(".pdf")]
if pdf_files:
    os.startfile(pdf_files[0])

# 5. Удалить файл test.txt.
os.remove("../test/test1/test.txt")

```

Протокол работы программы:

17.1

Имя: Степан

Телефон: +79998887777

Email: pochta@gmail.com

Возраст: 18

Пол: Мужчина

Личные качества: Обучение веб программирования

Любимые животные: ['Слон', 'Кошка', 'Краб']

17.2

Список файлов в PZ_11: ['file_1.txt', 'file_2.txt', 'PZ_11-(1).py', 'PZ_11-(2).py', 'text11-1.txt', 'text11-2.txt']

Файл: PZ_6-1.py, **Размер:** 433 байт

Файл: PZ_6-2.py, **Размер:** 1565 байт

Файл с самым коротким именем в PZ_11: file_1.txt

Вывод: В процессе выполнения практического задания выработал навыки составления программ с использованием GUI Tkinter в IDE PyCharm, изучил возможности модуля OS. Выполнены разработка кода, отладка, тестирование, оптимизация. Готовые программные коды выложены на GitHub.