

Практическое задание № 17

Наименование: 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.

Задача 1:

Разработать программу с применением пакета tk, взяв в качестве условия одну любую задачу из ПЗ № 2 - 9. (Выбрана ПЗ №5)

```
import tkinter as tk
from tkinter import ttk

import PZ_5 as PZ

def task_5(tab_control, history_text):
    def calculate_steps():
        num = int(entry_num.get())
        steps = PZ.find_steps_to_zero(num)
        result = f"Количество шагов до нуля числа {num}: {steps}"
        result_label.config(text=result)
        update_history("[5] " + result, history_text)

    tab_5 = ttk.Frame(tab_control)
    tab_control.add(tab_5, text='5')

    label_num = ttk.Label(tab_5, text="Введите число:")
    label_num.pack(pady=5)

    entry_num = ttk.Entry(tab_5)
    entry_num.pack(pady=5)

    calculate_button = ttk.Button(tab_5, text="Вычислить",
command=calculate_steps)
    calculate_button.pack(pady=5)

    result_label = ttk.Label(tab_5, text="")
    result_label.pack(pady=5)

def task_5_1(tab_control, history_text):
    def perform_shift():
        A1 = float(A1_entry.get())
        B1 = float(B1_entry.get())
        C1 = float(C1_entry.get())
        A2 = float(A2_entry.get())
        B2 = float(B2_entry.get())
        C2 = float(C2_entry.get())
```

```

A1, B1, C1 = PZ.ShiftLeft3(A1, B1, C1)
A2, B2, C2 = PZ.ShiftLeft3(A2, B2, C2)

result_1 = f"Числа 1 после сдвига: {A1}, {B1}, {C1}"
result_2 = f"Числа 2 после сдвига: {A2}, {B2}, {C2}"

result_label_1.config(text=result_1)
result_label_2.config(text=result_2)

update_history(f""[5.1] {result_1}\n          {result_2}""",
history_text)

tab_5_1 = ttk.Frame(tab_control)
tab_control.add(tab_5_1, text='5.1')

label_set_1 = ttk.Label(tab_5_1, text="Введите числа для
списка 1:")
label_set_1.pack(pady=5)

A1_entry = ttk.Entry(tab_5_1)
A1_entry.pack(pady=5)

B1_entry = ttk.Entry(tab_5_1)
B1_entry.pack(pady=5)

C1_entry = ttk.Entry(tab_5_1)
C1_entry.pack(pady=5)

label_set_2 = ttk.Label(tab_5_1, text="Введите числа для
списка 2:")
label_set_2.pack(pady=5)

A2_entry = ttk.Entry(tab_5_1)
A2_entry.pack(pady=5)

B2_entry = ttk.Entry(tab_5_1)
B2_entry.pack(pady=5)

C2_entry = ttk.Entry(tab_5_1)
C2_entry.pack(pady=5)

perform_button = ttk.Button(tab_5_1, text="Сдвиг",
command=perform_shift)
perform_button.pack(pady=5)

result_label_1 = ttk.Label(tab_5_1, text="")
result_label_1.pack(pady=5)

result_label_2 = ttk.Label(tab_5_1, text="")
result_label_2.pack(pady=5)

def update_history(result, history_text):
    history_text.config(state=tk.NORMAL)
    history_text.insert(tk.END, result + "\n")
    history_text.config(state=tk.DISABLED)

```

```

def main():
    root = tk.Tk()
    root.title("Задания")

    tab_control = ttk.Notebook(root)
    tab_control.pack(expand=1, fill='both')

    history_label = ttk.Label(root, text="История:")
    history_label.pack(pady=5)

    history_frame = ttk.Frame(root)
    history_frame.pack(pady=5, fill='both', expand=True)

    history_text = tk.Text(history_frame, height=10,
state=tk.DISABLED, wrap='word')
    history_text.pack(side='left', fill='both', expand=True)

    history_scrollbar = ttk.Scrollbar(history_frame,
orient='vertical', command=history_text.yview)
    history_scrollbar.pack(side='right', fill='y')

    history_text.config(yscrollcommand=history_scrollbar.set)

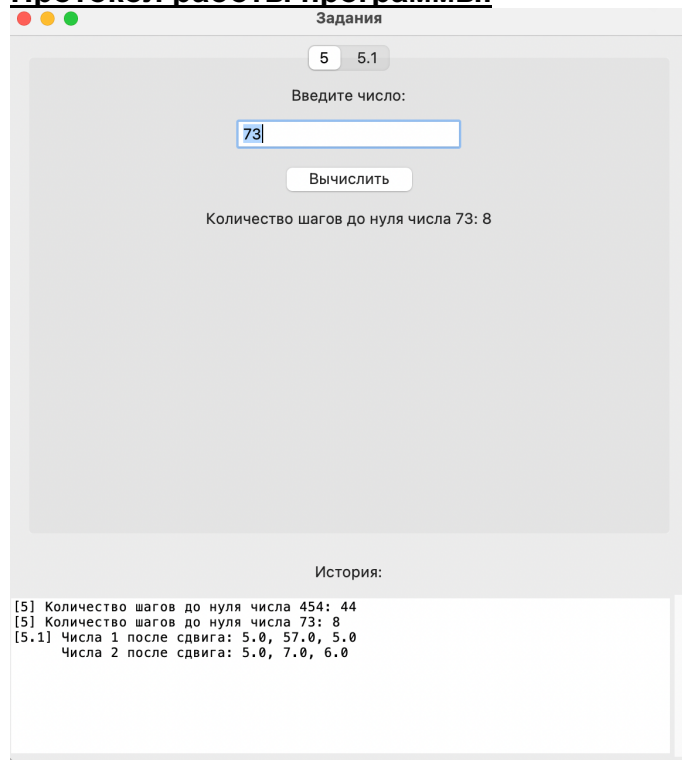
    task_5(tab_control, history_text)
    task_5_1(tab_control, history_text)

    root.mainloop()

if __name__ == "__main__":
    main()

```

Протокол работы программы:



Задания

5 5.1

Введите числа для списка 1:

5

5

57

Введите числа для списка 2:

6

5

7

Сдвиг

Числа 1 после сдвига: 5.0, 57.0, 5.0

Числа 2 после сдвига: 5.0, 7.0, 6.0

История:

[5] Количество шагов до нуля числа 454: 44
 [5] Количество шагов до нуля числа 73: 8
 [5.1] Числа 1 после сдвига: 5.0, 57.0, 5.0
 Числа 2 после сдвига: 5.0, 7.0, 6.0

Задача 2:

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
import sys
import subprocess

project_root = os.path.abspath('../')

paths = {
    "pz_6": os.path.join(project_root, 'PZ_6'),
    "pz_7": os.path.join(project_root, 'PZ_7', 'pz-7-1.py'),
```

```

    "pz_11": os.path.join(project_root, 'PZ_11'),
    "test": os.path.join(project_root, 'test'),
    "test1": os.path.join(project_root, 'test', 'test1'),
    "test_file": os.path.join(project_root, 'test', 'test1',
'test.txt'),
    "reports": os.path.join(project_root, 'reports'),
    "report_pdf": 'Отчет_pz7.pdf'
}

def open_file(filename):
    if sys.platform == "win32":
        os.startfile(filename)
    else:
        opener = "open" if sys.platform == "darwin" else "xdg-
open"
        subprocess.call([opener, filename])

def change_directory(path):
    if os.path.exists(path):
        os.chdir(path)
        return True
    else:
        print(f"Каталог {path} не найден")
        return False

def copy_file(source, destination):
    if os.path.exists(source):
        with open(source, 'rb') as f_src, open(destination, 'wb')
as f_dst:
            f_dst.write(f_src.read())
    else:
        print(f"Файл {source} не найден")

def list_files_in_directory(path):
    if os.path.exists(path):
        os.chdir(path)
        return [f for f in os.listdir() if os.path.isfile(f)]
    else:
        print(f"Каталог {path} не найден")
        return []

def create_directory(path):
    os.makedirs(path, exist_ok=True)

def print_file_sizes(directory):
    if os.path.exists(directory):
        files = [f for f in os.listdir(directory) if
os.path.isfile(os.path.join(directory, f))]
        for file in files:
            file_path = os.path.join(directory, file)
            print(f"Размер файла {file}:

```

```

{os.path.getsize(file_path)} байт")
    else:
        print(f"Каталог {directory} не найден")

os.chdir(project_root)

files_in_pz11 = list_files_in_directory(paths['pz_11'])

print("Файлы в каталоге PZ_11:", files_in_pz11)

create_directory(paths['test1'])

files_to_copy = ['pz-6-1.py', 'pz-6-2.py']
for file in files_to_copy:
    src = os.path.join(paths['pz_6'], file)
    dst = os.path.join(paths['test'], file)
    copy_file(src, dst)

copy_file(paths['pz_7'], paths['test_file'])

print_file_sizes(paths['test'])

if files_in_pz11:
    shortest_filename = min(files_in_pz11, key=len)
    print("Файл с самым коротким именем:",
os.path.basename(shortest_filename))

if change_directory(paths['reports']) and
os.path.exists(paths['report_pdf']):
    open_file(paths['report_pdf'])
else:
    print(f"PDF файл {paths['report_pdf']} не найден")

if os.path.exists(paths['test_file']):
    os.remove(paths['test_file'])
    print(f"Файл {paths['test_file']} успешно удален")
else:
    print(f"Файл {paths['test_file']} не найден для удаления")

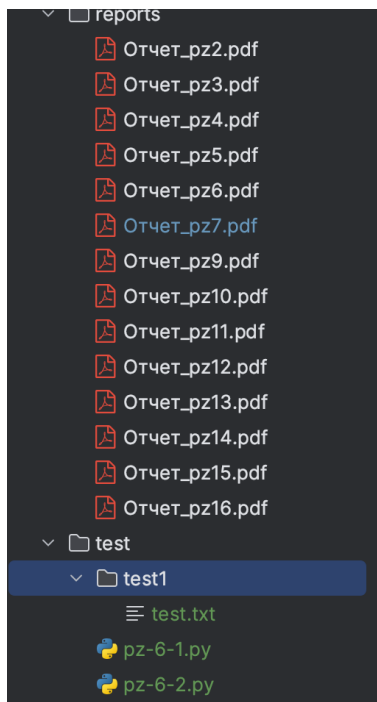
```

Протокол работы программы:

```

Файлы в каталоге PZ_11: ['__init__.py', 'pz-11.py', 'pz-11-1.py']
Размер файла pz-6-1.py: 435 байт
Размер файла pz-6-2.py: 860 байт
Файл с самым коротким именем: pz-11.py
Файл /Users/netia/Documents/python_projects/Нетяженко/test/test1/test.txt успешно удален

```



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