

## ПЗ\_17

### Первая часть

**Тема:** Составление программ с использованием GUI Tkinter, изучение возможностей модуля OS.

**Цели практического занятия:** закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ с использованием GUI Tkinter и изучить возможность модуля OS.

### №1

#### задачи:

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

**Тип алгоритма:** линейный

**Текст программы:**

### Вариант 12

```
import tkinter as tk
import random

def find_elements(lst):
    if len(lst) < 2:
        return None
    max_sum = lst[0] + lst[1]
    max_index = 0
    for i in range(len(lst)):
        if lst[i-1] + lst[i] > max_sum:
            max_sum = lst[i-1] + lst[i]
            max_index = i - 1
    return lst[max_index], lst[max_index + 1]

try:
    N = (random.randint(0, 15))
    lst = [random.randint(-100, 100) for _ in range(N)]
except:
    print('Введите число!')

def show_result():
    result_list = find_elements (lst)
    result_text.delete("1.0", "end")
```

```

result_text.insert("1.0", ''.join(str(result_list)))
result_text.insert("1.0", ''.join('Элементы с максимальной суммой: '))
result_text.insert("1.0", ''.join('\n'))
result_text.insert("1.0", ''.join(str(lst)))
result_text.insert("1.0", ''.join('Изначальный список: '))

root = tk.Tk()
root.title("список нечетных чисел")

personal_frame = tk.LabelFrame(root, text="Найти два соседних элемента, сумма которых максимальна и вывести эти элементы в порядке возрастания их индексов.")
personal_frame.grid(row=0, column=0, padx=5, pady=5, sticky="nsew")

result_text = tk.Text(personal_frame, height=30, width=85)
result_text.grid(row=0, column=0)

button = tk.Button(root, text="Сгенерировать список", command=show_result)
button.grid(row=1, column=0)

root.mainloop()

```

**Вывод:** В процессе работы я закрепил усвоенные знания, понятия, научился работать с GUI Tkinter и изучил возможности модуля OS.

## №2

### задачи:

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

**Тип алгоритма:** линейный

**Текст программы:**

### Вариант 12

```

import os
# task 1
print(os.listdir('C:\\Users\\tess\\.vscode\\IS-25\\PZ\\PZ_11'))
# task 2
try:
    os.mkdir('./test')
    os.mkdir('./test/test1')
except:
    pass

# task 3
os.rename('C:\\Users\\tess\\.vscode\\IS-25\\PZ\\PZ_6\\1.txt', './test/1.txt')

```

```
os.rename('C:\\Users\\tess\\.vscode\\IS-25\\PZ\\PZ_6\\2.txt', './test/2.txt')
os.rename('C:\\Users\\tess\\.vscode\\IS-25\\PZ\\PZ_7\\test.txt',
'./test/test1/test.txt')

# task 4
path = os.listdir('./test')
for i in path:
    print((os.path.getsize(f'./test/{i}')), 'bytes')

# task 5
path = os.listdir('C:\\Users\\tess\\.vscode\\IS-25\\PZ\\PZ_11')
path = min(path, key=len)

print(os.path.basename(f'C:\\Users\\tess\\.vscode\\IS-25\\PZ\\PZ_11\\{path}'))

# task 6

os.startfile('C:\\Users\\tess\\Desktop\\reports\\PZ_10.docx')

# task 7
os.remove('C:\\Users\\tess\\.vscode\\IS-25\\test\\test1\\test.txt')
```

**Вывод:** В процессе работы я закрепил усвоенные знания, понятия, научился работать с GUI Tkinter и изучил возможности модуля OS.