

### Практическое задание № 17

**Наименование:** составление программ с библиотеками tkinter и os в IDE PyCharm Community

**Цель:** закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ с библиотеками tkinter и os в IDE PyCharm Community.

**Задача 1.** Разработать программу с применением пакета tk, взяв в качестве условия одну любую задачу из ПЗ № 2-9. Для данной задачи был взят код из ПЗ №2 – code2.py

*# Дано трехзначное число. Вывести число, полученное при перестановке цифр сотен и десятков исходного числа. (например, 123 перейдет в 213).*

```
import tkinter as tk

def permute_digits():
    entered_num = entry.get()

    if not entered_num.isdigit() or len(entered_num) != 3:
        error_label.config(text="Ошибка: Введено не трёхзначное число!")
        return

    hundreds_digit = entered_num[0]
    tens_digit = entered_num[1]
    units_digit = entered_num[2]
    permuted_num = tens_digit + hundreds_digit + units_digit
    result_label.config(text=f"Переставленное число: {permuted_num}")

window = tk.Tk()
window.title("Перестановка цифр")

label = tk.Label(window, text="Введите трехзначное число:")
label.pack()

entry = tk.Entry(window)
entry.pack()

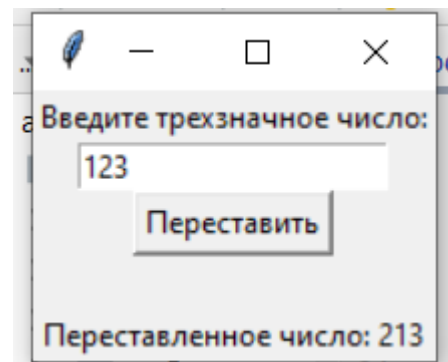
button = tk.Button(window, text="Переставить", command=permute_digits)
button.pack()

error_label = tk.Label(window, text="", fg="red")
error_label.pack()

result_label = tk.Label(window, text="")
result_label.pack()

window.mainloop()
```

#### Протокол работы программы:



**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', 'code7_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, 'pz_7'),
    "report_pdf": 'report7.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):
    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")

os.chdir(project_root)

files_in_pz11 = list_files_in_directory(paths['pz_11'])

print('Задание 1.')
print("Файлы в каталоге PZ_11:", files_in_pz11)
print()

create_directory(paths['test1'])

files_to_copy = ['code6_1.py', 'code6_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('Задание 2.')
print_file_sizes(paths['test'])
print()

print('Задание 3.')
if files_in_pz11:
    shortest_filename = min(files_in_pz11, key=len)
    print("Файл с самым коротким именем:",
os.path.basename(shortest_filename))
print()

change_directory(paths['reports']) and os.path.exists(paths['report_pdf'])
open_file(paths['report_pdf'])

print('Задание 5.')
os.path.exists(paths['test_file'])
os.remove(paths['test_file'])
print(f"Файл {paths['test_file']} успешно удален")

```

### Протокол работы программы:

Задание 1.

Файлы в каталоге PZ\_11: ['code11\_1.py', 'code11\_2.py', 'data\_3.txt', 'data\_4.txt', 'report11.pdf', 'text18-20-2.txt', 'text18-20.txt']

Задание 2.

Размер файла code6\_1.py: 920 байт

Размер файла code6\_2.py: 778 байт

Задание 3.

Файл с самым коротким именем: data\_3.txt

Задание 5.

Файл C:\Users\User\PycharmProjects\algoritms\pz\_codes\test\test1\test.txt  
успешно удален

ФАЙЛ report7.pdf УСПЕШНО ОТКРЫТ В БРАУЗЕРЕ

ПАПКИ test И test1 УСПЕШНО СОЗДАНЫ.

КОПИИ ФАЙЛОВ code6\_1.py И code6\_2.py УСПЕШНО СОЗДАНЫ

**Вывод:** в процессе выполнения практического занятия, я выработала навыки составления программ с библиотеками tkinter и os в IDE PyCharm Community. Выполнены разработка кода, отладка, тестирование, оптимизация, программного кода. Готовые программные коды выложены на GitHub.