Студент группы ИС-27 Овчаренко Евгений

Практическое задание №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 составление программ с использованием GUI Tkinter в IDE PyCharm Community, изучить возможности модуля OS.

**Постановка задачи: 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).

1. Разработать программу с применением пакета tk, взяв в качестве условия одну любую задачу из ПЗ №№ 2 – 9.
2. Задание предполагает, что у студента есть проект с практическими работами (№№ 2-13), оформленный согласно требованиям. Все задания выполняются c 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

Текст программы: 1)

import tkinter as tk from tkinter import ttk

def submit():

# Логика для обработки данных при нажатии кнопки Submit pass

def cancel():

# Логика для обработки данных при нажатии кнопки Cancel window.quit()

window = tk.Tk() window.title("Sign Up")

window.resizable(False, False) # Запрет на изменение размера окна

# Настройка стилей

style = ttk.Style()

style.configure('TButton', padding=6, relief="flat", background="#f0ad4e")

style.configure('TLabel', background='#2c3e50', foreground='white', font=('Helvetica', 10))

style.configure('TEntry', padding=6, relief="flat")

# Фрейм для размещения всех виджетов frame = tk.Frame(window, bg='#2c3e50')

frame.pack(padx=10, pady=(10, 0), fill='x')

# Заголовок

title = tk.Label(frame, text="Sign Up", bg='#f0ad4e', font=('Helvetica', 16))

title.grid(row=0, column=0, columnspan=2, pady=10, sticky='ew')

# Поля для ввода данных

labels = ['First Name', 'Last Name', 'Screen Name', 'Date of Birth', 'Gender', 'Country', 'E-mail', 'Phone', 'Password', 'Confirm Password']

entries = {}

for i, label\_text in enumerate(labels): label = ttk.Label(frame, text=label\_text)

label.grid(row=i+1, column=0, padx=5, pady=5, sticky='e') if label\_text == 'Date of Birth':

dob\_frame = tk.Frame(frame, bg='#2c3e50') dob\_frame.grid(row=i+1, column=1, pady=5, sticky='w') month = ttk.Combobox(dob\_frame, values=["January",

"February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December"], width=8)

month.set("May") month.pack(side='left')

day = ttk.Combobox(dob\_frame, values=list(range(1, 32)), width=3)

day.set(5) day.pack(side='left')

year = ttk.Combobox(dob\_frame, values=list(range(1900, 2024)), width=5)

year.set(1985) year.pack(side='left')

entries[label\_text] = (month, day, year) elif label\_text == 'Gender':

gender\_frame = tk.Frame(frame, bg='#2c3e50') gender\_frame.grid(row=i+1, column=1, pady=5, sticky='w') gender = tk.StringVar()

male\_rb = tk.Radiobutton(gender\_frame, text="Male", variable=gender, value="Male", bg='#2c3e50', foreground='white', selectcolor='#2c3e50')

male\_rb.pack(side='left')

female\_rb = tk.Radiobutton(gender\_frame, text="Female", variable=gender, value="Female", bg='#2c3e50', foreground='white', selectcolor='#2c3e50')

female\_rb.pack(side='left') gender.set("Male") entries[label\_text] = gender

elif label\_text == 'Country':

country = ttk.Combobox(frame, values=["USA", "Canada", "UK", "Australia", "Other"], width=17)

country.set("USA")

country.grid(row=i+1, column=1, pady=5, sticky='w') entries[label\_text] = country

else:

entry = ttk.Entry(frame, show='\*' if 'Password' in label\_text else '')

entry.grid(row=i+1, column=1, padx=5, pady=5, sticky='ew') entries[label\_text] = entry

# Checkbox

terms = tk.Checkbutton(frame, text="I agree to the Terms of Use", bg='#2c3e50', foreground='white', selectcolor='#2c3e50') terms.grid(row=len(labels)+1, column=0, columnspan=2, pady=5, sticky='w')

# Кнопки

button\_frame = tk.Frame(frame, bg='#2c3e50') button\_frame.grid(row=len(labels)+2, column=0, columnspan=2, pady=10)

submit\_button = ttk.Button(button\_frame, text="Submit", command=submit)

submit\_button.pack(side='left', padx=5)

cancel\_button = ttk.Button(button\_frame, text="Cancel", command=cancel)

cancel\_button.pack(side='left', padx=5) window.mainloop()

2)

import tkinter as tk

from tkinter import messagebox

def check\_values():

try:

a = int(entry\_a.get())

b = int(entry\_b.get())

c = int(entry\_c.get())

if a == -b or a == -c or b == -c: messagebox.showinfo("Результат", "True")

else:

messagebox.showinfo("Результат", "False") except ValueError:

messagebox.showerror("Ошибка", "Введите валидные числа")

# Создание основного окна window = tk.Tk()

window.title("Проверка чисел на противоположность")

# Метки и поля для ввода чисел

tk.Label(window, text="Введите первое число:").grid(row=0, column=0, padx=10, pady=5)

entry\_a = tk.Entry(window) entry\_a.grid(row=0, column=1, padx=10, pady=5)

tk.Label(window, text="Введите второе число:").grid(row=1, column=0, padx=10, pady=5)

entry\_b = tk.Entry(window) entry\_b.grid(row=1, column=1, padx=10, pady=5)

tk.Label(window, text="Введите третье число:").grid(row=2, column=0, padx=10, pady=5)

entry\_c = tk.Entry(window) entry\_c.grid(row=2, column=1, padx=10, pady=5)

# Кнопка для проверки значений

check\_button = tk.Button(window, text="Проверить", command=check\_values)

check\_button.grid(row=3, column=0, columnspan=2, pady=10)

window.mainloop()

3)

import os

# Получаем домашний каталог текущего пользователя home\_directory = os.path.expanduser('~')

# Путь к целевому каталогу target\_directory =

os.path.join('/home/student/Документы/uc27klipan/pz', 'pz 11')

# Проверяем существование каталога

if os.path.exists(target\_directory) and os.path.isdir(target\_directory):

# Получаем список всех файлов в целевом каталоге files = [f for f in os.listdir(target\_directory) if

os.path.isfile(os.path.join(target\_directory, f))]

# Выводим список файлов for file in files:

print(file)

else:

print(f"The directory '{target\_directory}' does not exist.")

3.2)

import os import shutil import random

# Определение путей

project\_root = os.path.abspath(os.path.dirname( file )) test\_folder = os.path.join(project\_root, 'test') test1\_folder = os.path.join(test\_folder, 'test1')

# Создание папок test и test/test1 os.makedirs(test1\_folder, exist\_ok=True)

home\_directory = os.path.expanduser('~')

# Определение путей к каталогам ПЗ6 и ПЗ7 pz6\_directory =

os.path.join('/home/student/Документы/uc27klipan/pz', 'pz 6') pz7\_directory = os.path.join('/home/student/Документы/uc27klipan/pz', 'pz 7')

# Проверка существования каталогов ПЗ6 и ПЗ7 if not os.path.exists(pz6\_directory):

print(f"The directory '{pz6\_directory}' does not exist.") exit(1)

if not os.path.exists(pz7\_directory):

print(f"The directory '{pz7\_directory}' does not exist.") exit(1)

# Получение списка всех файлов в каталогах ПЗ6 и ПЗ7 pz6\_files = []

for root, dirs, files in os.walk(pz6\_directory): for f in files:

pz6\_files.append(os.path.join(root, f))

pz7\_files = [os.path.join(pz7\_directory, f) for f in os.listdir(pz7\_directory) if os.path.isfile(os.path.join(pz7\_directory, f))]

# Проверка наличия достаточного количества файлов для перемещения if len(pz6\_files) < 2:

print(f"Not enough files in '{pz6\_directory}' to move.") exit(1)

if len(pz7\_files) < 1:

print(f"No files found in '{pz7\_directory}' to move.") exit(1)

# Выбор двух случайных файлов из ПЗ6 random\_pz6\_files = random.sample(pz6\_files, 2)

# Выбор одного случайного файла из ПЗ7 random\_pz7\_file = random.choice(pz7\_files)

# Перемещение выбранных файлов из ПЗ6 в папку test for file\_path in random\_pz6\_files:

shutil.move(file\_path, test\_folder)

# Перемещение и переименование выбранного файла из ПЗ7 в папку test/test1

shutil.move(random\_pz7\_file, os.path.join(test1\_folder))

# Получение списка файлов в папке test

test\_files = [f for f in os.listdir(test\_folder) if os.path.isfile(os.path.join(test\_folder, f))]

# Вывод информации о размере файлов в папке test print("File sizes in the 'test' folder:")

for file\_name in test\_files:

file\_path = os.path.join(test\_folder, file\_name) file\_size = os.path.getsize(file\_path) print(f"{file\_name}: {file\_size} bytes")

3.3)

import os

# Путь к корневой папке, где находятся pz1 до pz17 home\_directory = os.path.expanduser('~')

# Путь к папке PZ11 pz11\_directory =

os.path.join('/home/student/Документы/uc27klipan/pz', 'pz 11')

# Проверка существования папки PZ11 if not os.path.exists(pz11\_directory):

print(f"The directory '{pz11\_directory}' does not exist.") exit(1)

# Получение списка файлов в папке PZ11 files\_in\_pz11 = os.listdir(pz11\_directory)

py\_files = [f for f in os.listdir(pz11\_directory) if f.endswith('.py')]

# Инициализация переменной для хранения имени файла с самым коротким именем

shortest\_filename = None shortest\_length = None

# Поиск файла с самым коротким именем for filename in py\_files:

file\_path = os.path.join(pz11\_directory, filename) file\_length = len(filename) # Длина имени файла

if shortest\_length is None or file\_length < shortest\_length: shortest\_filename = filename

shortest\_length = file\_length

# Вывод имени файла с самым коротким именем, используя os.path.basename()

if shortest\_filename:

shortest\_basename = os.path.basename(shortest\_filename) print("Shortest filename in PZ11:", shortest\_basename)

else:

print("No .py files found in PZ11 directory.")

3.4)

import os import random

import subprocess import sys

# Путь к корневой папке, где находятся pz1 до pz17 home\_directory = os.path.expanduser('~')

# Путь к папке pz на рабочем столе в OneDrive home\_directory = os.path.join('/home/student/Документы/uc27klipan/pz')

# Список всех возможных папок pz\* от pz1 до pz17 pz\_folders = [f'pz {i}' for i in range(1, 18)]

# Выбор случайной папки из списка pz\* random\_pz\_folder = random.choice(pz\_folders)

# Путь к случайной папке pz\*

random\_pz\_directory = os.path.join(home\_directory, random\_pz\_folder)

# Путь к папке report в случайной папке pz\*

report\_directory = os.path.join(random\_pz\_directory, 'report')

# Проверка существования папки report if not os.path.exists(report\_directory):

print(f"The directory '{report\_directory}' does not exist.") exit(1)

# Поиск PDF-файла в папке report

pdf\_files = [os.path.join(report\_directory, f) for f in os.listdir(report\_directory) if f.endswith('.pdf')]

# Выбор случайного PDF-файла из найденных if pdf\_files:

random\_pdf\_file = random.choice(pdf\_files) try:

if os.name == 'posix': # macOS

if sys.platform == 'darwin': subprocess.call(('open', random\_pdf\_file))

# Linux else:

subprocess.call(('xdg-open', random\_pdf\_file)) elif os.name == 'nt':

os.startfile(random\_pdf\_file) except Exception as e:

print(f"Error opening file '{random\_pdf\_file}': {e}")

else:

print(f"No PDF files found in the 'report' directory of '{random\_pz\_folder}'.")

3.5)

import os

# Определение пути к файлу test.txt

file\_path = os.path.join('test', 'test1', 'test.txt')

# Проверка существования файла if os.path.exists(file\_path):

try:

os.remove(file\_path)

print(f"File '{file\_path}' successfully deleted.") except OSError as e:

print(f"Error deleting file '{file\_path}': {e}")

else:

print(f"File '{file\_path}' does not exist.")

Протокол работы программы:

1. /usr/bin/python3.9 /home/student/Документы/IS-27/Ovcharenko/pz 17/17.py Process finished with exit code 0
2. /usr/bin/python3.9 /home/student/Документы/ IS-27/Ovcharenko /pz 17/17.2.py Process finished with exit code 0
3. /usr/bin/python3.9 /home/student/Документы/ IS-27/Ovcharenko /pz 17/17.3.py 11.1.py

Text.txt LongestLine.txt data1.txt data2.txt 11.2.py

Process finished with exit code 0

* 1. /usr/bin/python3.9 /home/student/Документы/ IS-27/Ovcharenko /pz 17/17.3.2.py File sizes in the 'test' folder:

Студент группы ИС-27 Овчаренко Евгений.pdf: 96831 bytes pz 6.2.py: 769 bytes

Process finished with exit code 0

* 1. /usr/bin/python3.9 /home/student/Документы/ IS-27/Ovcharenko /pz 17/17.3.3.py Shortest filename in PZ11: 11.1.py

Process finished with exit code 0

* 1. /usr/bin/python3.9 /home/student/Документы/ IS-27/Ovcharenko /pz 17/17.3.4.py Process finished with exit code 0
  2. /usr/bin/python3.9 /home/student/Документы/ IS-27/Ovcharenko /pz 17/17.3.5.py File 'test/test1/test.txt' successfully deleted.

Process finished with exit code 0

Вывод:

Мы закрепили усвоенные знания, понятия, алгоритмы, основные принципы составления программ, мы приобрели навыки составление программ с использованием GUI Tkinter в IDE PyCharm Community, изучили возможности модуля OS.