

Практическое занятие № 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 составления

программ с ООП в IDE PyCharm Community

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.

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 `startfile()`.

☐ удалить файл test.txt

Тип алгоритма: разветвлённый

Текст программы 1

```

# 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).

import tkinter as tk
from tkinter import ttk
from PIL import Image, ImageTk

root = tk.Tk()
root.title("All Fields Form")

# Title
title_label = tk.Label(root, text="ALL FIELDS FORM", font=("Helvetica", 16,
"bold"), fg="blue", anchor="w")
title_label.grid(row=0, column=0, columnspan=2, padx=10, pady=10, sticky="w")

# Textfield
tk.Label(root, text="Textfield", font=("Helvetica", 12)).grid(row=1,
column=0, padx=10, pady=5, sticky="w")
textfield = tk.Entry(root, width=40)
textfield.grid(row=1, column=1, padx=10, pady=5, sticky="w")

# Textarea
tk.Label(root, text="Textarea", font=("Helvetica", 12)).grid(row=2, column=0,
padx=10, pady=5, sticky="w")
textarea = tk.Text(root, height=5, width=30)
textarea.grid(row=2, column=1, padx=10, pady=5)

# Email Address
tk.Label(root, text="Email Address", font=("Helvetica", 12)).grid(row=3,
column=0, padx=10, pady=5, sticky="w")
email_entry = tk.Entry(root, width=40)
email_entry.grid(row=3, column=1, padx=10, pady=5)

# Dropdown
tk.Label(root, text="Dropdown", font=("Helvetica", 12)).grid(row=4, column=0,
padx=10, pady=5, sticky="w")
options = ["Option 1", "Option 2", "Option 3"]
dropdown = ttk.Combobox(root, values=options)
dropdown.grid(row=4, column=1, padx=10, pady=5)
dropdown.current(0)

# Radio Buttons
tk.Label(root, text="Radio Button", font=("Helvetica", 12)).grid(row=5,
column=0, padx=10, pady=5, sticky="w")
radio_var = tk.StringVar()
tk.Radiobutton(root, text="Option 1", variable=radio_var, value="Option
1").grid(row=5, column=1, padx=10, pady=5, sticky="w")
tk.Radiobutton(root, text="Option 2", variable=radio_var, value="Option
2").grid(row=6, column=1, padx=10, pady=5, sticky="w")

# Checkboxes
tk.Label(root, text="Checkbox", font=("Helvetica", 12)).grid(row=7, column=0,
padx=10, pady=5, sticky="w")
chk_var1 = tk.BooleanVar()
chk_var2 = tk.BooleanVar()
chk_var3 = tk.BooleanVar()
tk.Checkbutton(root, text="Option 1", variable=chk_var1).grid(row=7,
column=1, padx=10, pady=5, sticky="w")
tk.Checkbutton(root, text="Option 2", variable=chk_var2).grid(row=8,
column=1, padx=10, pady=5, sticky="w")

```

```

tk.Checkbutton(root, text="Option 3", variable=chk_var3).grid(row=9,
column=1, padx=10, pady=5, sticky="w")

# Password field
tk.Label(root, text="Password", font=("Helvetica", 12)).grid(row=10,
column=0, padx=10, pady=5, sticky="w")
password_entry = tk.Entry(root, show="*", bg="yellow", width=40)
password_entry.grid(row=10, column=1, padx=10, pady=5)

# Number field
tk.Label(root, text="Number Field", font=("Helvetica", 12)).grid(row=11,
column=0, padx=10, pady=5, sticky="w")
number_entry = tk.Entry(root, width=40)
number_entry.grid(row=11, column=1, padx=10, pady=5)

tk.Label(root, text="Mathematical Captcha", font=("Helvetica",
12)).grid(row=12, column=0, padx=10, pady=5, sticky="w")
tk.Label(root, text="6 + 8 =", font=("Helvetica", 8, "bold")).grid(row=12,
column=1, padx=10, pady=5, sticky="w")

def on_entry_click():
    if captcha_entry.get() == 'Enter sum':
        captcha_entry.delete(0, "end") # Удалить все содержимое
        captcha_entry.insert(0, '') # Убедиться, что поле пустое
        captcha_entry.config(fg='black')

def on_focusout():
    if captcha_entry.get() == '':
        captcha_entry.insert(0, 'Enter sum')
        captcha_entry.config(fg='grey')

captcha_entry = tk.Entry(root, fg='grey', width=10)
captcha_entry.grid(row=12, column=1, padx=50, pady=5, sticky="w")
captcha_entry.insert(0, 'Enter sum')
captcha_entry.bind('<FocusIn>', on_entry_click)
captcha_entry.bind('<FocusOut>', on_focusout)

# Google Captcha
tk.Label(root, text="Google Captcha", font=("Helvetica", 12)).grid(row=13,
column=0, padx=10, pady=5, sticky="w")

# Placeholder для отображения капчи как на картинке
captcha_frame = tk.Frame(root, bd=1, relief="solid")
captcha_frame.grid(row=13, column=1, padx=10, pady=5, sticky="w")

# Checkbox для подтверждения
tk.Checkbutton(captcha_frame, text="I'm not a robot", font=("Helvetica",
12)).grid(row=0, column=0, padx=10, pady=10, sticky="w")


# Загружаемое изображение (например, логотип капчи)
captcha_image_path = "captcha_image.png"
img = Image.open(captcha_image_path)
img = img.resize((50, 50)) # Изменение размера изображения
img = ImageTk.PhotoImage(img)
image_label = tk.Label(captcha_frame, image=img)
image_label.grid(row=0, column=1, padx=10, pady=10, sticky="e")
image_label.image = img # Необходимо сохранить ссылку на изображение

# Submit button
submit_button = tk.Button(root, text="Submit", font=("Helvetica", 12),
fg="white", bg="dodgerblue")
submit_button.grid(row=14, column=1, padx=10, pady=20, sticky="w")

```

```
root.mainloop()
```

Протокол работы программы 1

 All Fields Form

ALL FIELDS FORM

Textfield

Textarea

Email Address

Dropdown

Option 1

Radio Button

☒ Option 1

☒ Option 2

Checkbox

☐ Option 1

☐ Option 2

☐ Option 3


Password

Number Field

Mathematical Captcha

6 + 8 =

Google Captcha

☐ I'm not a robot 

Submit

Текст программы 2

```

# Разработать программу с применением пакета tk, взяв в качестве условия одну
# любую задачу из ПЗ №№ 2 - 9
# я взял код с PZ_4_1
# -----
# # Дано вещественное число A и целое число N (>0). Используя один цикл,
# вывести все
# # целые степени числа A от 1 до N
#
# A = input('Введите число A (можно не целое): ')
# N = input('Введите число N (больше нуля): ')
# B = 1
#
# while type(A) != float or type(N) != int or B <= N: # [A != плавает], [N
# != целое], [1 <= N]
#     try:
#         A = float(A)
#         N = int(N)
#         if B <= N: # если B меньше или равно N
#             print('\n', f'{A} возводим в степень {B} и получаем = {int(A **
# B)})')
#             B = B + 1
#     except ValueError:
#         print('Вы ввели неправильно одно из чисел, исправьте!')
#         A = input('Введите число A (можно не целое): ')
#         N = input('Введите число (больше нуля): ')
#
import tkinter as tk
from tkinter import messagebox

def calculate_powers():
    try:
        A = float(entry_A.get())
        N = int(entry_N.get())
        if N <= 0:
            raise ValueError("N должно быть больше нуля.")

        results.delete("1.0", tk.END) # Очистить предыдущие результаты
        for B in range(1, N + 1):
            result = A ** B
            results.insert(tk.END, f'{A} возведенное в степень {B} =
{result}\n')
        except ValueError as e:
            messagebox.showerror("Ошибка", str(e))

root = tk.Tk()
root.title("Возведение в степень")

# Ввод числа A
tk.Label(root, text="Введите число A (можно не целое):").grid(row=0,
column=0, padx=10, pady=5, sticky="w")
entry_A = tk.Entry(root)
entry_A.grid(row=0, column=1, padx=10, pady=5, sticky="w")

# Ввод числа N
tk.Label(root, text="Введите число N (больше нуля):").grid(row=1, column=0,
padx=10, pady=5, sticky="w")
entry_N = tk.Entry(root)
entry_N.grid(row=1, column=1, padx=10, pady=5, sticky="w")

# Кнопка для выполнения вычисления
calculate_button = tk.Button(root, text="Вычислить",

```

```

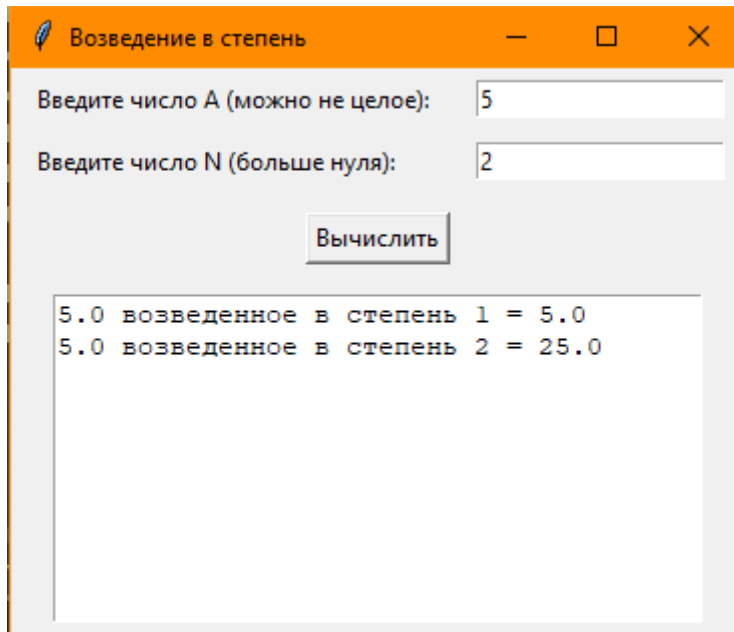
command=calculate_powers)
calculate_button.grid(row=2, column=0, columnspan=2, padx=10, pady=10)

# Окно для отображения результатов
results = tk.Text(root, height=10, width=40, wrap="word")
results.grid(row=3, column=0, columnspan=2, padx=10, pady=5)

root.mainloop()

```

Протокол работы программы 2



Текст программы 3

```

import os
import shutil
import tkinter as tk
from tkinter import messagebox, scrolledtext

def perform_operations():
    results.delete(1.0, tk.END) # Очистить предыдущие результаты

    try:
        # Перейти в каталог PZ11 и вывести список всех файлов
        os.chdir('C:\\Users\\User861\\Desktop\\PZ\\PZ_11')
        all_files = [f for f in os.listdir() if
os.path.isfile(os.path.join(os.getcwd(), f))]
        results.insert(tk.END, "Список всех файлов в каталоге PZ11:\n")
        results.insert(tk.END, f"{all_files}\n\n")

        # Перейти в корень проекта
        os.chdir('C:\\Users\\User861\\Desktop\\PZ')

        # Создать папку с именем test и вложенную папку test1
        if not os.path.exists('test'):
            os.mkdir('test')
        if not os.path.exists(os.path.join('test', 'test1')):
            os.mkdir(os.path.join('test', 'test1'))

        # Переместить два файла из PZ6 в папку test
        shutil.move('PZ_6/PZ_6_1.py', 'test/PZ_6_1.py')
        shutil.move('PZ_6/PZ_6_2.py', 'test/PZ_6_2.py')

```

```

        # Переместить один файл из ПЗ7 в папку test1 и переименовать его в
test.txt
        shutil.move('PZ_7/PZ_7_1.py', 'test/test1/test.txt')

        # Вывести информацию о размере файлов в папке test
        results.insert(tk.END, "Информация о размере файлов в папке test:\n")
        for filename in os.listdir('test'):
            filepath = os.path.join('test', filename)
            if os.path.isfile(filepath):
                results.insert(tk.END, f"{filename}:
{os.path.getsize(filepath)} bytes\n")

        # Перейти в папку с PZ11 и найти файл с самым коротким именем
        os.chdir('C:\\Users\\User861\\Desktop\\PZ\\PZ_11')
        shortest_file = min((f for f in os.listdir() if
os.path.isfile(os.path.join(os.getcwd(), f))),
                            key=lambda f: len(os.path.basename(f)))
        results.insert(tk.END, "\nФайл с самым коротким именем в PZ11:\n")
        results.insert(tk.END, f"{shortest_file}\n\n")

        # Перейти в любую папку где есть отчет в формате .pdf и «запустить»
файл
        pdf_file_path = 'C:\\Users\\User861\\Desktop\\PZ\\reports\\report
PZ_3.pdf'
        if os.path.exists(pdf_file_path):
            os.startfile(pdf_file_path)

        # Удаление файла test.txt из папки test1
        os.remove('C:\\Users\\User861\\Desktop\\PZ\\test\\test1\\test.txt')

    except Exception as e:
        messagebox.showerror("Ошибка", str(e))

root = tk.Tk()
root.title("Выполнение операций")

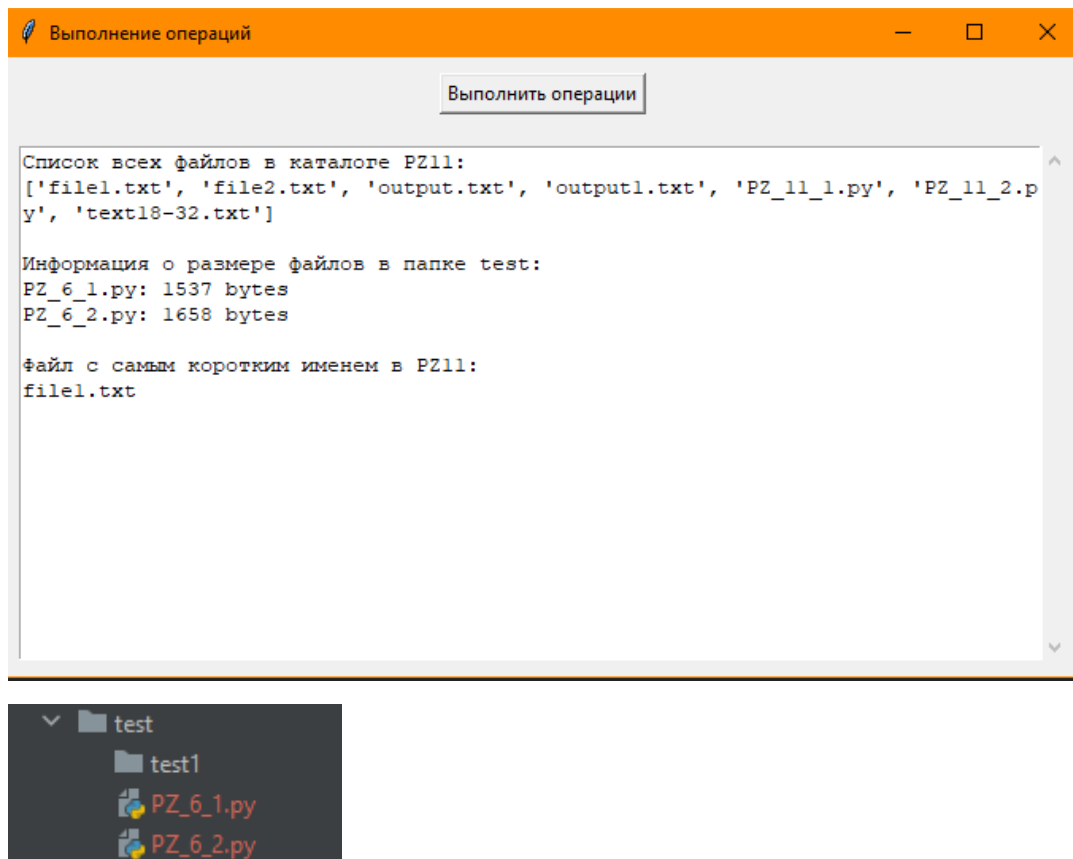
# Кнопка для запуска выполнения операций
perform_button = tk.Button(root, text="Выполнить операции",
command=perform_operations)
perform_button.pack(pady=10)

# Окно для отображения результатов
results = scrolledtext.ScrolledText(root, width=80, height=20)
results.pack(padx=10, pady=10)

root.mainloop()

```

Протокол работы программы 3



Вывод

закрепил усвоенные знания, понятия, алгоритмы,
основные принципы составления программ, приобрел навыки составление
программ с использованием GUI Tkinter в IDE PyCharm Community, изучил
возможности модуля OS.