

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 составления программ с использованием GUI Tkinter в IDE PyCharm Community, изучить возможности модуля OS.

PZ_17_1:

Постановка задачи:

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

Тип алгоритма: линейный

Текст программы:

```
"""
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
"""
import tkinter as tk
from tkinter import ttk

class PlaceholderEntry(ttk.Entry):
    def __init__(self, master=None, placeholder="PLACEHOLDER", color='grey', **kwargs):
        super().__init__(master, **kwargs)
        self.placeholder = placeholder
        self.placeholder_color = color
        self.default_fg_color = self['foreground']
        self._is_placeholder = True

        self.bind("<FocusIn>", self._clear_placeholder)
        self.bind("<FocusOut>", self._add_placeholder)

        self._add_placeholder()

    def _clear_placeholder(self, event=None):
        if self._is_placeholder:
            self.delete(0, tk.END)
            self['foreground'] = self.default_fg_color
            self._is_placeholder = False

    def _add_placeholder(self, event=None):
        if not self.get():
            self.insert(0, self.placeholder)
            self['foreground'] = self.placeholder_color
            self._is_placeholder = True
```

```

def sign_up():
    print("Sign Up button clicked")

root = tk.Tk()
root.title("Contact Us")
root.geometry("230x500")
root.resizable(False, False)

# Изменение цвета фона всего окна
root.configure(bg='lightgray')
style = ttk.Style()

style.configure('Custom.TFrame', background='lightgray')

frame = ttk.Frame(root, padding="10 10 10 10", style='Custom.TFrame')
frame.grid(column=0, row=1, sticky=(tk.W, tk.E, tk.N, tk.S))

title_frame = tk.Frame(root, bg='darkgray')
title_frame.grid(column=0, row=0, sticky=(tk.W, tk.E))

title_label = tk.Label(title_frame, text="Contact Us", font=('Helvetica', 18, 'bold'), bg='darkgray',
fg='black')
title_label.pack(padx=50, pady=15, fill=tk.X)

style.configure("Custom.TLabel", background='lightgray')

ttk.Label(frame, text="First Name", style="Custom.TLabel").grid(column=1, row=1,
sticky=tk.W, pady=(2, 0))
first_name = PlaceholderEntry(frame, placeholder="First Name")
first_name.grid(column=1, row=2, sticky=(tk.W, tk.E), pady=(0, 10))

ttk.Label(frame, text="Last Name", style="Custom.TLabel").grid(column=1, row=3,
sticky=tk.W, pady=(2, 0))
last_name = PlaceholderEntry(frame, placeholder="Smith")
last_name.grid(column=1, row=4, sticky=(tk.W, tk.E), pady=(0, 10))

ttk.Label(frame, text="Email", style="Custom.TLabel").grid(column=1, row=5, sticky=tk.W,
pady=(2, 0))
email = PlaceholderEntry(frame, placeholder="Email address")
email.grid(column=1, row=6, sticky=(tk.W, tk.E), pady=(0, 10))

ttk.Label(frame, text="Website", style="Custom.TLabel").grid(column=1, row=7, sticky=tk.W,
pady=(2, 0))
website = PlaceholderEntry(frame, placeholder="www.example.com")
website.grid(column=1, row=8, sticky=(tk.W, tk.E), pady=(0, 10))

ttk.Label(frame, text="Password", style="Custom.TLabel").grid(column=1, row=9, sticky=tk.W,
pady=(2, 0))
password = PlaceholderEntry(frame, placeholder="8-10 characters", show="*")

```

```
password.grid(column=1, row=10, sticky=(tk.W, tk.E), pady=(0, 10))

ttk.Label(frame, text="Password Confirmation", style="Custom.TLabel").grid(column=1,
row=11, sticky=tk.W, pady=(2, 0))
password_confirmation = PlaceholderEntry(frame, placeholder="Type your password again",
show="*")
password_confirmation.grid(column=1, row=12, sticky=(tk.W, tk.E), pady=(0, 10))

sign_up_button = ttk.Button(frame, text="Sign Up", command=sign_up, width=9)
sign_up_button.grid(column=1, row=13, sticky=tk.W, pady=(10, 0))

for child in frame.winfo_children():
    child.grid_configure(padx=5)

root.mainloop()
```

Протокол работы программы:

PZ_17_2:

Постановка задачи:

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

Тип алгоритма: циклический

Текст программы:

```
'''
Разработать программу с применением пакета tk, взяв в качестве условия одну
любую задачу из ПЗ №№ 2 – 9.
Я выбрал 5.2 Описать функцию SortInc3(A, B, C), меняющую содержимое переменных A,
B, C
таким образом, чтобы их значения оказались упорядоченными по возрастанию (A,
```

B, C - вещественные параметры, являющиеся одновременно входными и выходными). С помощью этой функции упорядочить по возрастанию два данных набора из трех чисел: (Ai, Bi, Ci) и (A2, B2, C2).

```
"""
import tkinter as tk
from tkinter import messagebox

def sort_inc3(A: float, B: float, C: float) -> tuple:
    arr = [A, B, C]
    sorted_arr = []
    num = float('infinity')
    for i in reversed(range(3)):
        for y in range(i+1):
            if arr[y] < num:
                index = y
                num = arr[y]
        sorted_arr.append(num)
        arr.pop(index)
        num = float('infinity')
    return sorted_arr[0], sorted_arr[1], sorted_arr[2]

def sort_numbers():
    try:
        A1 = float(entry_A1.get())
        B1 = float(entry_B1.get())
        C1 = float(entry_C1.get())
        A2 = float(entry_A2.get())
        B2 = float(entry_B2.get())
        C2 = float(entry_C2.get())

        sorted1 = sort_inc3(A1, B1, C1)
        sorted2 = sort_inc3(A2, B2, C2)

        result1_var.set(f"Sorted Set 1: {sorted1}")
        result2_var.set(f"Sorted Set 2: {sorted2}")
    except ValueError:
        messagebox.showerror("Input Error", "Введите все числа в правильном виде.")

app = tk.Tk()
app.title("pz 5_2 Сортировка")

tk.Label(app, text="Set 1:").grid(row=0, column=0, padx=5, pady=5)
tk.Label(app, text="A1:").grid(row=1, column=0, padx=5, pady=5)
entry_A1 = tk.Entry(app)
entry_A1.grid(row=1, column=1, padx=5, pady=5)

tk.Label(app, text="B1:").grid(row=2, column=0, padx=5, pady=5)
entry_B1 = tk.Entry(app)
entry_B1.grid(row=2, column=1, padx=5, pady=5)

tk.Label(app, text="C1:").grid(row=3, column=0, padx=5, pady=5)
entry_C1 = tk.Entry(app)
```

```

entry_C1.grid(row=3, column=1, padx=5, pady=5)

tk.Label(app, text="Set 2:").grid(row=0, column=2, padx=5, pady=5)
tk.Label(app, text="A2:").grid(row=1, column=2, padx=5, pady=5)
entry_A2 = tk.Entry(app)
entry_A2.grid(row=1, column=3, padx=5, pady=5)

tk.Label(app, text="B2:").grid(row=2, column=2, padx=5, pady=5)
entry_B2 = tk.Entry(app)
entry_B2.grid(row=2, column=3, padx=5, pady=5)

tk.Label(app, text="C2:").grid(row=3, column=2, padx=5, pady=5)
entry_C2 = tk.Entry(app)
entry_C2.grid(row=3, column=3, padx=5, pady=5)

sort_button = tk.Button(app, text="Sort", command=sort_numbers)
sort_button.grid(row=4, column=0, columnspan=4, padx=5, pady=5)

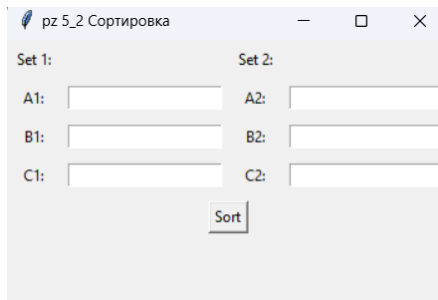
result1_var = tk.StringVar()
result2_var = tk.StringVar()

tk.Label(app, textvariable=result1_var).grid(row=5, column=0, columnspan=4, padx=5, pady=5)
tk.Label(app, textvariable=result2_var).grid(row=6, column=0, columnspan=4, padx=5, pady=5)

app.mainloop()

```

Протокол работы программы:



PZ_17_3:

Постановка задачи:

/ 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:

```
import os
import random

# 1
os.chdir('../PZ_11')

files = [f for f in os.listdir() if os.path.isfile(f)]
print("Список файлов в каталоге PZ11:", files)

# # 2
os.chdir('..')

os.makedirs('test/test1', exist_ok=True)

os.rename('PZ_6/file1.txt', 'test/file1.txt')
os.rename('PZ_6/file2.txt', 'test/file2.txt')

os.rename('PZ_7/file3.txt', 'test/test1/test.txt')

test_files = [f for f in os.listdir('test') if os.path.isfile(os.path.join('test', f))]
for file in test_files:
    size = os.path.getsize(os.path.join('test', file))
    print(f"Размер файла {file}: {size} байт")
# 3
os.chdir('PZ_11')

shortest_name = min([f for f in os.listdir() if os.path.isfile(f)], key=len)
print("Файл с самым коротким именем:", os.path.basename(shortest_name))
# 4
os.chdir('../reports')

pdf_file = [f for f in os.listdir() if f.endswith('.pdf')]
os.startfile(pdf_file[random.randint(0, len(pdf_file)-1)])

# 5
os.chdir('../test/test1')

os.remove('test.txt')
print("Файл test.txt был удален")
```

Протокол работы программы:

Список файлов в каталоге PZ11: ['PZ_11_1.py', 'PZ_11_2.py', 'result.txt', 'source_data.txt']

Размер файла file1.txt: 0 байт

Размер файла file2.txt: 0 байт

Файл с самым коротким именем: PZ_11_1.py

Файл test.txt был удален

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