

Практическая работа №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.

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).

<https://files.codegrape.com/62259/screenshot2.png>

Тип алгоритма: Линейный

Текст программы:

```
from tkinter import *

def submit():
    print("Submit Button!")

def choose_files():
    print("File Button!")

root = Tk()
root.title("Contact Form")
root.geometry("400x650+300+200")
root.resizable(False, False)

title_label = Label(root, text="Contact Form", font="Arial 16", anchor=W)
title_label.pack(fill='x', padx=10, pady=5)

line = Canvas(root, height=2, bd=0, highlightthickness=0, relief='ridge')
line.create_line(0, 1, 400, 1, fill="grey")
line.pack(fill='x', padx=10)

name_frame = Frame(root)
name_frame.pack(fill='x', padx=10, pady=5)
name_label_text = Label(name_frame, text="Name ", font="Arial 12", fg="black")
name_label_text.pack(side=LEFT)
name_label_star = Label(name_frame, text="*", font="Arial 12", fg="red")
name_label_star.pack(side=LEFT)
name_entry = Entry(root, font="Arial 12")
name_entry.pack(fill='x', padx=10, pady=5)

email_frame = Frame(root)
email_frame.pack(fill='x', padx=10, pady=5)
email_label_text = Label(email_frame, text="Email ", font="Arial 12", fg="black")
email_label_text.pack(side=LEFT)
email_label_star = Label(email_frame, text="*", font="Arial 12", fg="red")
```

```

email_label_star.pack(side=LEFT)
email_entry = Entry(root, font="Arial 12")
email_entry.pack(fill='x', padx=10, pady=5)

phone_frame = Frame(root)
phone_frame.pack(fill='x', padx=10, pady=5)
phone_label_text = Label(phone_frame, text="Phone Number ", font="Arial 12",
fg="black")
phone_label_text.pack(side=LEFT)
phone_label_star = Label(phone_frame, text="*", font="Arial 12", fg="red")
phone_label_star.pack(side=LEFT)
phone_entry = Entry(root, font="Arial 12")
phone_entry.pack(fill='x', padx=10, pady=5)

subject_frame = Frame(root)
subject_frame.pack(fill='x', padx=10, pady=5)
subject_label_text = Label(subject_frame, text="Subject ", font="Arial 12",
fg="black")
subject_label_text.pack(side=LEFT)
subject_label_star = Label(subject_frame, text="*", font="Arial 12", fg="red")
subject_label_star.pack(side=LEFT)
subject_entry = Entry(root, font="Arial 12")
subject_entry.pack(fill='x', padx=10, pady=5)

message_frame = Frame(root)
message_frame.pack(fill='x', padx=10, pady=5)
message_label_text = Label(message_frame, text="Leave us a few words ",
font="Arial 12", fg="black")
message_label_text.pack(side=LEFT)
message_label_star = Label(message_frame, text="*", font="Arial 12", fg="red")
message_label_star.pack(side=LEFT)
message_text = Text(root, height=5, font="Arial 12", wrap=WORD)
message_text.pack(fill='x', padx=10, pady=5)

file_frame = Frame(root)
file_frame.pack(anchor="w", padx=10, pady=5)
file_label = Label(file_frame, text="File Attachments", font="Arial 12", anchor=W)
file_label.pack(anchor="w", side=LEFT)

file_frame_inner = Frame(root)
file_frame_inner.pack(anchor="w", padx=10, pady=5)
file_button = Button(file_frame_inner, text="Choose Files", command=choose_files,
font="Arial 12")
file_button.pack(side=LEFT)
file_text = Label(file_frame_inner, text="No file chosen", font="Arial 12",
fg="gray")
file_text.pack(side=LEFT, padx=10)

captcha_check = Checkbutton(root, text="I'm not a robot", font="Arial 12")
captcha_check.pack(anchor="w", padx=10, pady=5)

submit_button = Button(root, text="Submit", command=submit, font="Arial 12",
bg="Blue", fg="white")
submit_button.pack(anchor="w", padx=10, pady=20)

root.mainloop()

```

Протокол работы программы:

Contact Form

Contact Form

Name *

Email *

Phone Number *

Subject *

Leave us a few words *

File Attachments

Choose Files

No file chosen

☐ I'm not a robot

Submit

File Button!
Submit Button!

Process finished with exit code 0

Постановка задачи №2.

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

Тип алгоритма: Линейный

Текст программы:

```
from tkinter import *
from PZ_5.PZ_5_2 import Swap

def swap_values():
    x = x_entry.get()
    y = y_entry.get()
    swapped_x, swapped_y = Swap(x, y)
    result_label.config(text=f"До: {x} | {y}\nПосле: {swapped_x} | {swapped_y}")

root = Tk()
root.title("Swap Values")
root.geometry("400x200")

x_label = Label(root, text="Введите значение для x:")
x_label.grid(row=0, column=0, padx=10, pady=5, sticky=E)
x_entry = Entry(root)
x_entry.grid(row=0, column=1, padx=10, pady=5, sticky=W)

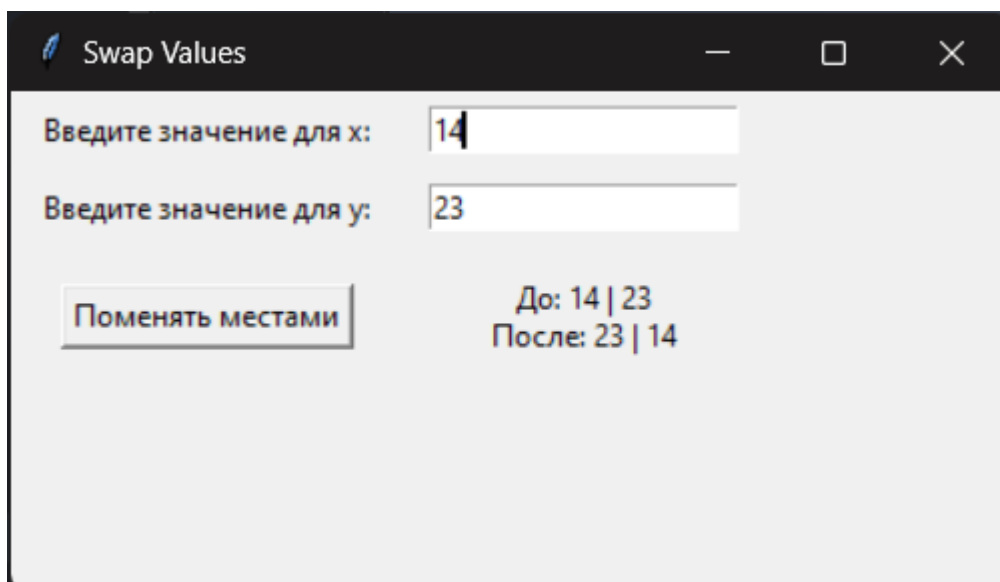
y_label = Label(root, text="Введите значение для y:")
y_label.grid(row=1, column=0, padx=10, pady=5, sticky=E)
y_entry = Entry(root)
y_entry.grid(row=1, column=1, padx=10, pady=5, sticky=W)

swap_button = Button(root, text="Поменять местами", command=swap_values)
swap_button.grid(row=2, column=0, pady=10)

result_label = Label(root, text="")
result_label.grid(row=2, column=1, pady=10)

root.mainloop()
```

Протокол работы программы:



Постановка задачи №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 os.startfile(). | удалить файл test.txt.

Тип алгоритма: Линейный

Текст программы:

```
import os
import random

path2 = "../test"

def os_files_operations(path: str, operation: str, end_path: str = None) -> None:
    try:
        if operation == "make":
            os.makedirs(path)
        elif operation == "move":
            os.replace(path, end_path)
        elif operation == "rename":
            os.rename(path, end_path)
        elif operation == "remove":
            os.remove(path)
    except FileNotFoundError:
        print("\nТакого файла не существует!\nЗапустите программу еще раз, чтобы
создать файл")
        with open(path, "w") as file:
            file.write("bom bom bom")
    except FileExistsError:
        # print("Папка уже существует")
        pass

def get_files_in_folder(path_to_folder: str) -> list[str]:
    return [file for file in os.listdir(path_to_folder) if
os.path.isfile(os.path.join(path_to_folder, file))]

files = get_files_in_folder("../PZ_11")
print(f"Список всех файлов в PZ_11: {files}\n")

os_files_operations("../test/test1", "make")
os_files_operations("../PZ_6/PZ_6(1).py", "move", "../test/PZ_6(1).py")
os_files_operations("../PZ_6/PZ_6(2).py", "move", "../test/PZ_6(2).py")
os_files_operations("../PZ_7/test1.txt", "move", "../test/test1/test1.txt")
os_files_operations("../test/test1/test1.txt", "rename", "../test/test1/test.txt")
print("Информацию о размере файлов в папке test:")
sizes = [print(f"{file} - {os.stat(os.path.join(path2, file)).st_size} байт") for
file in os.listdir(path2) if os.path.isfile(os.path.join(path2, file))]

test = min([f"../PZ_11/{file}" for file in files], key=lambda x: len(x))
```

```
print(f"\nФайл с самым коротким именем в PZ_11:\n{os.path.basename(test)} -  
{os.stat(test).st_size} байт")  
  
pdf_files = [file for file in os.listdir("../reports") if file.endswith(".pdf")]  
os.startfile(f"{os.path.join("../reports", random.choice(pdf_files))}")  
  
os_files_operations("../PZ_7/fake_PZ_7.txt", "remove")
```

Протокол работы программы:

Список всех файлов в PZ_11: ['test.txt', 'test123.txt', 'test12345.txt', '__init__.py']

Информацию о размере файлов в папке test:

PZ_6(1).py - 11 байт

PZ_6(2).py - 11 байт

Файл с самым коротким именем в PZ_11:

test.txt - 4 байт

Process finished with exit code 0

Вывод: В процессе выполнения практического занятия №17 закрепил усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрел навыки составления программ с использованием GUI Tkinter в IDE PyCharm Community, изучил возможности модуля OS.

Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные кода выложены на GitHub.

