

Отчет о практическом занятии

Практическое занятие № 17 Вариант 15

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.

```
# 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).
# Вариант 15 - https://www.webasyst.ru/wa-data/public/updates/img/09/209/7355/7355.970.jpg
import tkinter as tk
from tkinter import ttk

# Создание основного окна
root = tk.Tk()
root.title("Создайте заказ")

root.resizable(False, False) # Блокировка изменения размера окна
root.geometry('550x750') # Установка размеров окна, подберите размеры под
ваши нужды

# Создание заголовка
title_frame = tk.Frame(root, bg='#00838F', bd=10)
title_frame.pack(fill=tk.X)

title_label = tk.Label(title_frame, text="Создайте заказ", font=("Arial", 24),
bg='#00838F', fg='white')
title_label.pack()

# Секция 1: Информация о заказе
section1_frame = tk.Frame(root, bd=10)
section1_frame.pack(fill=tk.X, pady=10)

section1_label = tk.Label(section1_frame, text="1 Информация о заказе",
font=("Arial", 14))
section1_label.grid(row=0, column=0, columnspan=2, pady=10)

order_num_label = tk.Label(section1_frame, text="Номер заказа *")
order_num_label.grid(row=1, column=0, sticky=tk.W, pady=5)
order_num_entry = tk.Entry(section1_frame)
order_num_entry.grid(row=1, column=1, pady=5)

product_name_label = tk.Label(section1_frame, text="Название товара *")
product_name_label.grid(row=2, column=0, sticky=tk.W, pady=5)
product_name_entry = tk.Entry(section1_frame)
product_name_entry.grid(row=2, column=1, pady=5)
```

```

quantity_label = tk.Label(section1_frame, text="Количество *")
quantity_label.grid(row=3, column=0, sticky=tk.W, pady=5)
quantity_entry = tk.Entry(section1_frame)
quantity_entry.grid(row=3, column=1, pady=5)

# Секция 2: Контактная информация
section2_frame = tk.Frame(root, bd=10)
section2_frame.pack(fill=tk.X, pady=10)

section2_label = tk.Label(section2_frame, text="2 Контактная информация",
font=("Arial", 14))
section2_label.grid(row=0, column=0, columnspan=2, pady=10)

name_label = tk.Label(section2_frame, text="Ваше имя")
name_label.grid(row=1, column=0, sticky=tk.W, pady=5)
name_entry = tk.Entry(section2_frame)
name_entry.grid(row=1, column=1, pady=5)

email_label = tk.Label(section2_frame, text="Ваш email *")
email_label.grid(row=2, column=0, sticky=tk.W, pady=5)
email_entry = tk.Entry(section2_frame)
email_entry.grid(row=2, column=1, pady=5)

phone_label = tk.Label(section2_frame, text="Ваш телефон *")
phone_label.grid(row=3, column=0, sticky=tk.W, pady=5)
phone_entry = tk.Entry(section2_frame)
phone_entry.grid(row=3, column=1, pady=5)
phone_hint_label = tk.Label(section2_frame, text="Формат: +7 (999) 999-99-99")
phone_hint_label.grid(row=4, column=1, sticky=tk.W, pady=5)

# Секция 3: Информация о доставке
section3_frame = tk.Frame(root, bd=10)
section3_frame.pack(fill=tk.X, pady=10)

section3_label = tk.Label(section3_frame, text="3 Информация о доставке",
font=("Arial", 14))
section3_label.grid(row=0, column=0, columnspan=2, pady=10)

address_label = tk.Label(section3_frame, text="Адрес *")
address_label.grid(row=1, column=0, sticky=tk.W, pady=5)
address_entry = tk.Text(section3_frame, height=4, width=40)
address_entry.grid(row=1, column=1, pady=5)

delivery_time_label = tk.Label(section3_frame, text="Время доставки")
delivery_time_label.grid(row=2, column=0, sticky=tk.W, pady=5)

time_frame = tk.Frame(section3_frame)
time_frame.grid(row=2, column=1, pady=5)
hour_spinbox = ttk.Spinbox(time_frame, from_=0, to=23, width=2, wrap=True,
state="readonly", justify=tk.CENTER, format="%02.0f")
hour_spinbox.grid(row=0, column=0)
minute_spinbox = ttk.Spinbox(time_frame, from_=0, to=59, width=2, wrap=True,
state="readonly", justify=tk.CENTER, format="%02.0f")
minute_spinbox.grid(row=0, column=1)

# Кнопка "Подтвердить"
button_frame = tk.Frame(root, bd=10)
button_frame.pack(fill=tk.X, pady=10)

```

```
submit_button = tk.Button(button_frame, text="Подтвердить", font=("Arial",  
14))  
submit_button.pack()  
  
root.mainloop()
```

Протокол работы программы:

Создайте заказ

1 Информация о заказе

Номер заказа *

Название товара *

Количество *

2 Контактная информация

Ваше имя

Ваш email *

Ваш телефон *

Формат: +7 (999) 999-99-99

3 Информация о доставке

Адрес *

Время доставки

Подтвердить

Process finished with exit code 0

```
# Задание 2. Разработать программу с применением пакета tk, взяв в качестве  
условия одну любую задачу из ПЗ №№ 2 - 9.  
  
import tkinter as tk  
from tkinter import messagebox  
  
def calculate_digits():  
    try:  
        num = int(entry.get())  
        if 99 < num < 1000:  
            last_digit = num % 10  
            middle_digit = (num // 10) % 10
```

```
        result_last_digit.config(text=f"Последняя цифра: {last_digit}")
        result_middle_digit.config(text=f"Средняя цифра: {middle_digit}")
    else:
        messagebox.showerror("Ошибка ввода", "Введите трёхзначное число!")
except ValueError:
    messagebox.showerror("Ошибка ввода", "Введите допустимое число!")

# Создание основного окна
root = tk.Tk()
root.title("Подсчет цифр")
root.resizable(False, False)
root.geometry('300x200')

# Метка и поле ввода
prompt_label = tk.Label(root, text="Введите трёхзначное число:")
prompt_label.pack(pady=15)

entry = tk.Entry(root)
entry.pack(pady=5)

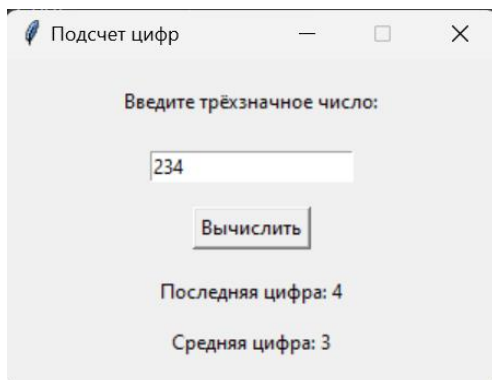
# Кнопка для запуска вычислений
calculate_button = tk.Button(root, text="Вычислить", command=calculate_digits)
calculate_button.pack(pady=10)

# Метки для отображения результата
result_last_digit = tk.Label(root, text="")
result_last_digit.pack(pady=5)

result_middle_digit = tk.Label(root, text="")
result_middle_digit.pack(pady=5)

# Запуск главного цикла событий
root.mainloop()
```

Протокол работы программы:



```

# 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 P36, а в папку test1 -
один файл из P37. Файл из P37 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

# Функция для безопасного перемещения в каталог
def safe_chdir(directory):
    if os.path.exists(directory) and os.path.isdir(directory):
        os.chdir(directory)
    else:
        print(f"Каталог {directory} не существует или не является
директорией.")
        return False
    return True

# Функция для копирования файлов
def copyfile(src, dest):
    with open(src, 'rb') as fsrc:
        with open(dest, 'wb') as fdest:
            while True:
                buf = fsrc.read(1024)
                if not buf:
                    break
                fdest.write(buf)

# Предполагаем, что текущий каталог это не корневой каталог проекта
'Kresichenko'.
os.chdir('..')
current_directory = os.getcwd()

# Перейти в каталог PZ_11 и вывести список всех файлов
if safe_chdir(os.path.join(current_directory, 'PZ_11')):
    files = [f for f in os.listdir() if os.path.isfile(f)]
    print("Файлы в PZ_11:", files)
    os.chdir(current_directory) # Вернуться в корень проекта

# Создать папку test и тестовую структуру внутри нее
test_dir = os.path.join(current_directory, 'test')
if not os.path.exists(test_dir):
    os.makedirs(os.path.join(test_dir, 'test1'))

# Переместить два файла из PZ_6 в папку test
pz6_files = [

```

```

os.path.join(current_directory, 'PZ_6', 'PZ_6_1.py'),
os.path.join(current_directory, 'PZ_6', 'PZ_6_2.py'),
]

for file in pz6_files:
    if os.path.isfile(file):
        copyfile(file, os.path.join(test_dir, os.path.basename(file)))
    else:
        print(f"Файл {file} не существует.")

test1_dir = os.path.join(current_directory, 'test/test1')

# Копировать файл из PZ_7 в папку test
pz7_files = [os.path.join(current_directory, 'PZ_7', 'PZ_7_1.py')]
for file in pz7_files:
    if os.path.isfile(file):
        copyfile(file, os.path.join(test1_dir, 'test.txt'))
    else:
        print(f"Файл {file} не существует.")

# Вывод информации о размере файлов в папке test
test_files_sizes = {f: os.path.getsize(os.path.join(test_dir, f)) for f in
os.listdir(test_dir) if os.path.isfile(os.path.join(test_dir, f))}
print("Размер файлов в test:", test_files_sizes)

# Перейти обратно в PZ_11 и найти файл с самым коротким именем
if safe_chdir(os.path.join(current_directory, 'PZ_11')):
    files = [f for f in os.listdir() if os.path.isfile(f)]
    shortest_name_file = min(files, key=len)
    print("Файл с самым коротким именем:",
os.path.basename(shortest_name_file))
    os.chdir(current_directory) # Вернуться в корень проекта

# Перейти в папку reports и открыть PDF файл
reports_dir = os.path.join(current_directory, 'reports') # Замените на вашу
папку
if safe_chdir(reports_dir):
    pdf_files = [f for f in os.listdir(reports_dir) if f.endswith('.pdf')]
    if pdf_files:
        os.startfile(os.path.join(reports_dir, pdf_files[0])) # Открыть
первый найденный PDF файл
    else:
        print("Нет PDF файлов в папке reports.")
        os.chdir(current_directory) # Вернуться в корень проекта

# Удалить файл test.txt
test_file = os.path.join(test1_dir, 'test.txt')
if os.path.isfile(test_file):
    os.remove(test_file)
    print("Файл test.txt удален.")
else:
    print(f"Файл {test_file} не существует.")

```

Протокол работы программы:

Файлы в PZ_11: ['new_text.txt', 'numbers.txt', 'processed_sequence.txt', 'PZ_11_1.py',

```
'text18-15.txt', '__init__.py']
```

Размер файлов в test: {'PZ_6_1.py': 533, 'PZ_6_2.py': 825}

Файл с самым коротким именем: PZ_11_1.py

Файл test.txt удален.

Process finished with exit code 0

Вывод: в процессе выполнения практического занятия выработал навыки составления программ с использованием GUI Tkinter в IDE PyCharm Community, изучил возможностей модуля OS.

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