

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

Задание 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.

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

root = tk.Tk()
root.title("Создание заказа")
root.geometry("450x520")

top_frame = tk.Frame(root, bg="#3c7b8f", width=600, height=50)
top_frame.pack(side="top", fill="x")

main_name = tk.Label(top_frame, text="Создайте заказ", font=("Arial", 14), bg="#3c7b8f", fg="white")
main_name.pack(pady=10)

section_1 = tk.Label(root, text="1. Информация о заказе", font=("Arial", 12, "bold"), fg="#3c7b8f")
section_1.place(x=5, y=90)

order_num = tk.Label(root, text="Номер заказа*", fg="black")
order_num.place(x=20, y=120)
order_num = tk.Entry(root, width=45)
order_num.place(x=150, y=120)

prod_name = tk.Label(root, text="Название товара", fg="black")
prod_name.place(x=20, y=150)
prod_name = tk.Entry(root, width=45)
prod_name.place(x=150, y=150)

amount = tk.Label(root, text="Количество*", fg="black")
amount.place(x=20, y=180)
amount = tk.Entry(root, width=10)
amount.place(x=150, y=180)

section_2 = tk.Label(root, text="2. Контактная информация", font=("Arial", 12, "bold"), fg="#3c7b8f")
```

```

section_2.place(x=5, y=220)

name = tk.Label(root, text="Ваше имя", fg="black")
name.place(x=20, y=250)
name = tk.Entry(root, width=45)
name.place(x=150, y=250)

email = tk.Label(root, text="Ваш email*", fg="black")
email.place(x=20, y=280)
email = tk.Entry(root, width=45)
email.place(x=150, y=280)

phone = tk.Label(root, text="Ваш телефон*", fg="black")
phone.place(x=20, y=310)
phone = tk.Entry(root, width=18)
phone.insert(0, "+7(")
phone.place(x=150, y=310)

section_3 = tk.Label(root, text="3. Информация о доставке", font=("Arial", 12, "bold"), fg="#3c7b8f")
section_3.place(x=5, y=350)

adres = tk.Label(root, text="Адрес*", fg="black")
adres.place(x=20, y=380)
adres = tk.Text(root, width=34, height=4)
adres.place(x=150, y=380)

time_dost = tk.Label(root, text="Время доставки", fg="black")
time_dost.place(x=20, y=460)

hours = [f"{i:02d}" for i in range(0, 24)]
time_dost_1 = ttk.Combobox(root, values=hours, width=5)
time_dost_1.place(x=150, y=460)
time_dost_1.set("00")

minutes = [f"{i:02d}" for i in range(0, 60, 15)]
time_dost_2 = ttk.Combobox(root, values=minutes, width=5)
time_dost_2.place(x=220, y=460)
time_dost_2.set("00")

root.mainloop()

```

Протокол программы:

Создание заказа

Создайте заказ

1. Информация о заказе

Номер заказа*

Название товара

Количество*

2. Контактная информация

Ваше имя

Ваш email*

Ваш телефон* +7(

3. Информация о доставке

Адрес*

Время доставки 00 00

Задание 2

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

Текст программы:

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

import tkinter as tk

def check_num():
    try:
        a = int(entry_a.get())
        b = int(entry_b.get())
        c = int(entry_c.get())
    except ValueError:
        label_res.config(text="Вы ввели неправильные данные!")
        return

    positive_count = 0
    if a > 0:
        positive_count += 1
    if b > 0:
        positive_count += 1
    if c > 0:
        positive_count += 1

    if positive_count == 1:
        label_res.config(text="Высказывание истинно")
    else:
        label_res.config(text="Высказывание ложно")

root = tk.Tk()
root.title("Проверка чисел")
root.geometry("300x300")

label_a = tk.Label(root, text="Введите число A:", font=("Arial", 12))
label_a.pack(pady=5)

entry_a = tk.Entry(root, font=("Arial", 12), width=10, justify=tk.CENTER)
entry_a.pack(pady=5)

label_b = tk.Label(root, text="Введите число B:", font=("Arial", 12))
label_b.pack(pady=5)

entry_b = tk.Entry(root, font=("Arial", 12), width=10, justify=tk.CENTER)
entry_b.pack(pady=5)

label_c = tk.Label(root, text="Введите число C:", font=("Arial", 12))
label_c.pack(pady=5)

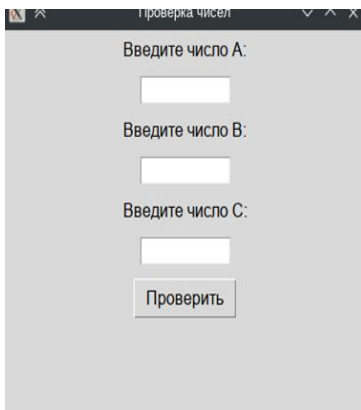
entry_c = tk.Entry(root, font=("Arial", 12), width=10, justify=tk.CENTER)
entry_c.pack(pady=5)

check = tk.Button(root, text="Проверить", command=check_num, font=("Arial", 12), pady=5)
check.pack(pady=5)

label_res = tk.Label(root, text="", font=("Arial", 12))
label_res.pack(pady=5)

root.mainloop()
```

Протокол программы:



Задание 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.

Текст программы:

```
import os
import subprocess

# 1. перейдите в каталог PZ11. Выведите список всех файлов в этом каталоге.
# Имена вложенных подкаталогов выводить не нужно.

os.chdir(r'/home/student/Документы/Kim Yana is-22/PZ_11')
files = [f for f in os.listdir() if os.path.isfile(f)]
print("Список файлов в каталоге PZ_11:")
for file in files:
    print(file)

# 2. 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.

os.chdir(r'/home/student/Документы/Kim Yana is-22')
os.makedirs('test/test1')
os.replace(r'PZ_6/PZ_6_1.py', 'test/PZ_6_1.py')
os.replace(r'PZ_6/PZ_6_2.py', 'test/PZ_6_2.py')
os.replace(r'PZ_7/PZ_7_1.py', 'test/test1/test.txt')

total_size = sum(os.path.getsize(os.path.join('test', f)) for f in os.listdir('test'))
print(f"Размер файлов в папке test: {total_size} байт")

# 3. перейти в папку с PZ11, найти там файл с самым коротким именем, имя вывести в
# консоль. Использовать функцию basename() (os.path.basename()).

os.chdir(r'/home/student/Документы/Kim Yana is-22/PZ_11')
files = os.listdir()
```

```

shortest_filename = None

for file in files:
    if shortest_filename is None or len(file) < len(shortest_filename):
        shortest_filename = file

print('Самое короткое имя имеет файл:', os.path.basename(shortest_filename))

# 4. перейти в любую папку, где есть отчет в формате .pdf и «запустите» файл в
# привязанной к нему программе. Использовать функцию os.startfile().

pdf_file = r'/home/student/Документы/Kim Yana is-22/PZ_13/PZ_13.pdf'
subprocess.call(['xdg-open', pdf_file])

# 5. удалить файл test.txt.

file = r'/home/student/Документы/Kim Yana is-22/test/test1/test.txt'

if os.path.exists(file):
    try:
        os.remove(file)
        print("Файл успешно удален")
    except OSError as e:
        print(f'Ошибка при удалении файла: {e}')
else:
    print("Ошибка: файл не найден")

```

Протокол программы:

Список файлов в каталоге PZ_11:

```

PZ_11_2.py
data.txt
data_1.txt
18-15.txt
PZ_11_1.py
18-15-1.txt

```

Самое короткое имя имеет файл: data.txt

Файл успешно удален

Process finished with exit code 1

Вывод: в процессе выполнения практического занятия выработала навыки составления программ с использованием GUI Tkinter в IDE PyCharm Community, изучение возможностей модуля OS. Выполнены разработка кода, отладка, тестирование программного кода. Готовые программные коды выложены на GitHub.