Практическая №17 Кайханиди Виктор ИС-27

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 составление 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).

Текст программы:

```
import tkinter as tk
from tkinter import ttk



def toggle_gender_button(button, label):
    if button['bg'] == 'white':
        button['bg'] = 'grey'
        label['fg'] = 'white'
    else:
        button['bg'] = 'white'
        label['fg'] = 'white'

def on_phone_entry_click(event):
    if phone_entry.get() == 'Phone':
        phone_entry.delete(0, "end")
        phone_entry.config(fg='black')

def on_phone_focusout(event):
    if phone_entry.get() == '':
        phone_entry.insert(0, 'Phone')
        phone_entry.config(fg='grey')

root = tk.Tk()

root.title('Python Window')
root.configure(bg='grey')
placeholders = ['Enter First Name', 'Enter Last Name', 'Enter Screen Name']
month_names = ['Январь', 'Февраль', 'Март', 'Апрель', 'Май', 'Июнь', 'Июль', 'Август', 'Сентябрь',
'Октябрь', 'Ноябрь', 'Декабрь']
countries = ['USA', 'Canada', 'Mexico', 'UK', 'France', 'Germany', 'Spain', 'Italy', 'Russia', 'China', 'Japan',
'Australia']

for i, placeholder in enumerate(placeholders):
    label = tk.Label(root, text=f'{placeholder}:', bg='grey', fg='white')
    label.grid(row=i, column=0)
    entry = tk.Entry(root)
    entry.insert(0, placeholder)
    entry.bind('<FocusIn>', lambda event, entry=entry, default_text=placeholder: on_entry_click(event,
entry, default_text))
    entry.bind('<FocusOut>', lambda event, entry=entry, default_text=placeholder: on_focusout(event,
```

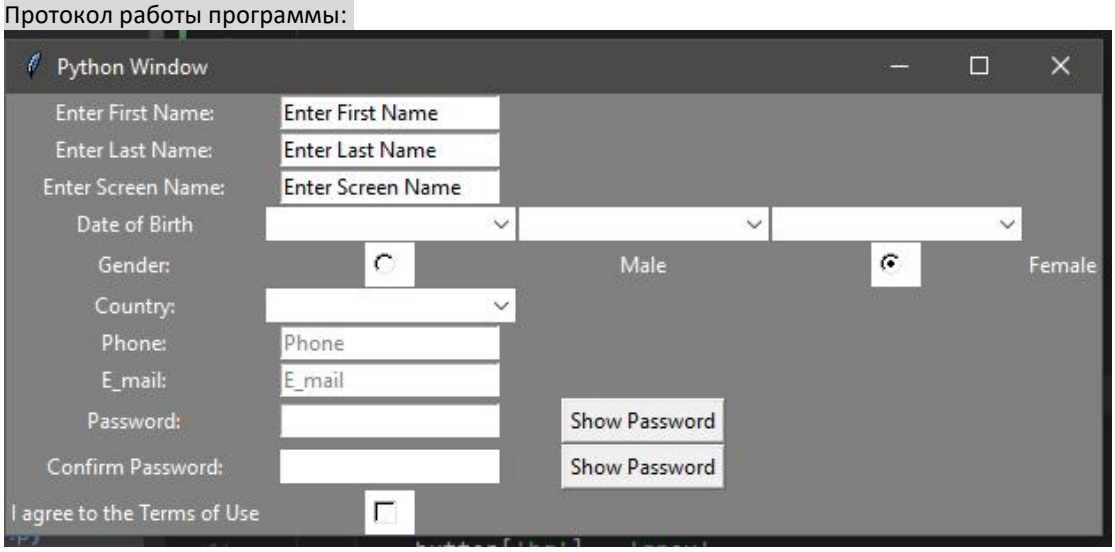
```
entry, default_text))
    entry.grid(row=i, column=1)


dob_label = tk.Label(root, text='Date of Birth', bg='grey', fg='white')
dob_label.grid(row=3, column=0)
day_entry = ttk.Combobox(root, state="readonly", values=list(range(1, 32)))
day_entry.grid(row=3, column=1)
month_entry = ttk.Combobox(root, state="readonly", values=month_names)
month_entry.grid(row=3, column=2)
year_entry = ttk.Combobox(root, state="readonly", values=list(range(1900, 2100)))
year_entry.grid(row=3, column=3)


gender_label = tk.Label(root, text='Gender:', bg='grey', fg='white')
gender_label.grid(row=4, column=0)
male_button = tk.Radiobutton(root, bg='white', value=1, command=lambda:
toggle_gender_button(male_button, male_label))
male_button.grid(row=4, column=1)
male_label = tk.Label(root, text='Male', bg='grey', fg='white')
male_label.grid(row=4, column=2)
female_button = tk.Radiobutton(root, bg='white', value=2, command=lambda:
toggle_gender_button(female_button, female_label))
female_button.grid(row=4, column=3)
female_label = tk.Label(root, text='Female', bg='grey', fg='white')
female_label.grid(row=4, column=4)


country_label = tk.Label(root, text='Country:', bg='grey', fg='white')
country_label.grid(row=5, column=0)
country_entry = ttk.Combobox(root, state="readonly", values=countries)
country_entry.grid(row=5, column=1)


phone_label = tk.Label(root, text='Phone:', bg='grey', fg='white')
phone_label.grid(row=6, column=0)
phone_entry = tk.Entry(root, fg='grey')
phone_entry.insert(0, 'Phone')
phone_entry.bind('<FocusIn>', on_phone_entry_click)
phone_entry.bind('<FocusOut>', on_phone_focusout)
phone_entry.grid(row=6, column=1)




def toggle_password_visibility():
    if password_entry.cget('show') == '*':
        password_entry.config(show='')
    else:
        password_entry.config(show='*')

# Password
password_label = tk.Label(root, text='Password:', bg='grey', fg='white')
password_label.grid(row=8, column=0)

password_entry = tk.Entry(root, show='*', fg='grey')
```

```
password_entry.grid(row=8, column=1)

password_button = tk.Button(root, text='Show Password', command=toggle_password_visibility)
password_button.grid(row=8, column=2)



def toggle_confirm_password():
    if confirm_password_entry.cget('show') == '*':
        confirm_password_entry.config(show='')
    else:
        confirm_password_entry.config(show='*')

# Confirm Password
confirm_password_label = tk.Label(root, text='Confirm Password:', bg='grey', fg='white')
confirm_password_label.grid(row=9, column=0)  # измените номера строк в соответствии с вашим местоположением

confirm_password_entry = ttk.Entry(root, show='*')
confirm_password_entry.grid(row=9, column=1)  # измените номера строк в соответствии с вашим местоположением

confirm_password_button = tk.Button(root, text='Show Password',
command=toggle_confirm_password)
confirm_password_button.grid(row=9, column=2)  # измените номера строк в соответствии с вашим местоположением


def on_email_entry_click(event):
    if E_mail_label_entry.get() == 'E_mail':
        E_mail_label_entry.delete(0, "end")
        E_mail_label_entry.config(fg='black')

def on_email_focusout(event):
    if E_mail_label_entry.get() == '':
        E_mail_label_entry.insert(0, 'E_mail')
        E_mail_label_entry.config(fg='grey')

E_mail_label = tk.Label(root, text='E_mail:', bg='grey', fg='white')
E_mail_label.grid(row=7, column=0)
E_mail_label_entry = tk.Entry(root, fg='grey')
E_mail_label_entry.insert(0, 'E_mail')
E_mail_label_entry.bind('<FocusIn>', on_email_entry_click)
E_mail_label_entry.bind('<FocusOut>', on_email_focusout)
E_mail_label_entry.grid(row=7, column=1)



def toggle_terms_agreement():
    if terms_checkbutton.cget('bg') == 'white':
        terms_checkbutton.config(bg='grey')
    else:
        terms_checkbutton.config(bg='white')


terms_label = tk.Label(root, text='I agree to the Terms of Use', bg='grey', fg='white')
```

```
terms_label.grid(row=10, column=0)

terms_checkbutton = tk.Checkbutton(root, bg='white', command=toggle_terms_agreement)
terms_checkbutton.grid(row=10, column=1)
```

```
def on_entry_click(event, entry, default_text):
    # если в поле ввода содержится текст по умолчанию, удаляем его
    if event.widget.get() == default_text:
        event.widget.delete(0, "end")

def on_focusout(event, entry, default_text):
    # если поле ввода пустое, вставляем текст по умолчанию
    if entry.get() == '':
        entry.insert(0, default_text)

root.mainloop()
```

Протокол работы программы:

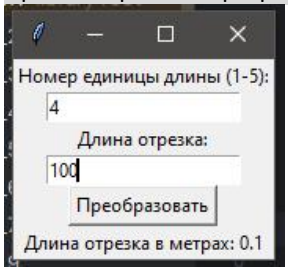


Постановка задачи №2: Разработать программу с применением пакета tk, взяв в качестве условия одну любую задачу из ПЗ №№ 2 – 9.

Текст программы:

```
import tkinter as tk


def convert_length():
    unit = int(entry_unit.get())
    length = float(entry_length.get())

    if unit == 1:
        result = length / 10
    elif unit == 2:
        result = length * 1000
    elif unit == 3:
        result = length
    elif unit == 4:
        result = length / 1000
    elif unit == 5:
        result = length / 100
    else:
        result = -1

    if result != -1:
        label_result.config(text=f"Длина отрезка в метрах: {result}")
    else:
        label_result.config(text="Неверный номер единицы длины")


root = tk.Tk()
root.title("Конвертер длины")

label_unit = tk.Label(root, text="Номер единицы длины (1-5):")
label_unit.pack()
entry_unit = tk.Entry(root)
entry_unit.pack()

label_length = tk.Label(root, text="Длина отрезка:")
label_length.pack()
entry_length = tk.Entry(root)
entry_length.pack()

convert_button = tk.Button(root, text="Преобразовать", command=convert_length)
convert_button.pack()

label_result = tk.Label(root, text="")
label_result.pack()

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 П36, а в папку test1 - один файл из П37. Файл из П37 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 shutil


os.chdir('C:\\Users\\victor\\Desktop\\Pzz\\pythonProject')
files_in_pz11 = [f for f in os.listdir() if os.path.isfile(f)]
print("PZ11:", files_in_pz11)


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


try:
    shutil.move('PZ_6/PZ_6.py', 'test/PZ_6.py')
except FileNotFoundError:
    print('Файл "PZ_6/PZ_6.py" не найден.')

try:
    shutil.move('PZ_6/PZ_6_2.py', 'test/PZ_6_2.py')
except FileNotFoundError:
    print('Файл "PZ_6/PZ_6_2.py" не найден.')

try:
    shutil.move('PZ_7/PZ_7.py', 'test/test1/test.txt')
except FileNotFoundError:
    print('Файл "PZ_7/PZ_7.py" не найден.')


os.chdir('test')
for file in os.listdir():
    if os.path.isfile(file):
        print(f"размер {file}: {os.path.getsize(file)}")


os.chdir('C:\\Users\\victor\\Desktop\\Pzz\\pythonProject\\PZ_11')
shortest_name = min(files_in_pz11, key=len)
print("короткий в PZ11:", shortest_name)


report_path = 'C:\\Users\\victor\\Desktop\\Pzz\\pythonProject\\Report'
os.chdir(report_path)
found = False
for file in os.listdir():
```

```
    if file.endswith('PZ_7.pdf'):
        os.startfile(file)
        found = True
        break

if not found:
    print(f'Файл "PZ_7.pdf" не найден в папке "{report_path}".')


try:
    os.remove('C:\\Users\\victor\\Desktop\\Pzz\\pythonProject\\test\\test1\\test.txt')
except FileNotFoundError:
    print('Файл "test.txt" не найден в папке "test/test1".')
```

Протокол работы программы:

```
PZ11: ['pythonProject.zip']
Файл "PZ_6/PZ_6.py" не найден.
Файл "PZ_6/PZ_6_2.py" не найден.
Файл "PZ_7/PZ_7.py" не найден.
размер PZ_6.py: 437
размер PZ_6_2.py: 822
короткий в PZ11: pythonProject.zip
Traceback (most recent call last):
  File "C:\Users\victor\Desktop\Pzz\pythonProject\PZ_17\PZ_17_3-1.py", line 62, in <module>
    os.startfile(file)
FileNotFoundError: [WinError 2] Не удается найти указанный файл: 'PZ_7.pdf'

Process finished with exit code 1
```

Вывод: я закрепил усвоенные знания, понятия, алгоритмы, основные принципы составления программ, и приобрёл навыки составление программ с использованием GUI Tkinter в IDE PyCharm Community, изучить возможности модуля OS.