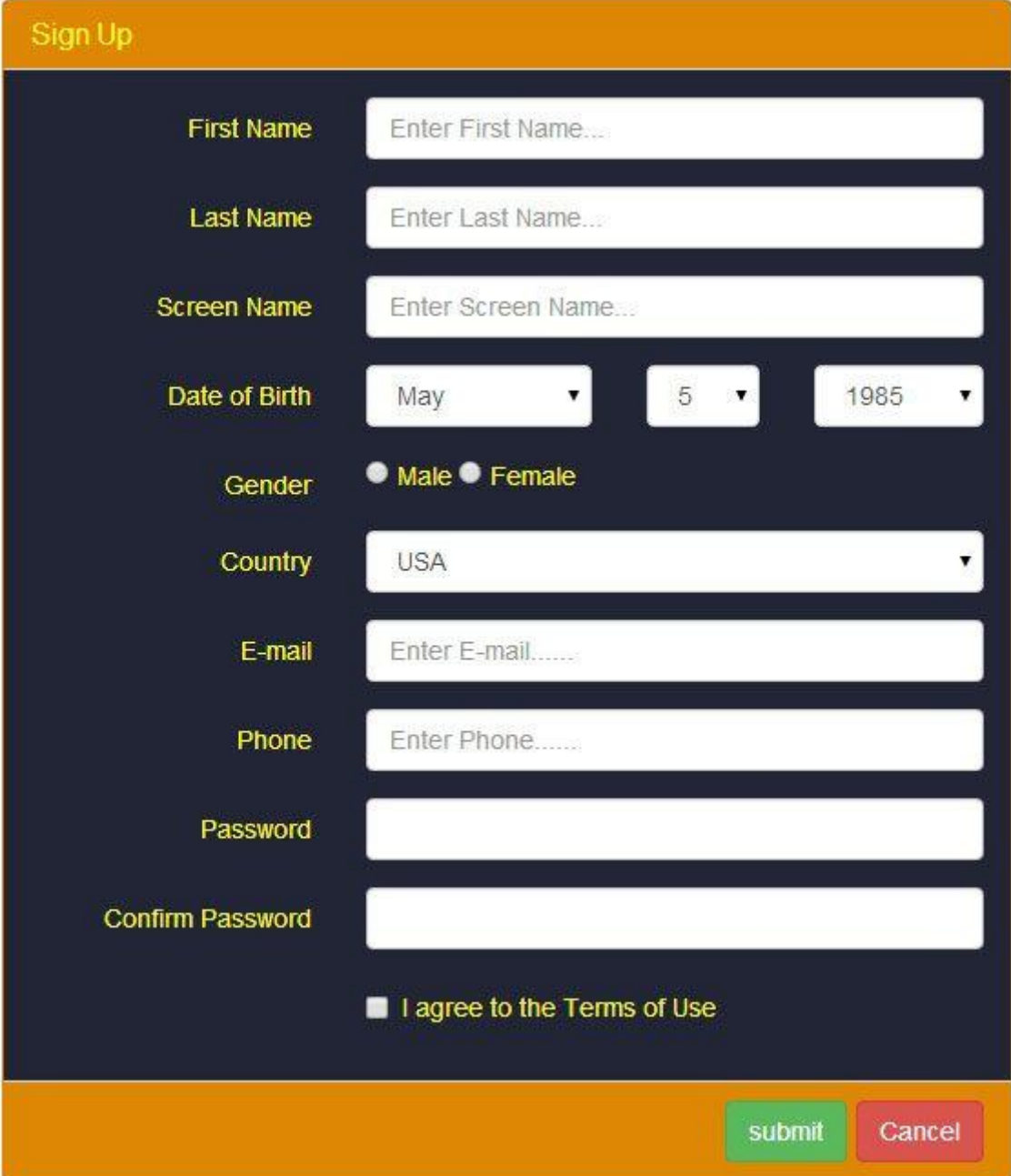


Практическое занятие №17

Тема: Составление программ с использованием tkinter в IDE PyCharm Professional.

Цель: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ с использованием tkinter в IDE PyCharm Professional.

Задание 1: создать десктоп приложение по фотографии:



Sign Up

First Name

Last Name

Screen Name

Date of Birth

Gender ☒ Male ☐ Female

Country

E-mail

Phone

Password

Confirm Password

☐ I agree to the Terms of Use

```
import tkinter as tk
from tkinter import ttk, messagebox

def submit():
```

```

        if not agree_var.get():
            messagebox.showwarning("Warning", "You must agree to the Terms of
Use")
            return
        data = {
            "first_name": first_name_entry.get(),
            "last_name": last_name_entry.get(),
            "screen_name": screen_name_entry.get(),
            "dob_month": dob_month_combobox.get(),
            "dob_day": dob_day_combobox.get(),
            "dob_year": dob_year_combobox.get(),
            "gender": gender_var.get(),
            "country": country_combobox.get(),
            "email": email_entry.get(),
            "phone": phone_entry.get(),
            "password": password_entry.get(),
            "confirm_password": confirm_password_entry.get()
        }
        print("Form submitted with data:", data)

def cancel():
    window.destroy()

window = tk.Tk()
window.title("Sign Up")

frame = ttk.Frame(window, padding="10 10 10 10")
frame.grid(row=0, column=0, sticky=(tk.W, tk.E, tk.N, tk.S))

gender_var = tk.StringVar(value="Male")
agree_var = tk.BooleanVar()

ttk.Label(frame, text="First Name").grid(row=1, column=0, sticky=tk.W)
first_name_entry = ttk.Entry(frame, width=30)
first_name_entry.grid(row=1, column=1, sticky=tk.W)

ttk.Label(frame, text="Last Name").grid(row=2, column=0, sticky=tk.W)
last_name_entry = ttk.Entry(frame, width=30)
last_name_entry.grid(row=2, column=1, sticky=tk.W)

ttk.Label(frame, text="Screen Name").grid(row=3, column=0, sticky=tk.W)
screen_name_entry = ttk.Entry(frame, width=30)
screen_name_entry.grid(row=3, column=1, sticky=tk.W)

ttk.Label(frame, text="Date of Birth").grid(row=4, column=0, sticky=tk.W)
dob_month_combobox = ttk.Combobox(frame,
                                   values=["January", "February", "March",
"April", "May", "June", "July", "August",
"September", "October", "November",
"December"], width=8)
dob_month_combobox.grid(row=4, column=1, sticky=tk.W)
dob_day_combobox = ttk.Combobox(frame, values=[str(i) for i in range(1, 32)],
width=5)
dob_day_combobox.grid(row=4, column=2, sticky=tk.W)
dob_year_combobox = ttk.Combobox(frame, values=[str(i) for i in range(1900,
2025)], width=7)
dob_year_combobox.grid(row=4, column=3, sticky=tk.W)

ttk.Label(frame, text="Gender").grid(row=5, column=0, sticky=tk.W)
ttk.Radiobutton(frame, text="Male", variable=gender_var,
value="Male").grid(row=5, column=1, sticky=tk.W)
ttk.Radiobutton(frame, text="Female", variable=gender_var,

```

```

value="Female").grid(row=5, column=2, sticky=tk.W)

ttk.Label(frame, text="Country").grid(row=6, column=0, sticky=tk.W)
country_combobox = ttk.Combobox(frame, values=["USA", "Canada", "UK",
"Australia"], width=30)
country_combobox.grid(row=6, column=1, sticky=tk.W)

ttk.Label(frame, text="E-mail").grid(row=7, column=0, sticky=tk.W)
email_entry = ttk.Entry(frame, width=30)
email_entry.grid(row=7, column=1, sticky=tk.W)

ttk.Label(frame, text="Phone").grid(row=8, column=0, sticky=tk.W)
phone_entry = ttk.Entry(frame, width=30)
phone_entry.grid(row=8, column=1, sticky=tk.W)

ttk.Label(frame, text="Password").grid(row=9, column=0, sticky=tk.W)
password_entry = ttk.Entry(frame, width=30, show="*")
password_entry.grid(row=9, column=1, sticky=tk.W)

ttk.Label(frame, text="Confirm Password").grid(row=10, column=0, sticky=tk.W)
confirm_password_entry = ttk.Entry(frame, width=30, show="*")
confirm_password_entry.grid(row=10, column=1, sticky=tk.W)

agree_checkbutton = ttk.Checkbutton(frame, text="I agree to the Terms of
Use", variable=agree_var)
agree_checkbutton.grid(row=11, column=1, sticky=tk.W)

button_frame = ttk.Frame(frame)
button_frame.grid(row=12, column=1, columnspan=2, pady=10, sticky=tk.E)

submit_button = ttk.Button(button_frame, text="Submit", command=submit)
submit_button.grid(row=0, column=0, padx=5)

cancel_button = ttk.Button(button_frame, text="Cancel", command=cancel)
cancel_button.grid(row=0, column=1, padx=5)

window.mainloop()

```

Вывод программы:

Задание 2: дано трехзначное число. Вывести вначале его последнюю цифру (единицы), а затем - его среднюю цифру (десятки)

```

import tkinter as tk
from tkinter import messagebox

def get_nums(number) -> None:
    last_digit = number % 10
    middle_digit = (number % 100) // 10
    result = f"Последняя цифра (единица): {last_digit}\nСредняя цифра (десятки): {middle_digit}"
    messagebox.showinfo("Результат", result)

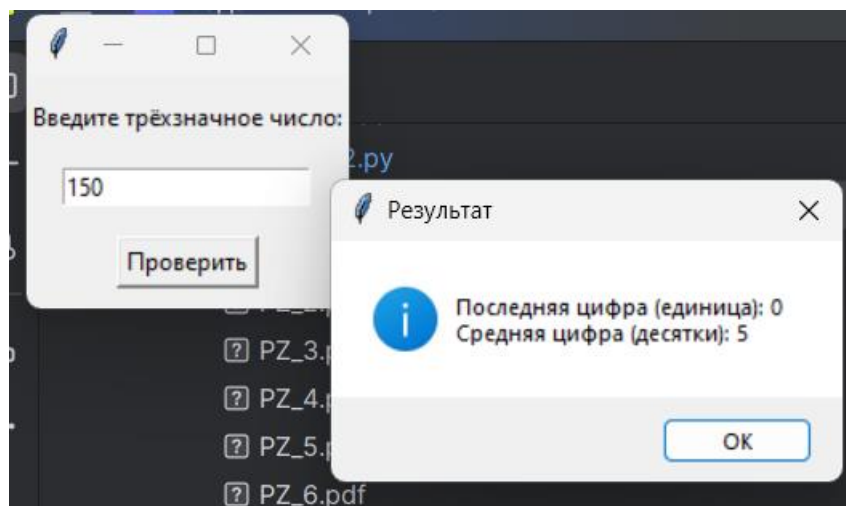
def check_number() -> None:
    number_str = entry.get()
    try:
        number = int(number_str)
        if len(str(number)) == 3:
            get_nums(number)
        else:
            messagebox.showerror("Ошибка", "Введенное вами число не трехзначное")
    except ValueError:
        messagebox.showerror("Ошибка", "Требуется ввести число")

root = tk.Tk()
root.title("Анализ трёхзначного числа")
label = tk.Label(root, text="Введите трёхзначное число:")
label.pack(pady=10)
entry = tk.Entry(root)
entry.pack(pady=5)
button = tk.Button(root, text="Проверить", command=check_number)
button.pack(pady=10)

root.mainloop()

```

Вывод программы:



Задание 3:

```

import os
import shutil

def task1():
    '''Перейдите в каталог PZ_11. Выведите список всех файлов в этом каталоге.'''

```

```

Имена вложенных подкаталогов выводить не нужно. '''
os.chdir("../PZ_11")
files_in_pz11 = [f for f in os.listdir() if os.path.isfile(f)]
print("Список файлов в каталоге PZ_11:", files_in_pz11)

def task2():
    '''перейти в корень проекта, создать папку с именем test. В ней создать
    еще одну папку
    test1. В папку test переместить два файла из ПЗ 6, а в папку test1 - один
    файл из ПЗ 7.
    Файл из ПЗ7 переименовать в PZ_7.1.py. Вывести в консоль информацию о
    размере
    файлов в папке test.'''
    os.chdir('../')
    os.makedirs('test/test1', exist_ok=True)

    file_from_pz6 = 'PZ_6/PZ_6_27.py'
    file_from_pz7 = 'PZ_7/PZ_7_27.py'

    shutil.move(file_from_pz6, 'test/')
    shutil.move(file_from_pz7, 'test/test1/test.txt')

    files_in_test = [f for f in os.listdir('test') if
os.path.isfile(os.path.join('test', f))]
    for file in files_in_test:
        file_size = os.path.getsize(os.path.join('test', file))
        print(f"Размер файла {file} в папке test: {file_size} байт")

def task4():
    '''Найти файл с самым коротким именем в папке PZ_11'''
    os.chdir('../PZ_11')
    shortest_name_file = 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_file))

def task5():
    '''перейти в любую папку где есть отчет в формате .pdf и «запустите» файл
    в
    # привязанной к нему программе. Использовать функцию os.startfile().'''
    os.startfile('../reports/PZ_5.pdf')

    '''удалить файл test.txt.'''
    os.remove('../reports/PZ_2.pdf')
    print("Файл test.txt удален")

```

Создаёт папки:



И выводит в консоль:

```

Список файлов в каталоге PZ_11: ['PZ_27_11.py', '__init__.py']
Файл с самым коротким именем: PZ_27_11.py
Файл test.txt удален

```

Вывод: В процессе выполнения практического занятия №17 я приобрел навыки понятия, алгоритмы, основные принципы составления программ, приобрел навыки составление программ с использованием tkinter в IDE PyCharm Professional.