Студент группы ИС-22 Дьяченко Л.А.

Практическое занятие № 17

Задача 1

Тема: составление программ в IDE PyCharm Community.

Цель: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составление программ с использованием GUI Tkinter в IDE PyCharm Community.

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

<https://i.pinimg.com/originals/73/c6/0d/73c60def8c55043f9fd27b370530a9cf.jpg>

Текст программы:

import tkinter as tk

from tkinter import ttk

def submit():

    print("Form Submitted")

def cancel():

    root.destroy()

# Создание главного окна

root = tk.Tk()

root.title("Sign Up")

# root.geometry("560x656")

root.resizable(width=False, height=False)

root.configure(bg='#e28406')

# Frame 1: Title

frame1 = tk.Frame(root, bg='#e28406', height=41)

frame1.grid(row=0, column=0, columnspan=2, sticky='ew')

label\_title = tk.Label(frame1, text="Sign Up", fg='#ebd216', bg='#e28406', font=('Arial', 16), anchor='w')

label\_title.pack(fill="both", padx=10, pady=10)

# Frame 2: Form Fields

frame2 = tk.Frame(root, bg='#242446')

frame2.grid(row=1, column=0, columnspan=2, padx=0, pady=0, sticky='nsew')

# First Name

label\_first\_name = tk.Label(frame2, text="First Name", bg='#242446', fg='#e9e59b', font=('Arial', 12))

label\_first\_name.grid(row=0, column=0, padx=10, pady=5, sticky='e')

entry\_first\_name = tk.Entry(frame2, width=30, font=('Arial', 12))

entry\_first\_name.insert(0, "Enter First Name...")

entry\_first\_name.grid(row=0, column=1, padx=10, pady=10, sticky='w')

# Last Name

label\_last\_name = tk.Label(frame2, text="Last Name", bg='#242446', fg='#e9e59b', font=('Arial', 12))

label\_last\_name.grid(row=1, column=0, padx=10, pady=5, sticky='e')

entry\_last\_name = tk.Entry(frame2, width=30, font=('Arial', 12))

entry\_last\_name.insert(0, "Enter Last Name...")

entry\_last\_name.grid(row=1, column=1, padx=10, pady=5, sticky='w')

# Screen Name

label\_screen\_name = tk.Label(frame2, text="Screen Name", bg='#242446', fg='#e9e59b', font=('Arial', 12))

label\_screen\_name.grid(row=2, column=0, padx=10, pady=5, sticky='e')

entry\_screen\_name = tk.Entry(frame2, width=30, font=('Arial', 12))

entry\_screen\_name.insert(0, "Enter Screen Name...")

entry\_screen\_name.grid(row=2, column=1, padx=10, pady=5, sticky='w')

# Date of Birth

label\_dob = tk.Label(frame2, text="Date of Birth", bg='#242446', fg='#e9e59b', font=('Arial', 12))

label\_dob.grid(row=3, column=0, padx=10, pady=5, sticky='e')

# Создание фрейма для выпадающих списков даты рождения

dob\_frame = tk.Frame(frame2, bg='#242446')

dob\_frame.grid(row=3, column=1, padx=10, pady=5, sticky='w')

months = ttk.Combobox(dob\_frame, values=["January", "February", "March", "April", "May", "June",

                                      "July", "August", "September", "October", "November", "December"], width=7)

months.current(4)

months.grid(row=0, column=0, padx=0, pady=5)

days = ttk.Combobox(dob\_frame, values=list(range(1, 32)), width=3)

days.current(4)

days.grid(row=0, column=1, padx=0, pady=5)

years = ttk.Combobox(dob\_frame, values=list(range(1900, 2024)), width=5)

years.current(85)

years.grid(row=0, column=2, padx=0, pady=5)

# Gender

label\_gender = tk.Label(frame2, text="Gender", bg='#242446', fg='#e9e59b', font=('Arial', 12))

label\_gender.grid(row=4, column=0, padx=10, pady=5, sticky='e')

gender\_frame = tk.Frame(frame2, bg='#242446')

gender\_frame.grid(row=4, column=1, columnspan=3, padx=5, pady=5, sticky='w')

male\_rb = tk.Radiobutton(gender\_frame, text="Male", variable=tk.IntVar(), value=1, bg='#242446', fg='#e9e59b', font=('Arial', 12))

male\_rb.pack(side='left')

female\_rb = tk.Radiobutton(gender\_frame, text="Female", variable=tk.IntVar(), value=2, bg='#242446', fg='#e9e59b', font=('Arial', 12))

female\_rb.pack(side='left')

# Country

label\_country = tk.Label(frame2, text="Country", bg='#242446', fg='#e9e59b', font=('Arial', 12))

label\_country.grid(row=5, column=0, padx=10, pady=5, sticky='e')

country = ttk.Combobox(frame2, values=["USA", "Canada", "UK", "Australia"], width=20)

country.current(0)

country.grid(row=5, column=1, columnspan=3, padx=10, pady=5, sticky='w')

# E-mail

label\_email = tk.Label(frame2, text="E-mail", bg='#242446', fg='#e9e59b', font=('Arial', 12))

label\_email.grid(row=6, column=0, padx=10, pady=5, sticky='e')

entry\_email = tk.Entry(frame2, width=30, font=('Arial', 12))

entry\_email.insert(0, "Enter E-mail...")

entry\_email.grid(row=6, column=1, columnspan=3, padx=10, pady=5, sticky='w')

# Phone

label\_phone = tk.Label(frame2, text="Phone", bg='#242446', fg='#e9e59b', font=('Arial', 12))

label\_phone.grid(row=7, column=0, padx=10, pady=5, sticky='e')

entry\_phone = tk.Entry(frame2, width=30, font=('Arial', 12))

entry\_phone.insert(0, "Enter Phone...")

entry\_phone.grid(row=7, column=1, columnspan=3, padx=10, pady=5, sticky='w')

# Password

label\_password = tk.Label(frame2, text="Password", bg='#242446', fg='#e9e59b', font=('Arial', 12))

label\_password.grid(row=8, column=0, padx=10, pady=5, sticky='e')

entry\_password = tk.Entry(frame2, width=30, font=('Arial', 12), show="\*")

entry\_password.grid(row=8, column=1, columnspan=3, padx=10, pady=5, sticky='w')

# Confirm Password

label\_confirm\_password = tk.Label(frame2, text="Confirm Password", bg='#242446', fg='#e9e59b', font=('Arial', 12))

label\_confirm\_password.grid(row=9, column=0, padx=10, pady=5, sticky='e')

entry\_confirm\_password = tk.Entry(frame2, width=30, font=('Arial', 12), show="\*")

entry\_confirm\_password.grid(row=9, column=1, columnspan=3, padx=10, pady=5, sticky='w')

# Agreement Checkbutton

# Creating a separate frame for the label and checkbutton

frame\_agree = tk.Frame(frame2, bg='#242446')

frame\_agree.grid(row=10, column=1, columnspan=4, padx=10, pady=5, sticky='e')

# Agreement Label

label\_agree = tk.Label(frame\_agree, text="I agree to the Terms of Use", bg='#242446', fg='#e9e59b', font=('Arial', 12))

label\_agree.grid(row=0, column=1, padx=10, pady=5, sticky='w')

# Agreement Checkbutton

agree\_var = tk.IntVar()

check\_agree = tk.Checkbutton(frame\_agree, variable=agree\_var, bg='#242446')

check\_agree.grid(row=0, column=0, padx=5, pady=5, sticky='w')

# Frame 3: Buttons

frame3 = tk.Frame(root, bg='#e28406')

frame3.grid(row=2, column=0, columnspan=2, sticky='ew')

cancel\_button = tk.Button(frame3, text="Cancel", command=cancel, fg="#ffffff", bg='#fc4c4c', font=('Arial', 14), width=10)

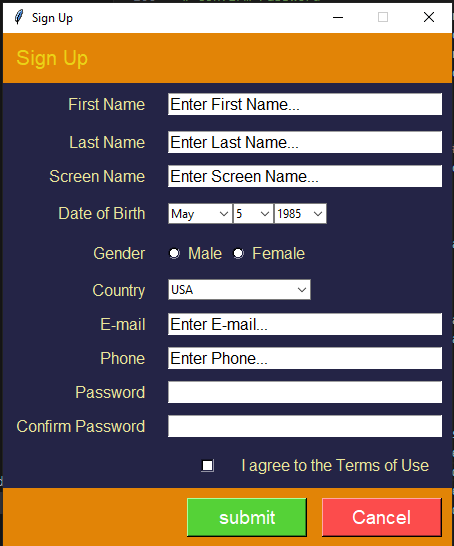
cancel\_button.pack(side='right', padx=10, pady=10)

submit\_button = tk.Button(frame3, text="submit", command=submit, fg="#ffffff", bg='#55d237', font=('Arial', 14), width=10)

submit\_button.pack(side='right', padx=5, pady=10)

root.mainloop()

Протокол работы программы:



Вывод: Я закрепил усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составление программ с использованием GUI Tkinter в IDE PyCharm Community.

PyCharm Community.

Задача 2

Тема: составление программ в IDE PyCharm Community.

Цель: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составление программ с использованием GUI Tkinter в IDE PyCharm Community.

Постановка задачи:

Задание 2. Разработать программу с применением пакета tk, взяв в качестве условия одну

любую задачу из ПЗ №№ 2 – 9.

Текст программы:

import tkinter as tk

from tkinter import ttk

from tkinter import messagebox

def generate\_string():

    try:

        n = int(entry\_n.get())

        if n <= 0:

            raise Exception("N должно быть больше 0")

        show\_result('Результат задачи 1', 'C' \* n)

    except Exception as exc:

        messagebox.showerror("Ошибка", str(exc))

def normalize\_spaces():

    try:

        string = entry\_string.get()

        show\_result('Результат задачи 2', ' '.join(string.split()))

    except Exception as exc:

        messagebox.showerror("Ошибка", str(exc))

def show\_result(title, result):

    result\_window = tk.Toplevel(root)

    result\_window.title(title)

    result\_label = tk.Label(result\_window, text=result, padx=20, pady=20)

    result\_label.pack()

# Создание главного окна

root = tk.Tk()

root.title("Tkinter Задачи")

# Создание вкладок

notebook = ttk.Notebook(root)

notebook.pack(pady=10, expand=True)

# Вкладка для задачи 1

frame1 = ttk.Frame(notebook)

notebook.add(frame1, text='Задача 1')

label\_n = tk.Label(frame1, text="Введите целое число N (>0):")

label\_n.pack(side=tk.LEFT, padx=10, pady=10)

entry\_n = tk.Entry(frame1)

entry\_n.pack(side=tk.LEFT, padx=10, pady=10)

button\_generate = tk.Button(frame1, text="Создать строку", command=generate\_string)

button\_generate.pack(side=tk.LEFT, padx=10, pady=10)

# Вкладка для задачи 2

frame2 = ttk.Frame(notebook)

notebook.add(frame2, text='Задача 2')

label\_string = tk.Label(frame2, text="Введите строку:")

label\_string.pack(side=tk.LEFT, padx=10, pady=10)

entry\_string = tk.Entry(frame2)

entry\_string.pack(side=tk.LEFT, padx=10, pady=10)

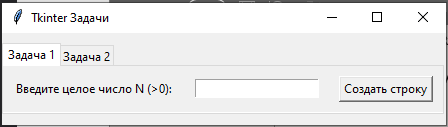
button\_normalize = tk.Button(frame2, text="Нормализовать пробелы", command=normalize\_spaces)

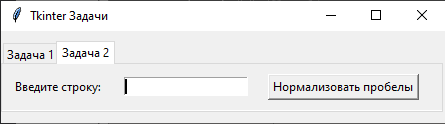
button\_normalize.pack(side=tk.LEFT, padx=10, pady=10)

# Запуск главного цикла приложения

root.mainloop()

Протокол работы программы:





Вывод: Я закрепил усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составление программ с использованием GUI Tkinter в IDE PyCharm Community.

PyCharm Community.

Задача 3

Тема: составление программ в IDE PyCharm Community.

Цель: закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составление программ с использованием GUI Tkinter в IDE PyCharm Community.

Постановка задачи:

Задание предполагает, что у студента есть проект с практическими работами (№№ 2-13),

оформленный согласно требованиям. Все задания выполняются c 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

Текст программы:

import os

import shutil

try:

    # 1. Перейти в каталог PZ11 и вывести список всех файлов

    os.chdir('PZ11')

    files\_in\_pz11 = [f for f in os.listdir() if os.path.isfile(f)]

    print("Files in PZ11:", files\_in\_pz11)

    # 2. Перейти в корень проекта, создать папку test и test1

    os.chdir('..')  # Переход в корень проекта

    os.makedirs('test/test1', exist\_ok=True)

    # 3. Переместить файлы из ПЗ6 и ПЗ7

    # Предположим, что файлы называются file1\_PZ6.txt, file2\_PZ6.txt и file\_PZ7.txt

    pz6\_files = ['PZ6/file1\_PZ6.txt', 'PZ6/file2\_PZ6.txt']

    pz7\_file = 'PZ7/file\_PZ7.txt'

    for file in pz6\_files:

        if os.path.exists(file):

            shutil.move(file, 'test/' + os.path.basename(file))

        else:

            print(f"File not found: {file}")

    if os.path.exists(pz7\_file):

        shutil.move(pz7\_file, 'test/test1/test.txt')

    else:

        print(f"File not found: {pz7\_file}")

    # 4. Вывести информацию о размере файлов в папке test

    test\_files = [f for f in os.listdir('test') if os.path.isfile(os.path.join('test', f))]

    for file in test\_files:

        file\_size = os.path.getsize(os.path.join('test', file))

        print(f"Size of {file}: {file\_size} bytes")

    # 5. Найти файл с самым коротким именем в PZ11

    os.chdir('PZ11')

    shortest\_name\_file = min(files\_in\_pz11, key=len)

    print("File with shortest name:", os.path.basename(shortest\_name\_file))

    # 6. Запустить отчет в формате .pdf

    pdf\_path = 'reports/report13.pdf'

    if os.path.exists(pdf\_path):

        os.startfile(pdf\_path)

    else:

        print(f"PDF report not found: {pdf\_path}")

    # 7. Удалить файл test.txt

    test\_txt\_path = 'test/test1/test.txt'

    if os.path.exists(test\_txt\_path):

        os.remove(test\_txt\_path)

    else:

        print(f"File not found: {test\_txt\_path}")

except Exception as e:

    print(f"An error occurred: {e}")

Протокол работы программы:

Files in PZ11: ['pz1.py', 'pz2.py']

Size of 1.py: 553 bytes

Size of 2.py: 799 bytes

File with shortest name: pz1.py

Вывод: Я закрепил усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составление программ с использованием GUI Tkinter в IDE PyCharm Community.