

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.

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.

The image shows a 'Sign Up' form with a dark blue background and orange borders. The form contains the following fields and elements:

- First Name:** Text input field with placeholder 'Enter First Name...'
- Last Name:** Text input field with placeholder 'Enter Last Name...'
- Screen Name:** Text input field with placeholder 'Enter Screen Name...'
- Date of Birth:** Three dropdown menus for month (May), day (5), and year (1985).
- Gender:** Radio buttons for 'Male' and 'Female'.
- Country:** Dropdown menu with 'USA' selected.
- E-mail:** Text input field with placeholder 'Enter E-mail.....'
- Phone:** Text input field with placeholder 'Enter Phone.....'
- Password:** Text input field.
- Confirm Password:** Text input field.
- I agree to the Terms of Use:** A checkbox.
- Buttons:** 'submit' (green) and 'Cancel' (red) buttons at the bottom right.

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 Вариант
28"""
from tkinter import *
from tkinter import ttk
root = Tk()
root.title('Sign up')
root.geometry('600x700')
root.resizable(width=False, height=False)

canvas = Canvas(root, width=600, height=700, bg='#2C2F48')
canvas.pack()

frame = Frame(canvas, bg='orange')
frame.place(relwidth=1, relheight=0.07)

title = Label(frame, text='Sign up', fg='gray', bg='orange',
font='Arial 14')
title.place(relx=0.05, rely=0.2)

label = Label(canvas, text="First Name", fg='yellow',
font=('Arial', 10, 'bold'), bg='#2C2F48')
label.place(relx=0.05, rely=0.15)

# Поле ввода
entry = Entry(canvas, font=('Arial', 10))
entry.insert(0, "Enter First Name...")
entry.place(relx=0.3, rely=0.15, relwidth=0.6)

label2 = Label(canvas, text="Last Name", fg='yellow',
font=('Arial', 10, 'bold'), bg='#2C2F48')
label2.place(relx=0.05, rely=0.2)

# Поле ввода "Last Name"
entry2 = Entry(canvas, font=('Arial', 10))
entry2.insert(0, "Enter Last Name...")
entry2.place(relx=0.3, rely=0.2, relwidth=0.6)
#-----
label3 = Label(canvas, text="Screen Name", fg='yellow',
font=('Arial', 10, 'bold'), bg='#2C2F48')
label3.place(relx=0.05, rely=0.25)

# Поле ввода "Screen Name"
entry3 = Entry(canvas, font=('Arial', 10))
entry3.insert(0, "Enter Scree Name...")
entry3.place(relx=0.3, rely=0.25, relwidth=0.6)

# Метка "Date of Birth"
```

```
label4 = Label(canvas, text="Date of Birth", fg='yellow',
font=('Arial', 10, 'bold'), bg='#2C2F48')
label4.place(relx=0.05, rely=0.3)

# Список месяцев
months = ["January", "February", "March", "April", "May",
"June", "July", "August", "September", "October", "November",
"December"]
month_var = StringVar(canvas)
month_var.set("May") # Установим значение по умолчанию
dropdown_month = OptionMenu(canvas, month_var, *months)
dropdown_month.place(relx=0.3, rely=0.3, relwidth=0.2)

# Список дней
days = [str(i) for i in range(1, 32)]
day_var = StringVar(canvas)
day_var.set("5")
dropdown_day = OptionMenu(canvas, day_var, *days)
dropdown_day.place(relx=0.52, rely=0.3, relwidth=0.1)

# Список годов
years = [str(i) for i in range(1985, 2024)]
year_var = StringVar(canvas)
year_var.set("1985")
dropdown_year = OptionMenu(canvas, year_var, *years)
dropdown_year.place(relx=0.64, rely=0.3, relwidth=0.26)

# Метка "Gender"
label5 = Label(canvas, text="Gender", fg='yellow',
font=('Arial', 10, 'bold'), bg='#2C2F48')
label5.place(relx=0.05, rely=0.35)

# Пол выбора пола
gender = IntVar()

male = Radiobutton(canvas, text='Male', variable=gender,
value=1, bg='#2C2F48', fg='yellow', font=('Arial', 10))
male.place(relx=0.3, rely=0.35)
female = Radiobutton(canvas, text='Female', variable=gender,
value=2, bg='#2C2F48', fg='yellow', font=('Arial', 10))
female.place(relx=0.4, rely=0.35)

# Метка "Country"
label6 = Label(canvas, text="Country", fg='yellow',
font=('Arial', 10, 'bold'), bg='#2C2F48')
label6.place(relx=0.05, rely=0.4)

# Список стран
dropdown_country = ttk.Combobox(root, values=["USA", "Russia",
"Germany"], width=20)
dropdown_country.place(relx=0.3, rely=0.4, relwidth=0.6)
```

```

label7 = Label(canvas, text="E-mail", fg='yellow',
font=('Arial', 10, 'bold'), bg='#2C2F48')
label7.place(relx=0.05, rely=0.47)

entry7 = Entry(canvas, font=('Arial', 10))
entry7.insert(0, "Enter E-mail...")
entry7.place(relx=0.3, rely=0.47, relwidth=0.6)

label8 = Label(canvas, text="Phone", fg='yellow', font=('Arial',
10, 'bold'), bg='#2C2F48')
label8.place(relx=0.05, rely=0.52)

# Поле ввода "Phone"
entry8 = Entry(canvas, font=('Arial', 10))
entry8.insert(0, "Enter Phone...")
entry8.place(relx=0.3, rely=0.52, relwidth=0.6)

# Метка "Password"
label9 = Label(canvas, text="Password", fg='yellow',
font=('Arial', 10, 'bold'), bg='#2C2F48')
label9.place(relx=0.05, rely=0.57)

# Поле ввода "Password"
entry9 = Entry(canvas, font=('Arial', 10), show='*')
entry9.place(relx=0.3, rely=0.57, relwidth=0.6)

# Метка "Confirm Password"
label10 = Label(canvas, text="Confirm Password", fg='yellow',
font=('Arial', 10, 'bold'), bg='#2C2F48')
label10.place(relx=0.05, rely=0.62)

# Поле ввода "Confirm Password"
entry10 = Entry(canvas, font=('Arial', 10), show='*')
entry10.place(relx=0.3, rely=0.62, relwidth=0.6)

agree_var = IntVar()
agree_checkbox = Checkbutton(canvas, text="I agree to the Terms
of Use", variable=agree_var, fg='yellow', bg='#2C2F48',
font=('Arial', 10))
agree_checkbox.place(relx=0.292, rely=0.68)

bottom_frame = Frame(canvas, bg='orange')
bottom_frame.place(relwidth=1, relheight=0.07, rely=0.93)

# Кнопка "Submit"
submit_button = Button(bottom_frame, text="Submit", bg='green',
fg='white', font=('Arial', 10))
submit_button.place(relx=0.4, rely=0.2, relwidth=0.2,
relheight=0.6)

# Кнопка "Cancel"
cancel_button = Button(bottom_frame, text="Cancel", bg='red',
fg='white', font=('Arial', 10))

```

```
cancel_button.place(relx=0.7, rely=0.2, relwidth=0.2,
relheight=0.6)
root.mainloop()
```

Протокол работы программы:

Постановка задачи 2:

С начала суток прошло N секунд (N – целое). Найти количество полных минут, прошедших с начала последнего часа.

Текст программы:

```
С начала суток прошло N секунд (N – целое). Найти количество
полных минут, прошедших с начала последнего часа.
import tkinter as tk
from tkinter import messagebox

def calculate_minutes():
    try:
        n = int(entry.get())
        minutes = (n // 60) % 60
        result_label.config(text=f'Количество полных минут:
```

```

{minutes}')
    except ValueError:
        messagebox.showerror("Ошибка", "Введите корректное
количество секунд.")

root = tk.Tk()
root.title("Калькулятор минут")
root.geometry('300x300+400+100')

entry_label = tk.Label(root, text="Введите количество секунд:")
entry_label.place(relx=0.25, rely=0.05)

entry = tk.Entry(root)
entry.place(relx=0.3, rely=0.15)

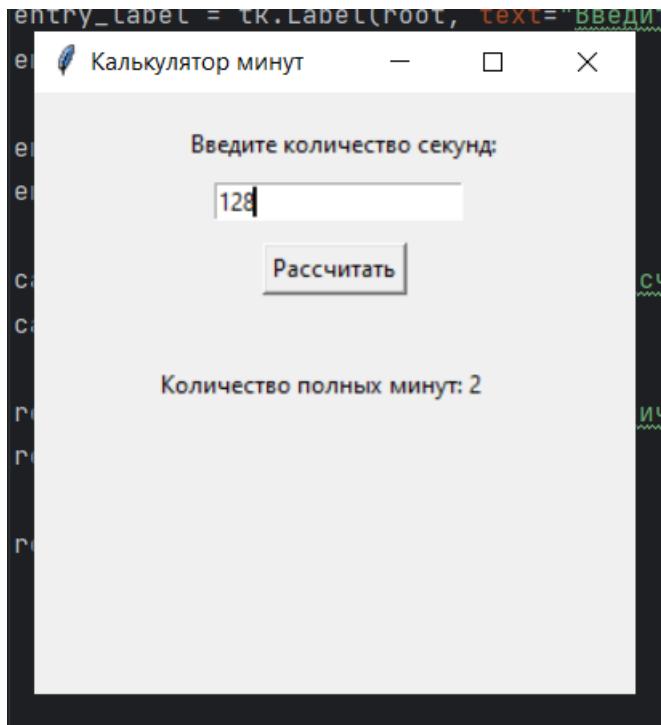
calc_button = tk.Button(root, text="Рассчитать",
command=calculate_minutes)
calc_button.place(relx=0.38, rely=0.25)

result_label = tk.Label(root, text="Количество полных минут: ")
result_label.place(relx=0.2, rely=0.45)

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 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:

```
"""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 PZ_7_1.py. 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 PZ_7_1.py."""
import os

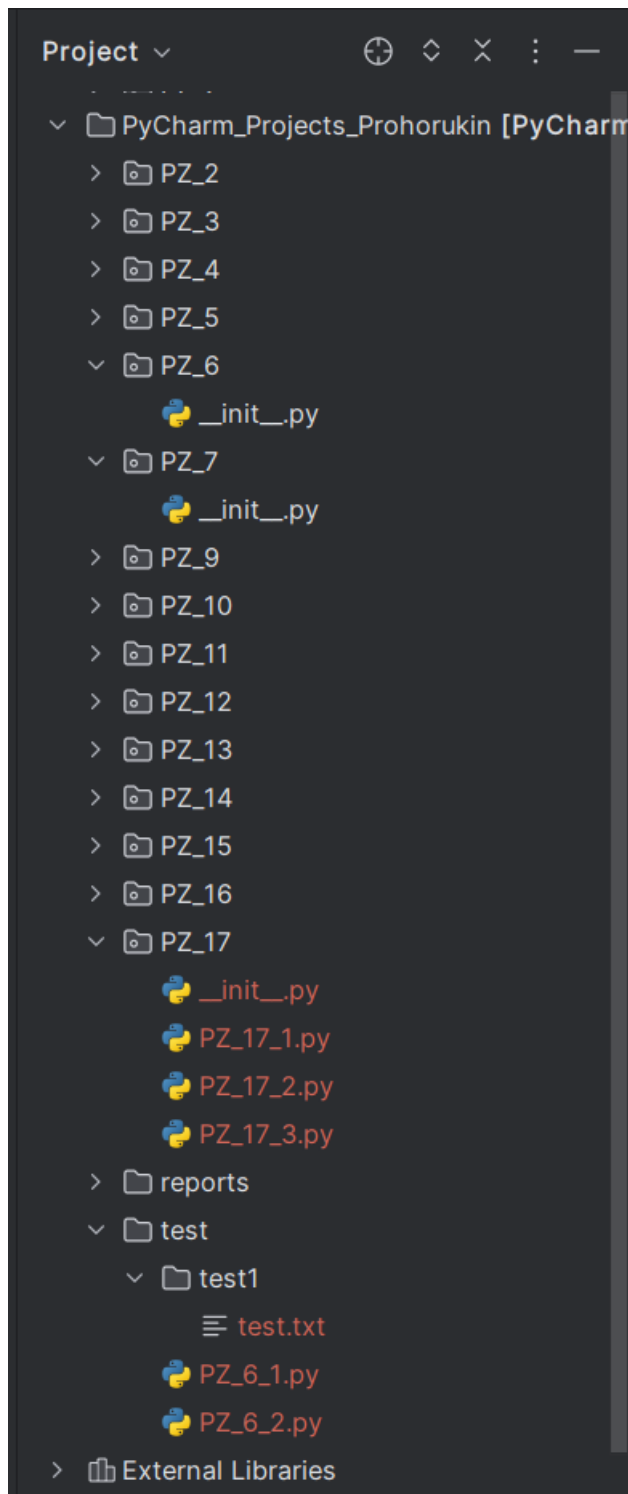
os.chdir('../PZ_11')
files_in_directory = [f for f in os.listdir()]
print('Все файлы ПЗ_11:')
for file in files_in_directory:
    print(file)
```

```

# -----
-----
os.chdir('..')
os.makedirs("test/test1", exist_ok=True)
os.chdir('./PZ_6')
os.replace('PZ_6_1.py', '../test/PZ_6_1.py')
os.replace('PZ_6_2.py', '../test/PZ_6_2.py')
os.chdir('../PZ_7')
os.replace('PZ_7_2.py', '../test/test1/test.txt')
os.chdir('../test')
print("Размер файла PZ_6_1:", os.stat("PZ_6_1.py").st_size)
print("Размер файла PZ_6_2:", os.stat("PZ_6_2.py").st_size)
#-----
os.chdir('../PZ_11')
files = os.listdir('.')
shortest_name_file = min(files, key=lambda x:
len(os.path.basename(x)))
print(f'\nФайл из ПЗ_11 с самым коротким именем:
{os.path.basename(shortest_name_file)}')
os.chdir('../reports')
pdf_file = 'PZ_3.pdf'
os.startfile(pdf_file)
#-----
-----
os.remove("../PZ_7/PZ_7_1.py")

```

Протокол работы программы:



Вывод: в процессе выполнения заданий я закрепил усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрел навыки составления программ с использованием GUI Tkinter в IDE PyCharm Community.

Выполнены разработка программного кода, отладка, тестирование, оптимизация кода. Используются языковые конструкции: `import tkinter`, `Tk()`, `Canvas()`, `Entry()`, `Label()`. Готовые программные коды выложены на GitHub