

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.

Задание 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

Тип алгоритма.

Линейный.

Текст программы.

```
from tkinter import *
from tkinter import ttk
from PIL import Image, ImageTk

root = Tk()
root.title('Registration')
root.geometry('500x600')
root.configure(bg='#2e2c2c')

screen_width = root.winfo_screenwidth()
screen_height = root.winfo_screenheight()

window_width = 440
window_height = 580

position_top = int(screen_height/2 - window_height/2)
position_right = int(screen_width/2 - window_width/2)

root.geometry(f'{window_width}x{window_height}+{position_right}+{position_top}')

def on_entry_click(event):
    if reg_entry.get() == 'Username':
        reg_entry.delete(0, "end")
        reg_entry.insert(0, "")
        reg_entry.config(fg='#2e2c2c')

def on_focusout(event):
    if reg_entry.get() == ";
```

```
reg_entry.insert(0, "")

reg_entry.config(fg='#2e2c2c')

# Style

style = ttk.Style()

style.configure('TEntry', fieldbackground='#2e2c2c', foreground='white', relief='flat', height=2, borderradius=5)

# FRAME TEXT

frame_text = Frame(root, bg='#2e2c2c', relief=SUNKEN)

frame_text.place(x=0, y=0, anchor='nw', width=440, height=100)

reg_text = Label(frame_text, text="Sign Up", font='arial 14', fg='#aba2a2', background='#545151', width='60', height=2)

reg_text.pack(anchor='nw', padx=25, pady=(25, 0))

# FRAME FORM

frame_form = Frame(root, bg='#171616', relief=SUNKEN)

frame_form.place(x=25, y=71, width=390, height=480)

# Field registration

reg_label = Label(frame_form, text="Username", fg='#aba2a2', font='arial 12 bold', background='#171616')

reg_label.grid(row=0, column=0, padx=29, pady=(15, 3), sticky='nw')

reg_entry = ttk.Entry(frame_form, width=40, style="TEntry")

reg_entry.insert(0, 'Username')

reg_entry.bind('<FocusIn>', on_entry_click)

reg_entry.bind('<FocusOut>', on_focusout)

reg_entry.grid(row=1, column=0, padx=30, pady=(0, 3))

# Field email

email_label = Label(frame_form, text="Email", fg='#aba2a2', font='arial 12 bold', background='#171616')

email_label.grid(row=3, column=0, padx=29, pady=(15, 3), sticky='nw')

email_entry = ttk.Entry(frame_form, width=40, style="TEntry")

email_entry.insert(0, 'Email')

email_entry.bind('<FocusIn>', on_entry_click)

email_entry.bind('<FocusOut>', on_focusout)

email_entry.grid(row=4, column=0, padx=30, pady=(0, 3))

# Field password

password_label = Label(frame_form, text="Password", fg='#aba2a2', font='arial 12 bold', background='#171616')

password_label.grid(row=6, column=0, padx=29, pady=(15, 3), sticky='nw')
```

```

password_entry = ttk.Entry(frame_form, width=40, style="TEntry")

password_entry.insert(0, 'Password')

password_entry.bind('<FocusIn>', on_entry_click)

password_entry.bind('<FocusOut>', on_focusout)

password_entry.grid(row=7, column=0, padx=30, pady=(0, 3))


# Field retype password

retype_password_label = Label(frame_form, text="Retype password", fg='#aba2a2', font='arial 12 bold', background='#171616')

retype_password_label.grid(row=9, column=0, padx=29, pady=(15, 3), sticky='nw')

retype_password_entry = ttk.Entry(frame_form, width=40, style="TEntry")

retype_password_entry.insert(0, 'Retype password')

retype_password_entry.bind('<FocusIn>', on_entry_click)

retype_password_entry.bind('<FocusOut>', on_focusout)

retype_password_entry.grid(row=10, column=0, padx=30, pady=(0, 3))


# Field check language

frame_accept = Frame(root, bg='#171616', relief=SUNKEN)

frame_accept.place(x=46, y=355, width=340, height=80)

accept_var1 = StringVar()

accept_check1 = Checkbutton(frame_accept, variable=accept_var1, offvalue="", font='arial 11', bg='#171616')

accept_check1.grid(row=0, column=0, padx=10, pady=(0, 3))

language_label = Label(frame_accept, text="Accept the terms and policies", font='arial 11', fg='#aba2a2', background='#171616')

language_label.grid(row=0, column=1, pady=(0, 3))

btn_lab = Button(root, text='SIGN UP', width=8, height=1, font='arial 13 bold', bg='#00b3ff', fg='white')

btn_lab.place(x=56, y=400)

text_lab = Label(root, text='Already have an account?', width=40, height=1, font='arial 11', bg='#171616', fg='white')

text_lab.place(x=56, y=450)

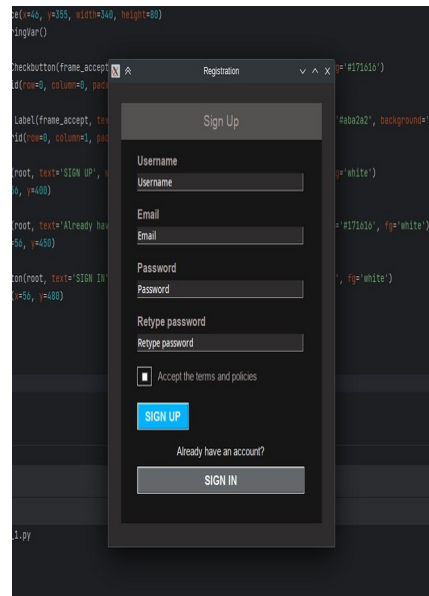
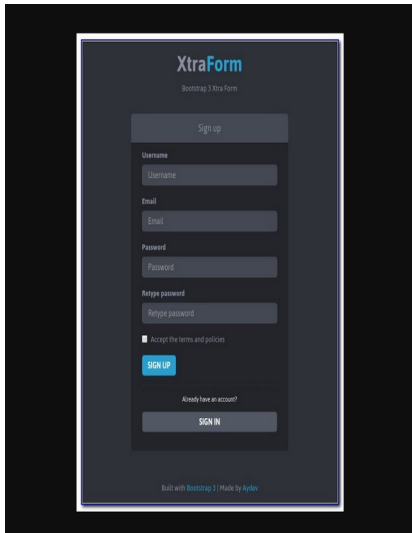
btn_signin = Button(root, text='SIGN IN', width=33, height=1, font='arial 13 bold', bg='#636769', fg='white')

btn_signin.place(x=56, y=480)

root.mainloop()

```

Протокол работы программы.



Задание 2.

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

Тип алгоритма.

Линейный.

Текст программы.

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

Даны три целых числа A, B, C.

Проверить истинность высказывания: "Хотя бы одно из чисел A, B, C положительное"

```
import tkinter as tk
```

```
def check_numbers():
```

```
    try:
```

```
        a = float(a_entry.get())
```

```
        b = float(b_entry.get())
```

```
        c = float(c_entry.get())
```

```
    if (a > 0) or (b > 0) or (c > 0):
```

```
        result_label.config(text="True")
```

```
    else:
```

```
        result_label.config(text="False")
```

```
except ValueError:
```

```
    result_label.config(text="Error!!! Введите число")
```

```
root = tk.Tk()
```

```
root.title("Проверка чисел")
```

```

a_label = tk.Label(root, text="Введите первое число:")
a_label.pack()

a_entry = tk.Entry(root)
a_entry.pack()

b_label = tk.Label(root, text="Введите второе число:")
b_label.pack()

b_entry = tk.Entry(root)
b_entry.pack()

c_label = tk.Label(root, text="Введите третье число:")
c_label.pack()

c_entry = tk.Entry(root)
c_entry.pack()

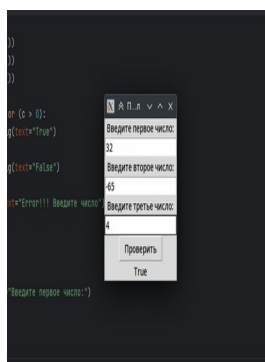
check_button = tk.Button(root, text="Проверить", command=check_numbers)
check_button.pack()

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

root.mainloop()

```

Протокол работы программы.



Задание 3.

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.

Тип алгоритма.

Линейный.

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/clone/PZ17')

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 PЗ6, а в папку test1 - один файл из PЗ7.

# Файл из PЗ7 переименовать в test.txt. Вывести в консоль информацию о размере

# файлов в папке test.

# os.chdir(r'/home/student/Документы/clone')

# os.makedirs('test/test1')

# os.replace(r'PZ6/PZ6_1.py', 'test/PZ6_1.py')

# os.replace(r'PZ6/PZ6_2.py', 'test/PZ6_2.py')

# os.replace(r'PZ7/pz7_2.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/clone/PZ11')

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/clone/reports/pract13.pdf'
subprocess.call(['xdg-open', pdf_file])

# 5. удалить файл test.txt.

file = r'/home/student/Документы/clone/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 работы программы.

/usr/bin/python3.9 /home/student/Документы/clone/PZ17/pz17_3.py

Список файлов в каталоге PZ_11:

data_1.txt

text18-13.txt

pz11_2.py

data_2.txt

new_file.txt

pz11_1.py

Самое короткое имя имеет файл: pz11_2.py

Файл успешно удален

Process finished with exit code 0

Вывод: закрепила усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрела навыки составления

программ с использованием GUI Tkinter в IDE PyCharm Community, изучила возможности модуля OS.