

Практическое занятие №12.

Тема: составление программ с использованием GUI Tkinter в IDE PyCharm Community.

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

Задача №1.

Постановка задачи: В соответствии с номером варианта (28) перейти по ссылке на прототип. Реализовать его в IDE PyCharm Community с применением пакета tk. Получить интерфейс максимально приближенный к оригиналу.

Текст программы:

```
from tkinter import *
from tkinter import ttk
from random import randint as random
from tkinter import messagebox

wincolor = '#DDD'
root = Tk()
root.title('Sign Up')
root.configure(bg=wincolor)
root.geometry('0x0')
root.minsize(width=585, height=365)
root.maxsize(width=585, height=365)

labwidth = 15
randint = random(0, 2)
randpic = ''
if randint == 0:
    randpic = '●'
elif randint == 1:
    randpic = '●'
elif randint == 2:
    randpic = '☆'
print('SYSTEM REPORT: PASSWORD')
print('\'+randpic+'\' will be in the password'
      if randint != 2 else 'Lucky one, \''+randpic+'\' will
be in the password')

namframe = Frame(root, background=wincolor)
firnam = Label(namframe, text='First Name', width=labwidth,
anchor='e', background=wincolor)
nametext = Entry(namframe, width=50, font='Arial 14',
background=wincolor)
```

```

lasnamframe = Frame(root, background=wincolor)
lasnam = Label(lasnamframe, text='Last Name', width=labwidth,
anchor='e', background=wincolor)
lasnametext = Entry(lasnamframe, width=50, font='Arial 14',
background=wincolor)

scrnframe = Frame(root, background=wincolor)
scrn = Label(scrnframe, text='Screen Name',
width=labwidth, anchor='e', background=wincolor)
scrnmetext = Entry(scrnframe, width=50, font='Arial 14',
background=wincolor)

birthframe = Frame(root, background=wincolor)
birthlab = Label(birthframe, text='Date of Birth',
justify='right', width=labwidth, anchor='e',
background=wincolor)
months = [
    u'January', u'February', u'March', u'April', u'May',
    u'June', u'July', u'August', u'September', u'October',
    u'November',
    u'December']
days = [str(i) for i in range(1, 32)]
years = [str(y) for y in range(1970, 2022)]
listheight = 5
listwidth = 12
monthlist = ttk.Combobox(birthframe, height=listheight,
width=listwidth, values=months)
daylist = ttk.Combobox(birthframe, height=listheight,
width=listwidth, values=days)
yearlist = ttk.Combobox(birthframe, height=listheight,
width=listwidth, values=years)

genframe = Frame(root, background=wincolor)
genvar = IntVar()
genam = Label(genframe, text='Gender', anchor='e',
width=labwidth, background=wincolor)
genmale = Radiobutton(genframe, text='Male', variable=genvar,
value=1)
genfem = Radiobutton(genframe, text='Female',
variable=genvar, value=2)

counframe = Frame(root, background=wincolor)
couname = Label(counframe, text='Country', width=labwidth,
anchor='e', background=wincolor)

```

```

counbox = ttk.Combobox(counframe, values=['USA', 'Russia',
'Poland', 'Sweden', 'Korea', 'China'])

mailframe = Frame(root, background=wincolor)
mailname = Label(mailframe, text='E-mail', width=labwidth,
anchor='e', background=wincolor)
mailtext = Entry(mailframe, width=50, font='Arial 14',
background=wincolor)

phoneframe = Frame(root, background=wincolor)
phname = Label(phoneframe, text='Phone', width=labwidth,
anchor='e', background=wincolor)
phonetext = Entry(phoneframe, width=50, font='Arial 14',
background=wincolor)

passframe = Frame(root, background=wincolor)
passname = Label(passframe, text='Password', width=labwidth,
anchor='e', background=wincolor)
passtext = Entry(passframe, width=50, font='Arial 14',
show=randpic, background=wincolor)

conframe = Frame(root, background=wincolor)
confname = Label(conframe, text='Confirm password',
width=labwidth, anchor='e', background=wincolor)
confcontext = Entry(conframe, width=50, font='Arial 14',
show=randpic, background=wincolor)

agreed = IntVar()
agrcheck = Checkbutton(root, text=u'I agree to the Terms of
Use', variable=agreed, onvalue=1, offvalue=0,
background=wincolor)

def submitclick():
    confirmations = 0
    # AGREEMENT
    if agreed.get() != 1:
        messagebox.showinfo("Information", "You haven't
agreed to the Terms of Use. \nWe can't use your info")
        root.grab_set()
    else:
        print('\nSYSTEM REPORT: AGREEMENT \nUser agreed to
the Terms of Use')
        confirmations += 1

```

```

        gotinfo = [nametext.get() if nametext.get() != "" else 0,
# NAME
                    lasnametext.get() if lasnametext.get() != ""
else 0, # LAST NAME
                    scrnametext.get() if scrnametext.get() != ""
else 0, # SCR NAME
                    daylist.get()+ ' ' +monthlist.get()+
'+yearlist.get() if #
BIRTH DATE
                    daylist.get() != "" and monthlist.get() != ""
and yearlist.get != "" else 0,
                    'Female' if genvar.get() == 2 else 'Male' if
genvar.get() == 1 else 0, # GENDER
                    counbox.get() if counbox.get() != "" else 0,
# COUNTRY
                    mailtext.get() if mailtext.get() != "" else 0,
# E-MAIL
                    phonetext.get() if phonetext.get() != "" else
0, # PHONE NUMBER
                    passtext.get() if passtext.get() ==
conftext.get() and
                    len(passtext.get()) >= 6 else 'PASSERROR' if
passtext != "" else 0] # PASSWORD
        if confirmations == 1:
            for i in gotinfo:
                if i == 0:
                    print('...but some fields appeared to be not
filled')
                    messagebox.showinfo("Information", "You
haven't filled the necessary fields. \nPlease, try again")
                    break
            else:
                print('\nSYSTEM REPORT: FIELDS\nEverything is
filled')
                if gotinfo[-1] == 'PASSERROR':
                    print('...but there is an error with the
password')
                    messagebox.showinfo("Information",
"Confirmation of password hasn't passed. Please, try again" +
"\nNote: Your password
must be more than 6 characters")
                else:
                    print('\nSYSTEM REPORT: PASSWORD\nPassword
confirmed')
                    confirmations += 1

```

```

        info = {'Name': nametext.get(), 'Last name':
lasnametext.get(), 'Screen name': scrnametext.get(),
                'Birth date': daylist.get()+
'+monthlist.get()+' '+yearlist.get(),
                'Gender': 'Female' if genvar.get() == 2 else
'Male',
                'Country': counbox.get(), 'Email':
mailtext.get(), 'Phone': phonetext.get(),
                'Password': passtext.get()}

    # RECHECK
    if confirmations == 2:
        if messagebox.askokcancel("Question", "Let's recheck
info given by you" +
                                "\nYou're "+info['Name']+"
"+info['Last name']+"." +
                                "\nYou were born in
"+info['Birth date']+"." +
                                "\n\nPAY ATTENTION to these
points:" +
                                "\nE-mail: "+info['Email']
+
                                "\nPhone number:
"+info['Phone']):
            print("\nUSER REGISTERED")
            root.destroy()
            root.quit()

submit = Button(root, text='Submit', width=7, height=2,
bg='#009900', fg='#005500', command=submitclick)
cancel = Button(root, text='Cancel', width=7, height=2,
bg='#990000', fg='#550000')

namframe.pack(fill=BOTH)
firnam.grid(column=0, row=0, padx=15)
nametext.grid(column=1, row=0)

lasnamframe.pack(fill=BOTH)
lasnam.grid(column=0, row=0, padx=15)
lasnametext.grid(column=1, row=0)

scrnamframe.pack(fill=BOTH)
scrnam.grid(column=0, row=0, padx=15)
scrnametext.grid(column=1, row=0)

```

```
birthframe.pack(fill=BOTH)
birthlab.grid(column=0, row=0, padx=15)
monthlist.grid(column=1, row=0)
daylist.grid(column=2, row=0)
yearlist.grid(column=3, row=0)

genframe.pack(fill=BOTH)
genam.grid(column=0, row=0, padx=15)
genmale.grid(column=1, row=0)
genfem.grid(column=2, row=0)

counframe.pack(fill=BOTH)
couname.grid(column=0, row=0, padx=15)
counbox.grid(column=1, row=0)
counbox.current(0)

mailframe.pack(fill=BOTH)
mailname.grid(column=0, row=0, padx=15)
mailtext.grid(column=1, row=0)

phoneframe.pack(fill=BOTH)
phname.grid(column=0, row=0, padx=15)
phonetext.grid(column=1, row=0)

passframe.pack(fill=BOTH)
passname.grid(column=0, row=0, padx=15)
passtext.grid(column=1, row=0)

conframe.pack(fill='both')
confname.grid(column=0, row=0, padx=15)
confcontext.grid(column=1, row=0)

agrcheck.place(x=170, y=280)
submit.place(x=375, y=310)
cancel.place(x=475, y=310)

root.mainloop()
```

Протокол работы программы:

Sign Up

First Name

Last Name

Screen Name

Date of Birth

Gender

Country

E-mail

Phone

Password

Confirm password

☐ Male

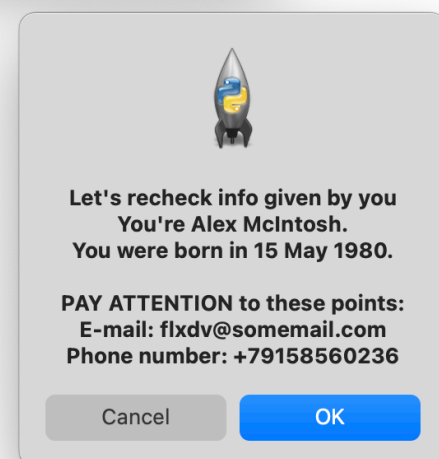
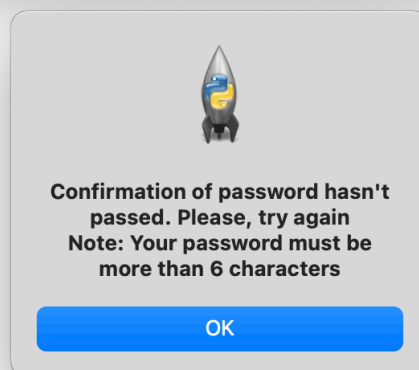
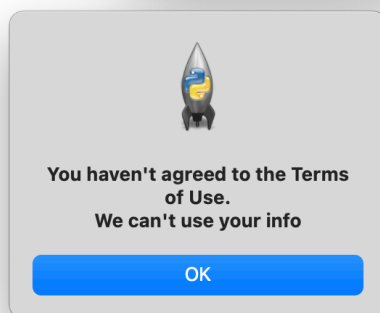
☐ Female

USA

☐ I agree to the Terms of Use

Submit

Cancel



Задача №2.

Постановка задачи: Разработать программу с применением пакета tk, взяв в качестве условия одну любую задачу из ПЗ №№ 3 – 8. (ПЗ-4.1)

Текст программы:

```
from tkinter import *

def calculate():
    try:
        n = int(entry.get())
        k = 0
        s = 0
        while k <= n:
```

```

        s += (n + k) * (n + k)
        k += 1
    answer['text'] = 'n^2 + (n+1)^2 + ... + 2n^2 = ' + str(s)
except ValueError:
    answer['text'] = 'Ошибка ввода данных. \nВведите целое число'

root = Tk()
root.title('n^2 + (n+1)^2 + (n+2)^2 ... 2n^2')
root.geometry('400x400')
root.minsize(width=400, height=400)
root.maxsize(width=500, height=100)

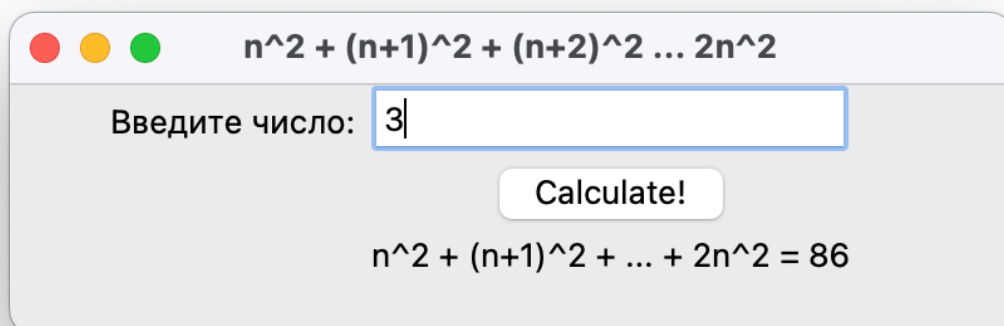
label = Label(root, text='Введите число:', anchor='e', width=15)
entry = Entry(root, width=20)
button = Button(root, text='Calculate!', command=calculate)
answer = Label(root)

label.grid(column=0, row=0)
entry.grid(column=1, row=0)
button.grid(column=1, row=1)
answer.grid(column=1, row=2)

root.mainloop()

```

Протокол работы программы:



Вывод: в ходе разработки закрепились усвоенные знания, понятия, алгоритмы, основные принципы составления программ, а также приобретены навыки составления программ с GUI Tkinter в IDE PyCharm Community. Выполнены разработка программы, отладка, тестирование и оптимизация кода. При разработке были использованы Tk(), множество виджетов Tkinter и методы-«упаковщики» grid() и pack(). Программы вместе с отчетом опубликованы в моем репозитории Github.