Худоян Мария вариант № 29

**Практическое занятие № 12**

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

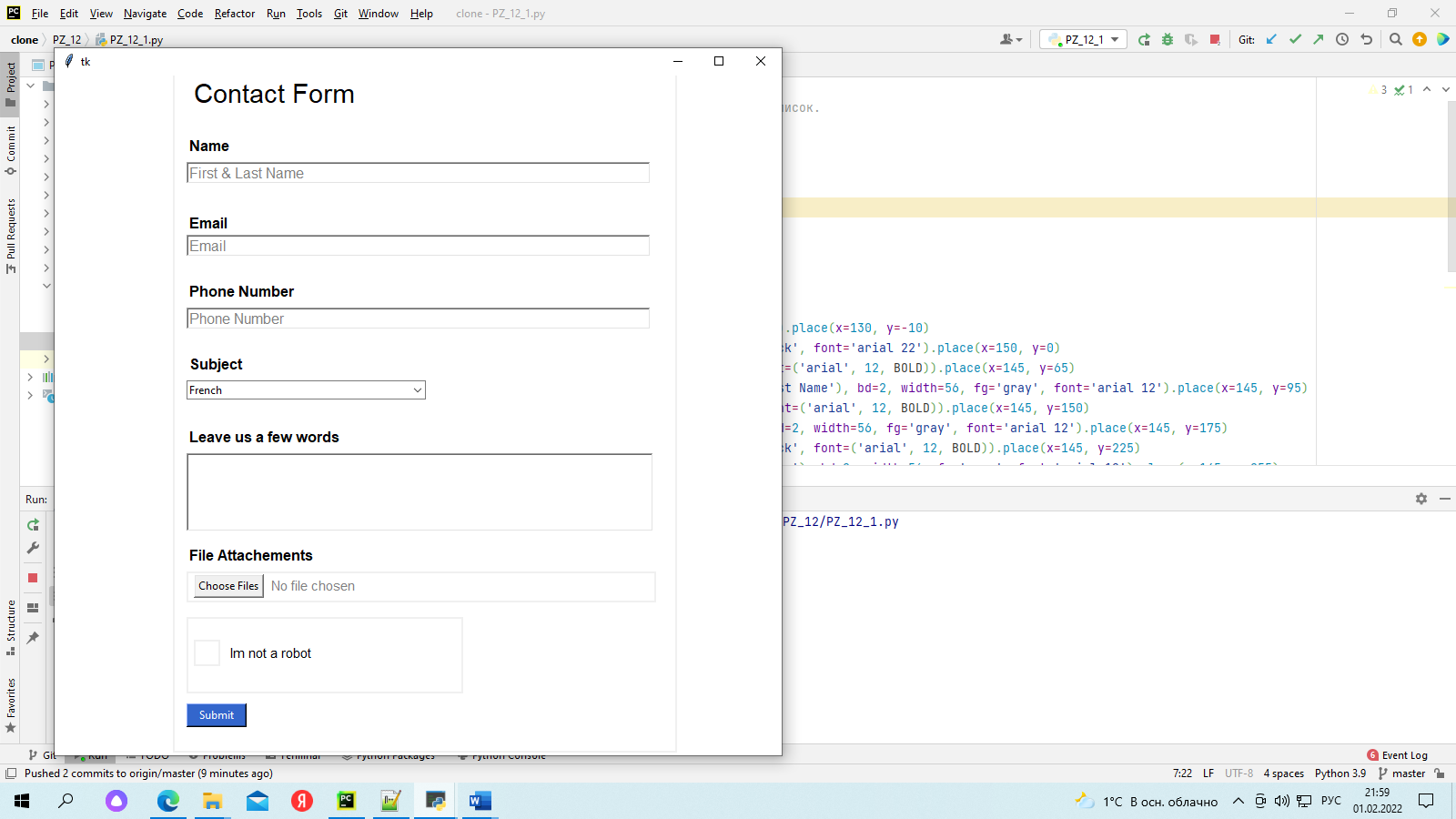
**Цели:** закрепить усвоенные знания, понятия, алгоритмы, основные принципы составления программ, приобрести навыки составления программ с с использованием GUI Tkinter в IDE PyCharm Community .

**Постановка для 1 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 (см. таблицу 1). <https://files.codegrape.com/62259/screenshot2.png>

**Текст программы:**

# -\*- coding: utf8 -\*-  
# Здесь с помощью combobox делаем выпадающий список.  
# А BOLD делает шрифт жирным.  
  
  
from tkinter import \*  
from tkinter.font import BOLD  
from tkinter.ttk import Combobox  
  
root = Tk()  
root.geometry('799x747')  
root['bg'] = 'white'  
Canvas(root, bg='white', height=750, width=550).place(x=130, y=-10)  
Label(text='Contact Form', bg='white', fg='black', font='arial 22').place(x=150, y=0)  
Label(text='Name', bg='white', fg='black', font=('arial', 12, BOLD)).place(x=145, y=65)  
Entry(textvariable=StringVar(value='First & Last Name'), bd=2, width=56, fg='gray', font='arial 12').place(x=145, y=95)  
Label(text='Email', bg='white', fg='black', font=('arial', 12, BOLD)).place(x=145, y=150)  
Entry(textvariable=StringVar(value='Email'), bd=2, width=56, fg='gray', font='arial 12').place(x=145, y=175)  
Label(text='Phone Number', bg='white', fg='black', font=('arial', 12, BOLD)).place(x=145, y=225)  
Entry(textvariable=StringVar(value='Phone Number'), bd=2, width=56, fg='gray', font='arial 12').place(x=145, y=255)  
Label(text='Subject', bg='white', fg='black', font=('arial', 12, BOLD)).place(x=145, y=305)  
combo = Combobox(root, width=40, values=['French','English','Russian','Spanish'])  
combo.current(0)  
combo.place(x=145, y=335)  
Label(text='Leave us a few words', bg='white', fg='black', font=('arial', 12, BOLD)).place(x=145, y=385)  
Entry(bd=2, width=13, fg='gray', font='arial 53').place(x=145, y=415)  
Label(text='File Attachements', bg='white', fg='black', font=('arial', 12, BOLD)).place(x=145, y=515)  
Canvas(root, bg='#fff', height=30, width=512).place(x=145, y=545)  
Button(text="Choose Files").place(x=153, y=548)  
Label(text='No file chosen', bg='white', fg='gray', font='arial 11').place(x=235, y=549)  
Canvas(root, bg='#fff', height=80, width=300).place(x=145, y=595)  
Canvas(root, bg='#fff', height=25, width=25).place(x=153, y=620)  
Label(text='Im not a robot', bg='white', fg='black', font='arial 11').place(x=190, y=623)  
Button(text="Submit", width=8, fg='white', bg='#36c').place(x=145, y=690)  
root.mainloop()

**Протокол работы программы:**

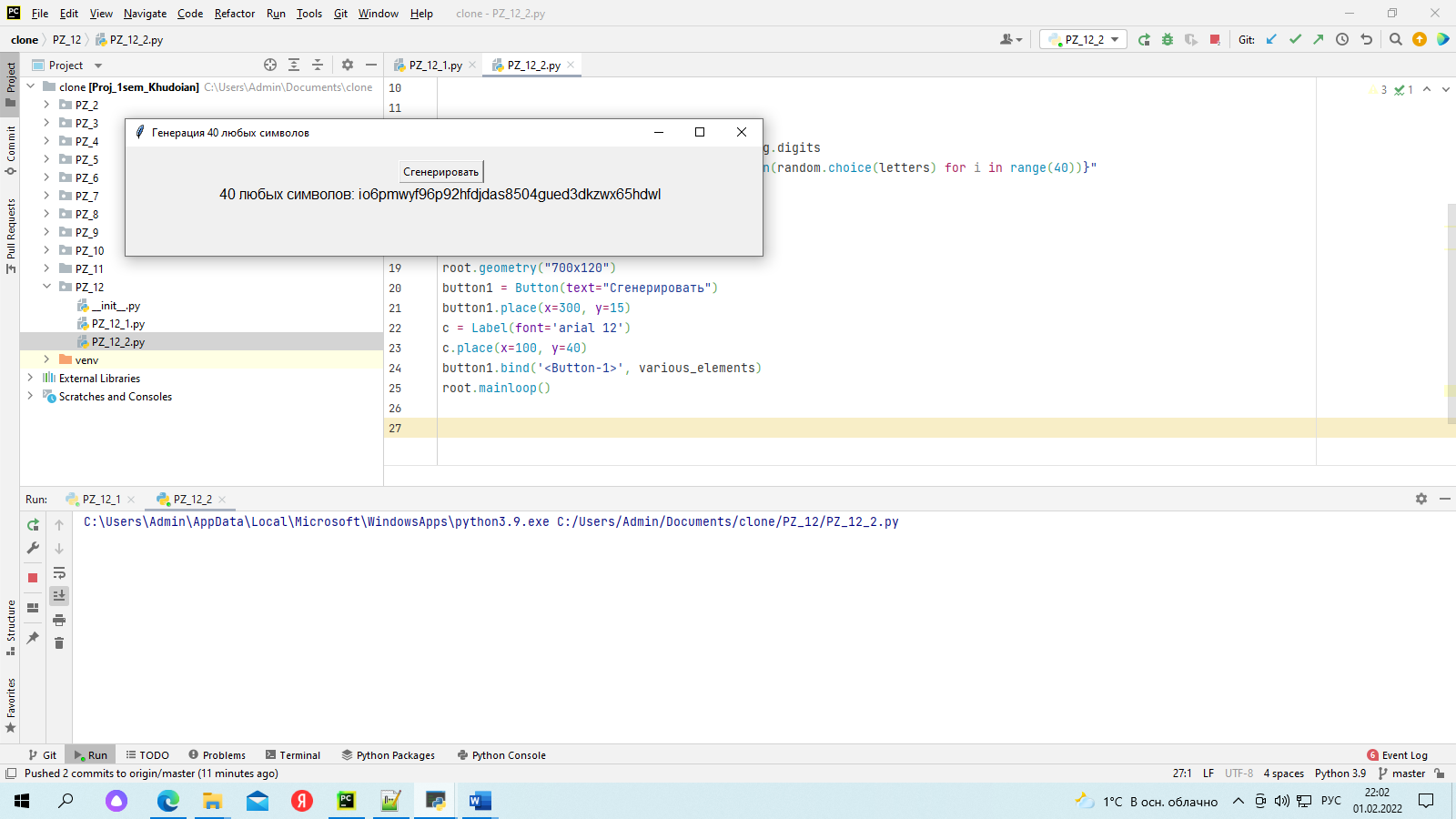


**Постановка для 2 задачи. .** Разработать программу с применением пакета tk, взяв в качестве условия одну любую задачу из ПЗ №№ 3 – 8. Я взяла задачу 1 из 5 практической.

**Текст программы:**

# -\*- coding: utf8 -\*-.  
# Задача 1 из 5 практической работы 29 варианта.  
# Здесь мы импортируем рандом для того чтобы вывести случайные 40 символов.  
# Ну и создаем окошко и кнопку с помощью root и button.  
  
  
from tkinter import \*  
import random  
import string  
  
  
def various\_elements(vvod):  
 letters = string.ascii\_lowercase + string.digits  
 c['text'] = f"40 любых символов: {''.join(random.choice(letters) for i in range(40))}"  
  
  
root = Tk()  
root.title('Генерация 40 любых символов')  
root.geometry("700x120")  
button1 = Button(text="Сгенерировать")  
button1.place(x=300, y=15)  
c = Label(font='arial 12')  
c.place(x=100, y=40)  
button1.bind('<Button-1>', various\_elements)  
root.mainloop()

**Протокол работы программы:**



**Вывод:**

В процессе выполнения практического занятия выработала навыки составления программ с использованием GUI Tkinter в IDE PyCharm Community . Были использованы языковые конструкции for in range, root, button, label, space, import, def, join, choice, canvas, entry, combo, BOLD . Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на GitHub.