

Практическое занятие № 12

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

Постановка первой 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.

HTML5 - Формы
localhost/web/html5_test.html

Форма заявки на работу в зоопарке

*Пожалуйста, заполните форму. Обязательные поля помечены **

Контактная информация
Имя *
Телефон
Email *

Персональная информация
Возраст *
Пол: Женцина
Перечислите личные качества

ВЫБЕРИТЕ ВАШИХ ЛЮБИМЫХ ЖИВОТНЫХ
☐ Зебра ☐ Кошак ☐ Анаконда ☐ Человек
☐ Слон ☐ Антилопа ☐ Голубь ☐ Краб

Отправить информацию

Текст первой программы:

```
# Вариант 2.  
# В соответствии с номером варианта перейти по ссылке  
(https://professorweb.ru/my/html/html5/level2/files/img46023.jpg)  
# на прототип. Реализовать его в IDE PyCharm Community с применением пакета  
tk. Получить интерфейс максимально  
# приближенный к оригиналу.  
from tkinter import *  
from tkinter.ttk import Combobox  
  
def close():  
    root.destroy()  
    root.quit()  
  
root = Tk()  
root.title("Формы")
```

```

root.geometry('700x650')

fr1 = Frame(bg='white', bd=2)
fr2 = Frame(bg='white', bd=2)
fr3 = Frame(bg='white', bd=2)

fr1.place(x=20, y=100, width=660, height=130)
fr2.place(x=20, y=260, width=660, height=185)
fr3.place(x=20, y=475, width=660, height=100)

lb0 = Label(text='Форма заявки на работу в зоопарке', font=('Times', 30))
lb1 = Label(text='Пожалуйста, заполните форму. Обязательные поля помечены',
font=('Times', 12, 'italic'))
lb2 = Label(text='Контактная информация', font=('Times', 13), bg='white')
lb3 = Label(fr1, text='Имя', font=('Times', 12), bg='white')
lb4 = Label(fr1, text='Телефон', font=('Times', 12), bg='white')
lb5 = Label(fr1, text='Email', font=('Times', 12), bg='white')
lb6 = Label(text='Персональная информация', font=('Times', 13), bg='white')
lb7 = Label(fr2, text='Возраст', font=('Times', 12), bg='white')
lb8 = Label(fr2, text='Пол', font=('Times', 12), bg='white')
lb9 = Label(fr2, text='Перечислите', font=('Times', 12), bg='white')
lb10 = Label(fr2, text='личные', font=('Times', 12), bg='white')
lb11 = Label(fr2, text='качества', font=('Times', 12), bg='white')
lb12 = Label(text='Выберите ваших любимых животных', font=('Times', 13),
bg='white')
lb13 = Label(text='*', fg='red')
lb14 = Label(fr1, text='*', bg='white', fg='red')
lb15 = Label(fr1, text='*', bg='white', fg='red')
lb16 = Label(fr2, text='*', bg='white', fg='red')

lb0.place(x=40, y=0) # Заголовок
lb1.place(x=40, y=50) # Подзаголовок
lb2.place(x=40, y=87) # Контактная информация
lb3.place(x=25, y=20) # Имя
lb4.place(x=25, y=50) # Телефон
lb5.place(x=25, y=80) # Email
lb6.place(x=40, y=248) # Персональная информация
lb7.place(x=25, y=20) # Возраст
lb8.place(x=25, y=50) # Пол
lb9.place(x=25, y=80) # Личные качества
lb10.place(x=25, y=100) # Личные качества
lb11.place(x=25, y=120) # Личные качества
lb12.place(x=40, y=464) # Животные
lb13.place(x=475, y=52) # *
lb14.place(x=56, y=20) # *
lb15.place(x=61, y=78) # *
lb16.place(x=81, y=20) # *

ent1 = Entry(fr1, bd=2, width=40)
ent2 = Entry(fr1, bd=2, width=40)
ent3 = Entry(fr1, bd=2, width=40)
ent4 = Entry(fr2, bd=2, width=40)

ent1.place(x=150, y=20)
ent2.place(x=150, y=50)
ent3.place(x=150, y=80)
ent4.place(x=150, y=20)

list1 = Combobox(fr2, height=0, width=37)
list1['values'] = ('Мужчина', 'Женщина')
list1.current(1)
list1.place(x=150, y=50)

tx1 = Text(fr2, bd=2, height=5, width=30)

```

```
tx1.place(x=150, y=80)

chk1 = Checkbutton(fr3, text='Зебра', font=('Times', 12), bg='white')
chk2 = Checkbutton(fr3, text='Слон', font=('Times', 12), bg='white')
chk3 = Checkbutton(fr3, text='Кошак', font=('Times', 12), bg='white')
chk4 = Checkbutton(fr3, text='Антилопа', font=('Times', 12), bg='white')
chk5 = Checkbutton(fr3, text='Анаконда', font=('Times', 12), bg='white')
chk6 = Checkbutton(fr3, text='Голубь', font=('Times', 12), bg='white')
chk7 = Checkbutton(fr3, text='Человек', font=('Times', 12), bg='white')
chk8 = Checkbutton(fr3, text='Краб', font=('Times', 12), bg='white')

chk1.place(x=25, y=25)
chk2.place(x=25, y=60)
chk3.place(x=170, y=25)
chk4.place(x=170, y=60)
chk5.place(x=315, y=25)
chk6.place(x=315, y=60)
chk7.place(x=460, y=25)
chk8.place(x=460, y=60)

bt1 = Button(text='Отправить информацию', width=23, height=2, font=('Arial',
10))
bt1.place(x=20, y=590)

root.mainloop()
```

Протокол работы первой программы:

Постановка второй задачи: составить функцию, которая напечатает сорок любых символов. Реализовать её в IDE PyCharm Community с применением пакета tk.

Текст второй программы:

```

# Вариант 2
# Составить функцию, которая напечатает сорок любых символов.
import random
import string
from tkinter import *
from tkinter.ttk import Combobox

rand_string = 0

def close():
    root.destroy()
    root.quit()

def generate(length): # Функция, генерирующая случайные символы
    global rand_string
    letters_and_digits = string.ascii_letters + string.digits
    rand_string = ''.join(random.sample(letters_and_digits, length))
    str(rand_string)
    lb0 = Label(text='')
    lb0.place(x=400, y=231, width=800, anchor='center')
    lb3 = Label(text=rand_string, font=('Times', 17))
    lb3.place(x=400, y=230, anchor='center')

root = Tk()
root.title("Генератор")
root.geometry('800x300')

lb1 = Label(text='Введите количество символов, которое нужно сгенерировать\n(не больше 62): ', font=('Times', 15))
lb2 = Label(text='Результат:', font=('Times', 15))

lb1.place(x=400, y=15, anchor='center') # Ввод
lb2.place(x=400, y=180, anchor='center') # Результат

ent1 = Entry(bd=2, width=24)
ent1.place(x=400, y=70, anchor='center')

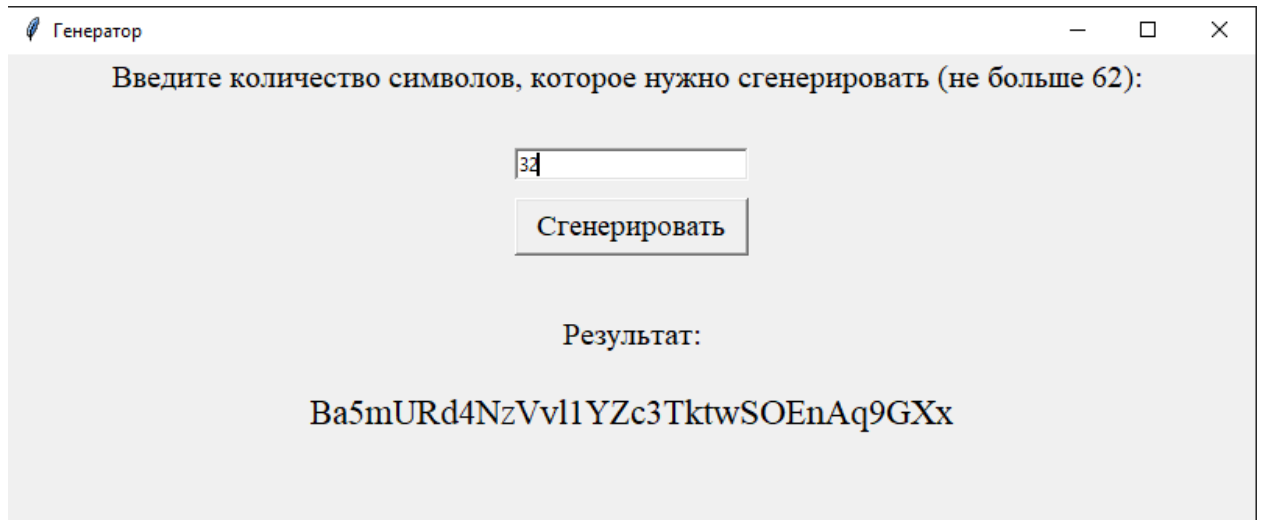
def btn_clicked():
    x = int(ent1.get())
    generate(x)

bt1 = Button(text='Сгенерировать', width=14, font=('Times', 14),
             command=btn_clicked)
bt1.place(x=400, y=110, anchor='center')

root.mainloop()

```

Протокол работы второй программы:



Вывод: в процессе выполнения практического занятия выработал навыки составления программ использованием GUI Tkinter в IDE PyCharm Community.

Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на GitHub.