

Практическое занятие №15

Тема: Составление программ для работы с использованием GUI Tkinter в IDE PyCharm Community

Цель: Цели практического занятия: 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 (см. таблицу 1).

Текст программы:

```
import tkinter as tk
from tkinter import ttk

win = tk.Tk()
win.geometry('800x800')

lab = tk.Label(win, text="Форма заявки на работу в зоопарке",
               font="Times New Roman" 22 bold',
               anchor='w')
lab.grid(row=0, column=0, padx=15, pady=[15, 0], sticky='we')

lab1 = tk.Label(win, text="Пожалуйста, заполните форму. Обязательные поля помечены *",
               font="Times New Roman" 14 italic', anchor='w').grid(row=1, column=0, padx=15, pady=0,
               sticky='we')

lab_contact = tk.Label(win, text="Контактная информация", font=("Times New Roman", 12),
               anchor='w')
lab_contact.grid(row=2, column=0, padx=30, pady= [15, 0], sticky='we')

frame_contact = tk.Frame(win, borderwidth=4, relief='ridge')
frame_contact.grid(row=4, column=0, padx=15, pady=15, sticky='wens')

lblname = tk.Label(frame_contact, text="Имя *", font=("Times New Roman", 12), width=15, anchor='w')
lblname.grid(row=2, column=0, padx=5, pady=5)
entry = tk.Entry(frame_contact)
entry.grid(row=2, column=1)

lbl_telefon = tk.Label(frame_contact, text="Телефон", font=("Times New Roman", 12), width=15,
               anchor='w')
lbl_telefon.grid(row=3, column=0, padx=5, pady=5)
tl_entry = tk.Entry(frame_contact)
tl_entry.grid(row=3, column=1)
```

```

lbl_Email = tk.Label(frame_contact, text="Email *", font=("Times New Roman", 12), width=15,
anchor='w')
lbl_Email.grid(row=4, column=0, padx=5, pady=5)
em_entry = tk.Entry(frame_contact)
em_entry.grid(row=4, column=1)

lab_percon_inf = tk.Label(win, text="Персональная информация", font=("Times New Roman", 12),
anchor='w')
lab_percon_inf.grid(row=5, column=0, padx=30, pady= [15, 0], sticky='we')

frame_contact_2 = tk.Frame(win, borderwidth=4, relief='ridge')
frame_contact_2.grid(row=7, column=0, padx=15, pady=0, sticky='wens')

lbl_age = tk.Label(frame_contact_2, text="Возраст *", font=("Times New Roman", 12), width=15,
anchor='w')
lbl_age.grid(row=8, column=0, padx=5, pady=5)
entry = tk.Entry(frame_contact_2)
entry.grid(row=8, column=1)

#
lblgender = tk.Label(frame_contact_2, text="Пол", font=("Times New Roman", 12), width=15,
anchor='w')
lblgender.grid(row=9, column=0, padx=5, pady=5)
genders = ["Женщина", "Мужчина"]
genders_var = tk.StringVar(value=genders[1])
combobox = ttk.Combobox(frame_contact_2,
textvariable=genders_var,
values=genders)
combobox.grid(row=9, column=1, padx=5, pady=5)

lbl_pers_qua = tk.Label(frame_contact_2, text="Перечислите \n личные \n качества ", font=("Times
New Roman", 12), width=15, anchor='w')
lbl_pers_qua.grid(row=10, column=0, padx=5, pady=5)

st = tk.Text(frame_contact_2, width=15, height=5)
st.grid(row=10, column=1)

lab_like_animal = tk.Label(win, text="Выберите ваших любимых животных", font=("Times New
Roman", 12), anchor='w')
lab_like_animal.grid(row=11, column=0, padx=30, pady= [15, 0], sticky='we')

frame_contact_3 = tk.Frame(win, borderwidth=4, relief='ridge')
frame_contact_3.grid(row=12, column=0, padx=15, pady=0, sticky='wens')

enabled = tk.IntVar()

```

```
options = ["Зебра", "Кошак", "Анаконда", "Человек", "Слон", "Антилопа", "Голубь", "Краб"]

checkboxes = {}
r = 0
c = 0
for option in options:
    var = tk.BooleanVar()
    chk = tk.Checkbutton(frame_contact_3, text=option, variable=var, font=("Times New Roman",
12), width=15, anchor='w')
    chk.grid(row=r, column=c)
    checkboxes[option] = var
    c += 1
    print(r, c)
    if c == 4:
        r += 1
        c = 0

button = tk.Button(win, text="Отправить информацию", anchor='w')
button.grid(padx=15, pady=5, row=15, column=0, sticky='w')

win.mainloop()
```

Протокол работы программы:

Форма заявки на работу в зоопарке

Пожалуйста, заполните форму. Обязательные поля помечены *

Контактная информация

Имя *

Телефон

Email *

Персональная информация

Возраст *

Пол

Мужчина ▼

Перечислите
личные
качества

Выберите ваших любимых животных

- | | | | |
|--------------------------------|-----------------------------------|-----------------------------------|----------------------------------|
| <input type="checkbox"/> Зебра | <input type="checkbox"/> Кошак | <input type="checkbox"/> Анаконда | <input type="checkbox"/> Человек |
| <input type="checkbox"/> Слон | <input type="checkbox"/> Антилопа | <input type="checkbox"/> Голубь | <input type="checkbox"/> Краб |

Отправить информацию

Постановка задачи №2

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

Текст программы:

```
import tkinter as tk
win = tk.Tk()
win.geometry('600x400')
lab = tk.Label(win, text="Проверка на четность", font=("Times New Roman", 22), justify=tk.RIGHT)
lab.pack()

proverka_lb = tk.Label(win, text="Введите число", font=("Times New Roman", 12), justify=tk.RIGHT)
proverka_lb.pack()
entry = tk.Entry()
entry.pack()
```

```

label_res = tk.Label(win, text="", font=("Times New Roman", 12,), foreground="red", justify=tk.RIGHT)
label_res.pack()
def check():
    chislo = int(entry.get())
    if chislo % 2 == 0:
        label_res.config(text='Четное')
    else:
        label_res.config(text="Нечетное")

button = tk.Button(win, text="Проверить", command=check)
button.pack()

win.mainloop()

```

Протокол работы программы:

Вторая задача

```

import tkinter as tk
win = tk.Tk()
win.geometry('600x400')
lab = tk.Label(win, text="Определение процентной ставки", font=("Times New Roman", 22),
justify=tk.RIGHT)
lab.pack()

proverka_lb = tk.Label(win, text="Введите сумму", font=("Times New Roman", 12), justify=tk.RIGHT)
proverka_lb.pack()
entry = tk.Entry()
entry.pack()

label_res = tk.Label(win, text="", font=("Times New Roman", 12,), foreground="red", justify=tk.RIGHT)
label_res.pack()

```

```
def vklad():
    vklad = int(entry.get())
    if vklad <= 50000:
        label_res.config(text="Процент вносимого вклада равен 4 %")
    elif vklad >= 50000 and vklad <= 1000000 :
        label_res.config(text="Процент вносимого вклада равен 5 %")
    elif vklad >= 100000 and vklad <= 1500000 :
        label_res.config(text="Процент вносимого вклада равен 6 %")
    elif vklad >= 150000 and vklad <= 2000000 :
        label_res.config(text="Процент вносимого вклада равен 7 %")
    else:
        label_res.config(text="Максимальный вклад равен 200000")

button = tk.Button(win, text="Определить", command=vklad)
button.pack()

win.mainloop()
```

Протокол работы программы:

Определение процентной ставки

Введите сумму

150000

Процент вносимого вклада равен 5 %

Определить

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:

```
import os
```

```
import subprocess, sys

opener = "open" if sys.platform == "darwin" else "xdg-open"

# Определение базового пути к проекту
base = '/home/student/Документы/Архипов ИС-27/Prog_Архипов/'
basepath = base + 'PZ_11'
test = base + 'test'
test1 = base + 'test/test1'

# Переход в каталог PZ11 и вывод списка файлов
os.chdir(basepath)
print('Текущая директория:', os.getcwd())
print('Список файлов в PZ11:', os.listdir(basepath))

# Создание папок test и test1
os.makedirs(test, exist_ok=True)
os.makedirs(test1, exist_ok=True)

# Перемещение файлов из PZ6 и PZ7
# os.replace(base + 'PZ_6/PZ_6.1.py', test + '/PZ_6.1.py')
# os.replace(base + 'PZ_6/PZ_6.2.py', test + '/PZ_6.2.py')
# os.replace(base + 'PZ_7/PZ_7.1.py', test1 + '/test.txt')

# Вывод информации о размере файлов в папке test
list_dir = os.listdir(test)
print('Список файлов в test:', list_dir)
for i in list_dir:
    print(i)
    razmer = os.path.getsize(os.path.join(test, i))
    print(f'Размер {i}: {razmer} байт')

# Нахождение файла с самым коротким именем в PZ11
os.chdir(basepath)
print('Текущая директория:', os.getcwd())
pz_11_list = os.listdir(basepath)
shortest_file = min(pz_11_list, key=len)
print('Файл с самым коротким именем:', shortest_file)

# Запуск PDF-файла
os.chdir(basepath)
for filename in os.listdir(basepath):
    if filename.endswith(".pdf"):
        subprocess.call([opener, os.path.join(basepath, filename)])
        break

# Запуск PDF-файла
os.chdir(basepath)
for filename in os.listdir(basepath):
    if filename.endswith(".pdf"):
        pdf_path = os.path.join(basepath, filename)
        if os.path.exists(pdf_path):
```

```
subprocess.call([opener, pdf_path])
break

# Удаление файла test.txt
# os.remove(test1 + '\\test.txt')
```

Протокол работы программы:

/home/student/Документы/Prog_Архипов/bin/python /home/student/Документы/Архипов
ИС-27/Prog_Архипов/PZ_17/PZ_17_3.py
Текущая директория: /home/student/Документы/Архипов ИС-27/Prog_Архипов/PZ_11
Список файлов в PZ11: ['pr11_2.txt', 'PZ_11_2.py', 'test', 'pr11.1.txt', 'text18-2.txt', 'Отчет
PZ_11.pdf', 'PZ_11_1.py', 'pr11.txt']
Список файлов в test: ['test1']
test1
Размер test1: 4096 байт
Текущая директория: /home/student/Документы/Архипов ИС-27/Prog_Архипов/PZ_11
Файл с самым коротким именем: test
kf.service.services: KApplicationTrader: mimeType "x-scheme-handler/file" not found

Process finished with exit code 0

Вывод: в процессе выполнения практического занятия выработал навыки составления программ с использованием GUI Tkinter в IDE PyCharm Community..
Выполнены разработка кода, отладка, тестирование, оптимизация программного кода.
Готовые программные коды выложены на GitHub.