

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 (img46023.jpg (706×775) (professorweb.ru)).

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

Текст программы(1):

```
import tkinter as tk
from tkinter import ttk

root = tk.Tk()
root.title("Форма заявки приема на работу в зоопарке")

mainframe = ttk.Frame(root, padding="3 3 12 12")
mainframe.grid(column=0, row=0, sticky=(tk.N, tk.W, tk.E, tk.S))
mainframe.columnconfigure(0, weight=1)
mainframe.rowconfigure(0, weight=1)

contact_info = ttk.Labelframe(mainframe, text="Контактная информация")
contact_info.grid(column=0, row=0, sticky=(tk.W, tk.E))
ttk.Label(contact_info, text="Имя*").grid(column=1, row=1, sticky=tk.W)
name_entry = ttk.Entry(contact_info)
name_entry.grid(column=3, row=1)
ttk.Label(contact_info, text="Телефон").grid(column=1, row=2, sticky=tk.W)
phone_entry = ttk.Entry(contact_info)
phone_entry.grid(column=3, row=2)
ttk.Label(contact_info, text="Email*").grid(column=1, row=3, sticky=tk.W)
email_entry = ttk.Entry(contact_info)
email_entry.grid(column=3, row=3)

personal_info = ttk.Labelframe(mainframe, text="Персональная информация")
personal_info.grid(column=0, row=1, sticky=(tk.W, tk.E))
ttk.Label(personal_info, text="Возраст*").grid(column=1, row=1, sticky=tk.W)
age_entry = ttk.Entry(personal_info)
age_entry.grid(column=2, row=1, sticky=(tk.W, tk.E))
ttk.Label(personal_info, text="Пол").grid(column=1, row=2, sticky=tk.W)
gender_combobox = ttk.Combobox(personal_info, values=["Женщина", "Мужчина"])
gender_combobox.grid(column=2, row=2, sticky=(tk.W, tk.E))
ttk.Label(personal_info, text="Перечислите личные качества",
```

```

wraplength=140).grid(column=1, row=3, sticky=tk.W)
qualities_text = tk.Text(personal_info, width=20, height=5)
qualities_text.grid(column=2, row=3, sticky=(tk.W, tk.E))

fav_animals = ttk.Labelframe(mainframe, text="Выберите ваших любимых
животных")
fav_animals.grid(column=0, row=2, sticky=(tk.W, tk.E))
zebra_checkbox = ttk.Checkbutton(fav_animals, text="Зебра")
zebra_checkbox.grid(column=1, row=1, sticky=tk.W)
cat_checkbox = ttk.Checkbutton(fav_animals, text="Кошка")
cat_checkbox.grid(column=2, row=1, sticky=tk.W)
cat_checkbox = ttk.Checkbutton(fav_animals, text="Анаконда")
cat_checkbox.grid(column=3, row=1, sticky=tk.W)
cat_checkbox = ttk.Checkbutton(fav_animals, text="Человек")
cat_checkbox.grid(column=4, row=1, sticky=tk.W)
cat_checkbox = ttk.Checkbutton(fav_animals, text="Слон")
cat_checkbox.grid(column=1, row=2, sticky=tk.W)
cat_checkbox = ttk.Checkbutton(fav_animals, text="Антилопа")
cat_checkbox.grid(column=2, row=2, sticky=tk.W)
cat_checkbox = ttk.Checkbutton(fav_animals, text="Голубь")
cat_checkbox.grid(column=3, row=2, sticky=tk.W)
cat_checkbox = ttk.Checkbutton(fav_animals, text="Краб")
cat_checkbox.grid(column=4, row=2, sticky=tk.W)

# Добавьте остальные животные аналогичным образом

submit_button = ttk.Button(mainframe, text="Отправить информацию")
submit_button.grid(column=0, row=3)

style = ttk.Style()
style.configure('.', font=('Arial', 14))

name_entry.focus()

root.mainloop()

root.mainloop()

```

Протокол работы программы:

Process finished with exit code 0

Текст программы(2):

```
# Дано целое число А. Проверить истинность высказывания: «Число А является
# нечетным»
import tkinter as tk
from tkinter import ttk

def check_number():
    input_value = entry.get()
    try:
        number = int(input_value)
        if number % 2 == 0:
            result_label["text"] = "Число {} четное".format(number)
        else:
            result_label["text"] = "Число {} нечетное".format(number)
    except ValueError:
        result_label["text"] = "Неправильный формат числа"

root = tk.Tk()
root.title("Проверка на четность")
root.geometry('300x150')

input_label = ttk.Label(root, text="Введите целое число:")
input_label.pack()

entry = ttk.Entry(root)
entry.pack()

check_button = ttk.Button(root, text="Проверить", command=check_number)
check_button.pack()

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

style = ttk.Style()
style.configure('.', font=('Arial', 12))

root.mainloop()
```

Протокол работы программы:

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.