

Студент группы ИС-23 Климов Я.В.

Практическое занятие №17

Тема: Составление программ с ООП в IDE PyCharm Community.

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

Задача №1

Постановка 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).

Текст программы:

```
def validate_password(password) :  
    """Validate the password length."""  
    if len(password) < 8: # Проверяет длина пароля  
меньше 8 символов  
        return False # Если условие истинно функция  
возвращает False  
    return True # Если условие ложно функция  
возвращает True  
  
def validate_password_confirmation(password,  
confirmation):  
    """Validate the password confirmation."""  
    if password != confirmation: # Проверяет пароль  
        return False  
    return True
```

```
def submit_form():
    """Submit the form and display a message box."""
    first_name = first_name_entry.get()
    last_name = last_name_entry.get()
    email = email_entry.get()
    website = website_entry.get()
    password = password_entry.get()
    confirmation = confirmation_entry.get()

    if not first_name:
        messagebox.showerror("Mistake", "Please enter
your name.")
    elif not last_name:
        messagebox.showerror("Mistake", "Please enter
your last name.")
    elif not email:
        messagebox.showerror("Mistake", "Please enter
your email address.")
    elif not website:
        messagebox.showerror("Mistake", "Please enter
your website.")
    elif not password:
        messagebox.showerror("Mistake", "Please enter
your password.")
    elif not confirmation:
        messagebox.showerror("Mistake", "Please
confirm your password.")
    elif not validate_password(password):
        messagebox.showerror("Mistake", "The password
must be 8-10 characters long.")
    elif not validate_password_confirmation(password,
confirmation):
        messagebox.showerror("Mistake", "The passwords
don't match.")
    else:
        messagebox.showinfo("Success", "The form has
been submitted successfully.")
```

```
root = tk.Tk() #Создает основное окно приложения.
root.title("Contacts")

first_name_label = tk.Label(root, text="First Name ")
# Создает метку для поля имени
first_name_label.pack() # Добавляет метку в окно
first_name_entry = tk.Entry(root) # Создает поле
ввода для имени
first_name_entry.pack() # Добавляет поле ввода в окно

last_name_label = tk.Label(root, text="Last Name ")
last_name_label.pack()
last_name_entry = tk.Entry(root)
last_name_entry.pack()

email_label = tk.Label(root, text="Email")
email_label.pack()
email_entry = tk.Entry(root)
email_entry.pack()

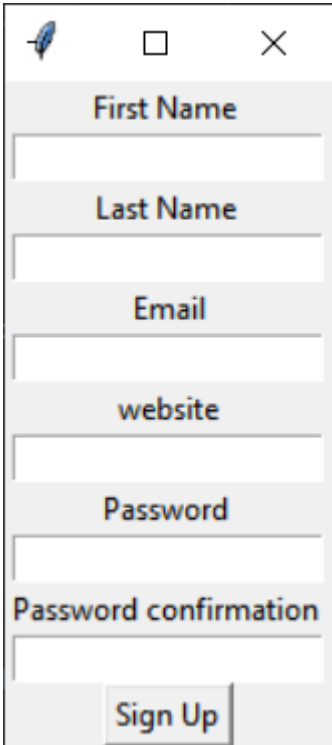
website_label = tk.Label(root, text="website")
website_label.pack()
website_entry = tk.Entry(root)
website_entry.pack()

password_label = tk.Label(root, text="Password ")
password_label.pack()
password_entry = tk.Entry(root, show="*")
password_entry.pack()

confirmation_label = tk.Label(root, text="Password
confirmation ")
confirmation_label.pack()
confirmation_entry = tk.Entry(root, show="*")
confirmation_entry.pack()
```

```
submit_button = tk.Button(root, text="Sign Up",  
command=submit_form) # Создает кнопку с текстом  
"Регистрация" которая вызывает функцию submit_form  
при клике  
submit_button.pack() # Добавляет кнопку в окно  
  
root.mainloop()
```

Протокол работы программы:



Задача №2

Постановка задачи.

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

Текст программы:

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

```
import tkinter as tk
from tkinter import messagebox

def check_number():
    try:
        number = int(entry.get())
        if 1 <= number <= 999: # Проверяем, находится
            # ли число в диапазоне от 1 до 999
            if number % 2 == 0: # Проверяем, является
                # ли число четным
                if 1 <= number < 10:
                    result = "Четное однозначное
число"
                elif 10 <= number < 100:
                    result = "Четное двузначное число"
                else:
                    result = "Четное трехзначное
число"
            else:
                if 1 <= number < 10:
                    result = "Нечетное однозначное
число"
                elif 10 <= number < 100:
                    result = "Нечетное двузначное
число"
                else:
                    result = "Нечетное трехзначное
число"
            else:
                result = "Число вне диапазона 1-999"
    except ValueError:
        result = "Введено не целое число"
    messagebox.showinfo("Результат", result)

root = tk.Tk() # Создаем новое окно tkinter
root.title("Проверка четности числа")
```

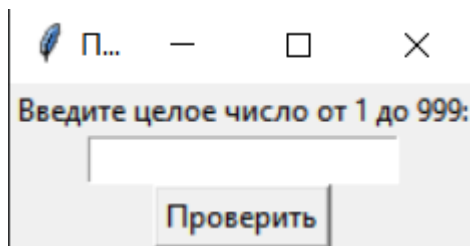
```
label = tk.Label(root, text="Введите целое число от 1 до 999:") # Создаем метку с указанным текстом
label.pack() # Добавляем метку в окно

entry = tk.Entry(root) # # Создаем поле ввода
entry.pack()

button = tk.Button(root, text="Проверить",
command=check_number) # Создаем кнопку с указанным
текстом и связываем ее с функцией check_number
button.pack()

root.mainloop()
```

Протокол работы программы:



Вывод: В процессе выполнения практического занятия выработал навыки Составление программ с использованием GUI Tkinter в IDE PyCharm Community, изучить возможности модуля OS.. Были использованы языковые конструкции `def, if, return False, return True, if not, elif not, else, text, show, command, import, as, from, try, except.`

Выполнены разработка кода, отладка, тестирование, оптимизация программного кода.

Готовые программные коды выложены на GitHub.

