

Практическое занятие №17

Тема: закрепил усвоенные знания, понятия, алгоритмы, основные принципы составления программ с использованием GUI Tkinter в IDE PyCharm Community. Изучил 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).  
# Разработать программу с применением пакета tk, взяв в качестве условия  
одну  
# любую задачу из ПЗ NoNo 2 - 9.
```

Текст программы:

```
import tkinter as tk  
from tkinter import ttk  
  
root = tk.Tk()  
root.title("Форма регистрации пользователя")  
  
frame = tk.Frame(root, borderwidth=2, relief="groove", padx=10, pady=10)  
frame.grid(row=0, column=0, padx=10, pady=10)  
  
title_label = tk.Label(frame, text="Форма регистрации пользователя",  
font=("Helvetica", 16, "bold"))  
title_label.grid(row=0, column=0, columnspan=2, pady=10)  
  
tk.Label(frame, text="Ваше имя:", font=("Helvetica", 12)).grid(row=1,  
column=0, pady=5, sticky="e")  
entry_name = tk.Entry(frame, width=40)  
entry_name.grid(row=1, column=1, pady=5)  
  
tk.Label(frame, text="Пароль:", font=("Helvetica", 12)).grid(row=2,  
column=0, pady=5, sticky="e")  
entry_password = tk.Entry(frame, show='*', width=40)  
entry_password.grid(row=2, column=1, pady=5)
```

```

tk.Label(frame, text="Возраст:", font=("Helvetica", 12)).grid(row=3,
column=0, pady=5, sticky="e")
entry_age = tk.Entry(frame, width=40)
entry_age.grid(row=3, column=1, pady=5)
tk.Label(frame, text="Пол:", font=("Helvetica", 12)).grid(row=4, column=0,
pady=5, sticky="e")
gender_frame = tk.Frame(frame)
gender_frame.grid(row=4, column=1, pady=5, sticky="w", padx=70)
gender_var = tk.StringVar(value="Мужской")
tk.Radiobutton(gender_frame, text="Мужской", variable=gender_var,
value="Мужской").pack(side="left")
tk.Radiobutton(gender_frame, text="Женский", variable=gender_var,
value="Женский").pack(side="left")

tk.Label(frame, text="Ваши увлечения:", font=("Helvetica", 12)).grid(row=5,
column=0, pady=5, sticky="e")
hobbies_frame = tk.Frame(frame)
hobbies_frame.grid(row=5, column=1, pady=5, sticky="w", padx=70)
music_var = tk.BooleanVar()
video_var = tk.BooleanVar()
drawing_var = tk.BooleanVar()
tk.Checkbutton(hobbies_frame, text="Музыка",
variable=music_var).pack(side="left")
tk.Checkbutton(hobbies_frame, text="Видео",
variable=video_var).pack(side="left")
tk.Checkbutton(hobbies_frame, text="Рисование",
variable=drawing_var).pack(side="left")
tk.Label(frame, text="Ваша страна:", font=("Helvetica", 12)).grid(row=6,
column=0, pady=5, sticky="e")
country_combobox = ttk.Combobox(frame, values=[""], width=37)
country_combobox.grid(row=6, column=1, pady=5)
country_combobox.current(0)

tk.Label(frame, text="Ваш город:", font=("Helvetica", 12)).grid(row=7,
column=0, pady=5, sticky="e")
city_combobox = ttk.Combobox(frame, values=[""], width=37)
city_combobox.grid(row=7, column=1, pady=5)
city_combobox.current(0)

tk.Label(frame, text="Кратко о себе:", font=("Helvetica", 12)).grid(row=8,
column=0, pady=5, sticky="e")
about_text = tk.Text(frame, height=4, width=30, wrap="word")
about_text.grid(row=8, column=1, pady=5)
about_text.insert("1.0", "Краткая информация о ваших увлечениях")
tk.Label(frame, text="Решите пример, запишите результат в поле ниже:",
font=("Helvetica", 12)).grid(row=9, column=1, pady=5, sticky="e")

captcha_entry = tk.Entry(frame, width=40)
captcha_entry.grid(row=10, column=1, pady=5, padx=20, sticky="w")

buttons_frame = tk.Frame(frame)

```

```

buttons_frame.grid(row=11, column=0, columnspan=2, pady=10)
cancel_button = tk.Button(buttons_frame, text="Отменить ввод", width=20)
cancel_button.pack(side="left", padx=5)
submit_button = tk.Button(buttons_frame, text="Данные подтверждаю",
width=20)
submit_button.pack(side="left", padx=5, pady=5)

root.mainloop()

```

Протокол работы программы:

The screenshot shows a graphical user interface for a user registration form. The window title is "Форма регистрации пользователя". The form contains the following elements:

- Title:** "Форма регистрации пользователя" (User Registration Form)
- Fields:**
 - "Ваше имя:" (Your name) with a text input field.
 - "Пароль:" (Password) with a text input field.
 - "Возраст:" (Age) with a text input field.
 - "Пол:" (Gender) with two radio buttons: "Мужской" (Male) and "Женский" (Female).
 - "Ваши увлечения:" (Your hobbies) with three checkboxes: "Музыка" (Music), "Видео" (Video), and "Рисование" (Drawing).
 - "Ваша страна:" (Your country) with a dropdown menu.
 - "Ваш город:" (Your city) with a dropdown menu.
 - "Кратко о себе:" (Briefly about yourself) with a text area containing the placeholder text "Краткая информация о ваших увлечениях" (Brief information about your hobbies).
- Instructions:** "Решите пример, запишите результат в поле ниже:" (Solve the example, write the result in the field below:)
- Result Field:** A text input field for the user to enter the result of the example.
- Buttons:**
 - "Отменить ввод" (Cancel input)
 - "Данные подтверждаю" (I confirm the data)

Задача 2

Постановка задачи.

Дано расстояние L в сантиметрах. Используя операцию деления нацело, найти количество полных метров в нем (1 метр =100 см).

Текст программы:

```
import tkinter as tk

def to_meters():
    l = float(entry_1.get())
    meters = l / 100
    whole_meters = int(meters)
    meters_result.config(text=f"Количество полных метров: {whole_meters}.")

root = tk.Tk()
root.title("Сантиметры в метры")

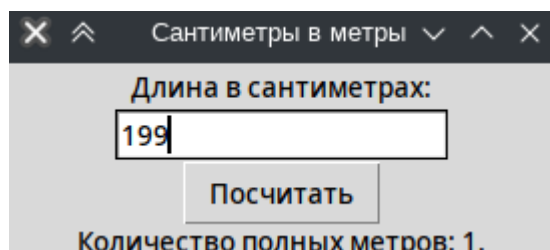
label_1 = tk.Label(root, text="Длина в сантиметрах:")
label_1.pack()
entry_1 = tk.Entry(root)
entry_1.pack()

button_calculate = tk.Button(root, text="Посчитать", command=to_meters)
button_calculate.pack()

meters_result = tk.Label(root, text="")
meters_result.pack()

root.mainloop()
```

Протокол работы программы:



Задача 3

Постановка задачи.

```
# Задание предполагает, что у студента есть проект с практическими работами
(NoNo 2-13),
# оформленный согласно требованиям. Все задания выполняются с использованием
модуля
# OS
```

Текст программы:

```
import os

def size(pathf):
    for i in os.listdir(pathf):
        if os.path.isfile(i):
            os.path.getsize(i)

pz_11 = os.listdir('PZ_11')
print(pz_11)

os.makedirs('../test/test1')

os.replace('6_4_1.py', '/home/student/Документы/PycharmProject/IS-27/Proj 1sem Vasilchenko/Rep_for_PZ/test/6_4_1.py')
os.replace('6_4_2.py', '/home/student/Документы/PycharmProject/IS-27/Proj 1sem Vasilchenko/Rep_for_PZ/test/6_4_2.py')
os.replace('7_4_1.py', '/home/student/Документы/PycharmProject/IS-27/Proj 1sem Vasilchenko/Rep_for_PZ/test/test1/7_4_1.py')
os.rename('/home/student/Документы/PycharmProject/IS-27/Proj 1sem Vasilchenko/Rep_for_PZ/test/test1/7_4_1.py',
'/home/student/Документы/PycharmProject/IS-27/Proj 1sem Vasilchenko/Rep_for_PZ/test/test1/test.txt')

for i in os.listdir():
    if os.path.isfile(i):
        os.path.getsize(i)

size('../test')
size('../test/test1')

short_file = '111111111111111111111111111111'
for i in pz_11:
    if len(i) < len(short_file):
        short file = i
```

```
print(short_file)
```

```
os.remove('/home/student/Документы/PycharmProject/IS-27/Proj 1sem  
Vasilchenko/Rep_for_PZ/test/test1/test.txt')
```

Протокол работы программы:

```
['Test-input.txt', 'Test-output.txt', '11_4_1.py', '11_4_2.py', 'text18-14.txt', 'LowText18-14.txt']  
11_4_1.py
```

Вывод: в процессе выполнения практического занятия выработал навыки составления программ с использованием GUI Tkinter в IDE PyCharm Community. Изучил возможности модуля OS.

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