

## Практическое занятие №17

**Тема:** Составление программ с использованием GUI Tkinter 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.

### 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).

### Постановка задачи 2.

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

### 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 1:

```
import tkinter as tk
from tkinter import ttk

# Create the main application window
root = tk.Tk()
root.title("Zoo Keeper Application Form")

# Add a title label
title_label = tk.Label(root, text="Zoo Keeper Application Form", font=("Helvetica", 16, "bold"))
title_label.grid(row=0, column=0, columnspan=2, pady=10)

# Add a description label
desc_label = tk.Label(root, text="Please complete the form. Mandatory fields are marked with a *",
font=("Helvetica", 10))
desc_label.grid(row=1, column=0, columnspan=2, pady=5)

# Add contact details frame
contact_frame = tk.LabelFrame(root, text="CONTACT DETAILS", padx=10, pady=10)
contact_frame.grid(row=2, column=0, columnspan=2, padx=10, pady=10, sticky="ew")

# Contact details labels and entries
tk.Label(contact_frame, text="Name*").grid(row=0, column=0, sticky="w")
tk.Entry(contact_frame).grid(row=0, column=1, padx=5, pady=5)

tk.Label(contact_frame, text="Telephone*").grid(row=1, column=0, sticky="w")
tk.Entry(contact_frame).grid(row=1, column=1, padx=5, pady=5)

tk.Label(contact_frame, text="Email*").grid(row=2, column=0, sticky="w")
```

```

tk.Entry(contact_frame).grid(row=2, column=1, padx=5, pady=5)

# Add personal information frame
personal_frame = tk.LabelFrame(root, text="PERSONAL INFORMATION", padx=10, pady=10)
personal_frame.grid(row=3, column=0, columnspan=2, padx=10, pady=10, sticky="ew")

# Personal information labels and entries
tk.Label(personal_frame, text="Age*").grid(row=0, column=0, sticky="w")
tk.Entry(personal_frame).grid(row=0, column=1, padx=5, pady=5)

tk.Label(personal_frame, text="Gender").grid(row=1, column=0, sticky="w")
gender_combobox = ttk.Combobox(personal_frame, values=["Female", "Male", "Other"])
gender_combobox.grid(row=1, column=1, padx=5, pady=5)

tk.Label(personal_frame, text="When did you first know you wanted to be a zoo-keeper?").grid(row=2,
column=0, sticky="w")
tk.Entry(personal_frame).grid(row=2, column=1, padx=5, pady=5)

# Add favorite animals frame
animals_frame = tk.LabelFrame(root, text="PICK YOUR FAVORITE ANIMALS", padx=10, pady=10)
animals_frame.grid(row=4, column=0, columnspan=2, padx=10, pady=10, sticky="ew")

# Favorite animals checkboxes
animals = ["Zebra", "Cat", "Anaconda", "Human", "Elephant", "Wildebeest", "Pigeon", "Crab"]
for idx, animal in enumerate(animals):
    tk.Checkbutton(animals_frame, text=animal).grid(row=idx // 2, column=idx % 2, sticky="w")

# Add submit button
submit_button = tk.Button(root, text="Submit Application")
submit_button.grid(row=5, column=0, columnspan=2, pady=10)

# Run the application
root.mainloop()

```

## Текст программы 2:

```

import tkinter as tk

def inverse_case(s):
    return s.swapcase()

def main():
    root = tk.Tk()
    root.title("Преобразователь обратного регистра")

    label = tk.Label(root, text="Введите строку:")
    label.pack()

    entry = tk.Entry(root, width=50)
    entry.pack()

    def convert():
        string = entry.get()
        result = inverse_case(string)
        result_label.config(text=result)

    button = tk.Button(root, text="Преобразовать", command=convert)
    button.pack()

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

    root.mainloop()

```

```
if __name__ == "__main__":  
    main()
```

### Текст программы 3:

```
import os
```

```
os.chdir("..")  
os.chdir("PZ_11")
```

```
files = os.listdir()  
print("Файлы в каталоге PZ_11:")  
for file in files:  
    if os.path.isfile(file):  
        print(file)
```

```
os.chdir("..")  
os.makedirs("test/test1", exist_ok=True)
```

```
os.rename("PZ_6/PZ_6.1.py", "test/PZ_6.1.py")  
os.rename("PZ_6/PZ_6.2.py", "test/PZ_6.2.py")
```

```
os.rename("PZ_7/PZ_7.1.py", "test/test1/test.txt")
```

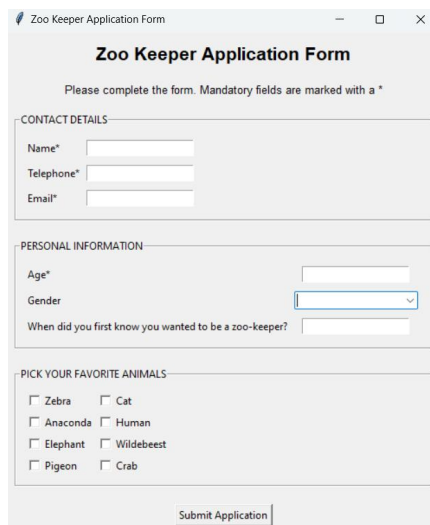
```
os.chdir("test")  
for file in os.listdir():  
    if os.path.isfile(file):  
        print(f"{file}: {os.path.getsize(file)} байт")
```

```
os.chdir("..")  
os.chdir("PZ_11")  
shortest_file = min(files, key=len)  
print("Файл с самым коротким именем в PZ_11:", os.path.basename(shortest_file))
```

```
os.chdir("..")  
for file in os.listdir():  
    if file.endswith(".pdf"):  
        os.startfile(file)
```

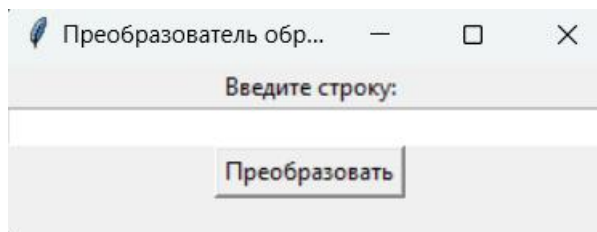
```
os.remove("test/test1/test.txt")
```

### Протокол работы программы 1:



The screenshot shows a web browser window with the title 'Zoo Keeper Application Form'. The page has a light gray background and a white form area. At the top, the title is centered, followed by a subtitle: 'Please complete the form. Mandatory fields are marked with a \*'. The form is divided into three sections: 'CONTACT DETAILS', 'PERSONAL INFORMATION', and 'PICK YOUR FAVORITE ANIMALS'. The 'CONTACT DETAILS' section has three input fields: 'Name\*', 'Telephone\*', and 'Email\*'. The 'PERSONAL INFORMATION' section has three fields: 'Age\*' (a text input), 'Gender' (a dropdown menu), and 'When did you first know you wanted to be a zoo-keeper?' (a text input). The 'PICK YOUR FAVORITE ANIMALS' section has eight checkboxes arranged in two columns: Zebra, Cat, Anaconda, Human, Elephant, Wildebeest, Pigeon, and Crab. At the bottom of the form is a 'Submit Application' button.

### Протокол работы программы 2:



The screenshot shows a web browser window with the title 'Преобразователь обр...'. The page has a light gray background and a white form area. At the top, the title is centered, followed by a subtitle: 'Введите строку:'. Below the subtitle is a large text input field. At the bottom of the form is a 'Преобразовать' button.

### Протокол работы программы 3:

Файлы в каталоге PZ\_11:

output.txt

processed\_output.txt

PZ\_11.1.py

PZ\_11.2.py

reversed\_poem.txt

Отчет к ПЗ-11.pdf

Process finished with exit code 1

**Вывод:** В процессе выполнения практического занятия выработала навыки составления программ с использованием регулярных выражений в IDE PyCharm Community. Выполнены разработка кода, отладка, тестирование, оптимизация программного кода. Готовые программные коды выложены на GitHub.