

Задача 1.

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 (см. таблицу 1).

The screenshot shows a web browser window titled 'Обработка формы - Mozilla Firefox'. The address bar shows 'Firefox' and 'Обработка формы'. The main content area displays a registration form titled 'Форма регистрации пользователя'. The form includes the following fields and controls:

- Ваше имя: [text input]
- Пароль: [text input]
- Возраст: [text input]
- Пол: Мужской ☐ Женский ☐
- Ваши увлечения: Музыка ☐ Видео ☐ Рисование ☐
- Ваша страна: [dropdown menu]
- Ваш город: [dropdown menu]
- Кратко о себе: [text area with placeholder 'краткая информация о ваших увлечениях']
- Решите пример, запишите результат в поле ниже: [text input]
- Buttons: 'Отменить ввод' and 'Данные подтверждаю'

The screenshot shows a Tkinter application window titled 'Форма регистрации пользователя'. The window contains a registration form with the following elements:

- Ваше имя: [text input]
- Пароль: [text input]
- Возраст: [text input]
- Пол: ☐ Мужской ☐ Женский
- Ваши увлечения: ☐ Музыка ☐ Видео ☐ Рисование
- Ваша страна: [dropdown menu]
- Ваш город: [dropdown menu]
- Кратко о себе: [text area]
- Решите пример, запишите результат в поле ниже: [text input]
- Buttons: 'Отменить ввод' and 'Данные подтверждаю'

```
import tkinter as tk
from tkinter import ttk
```

```
def submit():
    print("Data submitted")
```

```

def cancel():
    print("Data entry cancelled")

root = tk.Tk()
root.title("Форма регистрации пользователя")

mainframe = ttk.Frame(root, padding="10")
mainframe.grid(row=0, column=0, sticky=(tk.W, tk.E, tk.N, tk.S))

ttk.Label(mainframe, text="Ваше имя:").grid(row=1, column=1, sticky=tk.W)
name_entry = ttk.Entry(mainframe, width=30)
name_entry.grid(row=1, column=2, sticky=(tk.W, tk.E))

ttk.Label(mainframe, text="Пароль:").grid(row=2, column=1, sticky=tk.W)
password_entry = ttk.Entry(mainframe, width=30, show="*")
password_entry.grid(row=2, column=2, sticky=(tk.W, tk.E))

ttk.Label(mainframe, text="Возраст:").grid(row=3, column=1, sticky=tk.W)
age_entry = ttk.Entry(mainframe, width=30)
age_entry.grid(row=3, column=2, sticky=(tk.W, tk.E))

ttk.Label(mainframe, text="Пол:").grid(row=4, column=1, sticky=tk.W)
gender = tk.StringVar()
ttk.Radiobutton(mainframe, text="Мужской", variable=gender,
                 value="Мужской").grid(row=4, column=2, sticky=tk.W)
ttk.Radiobutton(mainframe, text="Женский", variable=gender,
                 value="Женский").grid(row=4, column=3, sticky=tk.W)

ttk.Label(mainframe, text="Ваши увлечения:").grid(row=5, column=1,
sticky=tk.W)
music_var = tk.BooleanVar()
video_var = tk.BooleanVar()
drawing_var = tk.BooleanVar()
ttk.Checkbutton(mainframe, text="Музыка", variable=music_var).grid(
    row=5, column=2, sticky=tk.W)
ttk.Checkbutton(mainframe, text="Видео", variable=video_var).grid(
    row=5, column=3, sticky=tk.W)
ttk.Checkbutton(mainframe, text="Рисование", variable=drawing_var).grid(
    row=5, column=4, sticky=tk.W)

ttk.Label(mainframe, text="Ваша страна:").grid(row=6, column=1, sticky=tk.W)
country_combobox = ttk.Combobox(
    mainframe, values=["Россия", "США", "Канада", "Другие"], state='readonly')
country_combobox.grid(row=6, column=2, sticky=(tk.W, tk.E))

ttk.Label(mainframe, text="Ваш город:").grid(row=7, column=1, sticky=tk.W)
city_entry = ttk.Entry(mainframe, width=30)

```

```

city_entry.grid(row=7, column=2, sticky=(tk.W, tk.E))

ttk.Label(mainframe, text="Кратко о себе:").grid(row=8, column=1, sticky=tk.W)
about_text = tk.Text(mainframe, width=40, height=4)
about_text.grid(row=8, column=2, columnspan=2, sticky=(tk.W, tk.E))

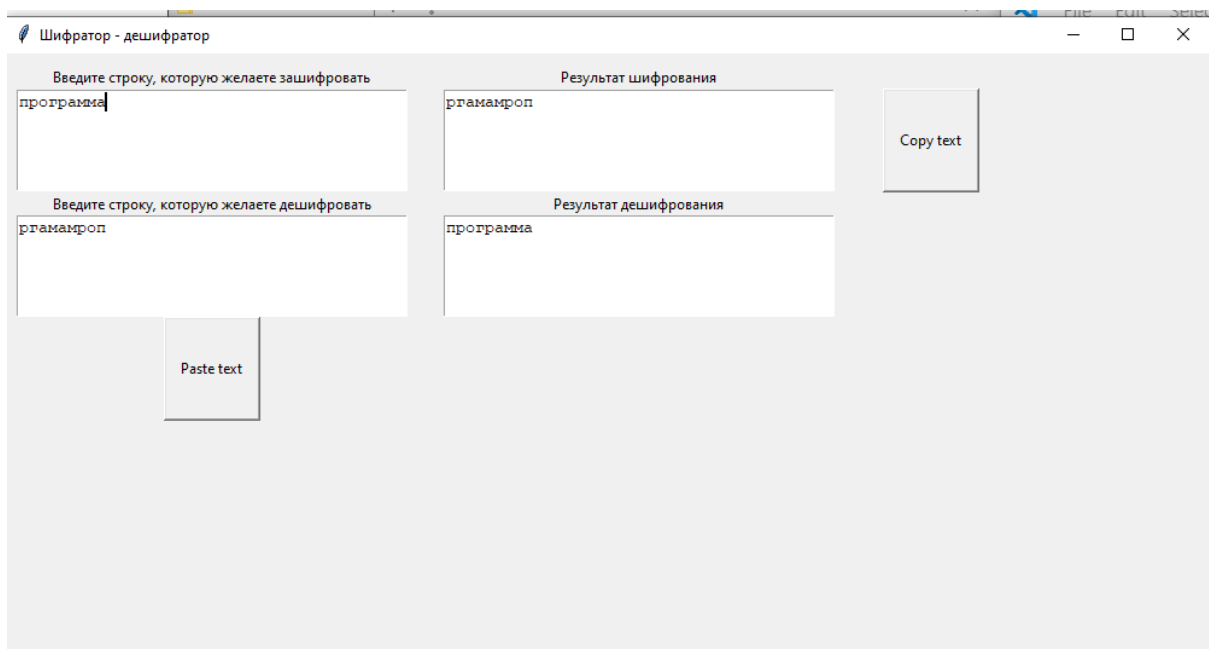
ttk.Label(mainframe, text="Решите пример, запишите результат в поле
ниже:").grid(
    row=9, column=1, sticky=tk.W)
example_entry = ttk.Entry(mainframe, width=30)
example_entry.grid(row=9, column=2, sticky=(tk.W, tk.E))

ttk.Button(mainframe, text="Отменить ввод", command=cancel).grid(
    row=10, column=1, sticky=tk.W)
ttk.Button(mainframe, text="Данные подтверждаю",
    command=submit).grid(row=10, column=2, sticky=tk.W)

root.mainloop()

```

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



encode and decode programm using tkinter

```

import tkinter as tk
from tkinter import ttk

```

```

def encode_func(event):
    encode_text_field.delete('1.0', tk.END)
    text = original_text_field.get("1.0", tk.END).strip('\n')
    encode_text = text[1::2] + text[::2][::-1]
    encode_text_field.insert("1.0", encode_text)
    original_text_field.edit_modified(False)

def decode_func(event):
    after_decode_field.delete('1.0', tk.END)
    text = before_decode_field.get("1.0", tk.END).strip('\n')

    length = len(text)
    mid = (length) // 2
    even_chars = text[:mid].strip()
    odd_chars = text[mid:][::-1]
    decrypted_str = []
    for i in range(mid):
        try:
            decrypted_str.append(odd_chars[i])
            decrypted_str.append(even_chars[i])
        except:
            print('some error')
    if len(text) % 2 != 0:
        decrypted_str.append(odd_chars[-1])
    decode_text = ''.join(decrypted_str)
    after_decode_field.insert("1.0", decode_text)
    before_decode_field.edit_modified(False)

def copy_to_clipboard():
    text = encode_text_field.get("1.0", tk.END)
    root.clipboard_clear()
    root.clipboard_append(text)

def paste_from_clipboard():
    clipboard_text = root.clipboard_get()
    # Вставляем текст в текстовое поле
    before_decode_field.insert(tk.END, clipboard_text)

root = tk.Tk()
root.title("Шифратор - дешифратор")
root.geometry("1000x500")
mainframe = ttk.Frame(root, padding="10")
mainframe.grid(row=0, column=0, sticky=(tk.W, tk.E, tk.N, tk.S))

```

```

ttk.Label(mainframe, text='Введите строку, которую желаете
зашифровать').grid(row=1, column=0)
ttk.Label(mainframe, text='Результат шифрования').grid(row=1, column=2,
padx=10)

original_text_field = tk.Text(mainframe, width=40, height=5)
original_text_field.grid(row=2, column=0)
original_text_field.bind("<<Modified>>", encode_func)

encode_text_field = tk.Text(mainframe, width=40, height=5)
encode_text_field.grid(row=2, column=2, padx=30)

ttk.Label(mainframe, text='Введите строку, которую желаете дешифровать').grid(
    row=3, column=0)
ttk.Label(mainframe, text='Результат дешифрования').grid(
    row=3, column=2, padx=10)

before_decode_field = tk.Text(mainframe, width=40, height=5)
before_decode_field.grid(row=4, column=0)
before_decode_field.bind("<<Modified>>", decode_func)

after_decode_field = tk.Text(mainframe, width=40, height=5)
after_decode_field.grid(row=4, column=2, padx=30)

copy_button = tk.Button(mainframe, width=10, height=5,
    text='Copy text', command=copy_to_clipboard)
copy_button.grid(row=2, column=3, padx=10)

paste_button = tk.Button(mainframe, width=10, height=5,
    text='Paste text', command=paste_from_clipboard)
paste_button.grid(row=5, column=0)

root.mainloop()

```

Задание 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

```
import os
```

```
print(''Пункт 1.
```

Перейдите в каталог PZ11. Выведите список всех файлов в этом каталоге. Имена вложенных подкаталогов выводить не нужно.

```
Результат:''')
```

```
print(os.listdir('../PZ-11'))
```

```
print(''
```

```
Пункт 2.
```

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.

```
''')
```

```
if not os.path.exists('../test/test1'):
```

```
    os.mkdir('../test/test1')
```

```
with open('../PZ-6/Untitled - Frame 4.jpg', 'rb') as file1:
```

```
    with open('../test/Untitled - Frame 4.jpg', 'wb') as file2:
```

```
        file2.write(file1.read())
```

```
with open('../PZ-6/var4_2.py', 'rb') as file1:
```

```
    with open('../test/var4_2.py', 'wb') as file2:
```

```
        file2.write(file1.read())
```

```
with open('../PZ-7/pz_7_var4_1.py', 'rb') as file1:
```

```
    with open('../test/test1/test.txt', 'wb') as file2:
```

```
        file2.write(file1.read())
```

```
def get_directory_size(directory):
```

```
    total_size = 0
```

```
    for dirpath, dirnames, filenames in os.walk(directory):
```

```
        for filename in filenames:
```

```
            filepath = os.path.join(dirpath, filename)
```

```
            if os.path.isfile(filepath):
```

```
                total_size += os.path.getsize(filepath)
```

```
    return total_size
```

```
print(f'Размер файлов в папке test: {get_directory_size("../test")}\n\n')
```

```
print(''Пункт 3.
```

```
Перейти в папку с PZ11, найти там файл с самым коротким именем, имя вывести в консоль. Использовать функцию basename (os.path.basename()).
```

```
'')
```

```
shortest_file = min(os.listdir('../PZ-11'), key=len)
```

```
print(f'Файл с самым коротким названием: {os.path.basename(shortest_file)}')
```

```
print(''Пункт 4.
```

```
Перейти в любую папку где есть отчет в формате .pdf и «запустите» файл в привязанной к нему программе. Использовать функцию os.startfile().
```

```
'')
```

```
os.chdir('../PZ-2')
```

```
os.startfile('otchet.pdf')
```

```
print('Файл успешно запущен')
```

```
print(''Пункт 5.
```

```
Удалить файл test.txt
```

```
'')
```

```
os.chdir('../test/test1')
```

```
print(os.listdir())
```

```
os.remove('test.txt')
```

```
print(os.listdir())
```

```
# Пункт 1.
```

```
# Перейдите в каталог PZ11. Выведите список всех файлов в этом каталоге. Имена  
# вложенных подкаталогов выводить не нужно.
```

```
# Результат:
```

```
# ['new_file.txt', 'otchet_pz11.docx', 'otchet_pz11.pdf',  
#  'otchet_pz11_v1.pdf',
```

```
#    'pz_11_var4_1.py', 'pz_11_var4_2.py', 'result.txt', 'some.txt', 'text18-  
# 4.txt']
```

```
# Пункт 2.
```

```
# 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.
```

```
# Размер файлов в папке test: 420240
```

```
# Пункт 3.
```

```
# Перейти в папку с PZ11, найти там файл с самым коротким именем, имя вывести
в
# консоль. Использовать функцию basename(os.path.basename()).
```

```
# Файл с самым коротким названием: some.txt
```

```
# Пункт 4.
```

```
# Перейти в любую папку где есть отчет в формате .pdf и «запустите» файл в
# привязанной к нему программе. Использовать функцию os.startfile().
```

```
# Файл успешно запущен
```

```
# Пункт 5.
```

```
# Удалить файл test.txt
```

```
# ['test.txt', 'var4_1.py']
```

```
# ['var4_1.py']
```