

Создание окна авторизации с помощью библиотеки TKinter

Романов Даниил Алексеевич

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

Целью данной статьи является, создание программы с графическим интерфейсом в виде окна авторизации. Графический интерфейс создан с помощью библиотеки TKinter языка программирования Python. Результатом исследования является программа способная авторизовать пользователя, которую в будущем можно использовать для создания полноценного приложения с функционалом. Программа создана с помощью языка программирования Python и его библиотек TKinter и pickle в среде программирования PyCharm

Ключевые слова: окно авторизации, Python, TKinter, pickle, Pycharm

Creating an authorization window using the TKinter library

Romanov Daniil Alekseevich

Sholom-Aleichem Priamursky State University

Student

Abstract

The purpose of this article is to create a program with a graphical interface in the form of an authorization window. The graphical interface is created using the TKinter library of the Python programming language. The result of the study is an authorization window that can be used in the future to create a full-fledged application with functionality. The program was created using the Python programming language and its TKinter library in the PyCharm programming environment

Keywords: authorization window, Python, TKinter, pickle, Pycharm

1 Введение

1.1 Актуальность

Язык программирования Python является поистине многозадачным языком программирования способным выполнять множество задач. В данной статье рассматривается лишь одна из многочисленных возможностей данного языка программирования. В данной статье будет использоваться библиотека TKinter. С помощью библиотеки TKinter, которая даёт возможность создавать графический интерфейс для программы, можно создавать различные приложения. Она заслуживает внимания благодаря своей простоте и возможностям, которые она предоставляет. Например, с

помощью Tkinter и pickle можно создать полноценное окно регистрации, которое используется во многих приложениях.

1.2 Обзор исследований

В своей работе D. Veniz описывает возможности библиотеки Tkinter и способы её применения для создания приложений [1]. M. J. Conway в своей статье рассказывает о применении Tkinter для создания 2D панели графического интерфейса и диалоговых окон его программы [2]. A. Cereto-Massagué рассматривал методы создания GUI на основе языка программирования Python [3]. D.C. Price в своём исследовании описывает возможности библиотеки pickle и способы её применения [4].

1.3 Цель исследования

Цель исследования - рассмотрение библиотеки Tkinter и pickle, а также создание на их основе окна авторизации пользователей.

2 Материалы и методы

Для создания программы потребуется несколько вещей. Во-первых, это язык программирования Python [5], библиотеки Tkinter [6], pickle [7] и среда программирования PyCharm [8].

3 Результаты и обсуждение

В первую очередь создаём новый проект в PyCharm. Устанавливаем и импортируем необходимые модули такие как Tkinter для графического интерфейса программы и pickle для сохранения зарегистрированных данных (рис.1).

```
from tkinter import *
from tkinter import messagebox
import pickle
```

Рисунок 1 - Импорт всех необходимых компонентов

Создаём окно. Для этого кладём объект "Tk()" в переменную "root", обозначаем размеры окна и название программы (рис.2).

```
root = Tk()
root.geometry("300x500")
root.title("Войти в систему")
```

Рисунок 2 - Создание окна

Далее создаём функцию регистрации. В ней прописываем кнопки, текст и поля ввода. Для более удобного размещения атрибутов используется метод “pack()” (рис.3).

```
def registration():
    text = Label(text="Для входа в систему - зарегистрируйтесь!")
    text_log = Label(text="Введите ваш логин: ")
    registr_lodin = Entry()
    text_password1 = Label(text="Введите ваш пароль: ")
    registr_password1 = Entry()
    text_password2 = Label(text="Ещё раз пароль: ")
    registr_password2 = Entry(show="*")
    button_registr = Button(text='Зарегистрироваться!', command = lambda: save())
    text.pack()
    text_log.pack()
    registr_lodin.pack()
    text_password1.pack()
    registr_password1.pack()
    text_password2.pack()
    registr_password2.pack()
    button_registr.pack()
```

Рисунок 3 - Основная часть функции регистрации

Внутри функции регистрации создаём функцию, которая будет сохранять зарегистрированные данные используя модуль pickle. Этот модуль позволяет сохранять, записывать и выводить объекты Python. Другими словами, он способен преобразовывать объекты Python в байты и наоборот. Внутри функции “save()” создаём словарь и записываем в него зарегистрированные данные. В переменной “text” создаём текстовый файл, который будет использовать для хранения словаря с помощью метода “pickle”. Данная функция будет срабатывать при нажатии на кнопку “зарегистрироваться” (рис.4).

```
def save():
    login_pass_save = {}
    login_pass_save[registr_lodin.get()]=registr_password1.get()
    f = open("login.txt", "wb")
    pickle.dump(login_pass_save, f)
    f.close()
    login()
```

Рисунок 4 - Дополнительная функция сохранения данных

Переходим к созданию функции входа в систему. Прописываем в ней графические атрибуты и размещаем их, используя метод “pack()” (рис.5).

```
def login():
    text_log = Label(text="Поздравляем! Теперь вы можете войти в систему!")
    text_enter_login = Label(text="Введите ваш логин: ")
    enter_login = Entry()
    text_enter_pass = Label(text="Введите ваш пароль:")
    enter_password = Entry(show="*")
    button_enter = Button(text="Войти", command=lambda: log_pass())
    text_log.pack()
    text_enter_login.pack()
    enter_login.pack()
    text_enter_pass.pack()
    enter_password.pack()
    button_enter.pack()
```

Рисунок 5 - Основная часть функции "login()"

Далее внутри функции "login()" создаём функцию проверки правильности введённого пароля и логина. Функция сравнивает введённые данные с сохранёнными в текстовом файле. Для этого открываем файл и преобразуем его с помощью метода "pickle()". В логической части функции происходит сравнение данных, если данные не совпадают, то программа выдаст ошибку (рис.6).

```
def log_pass():
    f = open("login.txt", "rb")
    a = pickle.load(f)
    f.close()
    if enter_login.get() in a:
        if enter_password.get() == a[enter_login.get()]:
            messagebox.showinfo("Вход выполнен", "Привет! У тебя новое сообщение!")
        else:
            messagebox.showerror("Ошибка!", "Вы ввели неверный логин или пароль")
    else:
        messagebox.showerror("Ошибка!", "Неверный логин!")
```

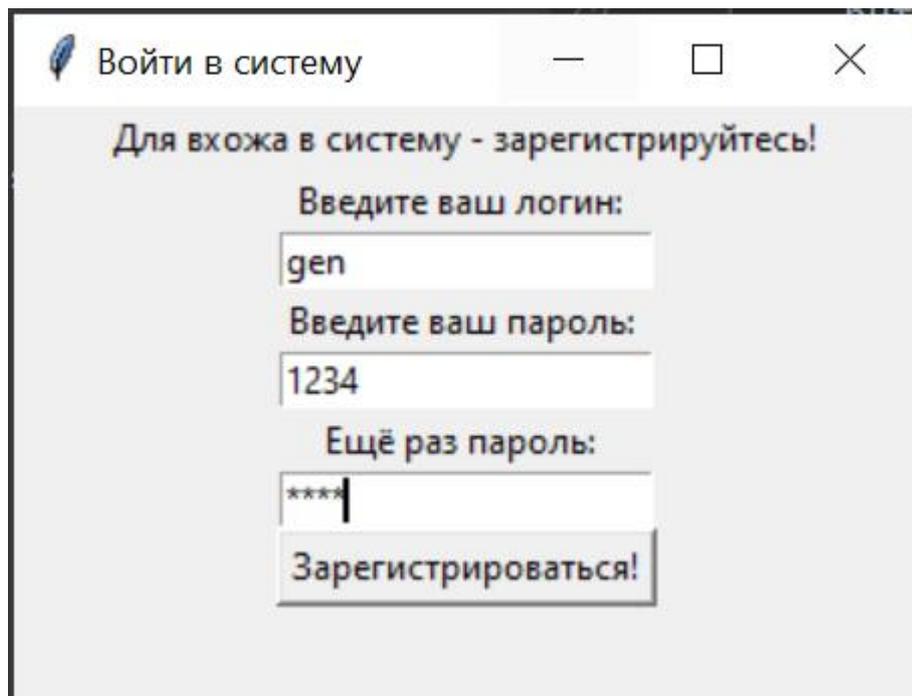
Рисунок 6 - Функция проверки правильности введённого пароля и логина

В конце программы вызываем функцию регистрации и команду запуска окна (рис.7).

```
registration()
root.mainloop()
```

Рисунок 7 - Завершающая часть программы

Переходим к результатам работы. При запуске программы появится окно регистрации (рис.8).



Войти в систему

Для входа в систему - зарегистрируйтесь!

Введите ваш логин:
gen

Введите ваш пароль:
1234

Ещё раз пароль:

Зарегистрироваться!

Рисунок 8 - Окно регистрации

Нажав на кнопку зарегистрироваться, данные будут сохранены и появится дополнительные атрибуты для входа (рис.9).

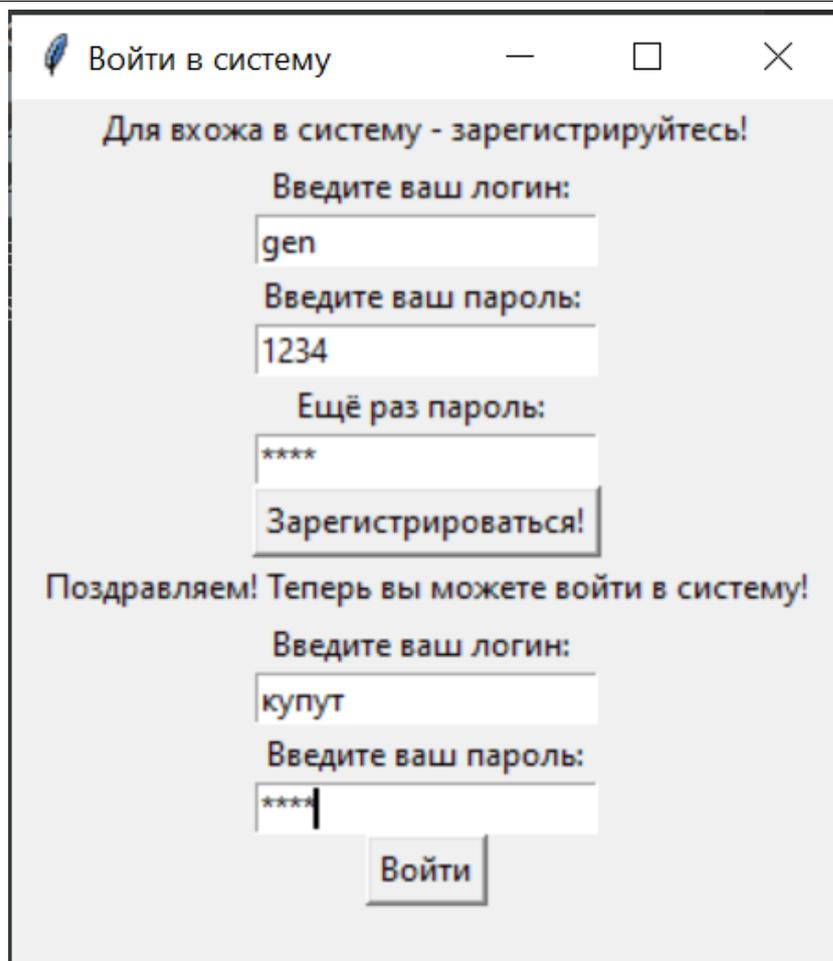


Рисунок 9 - Панель входа в какую-либо систему

Если введённые данные будут неверными, программа выдаст ошибку (рис.10).

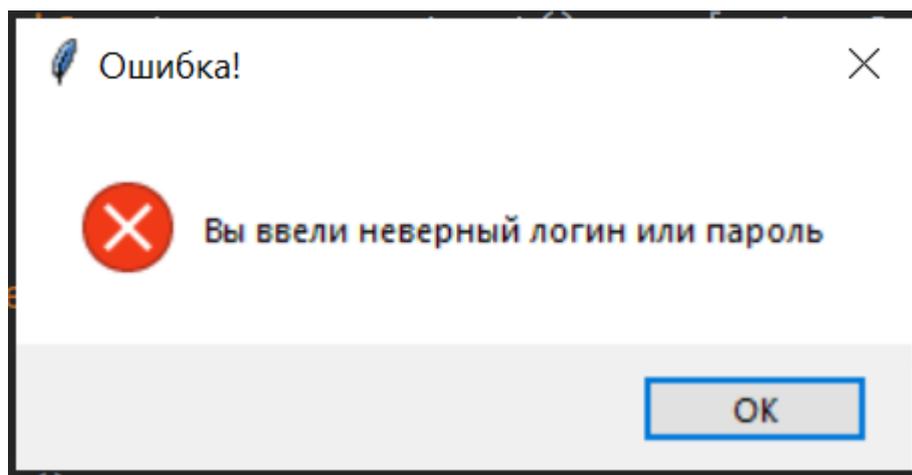


Рисунок 10 - Ошибка неправильных данных

Выводы

В данной работе была создана программа в виде окна регистрации пользователя. Программа способна сохранять данные пользователь и делать проверку на корректность введённого логина или пароля. Программу можно

использовать как шаблон для создания других приложений с более продвинутым функционалом.

Библиографический список

1. Beniz D. et al. Using Tkinter of python to create graphical user interface (GUI) for scripts in LNLS // WEPOPRPO25. 2016. Т. 9. С. 25-28.
2. Conway M. J. Python: a GUI development tool // Interactions. 1995. Т. 2. № 2. С. 23-28.
3. Cereto-Massagué A. et al. DecoyFinder: an easy-to-use python GUI application for building target-specific decoy sets //Bioinformatics. 2012. Т. 28. №12. С. 1661-1662.
4. Price D. C. et al. Hickie: A HDF5-based python pickle replacement // Journal of Open Source Software. 2018. Т. 3. №. 32. С. 1115.
5. Python URL: <https://www.python.org/downloads>
6. TKinter URL: <https://pypi.org/project/tkinter-page>
7. pickle URL: <https://pypi.org/project/pickle5>
8. PyCharm URL: <https://www.jetbrains.com/ru-ru/pycharm/download>