

Создание графического интерфейса программы с помощью TKinter

Романов Даниил Алексеевич

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

Целью данной статьи является, создание графического интерфейса программы с помощью библиотеки TKinter языка программирования Python. Результатом исследования является шаблон графического интерфейса с диалоговым окном, который в будущем можно использовать для создания полноценного приложения с функционалом. Шаблон создан с помощью языка программирования Python и его библиотеки TKinter в среде программирования PyCharm.

Ключевые слова: Python, TKinter, графический интерфейс, шаблон, PyCharm

Creating a graphical program interface using TKinter

Romanov Daniil Alekseevich

Sholom-Aleichem Priamursky State University

Student

Abstract

The purpose of this article is to create a graphical program interface using the TKinter library of the Python programming language. The result of the study is a graphical interface template with a dialog box, which in the future can be used to create a full-fledged application with functionality. The template was created using the Python programming language and its TKinter library in the PyCharm programming environment.

Keywords: Python, TKinter, GUI, Template, PyCharm

1 Введение

1.1 Актуальность

Разработка на Python не сводится лишь к созданию web-сайтов или программированию машинного обучения. С помощью библиотеки TKinter, которая даёт возможность создавать графический интерфейс для программы, можно создавать различные приложения. Стоит отметить, что TKinter в списке похожих библиотек не является лидером, но он явно заслуживает внимания благодаря своей простоте и возможностям, которые он предоставляет.

1.2 Обзор исследований

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

1.3 Цель исследования

Цель исследования - рассмотрение библиотеки Tkinter и создание на её основе графический интерфейс программы.

2 Материалы и методы

Для создания программы потребуется несколько вещей. Во-первых, это язык программирования Python [4], библиотека Tkinter [5] и среда программирования PyCharm [6].

3 Результаты и обсуждения

В первую очередь создаём новый проект в PyCharm. Импортируем все классы из библиотеки Tkinter и создаём глобальный объект на основе класса Tk. Он будет являться полноценным окном приложения, к которому можно добавить определённые характеристики (рис.1).

```
from tkinter import *
from tkinter import messagebox
|
root = Tk()
```

Рисунок 1 - Импортирование классов библиотеки Tkinter и создание глобального объекта

Далее создаём функцию, она будет отслеживать нажатие кнопки, а затем будет получать данные из текстовых полей и показывать их значения в сплывающем окне (рис.2).

```
def btn_click():
    login = loginInput.get()
    password = passField.get()

    info_str = f'Данные: {str(login)}, {str(password)}'
    messagebox.showinfo(title='Название', message=info_str)
```

Рисунок 2 - Функция, отслеживающая нажатия кнопки

Дадим окну приложения определённые характеристики, такие как: задний фон, название окна, прозрачность окна, задать размеры окна и даже запретить пользователю изменять размер она с помощью метода `resizable` (рис.3).

```
root['bg'] = '#fafafa'  
root.title('Название проги')  
root.wm_attributes('-alpha', 0.7)  
root.geometry('300x250')  
  
root.resizable(width=False, height=False)
```

Рисунок 3 - Придаём окну приложения характеристики

Создаём объект `canvas`. Он не является обязательным, но позволяет на форме окна рисовать различные объекты (рис.4).

```
canvas = Canvas(root, height=300, width=250)  
canvas.pack()
```

Рисунок 4 - Создание объекта `canvas`

Далее создаём `Frame`. Это рамка, что содержит в себе другие визуальные компоненты. Создаём объект на основе класса `Frame`, указываем к какому окну он принадлежит и так же можем указать цвет фона, высоту и ширину (рис.5).

```
frame = Frame(root, bg='red')  
frame.place(relx=0.15, rely=0.15, relwidth=0.7, relheight=0.7)
```

Рисунок 5 - Создание объекта `Frame`

Теперь создадим текстовую надпись. Она создаётся на основе класса `Label`. При создании надписи указываем параметры такие как: окно, к которому принадлежит надпись, текст надписи и её цвет. Так же по желанию можно заменить шрифт и прочие настройки. Добавим кнопку, она создаётся на основе класса `Button`. В её параметрах нужно указать функцию, которая отслеживает нажатие кнопки. Чтобы виджеты добавились в окно, после их создание нужно вызывать функцию `place`, которая позволяет расположить объект в окне и указать, где именно он будет расположен (рис.6).

```
title = Label(frame, text='Текст подсказка', bg='gray', font=40)
title.pack()
btn = Button(frame, text='Кнопка', bg='yellow', command=btn_click)
btn.pack()
```

Рисунок 6 - Создание текстовой надписи и кнопки

Добавим в программу текстовое поле. Оно создаётся по тому же принципу что и предыдущие объекты, только в этот раз используем класс Entry. Далее создаём поле-пароль по тому же самому принципу, только добавляем параметр show, чтобы указать какие символы будут показываться вместо текста пользователя (рис.7).

```
loginInput = Entry(frame, bg='white')
loginInput.pack()
passField = Entry(frame, bg='white', show='*')
passField.pack()
```

Рисунок 7 - Создание текстовых полей

Осталось прописать запуск постоянного цикла, без которого программа работать не будет. Пока цикл будет запущен, то и программа будет работать (рис.8).

```
root.mainloop()
```

Рисунок 8 - Запуск постоянного цикла

Переходим к итогу работы. При запуске появляется окно программы, которое получает данные от пользователя, а затем выводит эти данные в сплывающем окне (рис.9).

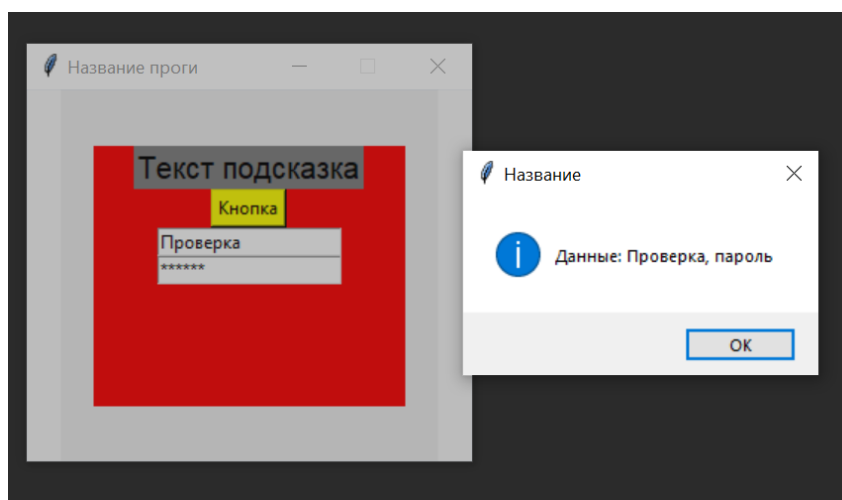


Рисунок 9 - Итог работы, графический интерфейс программы с функционалом

Выводы

В данной работе была создана программа с графическим интерфейсом для пользователей, которая получает данные от пользователя и выводит их в диалоговом окне. 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. Python URL: <https://www.python.org/downloads>
5. TKinter URL: <https://pypi.org/project/tkinter-page>
6. PyCharm URL: <https://www.jetbrains.com/ru-ru/pycharm/download/?ysclid=14e394tnm7260579901#section=windows>