УДК 004

Создание метеорологического GUI приложения с помощью TKinter

Романов Даниил Алексеевич
Приамурский государственный университет имени Шолом-Алейхема
студент

Аннотация
Целью данной статьи является, создание метеорологического GUI приложения с помощью языка программирования Python и его библиотеки TKinter. Результатом исследования является приложение, которое отслеживает погодную информацию в погодном городе. Программа написана в среде программирования PyCharm. Используются гео-данные с сайта OpenWeatherMap.
Ключевые слова: Python, TKinter, GUI, метеоданные, OpenWeatherMap, PyCharm

Creating a meteorological GUI application using TKinter

Romanov Daniil Alekseevich
Sholom-Aleichem Priamursky State University
Student

Abstract
The purpose of this article is to create a meteorological GUI application using the Python programming language and its TKinter library. The result of the study is an application that tracks weather information in a weather city. The program is written in the PyCharm programming environment. Geo-data from the OpenWeatherMap website is used.
Keywords: Python, TKinter, GUI, Weather data, OpenWeatherMap, PyCharm

1 Введение

1.1 Актуальность

Язык программирования Python даёт возможность создавать большой спектр приложений. Для создания графического интерфейса используются такие популярные библиотеки как TKinter, Kivy, PyQt, Turtle, WxPython, PyGUI. В данной статье будет использоваться библиотека TKinter. Она заслуживает внимания благодаря своей простоте и возможностям, которые она предоставляет. На её основе будет создано приложение с дизайном для отслеживания погоды.

1.2 Обзор исследований

В своей работе D. Beniz 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]. В своей работе C. Dewi и R. C. Chen описывали возможности сайта OpenWeather и его применение в различных веб-проектах [4]. Р. Р. Крапивин, Г. А. Гареева исследовали способ получения доступа к данным путем авторизации аккаунта с помощью библиотеки Requests в языке Python [5].

1.3 Цель исследования

Цель исследования - создать метеорологическое приложение с графическим интерфейсом с помощью TKinter и понять принцип его работы.

2 Материалы и методы

Для создания программы потребуется несколько вещей. Во-первых, это язык программирования Pynthon [6], библиотека TKinter [7], среда программирования PyCharm [8] и сайт OpenWeatherMap [9]. Данный сайт предоставляет нам API при помощи которого можно получать информацию о погоде в разных городах мира. Так же рекомендуется иметь готовый шаблон графического интерфейса, создать который можно по статье “Создание графического интерфейса программы с помощью TKinter” [10].

3 Результаты и обсуждения

Для начала работы нужно зайти на сайт OpenWeatherMap и зарегистрироваться на нём. После регистрации в кабинете пользователя можно найти личный API ключ, который будет использоваться для получения данных о погоде (рис.1).

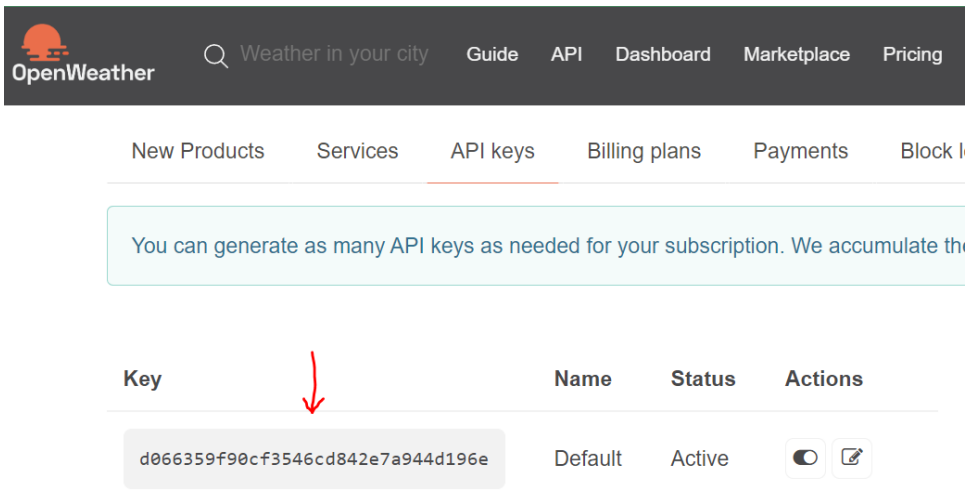

Рисунок 1 - Личный API ключ

Открываем PyCharm и создаём новый проект. Первым делом импортируем все из библиотеки TKinter и подключаем библиотеку requests. Эта библиотека нужна для работы с отправкой URL запросов. Затем создаём главный объект, по сути, окно приложения (рис.2).

```
from tkinter import *
import requests
root = Tk()
```

Рисунок 2 - Подключение необходимых библиотек

Приступаем к созданию функции. Эта функция срабатывает при нажатии на кнопку "Посмотреть погоду". В ней мы получаем данные от пользователя и отправляем запрос по определённому URL-адресу, а в ответ будет приходить JSON-формат с данными про погоду. В переменную key нужно вставить свой API ключ. Все параметры прописываем в отдельный словарь и обращаемся через библиотеку requests по URL-адресу с передачей всех данных. В качестве города используются данные, полученные от пользователя. Полученные данные добавляем в текстовую надпись для отображения пользователю (рис.3).

```
def get_weather():
    city = cityField.get()

    key = 'd066359f90cf3546cd842e7a944d196e'
    url = 'http://api.openweathermap.org/data/2.5/weather'
    params = {'APPID': key, 'q': city, 'units': 'metric'}
    result = requests.get(url, params=params)
    weather = result.json()

    info['text'] = f'{str(weather["name"])}: {weather["main"]["temp"]}'
```

Рисунок 3 - Функция, отслеживающая нажатие кнопки

Настраиваем главное окно. Указываем фоновый цвет, название окна, размер окна и делаем невозможным менять размер окна (рис.4).

```
root['bg'] = '#fafafa'
root.title('Погодное приложение')
root.geometry('300x250')
root.resizable(width=False, height=False)
```

Рисунок 4 - Настройка главного окна

Создаем фрейм, он является областью для размещения других объектов. Указываем к какому окну он принадлежит, какой у него фон и какая обводка. Также указываем его расположение. Этот фрейм отвечает за текстовое поле и кнопку (рис.5).

```
frame_top = Frame(root, bg='#ffb700', bd=5)
frame_top.place(relx=0.15, rely=0.15, relwidth=0.7, relheight=0.25)
```

Рисунок 5 - Создание фрейма

Делаем всё тоже самое, но для второго фрейма. Этот фрейм отвечает за текстовую надпись для вывода в ней информации (рис.6).

```
frame_bottom = Frame(root, bg='#ffb700', bd=5)
frame_bottom.place(relx=0.15, rely=0.55, relwidth=0.7, relheight=0.1)
```

Рисунок 6 - Создание второго фрейма

Создаем текстовое поле для получения данных от пользователя. Размещение этого объекта, всегда нужно прописывать (рис.7).

```
cityField = Entry(frame_top, bg='white', font=30)
cityField.pack()
```

Рисунок 7 - Создание текстового поля

Создаем кнопку и при нажатии будет срабатывать метод get_weather (рис.8).

```
btn = Button(frame_top, text='Посмотреть погоду', command=get_weather)
btn.pack()
```

Рисунок 8 - Создание кнопки

Создаем текстовую надпись, в которую будет выводиться информация о погоде (рис.9).

```
info = Label(frame_bottom, text='Погодная информация', bg='#ffb700', font=40)
info.pack()
```

Рисунок 9 - Создание текстовой надписи

Запускаем постоянный цикл, чтобы программа работала (рис.10)

```
root.mainloop()
```

Рисунок 10 - Запуск постоянного цикла

И наконец переходим к итогу работы. При запуске появляется окно программы, которое получает данные от пользователя, то есть название города, а затем выводит информацию о погоде в нём (рис.11).

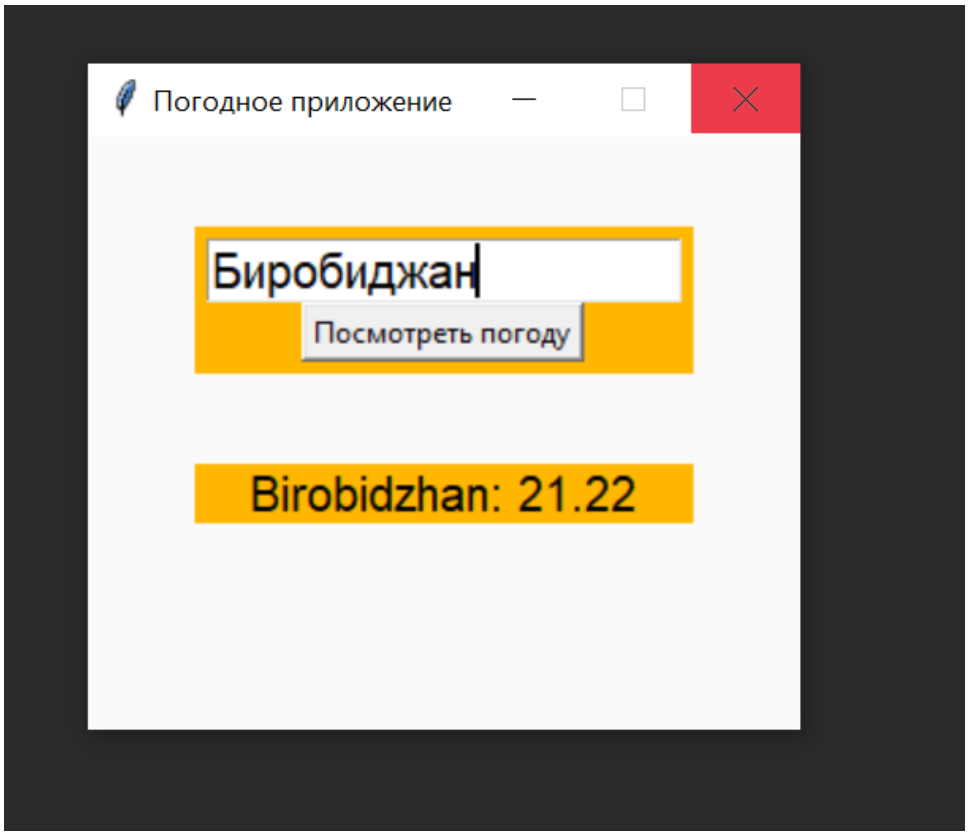

Рисунок 11 - Итог работы, метеорологическое GUI приложение

Выводы

В данной работе была создана полноценная программа, которая имеет неплохой дизайн, а главное полный функционал что позволяет получать данные про погоду в городе, который введёт сам пользователь.

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. Dewi C., Chen R. C. Integrating Real-Time Weather Forecasts Data Using OpenWeatherMap and Twitter //International Journal of Information Technology and Business. – 2019. – Т. 1. – №. 2. – С. 48-52.
5. Крапивин Р. Р., Гареева Г. А. Получение доступа к данным путем авторизации в аккаунт с помощью библиотеки Requests в языке Python

//Инновационные технологии, экономика и менеджмент в промышленности. – 2021. – С. 206-208.
6. Python URL: https://www.python.org/downloads
7. TKinter URL: https://pypi.org/project/tkinter-page
8. PyCharm URL: https://www.jetbrains.com/ru-ru/pycharm/download/?ysclid=l4e394tnm7260579901#section=windows
9. OpenWeatherMap URL: https://openweathermap.org
10. Создание графического интерфейса программы с помощью TKinter URL: https://1drv.ms/w/s!AtdxmxWoArjpgXyiBivuqn-bT-c0?e=cEeBmN