

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

ФАКУЛЬТЕТ ЭЛЕКТРОННО-ИНФОРМАЦИОННЫХ СИСТЕМ

Кафедра интеллектуальных информационных технологий

Отчёт по лабораторной работе №4

Специальность ПО

Выполнил

В. С. Юрашевич,

студент группы ПО-11

Проверил

И. Д. Козик

Брест 2025

Цель работы: научиться работать с Github API, приобрести практические навыки написания программ для работы с REST API или GraphQL API.

Ход работы

Общее задание

Используя Github API, реализовать предложенное задание на языке Python. Выполнить визуализацию результатов, с использованием графика или отчета. Можно использовать как REST API (рекомендуется), так и GraphQL.

Код программы

```
import requests

import datetime

from datetime import timezone

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

language = input("Введите язык программирования: ")

days = int(input("Выберите период (7 / 30 дней): "))

min_stars = input("Минимальное количество звёзд (по желанию, Enter – пропустить): ")

min_stars = int(min_stars) if min_stars else 0

print(f"Анализируем популярные репозитории на {language} за последние {days} дней...")

today = datetime.datetime.now(timezone.utc)

since = (today - datetime.timedelta(days=days)).strftime('%Y-%m-%d')

query = f"language:{language} created:>{since}"

url = "https://api.github.com/search/repositories"

params = {

    "q": query,

    "sort": "stars",

    "order": "desc",

    "per_page": 100

}

headers = {
```

```

    "Accept": "application/vnd.github+json"
}

response = requests.get(url, params=params, headers=headers)
data = response.json()
repos = []
for repo in data.get('items', []):
    stars = repo['stargazers_count']
    if stars >= min_stars:
        repos.append({
            'name': repo['name'],
            'owner': repo['owner']['login'],
            'stars': repo['stargazers_count'],
            'forks': repo['forks_count'],
            'language': repo['language'],
            'description': repo['description'],
            'url': repo['html_url'],
        })

df = pd.DataFrame(repos)
df_sorted = df.sort_values(by='stars', ascending=False).head(5)
print(df_sorted)
print("ТОП-5 самых быстрорастущих проектов:")
for i, row in enumerate(df_sorted.itertuples(), 1):
    print(f"{i}. **{row.name}** (+{row.stars} ⭐) - {row.description}")

sns.set(style="whitegrid")
plt.figure(figsize=(10, 6))

sns.barplot(data=df_sorted, x='stars', y='name', hue='name', palette='viridis',
            legend=False)

plt.xlabel('Количество звёзд за период')
plt.ylabel('Репозиторий')
plt.title(f"ТОП-5 популярных {language}-репозитория за {days} дней")
plt.tight_layout()

```

```
plt.savefig(f"trending_{language.lower()}.png")  
plt.show()
```

Спецификация ввода

>python language<язык программирования>

>python days<период в днях>

>python min_stars<минимальное количество звёзд>

Пример

>python language Python

>python days 7

>python min_stars 100

Спецификация вывода

<5 строк с самыми популярными репозиториями>

Пример

ТОП-5 самых быстрорастущих проектов:













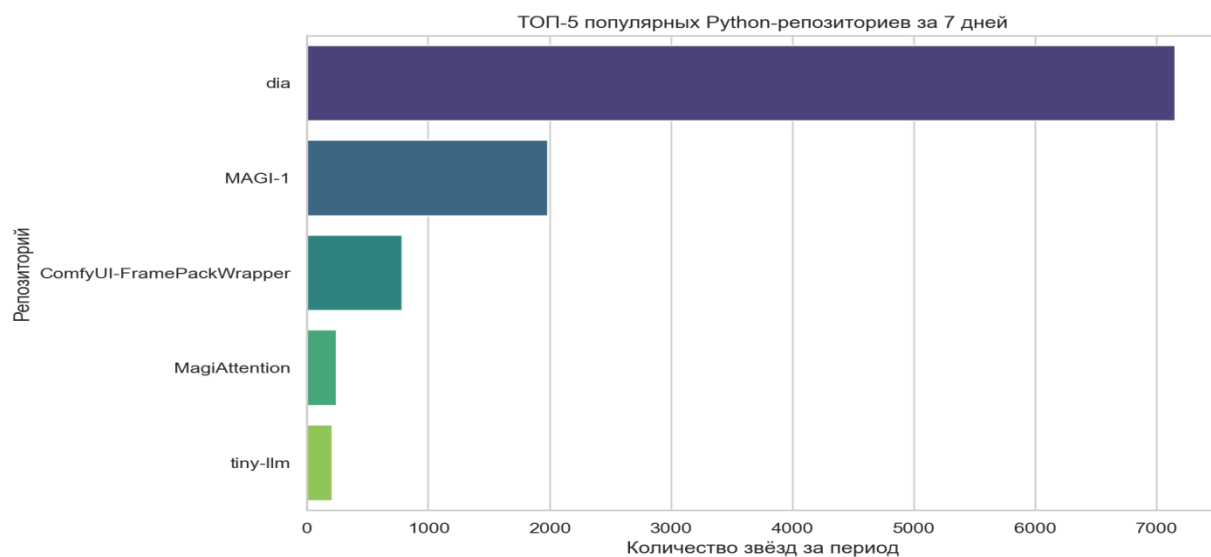
1. ****dia**** (+7153 ) - A TTS model capable of generating ultra-realistic dialogue in one pass.
2. ****MAGI-1**** (+1988 ) - MAGI-1: Autoregressive Video Generation at Scale
3. ****ComfyUI-FramePackWrapper**** (+783 ) - None
4. ****MagiAttention**** (+244 ) - A Distributed Attention Towards Linear Scalability for Ultra-Long Context, Heterogeneous Data Training
5. ****tiny-llm**** (+208 ) - ( WIP) LLM serving with MLX tutorial for systems engineers.

Рисунок с результатами работы программы

```
ТОП-5 самых быстрорастущих проектов:  
1. **dia** (+7153 ) - A TTS model capable of generating ultra-realistic dialogue in one pass.  
2. **MAGI-1** (+1988 ) - MAGI-1: Autoregressive Video Generation at Scale  
3. **ComfyUI-FramePackWrapper** (+783 ) - None  
4. **MagiAttention** (+244 ) - A Distributed Attention Towards Linear Scalability for Ultra-Long Context, Heterogeneous Data Training  
5. **tiny-llm** (+208 ) - ( WIP) LLM serving with MLX tutorial for systems engineers.
```



Вывод: научился работать с Github API, приобрел практические навыки написания программ для работы с REST API.