

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
ФАКУЛЬТЕТ ЭЛЕКТРОННО-ИНФОРМАЦИОННЫХ СИСТЕМ
Кафедра интеллектуальных информационных технологий

Отчёт по лабораторной работе №4
Специальность ПО11

Выполнил
Д. М. Андросюк
студент группы ПО11

Проверил
А. А. Крощенко
ст. преп. кафедры ИИТ,
05.04.2025 г.

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.

Задание 1

Автоматический анализ трендов популярных репозиторий.

Напишите Python-скрипт, который анализирует тренды популярных GitHub-репозиторий за определённый период (например, за последние 7 или 30 дней).

1. Запрашивает у пользователя:

Язык программирования (например, Python, JavaScript, Go)

Период анализа (за неделю, за месяц)

(Опционально) Минимальное количество звезд, чтобы фильтровать менее популярные проекты

2. Использует GitHub API для получения самых быстрорастущих репозиторий по звёздам (stars).

3. Для каждого найденного репозитория собирает данные:

Название

Автор

Количество звёзд за период

Общее количество звёзд

Количество форков

Основной язык

Описание

Ссылка на репозиторий

4. Определяет самые динамично растущие проекты (по количеству новых звёзд за период).

5. Визуализирует динамику роста популярных проектов (matplotlib, seaborn).

Код программы:

```
import requests
import matplotlib.pyplot as plt
import seaborn as sns
from datetime import datetime, timedelta
import argparse
import os
import matplotlib
import warnings
```

```
warnings.filterwarnings("ignore", category=UserWarning, module="matplotlib")
warnings.filterwarnings("ignore", category=UserWarning, module="urllib3")
```

```
matplotlib.use('Agg')
```

```

try:
    plt.rcParams['font.family'] = 'DejaVu Sans'
    plt.rcParams['axes.unicode_minus'] = False
except:
    pass

def get_trending_repos(language, days, min_stars=None):
    since_date = (datetime.now() - timedelta(days=days)).strftime('%Y-%m-%d')

    url = "https://api.github.com/search/repositories"
    params = {
        'q': f'language:{language} created:>={since_date}',
        'sort': 'stars',
        'order': 'desc',
        'per_page': 50
    }

    if min_stars:
        params['q'] += f' stars:>={min_stars}'

    headers = {'Accept': 'application/vnd.github.v3+json'}

    try:
        response = requests.get(url, params=params, headers=headers, timeout=10)
        response.raise_for_status()
        return response.json()['items']
    except Exception as e:
        raise Exception(f"Ошибка GitHub API: {str(e)}")

def process_repo_data(repos, since_date):
    repo_data = []
    headers = {'Accept': 'application/vnd.github.v3+json'}

    for repo in repos:
        try:
            created_at = datetime.strptime(repo['created_at'], '%Y-%m-%dT%H:%M:%SZ')
            days_existed = (datetime.now() - created_at).days
            days_existed = max(days_existed, 1)
            new_stars = int(repo['stargazers_count'] * (min(days_existed, 30)/30))

            repo_data.append({
                'name': repo['name'],
                'author': repo['owner']['login'],
                'new_stars': new_stars,
                'total_stars': repo['stargazers_count'],
                'forks': repo['forks_count'],
                'language': repo['language'],
                'description': repo['description'] or "Без описания",
                'url': repo['html_url']
            })
        except Exception as e:
            print(f"Пропускаем репозиторий {repo['name']} из-за ошибки: {str(e)}")

    return sorted(repo_data, key=lambda x: x['new_stars'], reverse=True)

```

```

def visualize_trends(repos, language, days, filename=None):
    if not filename:
        filename = f'trending_{language.lower()}.png'
        filename = os.path.abspath(filename)

    if not repos:
        print("Нет данных для визуализации")
        return

    sns.set_theme(style="whitegrid")
    plt.figure(figsize=(14, 8))

    top_repos = repos[:10]
    names = [repo['name'][:25] + ('...' if len(repo['name']) > 25 else '') for repo in top_repos]
    new_stars = [repo['new_stars'] for repo in top_repos]
    total_stars = [repo['total_stars'] for repo in top_repos]

    colors = sns.color_palette("husl", len(top_repos))

    bars = plt.barh(
        y=names,
        width=new_stars,
        color=colors,
        edgecolor='black',
        linewidth=0.7,
        alpha=0.8
    )

    for bar, total in zip(bars, total_stars):
        width = bar.get_width()
        plt.text(
            width + max(new_stars)*0.01,
            bar.get_y() + bar.get_height()/2,
            f'Всего: {total:,}',
            va='center',
            ha='left',
            fontsize=10,
            bbox=dict(facecolor='white', alpha=0.7, edgecolor='none')
        )

    plt.title(
        f'Топ-{len(top_repos)} популярных репозиторий на {language}\n'
        f'за последние {days} дней',
        fontsize=14,
        pad=20
    )
    plt.xlabel("")
    plt.ylabel("")
    plt.grid(axis='x', alpha=0.4)

    try:
        plt.tight_layout()
        plt.savefig(filename, dpi=300, bbox_inches='tight', facecolor='white')
        print(f"График успешно сохранён: {filename}")

```

```

except Exception as e:
    print(f"Ошибка при сохранении: {str(e)}")
finally:
    plt.close()

def main():
    parser = argparse.ArgumentParser(description='Анализатор трендов GitHub')
    parser.add_argument('--language', type=str, help='Python/Java/JavaScript')
    parser.add_argument('--days', type=int, choices=[7, 30], help='7 или 30 дней')
    parser.add_argument('--min-stars', type=int, help='Минимальное количество звёзд')

    args = parser.parse_args()

    if not args.language or not args.days:
        print("\nАнализатор трендов GitHub")
        args.language = input("Введите язык программирования: ")
        args.days = int(input("Период анализа (7/30 дней): "))
        min_stars_input = input("Минимальное количество звёзд (опционально, Enter чтобы пропустить): ")
        args.min_stars = int(min_stars_input) if min_stars_input.strip() else None

    print(f"\nАнализ популярных репозиторий на {args.language} за последние {args.days} дней...")

    try:
        repos = get_trending_repos(args.language, args.days, args.min_stars)
        if not repos:
            print("\nНе найдено репозиторий по заданным критериям.")
            return

        since_date = (datetime.now() - timedelta(days=args.days)).strftime('%Y-%m-%d')
        processed_data = process_repo_data(repos, since_date)

        print("\nТОП-5 самых быстрорастущих проектов:")
        for i, repo in enumerate(processed_data[:5], 1):
            print(f"\n{i}. {repo['name']} (+{repo['new_stars']:,} ★)")
            print(f"  Автор: {repo['author']}")
            print(f"  Описание: {repo['description']}")
            print(f"  Всего звёзд: {repo['total_stars']:,} | Форков: {repo['forks']}")
            print(f"  Ссылка: {repo['url']}")

        visualize_trends(processed_data, args.language, args.days)

    except Exception as e:
        print(f"\nПроизошла ошибка: {str(e)}")

if __name__ == "__main__":
    main()

```

Рисунки с результатами работы программы:

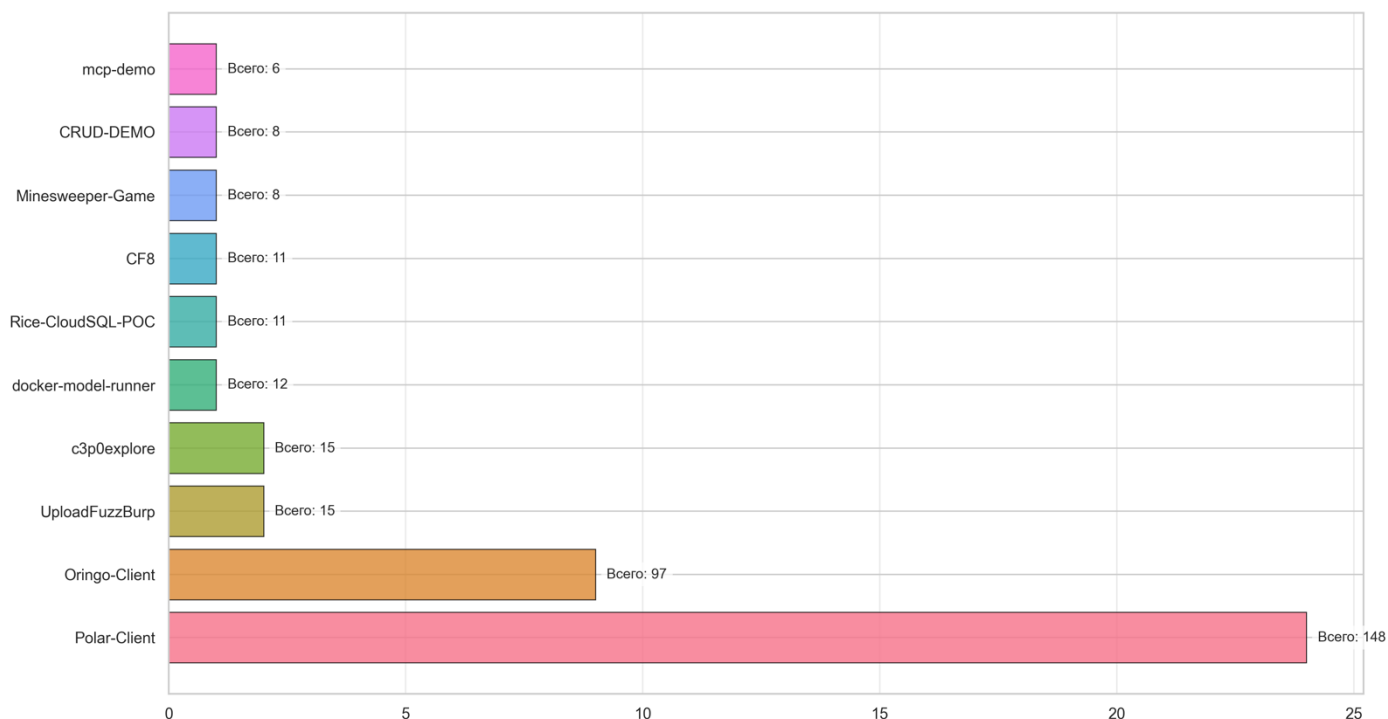
Введите язык программирования: Java
Период анализа (7/30 дней): 7
Минимальное количество звёзд (опционально, Enter чтобы пропустить): 3

Анализ популярных репозиторий на Java за последние 7 дней...

ТОП-5 самых быстрорастущих проектов:

1. Polar-Client (+24 ★)
Автор: PacketDev
Описание: The best QOL mod for mining at hypixel skyblock
Всего звёзд: 148 | Форков: 83
Ссылка: <https://github.com/PacketDev/Polar-Client>
 2. Oringo-Client (+9 ★)
Автор: Queso081
Описание: Без описания
Всего звёзд: 97 | Форков: 90
Ссылка: <https://github.com/Queso081/Oringo-Client>
 3. UploadFuzzBurp (+2 ★)
Автор: eeeeeeeee-code
Описание: burp 文件上传 fuzz插件
Всего звёзд: 15 | Форков: 2
Ссылка: <https://github.com/eeeeeeee-code/UploadFuzzBurp>
 4. c3p0explore (+2 ★)
Автор: unam4
Описание: c3p0 new gadget
Всего звёзд: 15 | Форков: 0
Ссылка: <https://github.com/unam4/c3p0explore>
 5. docker-model-runner (+1 ★)
Автор: danvega
Описание: Без описания
Всего звёзд: 12 | Форков: 7
Ссылка: <https://github.com/danvega/docker-model-runner>
- График успешно сохранён: /Users/dmitrijandrosuk/Downloads/SPP/Lab4/trending_java.png
(myenv) dmitrijandrosuk@MacBook-Pro-Dmitrij Lab4 %

Топ-10 популярных репозиторий на Java
за последние 7 дней



Вывод: Закрепил навыки по работе с Github API, приобрел практические

навыки написания программ для работы с REST API или GraphQL API.