

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
П. А. Захарчук
студент группы ПО11

Проверил
А. А. Крощенко
ст. преп. кафедры ИИТ,
25.04.2025 г.

Брест 2025

Цель работы: приобрести навыки применения паттернов проектирования при решении практических задач с использованием языка Python.

Задание 1: 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 datetime
from dateutil.relativedelta import relativedelta
import matplotlib.pyplot as plt
import seaborn as sns
from urllib.parse import quote

# Настройка стиля графиков
sns.set(style="whitegrid")

def get_user_input():
    """Получение входных данных от пользователя"""
    print("Введите язык программирования (например, Python, JavaScript, Go):")
    language = input().strip()

    print("Выберите период анализа (7 или 30 дней):")
    period = input().strip()
    while period not in ['7', '30']:
        print("Пожалуйста, выберите 7 или 30 дней:")
        period = input().strip()
    period = int(period)

    print("Введите минимальное количество звёзд (или нажмите Enter для пропуска):")
    min_stars = input().strip()
    min_stars = int(min_stars) if min_stars.isdigit() else 0
```

```

return language, period, min_stars

def get_trending_repos(language, period, min_stars):
    """Получение трендовых репозиторий через GitHub API"""
    # Рассчитываем дату начала периода
    end_date = datetime.datetime.now()
    start_date = end_date - relativedelta(days=period)
    date_query = f"created:>{start_date.strftime('%Y-%m-%d')}}"

    # Формируем запрос
    query = f"language:{quote(language)} {date_query}"
    if min_stars > 0:
        query += f" stars:>{min_stars}"

    url = "https://api.github.com/search/repositories"
    params = {
        "q": query,
        "sort": "stars",
        "order": "desc",
        "per_page": 10 # Ограничимся топ-10 для примера
    }

    try:
        response = requests.get(url, params=params)
        response.raise_for_status()
        return response.json().get('items', [])
    except requests.RequestException as e:
        print(f"Ошибка при запросе к API: {e}")
        return []

def process_repo_data(repos):
    """Обработка данных репозиторий"""
    repo_data = []
    for repo in repos:
        repo_info = {
            'name': repo['name'],
            'author': repo['owner']['login'],
            'stars': repo['stargazers_count'],
            'forks': repo['forks_count'],
            'language': repo['language'],
            'description': repo['description'] or "Нет описания",
            'url': repo['html_url']
        }
        repo_data.append(repo_info)
    return repo_data

def visualize_trends(repo_data, language, period):
    """Визуализация трендов"""
    if not repo_data:
        print("Нет данных для визуализации")
        return

    # Подготовка данных для графика
    names = [repo['name'] for repo in repo_data[:5]] # Топ-5
    stars = [repo['stars'] for repo in repo_data[:5]]

    # Создание графика
    plt.figure(figsize=(10, 6))
    sns.barplot(x=stars, y=names, hue=names, palette="viridis", legend=False)

```

```

plt.title(f"Топ-5 популярных {language} репозиторийев за последние {period} дней")
plt.xlabel("Количество звёзд")
plt.ylabel("Репозиторий")

# Сохранение графика
filename = f"trending_{language.lower()}.png"
plt.savefig(filename, bbox_inches='tight')
plt.close()
print(f"График сохранён как {filename}")

def print_report(repo_data, language, period):
    """Вывод отчёта"""
    print(f"\nАнализируем популярные репозитории на {language} за последние {period} дней...")
    print("ТОП-5 самых быстрорастущих проектов:")

    for i, repo in enumerate(repo_data[:5], 1):
        print(f"{i}. **{repo['name']}** (+{repo['stars']} ) - {repo['description']}")
        print(f"  Автор: {repo['author']}, Форков: {repo['forks']}, URL: {repo['url']}")

def main():
    # Получение пользовательского ввода
    language, period, min_stars = get_user_input()

    # Получение данных
    print("Получаем данные с GitHub...")
    repos = get_trending_repos(language, period, min_stars)

    if not repos:
        print("Не удалось получить данные. Проверьте подключение или параметры.")
        return

    # Обработка данных
    repo_data = process_repo_data(repos)

    # Вывод отчёта
    print_report(repo_data, language, period)

    # Визуализация
    visualize_trends(repo_data, language, period)

if __name__ == "__main__":
    main()

```

Спецификация ввода:

Введите язык программирования: <1-й элемент>

Выберите период анализа: <1-й элемент>

Введите минимальное количество звезд: <1-й элемент>

Пример:

Введите язык программирования (например, Python, JavaScript, Go): JavaScript

Выберите период анализа (7 или 30 дней): 30

Введите минимальное количество звёзд (или нажмите Enter для пропуска): 50

Спецификация вывода:

Пример:

Получаем данные с GitHub...

Анализируем популярные репозитории на JavaScript за последние 30 дней...
ТОП-5 самых быстрорастущих проектов:

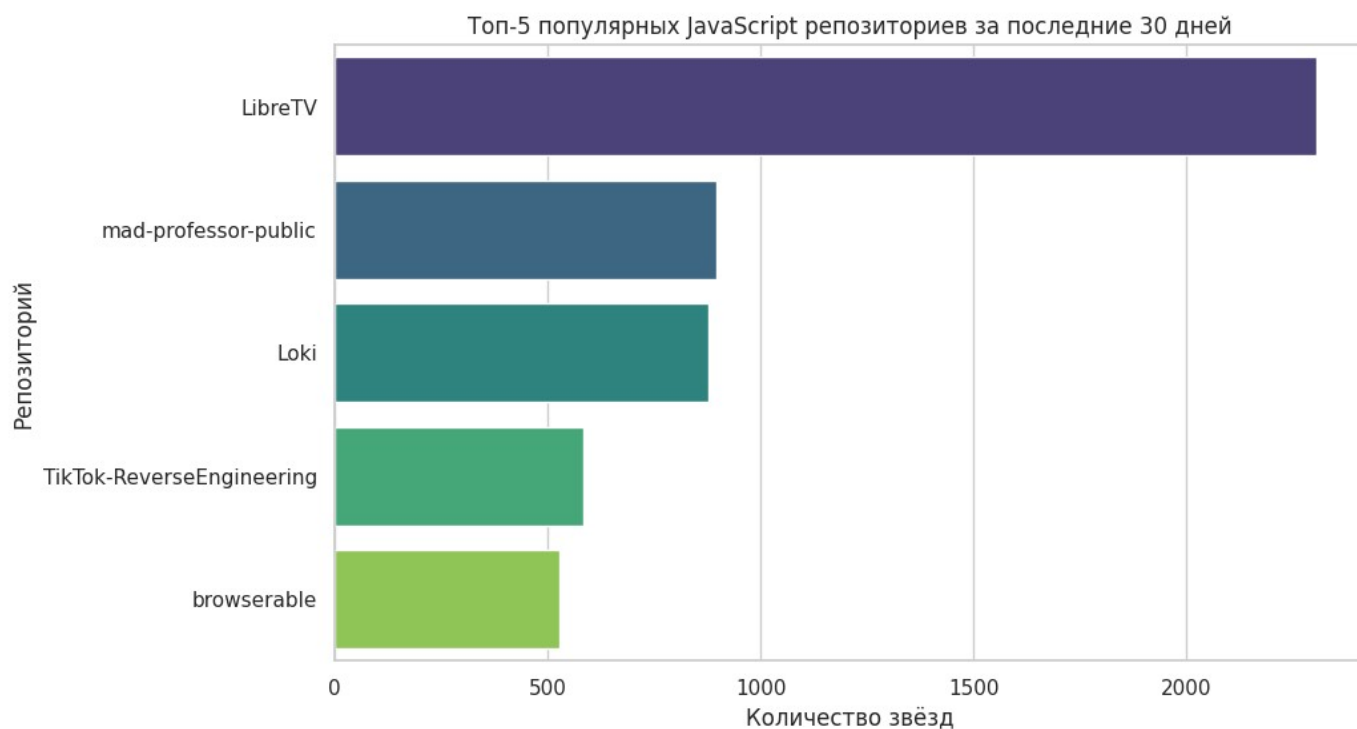
1. **LibreTV** (+2306) - 一分钟搭建影视站，支持 Vercel/Docker 等部署方式
Автор: LibreSpark, Форков: 1822, URL: <https://github.com/LibreSpark/LibreTV>
 2. **mad-professor-public** (+898) - An AI companion for reading papers.
Автор: LYiHub, Форков: 105, URL: <https://github.com/LYiHub/mad-professor-public>
 3. **Loki** (+878) - Node.js Command & Control for Script-Jacking Vulnerable Electron Applications
Автор: boku7, Форков: 144, URL: <https://github.com/boku7/Loki>
 4. **TikTok-ReverseEngineering** (+583) - Нет описания
Автор: LukasOgunfeitimi, Форков: 77, URL: <https://github.com/LukasOgunfeitimi/TikTok-ReverseEngineering>
 5. **browserable** (+528) - Open source and self-hostable browser automation library for AI agents
Автор: browserable, Форков: 49, URL: <https://github.com/browserable/browserable>
- График сохранён как trending_javascript.png

Рисунки с результатами работы программы:

```
/home/twinkle/PycharmProjects/pythonProject/venv/bin/python /home/twinkle/PycharmProjects/pythonProject/lab4.py
Введите язык программирования (например, Python, JavaScript, Go):
JavaScript
Выберите период анализа (7 или 30 дней):
30
Введите минимальное количество звёзд (или нажмите Enter для пропуска):
50
Получаем данные с GitHub...

Анализируем популярные репозитории на JavaScript за последние 30 дней...
ТОП-5 самых быстрорастущих проектов:
1. LibreTV (+2296 *) - 一分钟搭建影视站，支持Vercel/Docker等部署方式
   Автор: LibreSpark, Форков: 1809, URL: https://github.com/LibreSpark/LibreTV
2. mad-professor-public (+889 *) - An AI companion for reading papers.
   Автор: LYiHub, Форков: 104, URL: https://github.com/LYiHub/mad-professor-public
3. Loki (+877 *) - 🐙 Node.js Command & Control for Script-Jacking Vulnerable Electron Applications
   Автор: boku7, Форков: 144, URL: https://github.com/boku7/Loki
4. TikTok-ReverseEngineering (+582 *) - Нет описания
   Автор: LukasOgunfeitimi, Форков: 76, URL: https://github.com/LukasOgunfeitimi/TikTok-ReverseEngineering
5. browserable (+528 *) - Open source and self-hostable browser automation library for AI agents
   Автор: browserable, Форков: 49, URL: https://github.com/browserable/browserable
График сохранён как trending_javascript.png

Process finished with exit code 0
```



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