

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ  
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ  
“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”  
Кафедра интеллектуальных информационных технологий

Отчёт по лабораторной работе №4  
Специальность ПО-11

Выполнил:  
А. А. Билялова  
студент группы ПО11

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

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

**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

### **Вариант 3. Автоматическое отслеживание новых релизов и обновлений в репозитории GitHub.**

Напишите Python-скрипт, который:

1. Запрашивает у пользователя список репозиториях GitHub для мониторинга (например, django/django, fastapi/fastapi, pallets/flask).
2. Использует GitHub API для получения информации о последних релизах и тегах в каждом репозитории.
3. Проверяет, появились ли новые версии или обновления с момента последнего запуска скрипта.
4. Если есть обновления, собирает следующую информацию:
  - Номер новой версии
  - Дата релиза
  - Список изменений (changelog)
  - Ссылку на страницу релиза

### **Код программы:**

```
def check_for_updates(repos):
    config = load_config()
    new_releases = []

    for repo in repos:
        print(f"\nПроверяем обновления для {repo}...")
        latest_release = fetch_latest_release(repo)

        if not latest_release:
            continue

        release_info = format_release_info(latest_release)
        repo_data = config["repositories"].get(repo, {})

        if not repo_data or repo_data.get("version") != release_info["version"]:
            print(f"Найден новый релиз: {release_info['version']} ({release_info['date']})")
            print(f"Ссылка: {release_info['url']}")
            print(f"Основные изменения: {release_info['changes'][:200]}...")

            config["repositories"][repo] = {
                "version": release_info["version"],
                "date": release_info["date"],
                "url": release_info["url"],
                "last_checked": datetime.now().strftime("%Y-%m-%d %H:%M:%S")
            }
            new_releases.append({
                "repo": repo,
                "version": release_info["version"],
                "date": release_info["date"],
                "url": release_info["url"],
                "changes": release_info["changes"]
            })
```

```

    })
    else:
        print(f"Новых релизов для {repo} не обнаружено")

    config["last_check"] = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
    save_config(config)

    create_releases_plot(config["repositories"])

    return new_releases

def main():
    print("GitHub Releases Tracker with Visualization")
    print("=" * 50)

    repos = get_user_repositories()
    if not repos:
        print("Не указаны репозитории для отслеживания")
        return

    new_releases = check_for_updates(repos)

    if new_releases:
        print("\nОбнаружены новые релизы:")
        for release in new_releases:
            print(f"- {release['repo']}: {release['version']} ({release['date']})")
    else:
        print("\nНовых релизов не обнаружено")

if __name__ == "__main__":
    main()

```

## Рисунки с результатами работы программы

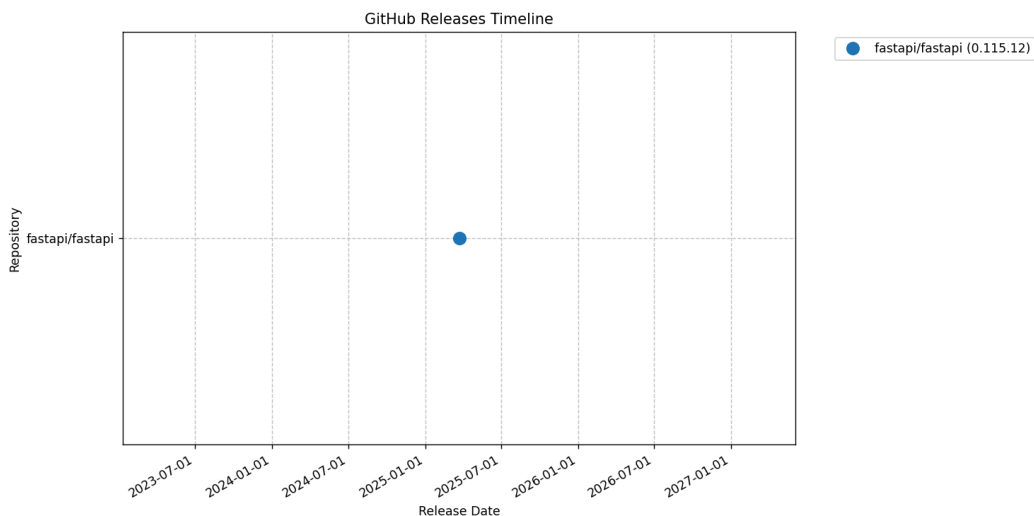
```

PS C:\Users\bilya\OneDrive\Documents\GitHub\spp_po11\reports\Bilyalova\4\src> python test.py
GitHub Releases Tracker
=====
Введите репозитории для отслеживания (через запятую): django/django, fastapi/fastapi

Проверяем обновления для django/django...
Репозиторий django/django не найден или не имеет релизов

Проверяем обновления для fastapi/fastapi...
Найден новый релиз: 0.115.12 (2025-03-23)
Ссылка: https://github.com/fastapi/fastapi/releases/tag/0.115.12
Основные изменения: ### Fixes

```



**Вывод:** научилась работать с Github API, приобрела практические навыки написания программ для работы с REST API или GraphQL API.