

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт прогрессивной инженерии
Департамент цифровых, робототехнических систем и электроники

**ОТЧЕТ
ПО ПРАКТИЧЕСКОЙ РАБОТЕ №1
дисциплины «Программирование на Python»**

Выполнил:
Мыцыков Егор Сергеевич
2 курс, группа ИВТ-б-о-24-1,
09.03.01 «Информатика и
вычислительная техника», очная
форма обучения

(подпись)

Руководитель практики:
Воронкин Р. А.

(подпись)

Отчет защищен с оценкой _____ Дата защиты_____

Ставрополь, 2025 г.

Тема: Исследование основных возможностей Git и GitHub.

Цель: Исследовать базовые возможности системы контроля версий Git и веб-сервиса для хостинга IT-проектов GitHub.

Ссылка на репозиторий: <https://github.com/Egorchik2006/La.git>

Порядок выполнения работы:

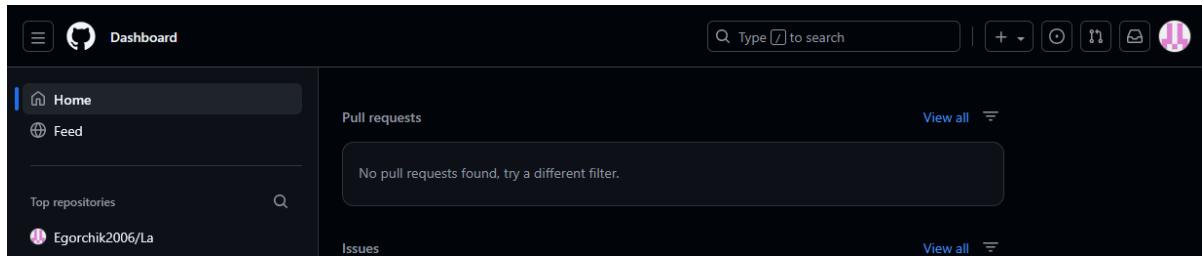


Рисунок 1. Новые аккаунт и репозиторий GitHub

```
C:\Users\Erop>git version  
git version 2.51.0.windows.2
```

Рисунок 2. Установка git и проверка версии

```
C:\Users\Erop>git config --global user.name "Egorchik2006"  
C:\Users\Erop>git config --global user.email "egorm4546@gmail.com"
```

Рисунок 3. Добавление в настройки имени аккаунта и электронной почты

```
C:\Users\Erop>git clone https://github.com/Egorchik2006/La.git  
Cloning into 'La'...  
remote: Enumerating objects: 5, done.  
remote: Counting objects: 100% (5/5), done.  
remote: Compressing objects: 100% (4/4), done.  
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)  
Receiving objects: 100% (5/5), done.
```

Рисунок 4. Клонирование репозитория

```
C:\Users\Erop>cd La
```

Рисунок 5. Переход к папке репозитория

```
C:\Users\Erop\La>git status  
On branch main  
Your branch is up to date with 'origin/main'.  
nothing to commit, working tree clean
```

Рисунок 6. Проверка статуса репозитория



Рисунок 7. Изменение файла README.md

```
print("Калькулятор")

a = input("Введите делимое")
b = input("Введите делитель")

c = a/b
d = a*b
e = a+b
f = a-b

print("Частное: ", c, "\n", "Произведение: " d, "\n", "Сумма: ", e, "\n", "Разность: ", f)print("Калькулятор")
```

Рисунок 8. Создание простой программы на языке Python



Рисунок 9. Добавление README.md

```
C:\Users\Egor\La>git add Program.py
C:\Users\Egor\La>git commit -m "Ввод а"
[main eba38f4] Ввод а
 1 file changed, 11 insertions(+)

C:\Users\Egor\La>add Program.py
"add" не является внутренней или внешней
командой, исполняемой программой или пакетным файлом.

C:\Users\Egor\La>git add Program.py
C:\Users\Egor\La>git commit -m "Ввод а"
On branch main
Your branch is ahead of 'origin/main' by 2 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

C:\Users\Egor\La>git add Program.py
C:\Users\Egor\La>git commit -m "Ввод а"
[main cd064fa] Ввод а
 1 file changed, 11 deletions(-)

C:\Users\Egor\La>git add Program.py
C:\Users\Egor\La>git commit -m "Ввод б"
[main ba1705f] Ввод б
 1 file changed, 4 insertions(+)

C:\Users\Egor\La>git add Program.py
C:\Users\Egor\La>git commit -m "Частное"
[main e8983c3] Частное
```

Рисунок 10. Построчное добавление коммитов в процессе написания кода

```
C:\Users\Egor\La>git add Program.py
C:\Users\Egor\La>git commit -m "Сумма"
[main 07a3d06] Сумма
 1 file changed, 2 insertions(+), 1 deletion(-)
C:\Users\Egor\La>git add Program.py
C:\Users\Egor\La>git commit -m "Разность"
[main 91d58ea] Разность
 1 file changed, 3 insertions(+), 1 deletion(-)
C:\Users\Egor\La>git add Program.py
C:\Users\Egor\La>git commit -m "Вывод значений"
[main 32cd647] Вывод значений
 1 file changed, 3 insertions(+), 1 deletion(-)
C:\Users\Egor\La>
```

Рисунок 11. Построчное добавление коммитов в процессе написания кода

```
C:\Users\Egor>cd La
C:\Users\Egor\La>git status
On branch main
Your branch is ahead of 'origin/main' by 8 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

C:\Users\Egor\La>git push
info: please complete authentication in your browser...
Enumerating objects: 25, done.
Counting objects: 100% (25/25), done.
Delta compression using up to 12 threads
Compressing objects: 100% (21/21), done.
Writing objects: 100% (23/23), 2.00 KiB | 1022.00 KiB/s, done.
Total 23 (delta 12), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (12/12), done.
To https://github.com/Egorchik2006/La.git
  934acfe..32cd647  main -> main

C:\Users\Egor\La>
```

Рисунок 12. Полная синхронизация локального и удалённого репозитория

Ответы на контрольные вопросы

1. Что такое СКВ (git) и каково ее назначение?

Система контроля версий (СКВ) — это программа для отслеживания изменений в файлах. Позволяет сохранять историю, возвращаться к предыдущим версиям и работать в команде.

2. В чем недостатки локальных и централизованных СКВ?

Локальные: нет совместной работы, риск потери данных.

Централизованные: единый сервер — если он упадет, работа остановится.

3. К какой СКВ относится Git?

К распределенной (каждый разработчик имеет полную копию репозитория).

4. В чем концептуальное отличие Git от других СКВ?

Git хранит не изменения файлов, а снимки всего проекта на каждый коммит.

5. Как обеспечивается целостность хранимых данных в Git?

Через хеш-суммы (SHA-1). Каждый коммит и файл имеют уникальный хеш.

6. В каких состояниях могут находиться файлы в Git?

modified (изменен), staged (подготовлен к коммиту), committed (сохранен в репозитории).

7. Что такое профиль пользователя в GitHub?

Это ваша учетная запись на github.com с репозиториями, настройками и историей действий.

8. Какие бывают репозитории в GitHub?

Public (открытые для всех),

Private (только для вас и collaborators).

9. Основные этапы модели работы с GitHub:

Клонировать(git clone) → Изменить(modify) → Добавить (git add) → Закоммитить (git commit) → Запушить (git push).

10. Как осуществляется первоначальная настройка Git после установки?

git config --global user.name "Ваше имя"

git config --global user.email "ваша@почта.com"

11. Этапы создания репозитория в GitHub:

Зарегистрироваться → Нажать «New repository» → Ввести имя → Выбрать лицензию → Create.

12. Какие типы лицензий поддерживаются GitHub?

MIT, GPL, Apache — они определяют правила использования вашего кода.

13. Как осуществляется клонирование репозитория?

`git clone https://github.com/username/repo.git`

Нужно, чтобы скопировать проект с GitHub на компьютер.

14. Как проверить состояние локального репозитория?

`git status`

15. Как изменяется состояние после операций?

`git add` — файлы из modified переходят в staged.

`git commit` — файлы из staged переходят в committed.

`git push` — изменения отправляются на сервер.

16. Как синхронизировать два компьютера с GitHub?

На каждом компьютере:

`git clone https://github.com/username/repo.git`

После изменений:

`git pull` - получить обновления

`git add . && git commit -m "update" && git push` отправить свои

17. Другие сервисы, кроме GitHub?

GitLab: больше функций для CI/CD, можно hosting свой. Вэб-платформа для управления репозиториями, проектами.

Bitbucket: бесплатные private-репозитории.

18. Программы с графическим интерфейсом для Git?

GitKraken, Sourcetree, GitHub Desktop.

Вместо команд add/commit — выделяете файлы и жмете кнопки «Stage» и «Commit».

Выход:

В ходе данной работы студентом были исследованы базовые возможности системы контроля версий Git и веб-сервиса