

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

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №1
дисциплины «Программирование на Python»

Выполнила:
Беседина Инга Олеговна
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника», очная
форма обучения

(подпись)

Проверил:
Воронкин Р. А., канд. технических
наук, доцент, доцент кафедры
инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Тема: Исследование основных возможностей Git и GitHub

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

Порядок выполнения работы:

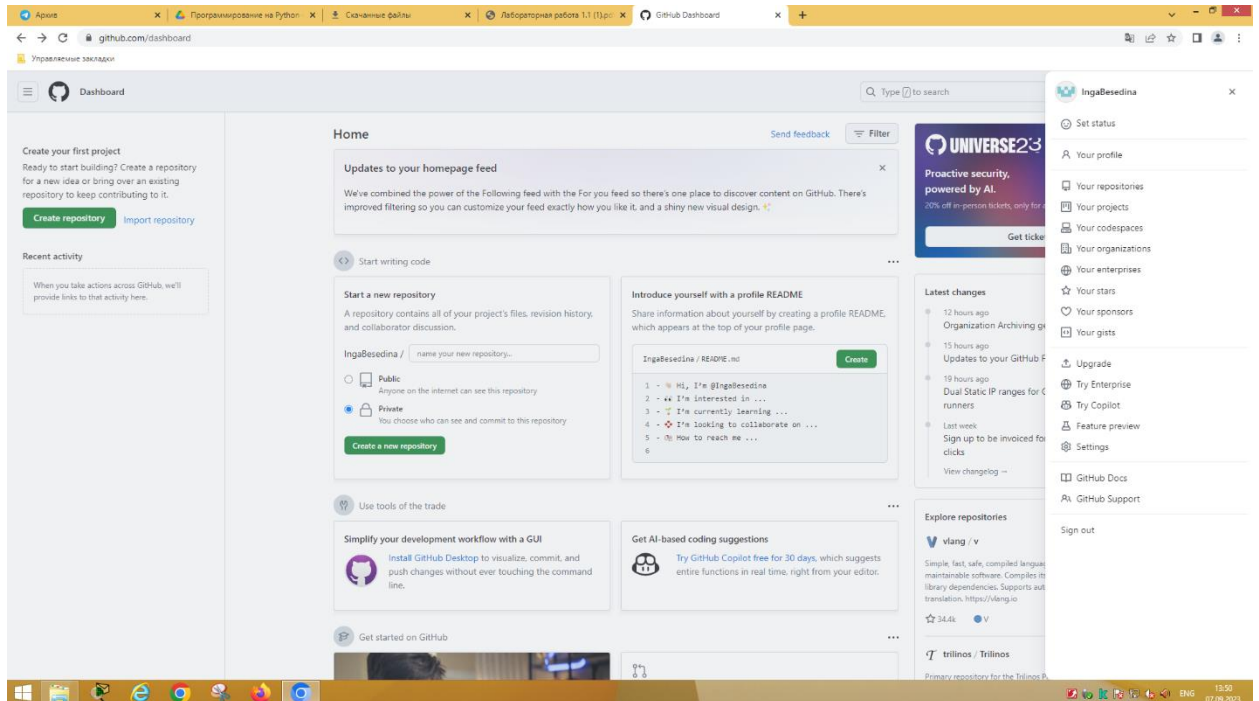


Рисунок 1. Профиль пользователя на GitHub



Рисунок 2. Добавление в настройки Git имени и адреса электронной почты

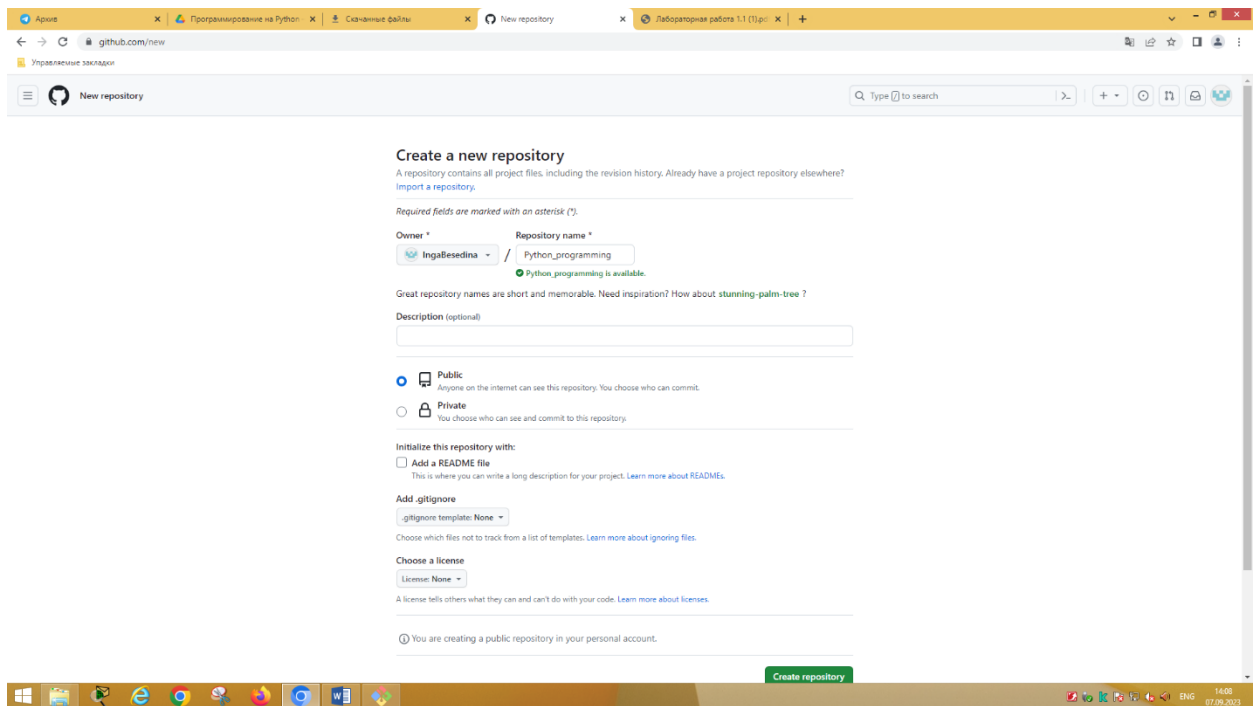


Рисунок 3. Добавление репозитория

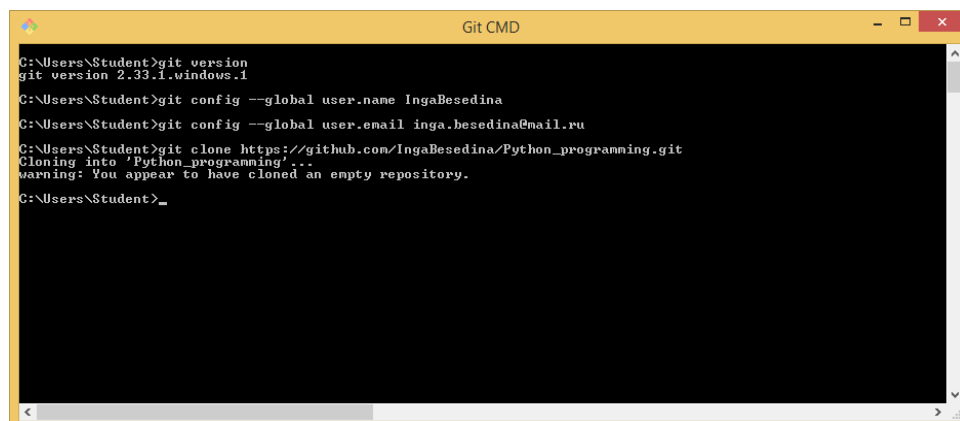


Рисунок 4. Копирование репозитория

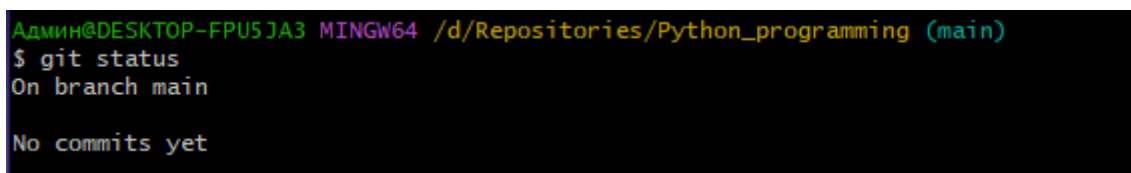


Рисунок 5. Проверка состояния репозитория

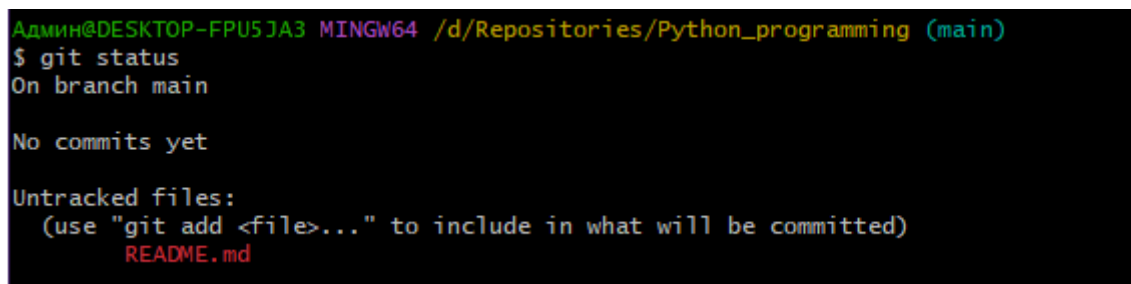


Рисунок 6. Изменения в файле README.md

```
Админ@DESKTOP-FPU5JA3 MINGW64 /d/Repositories/Python_programming (main)
$ git add README.md
```

Рисунок 7. Добавление изменений в файл

```
Админ@DESKTOP-FPU5JA3 MINGW64 /d/Repositories/Python_programming (main)
$ git commit -m "Add information about local repository in readme file"
[main (root-commit) fc32782] Add information about local repository in readme fi
le
1 file changed, 2 insertions(+)
create mode 100644 README.md
```

Рисунок 8. Коммит

```
Админ@DESKTOP-FPU5JA3 MINGW64 /d/Repositories/Python_programming (main)
$ git push --set-upstream origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 298 bytes | 298.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/IngaBesedina/Python_programming.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

Рисунок 9. Внесение изменений в исходный репозиторий с помощью push

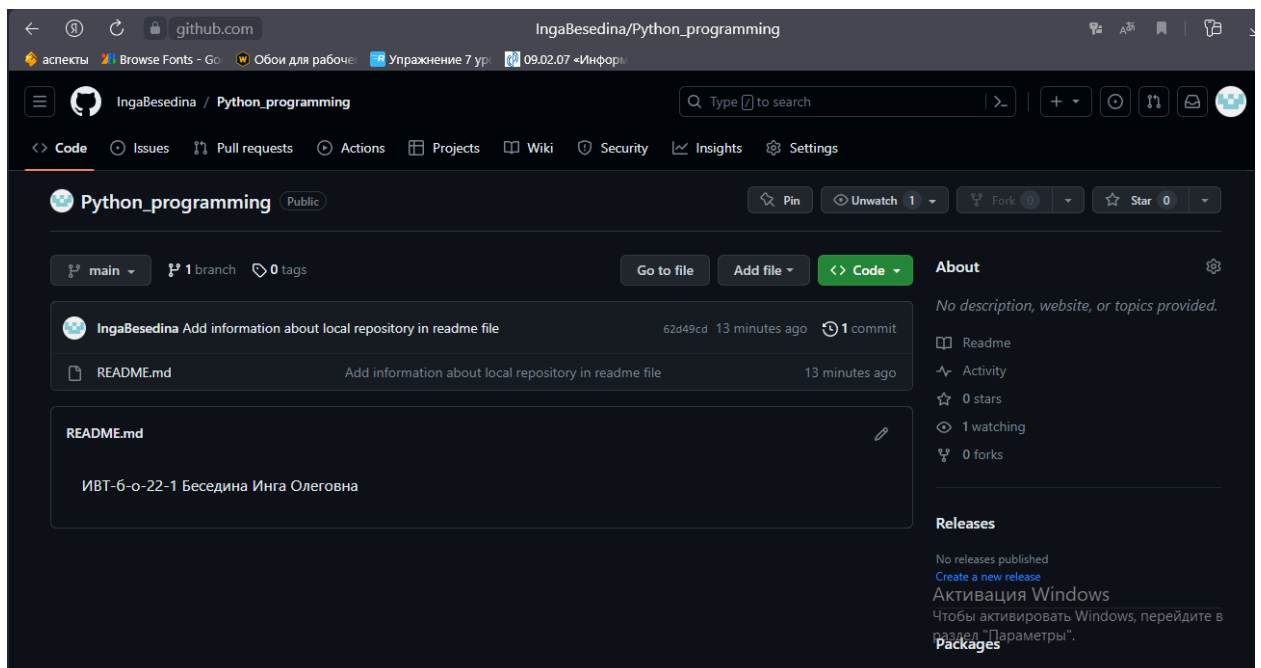


Рисунок 10. Результат изменений в репозитории

```
Админ@DESKTOP-FPU5JA3 MINGW64 /d/Repositories/PythonProgramming (main)
$ git add PythonProject/Project.py

Админ@DESKTOP-FPU5JA3 MINGW64 /d/Repositories/PythonProgramming (main)
$ git commit -m "Add python project"
[main a94cea6] Add python project
1 file changed, 2 insertions(+)
create mode 100644 PythonProject/Project.py
```

Рисунок 11. Добавление программы на языке Python

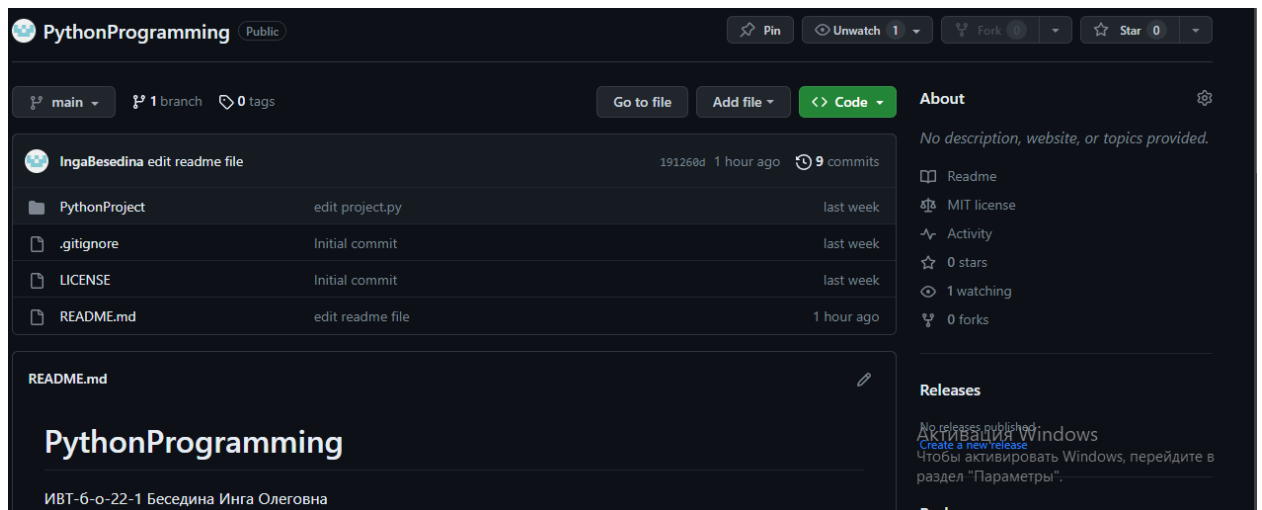


Рисунок 12. Результат внесённых изменений

Контрольные вопросы:

1. Система контроля версий (СКВ) — это система, регистрирующая изменения в одном или нескольких файлах с тем, чтобы в дальнейшем была возможность вернуться к определённым старым версиям этих файлов.

2. Недостатком локальных СКВ является возможность легко забыть, в какой директории вы находитесь, и случайно изменить не тот файл или скопировать не те файлы, которые вы хотели, также если вся история проекта хранится в одном месте, вы рискуете потерять всё. Самый очевидный минус централизованных СКВ — это единая точка отказа, представленная централизованным сервером.

3. Git относится к распределённым системам контроля версий.

4. Концептуально, большинство других систем хранят информацию в виде списка изменений в файлах. Git не хранит и не обрабатывает данные таким способом. Вместо этого, подход Git к хранению данных больше похож на набор снимков миниатюрной файловой системы. Каждый раз, когда делается коммит, система запоминает, как выглядит каждый файл в этот момент, и сохраняет ссылку на этот снимок. Для увеличения эффективности, если файлы не были изменены, Git не запоминает эти файлы вновь, а только создаёт ссылку на предыдущую версию идентичного файла, который уже сохранён.

5. В Git для всего вычисляется хеш-сумма, и только потом происходит сохранение. В дальнейшем обращение к сохранённым объектам происходит по этой хеш-сумме. Это значит, что невозможно изменить содержимое файла или директории так, чтобы Git не узнал об этом. Вы не потеряете информацию во время её передачи и не получите повреждённый файл без ведома Git

6. У Git есть три основных состояния, в которых могут находиться ваши файлы: зафиксированное (committed), изменённое (modified) и подготовленное (staged). Зафиксированный значит, что файл уже сохранён в вашей локальной базе. К изменённым относятся файлы, которые поменялись, но ещё не были зафиксированы. Подготовленные файлы — это изменённые файлы, отмеченные для включения в следующий коммит

7. Профиль пользователя - это публичная страница на GitHub

8. В GitHub бывают публичные и приватные репозитории

9. Создание локального клон удаленного репозитория; работа с ним локально; периодические внесения изменения в удаленный репозиторий.

10. Первое, что следует сделать после установки Git — указать имя и адрес электронной почты. Это важно, потому что каждый коммит в Git содержит эту информацию, и она включена в передаваемые коммиты, и не может быть далее изменена

11. Этапы создания репозитория в GitHub: в профиле пользователя нажать на кнопку добавить репозиторий; указать имя репозитория и описание(необязательно); указать публичным или приватным будет репозиторий; добавить README, .gitignore, и LICENSE; нажать кнопку создать репозиторий

12. MIT License, Apache License 2.0, GPL, LGPL, BSD License, Creative Commons, Unlicense, WTFPL.

13. Открыть командную строку или терминал и перейти в каталог, в который нужно скопировать хранилище. Затем написать `git clone` и ввести

адрес. Клонировать репозиторий нужно для того чтобы работать с файлами локально.

14. Проверить состояние локального репозитория можно командой `git status`

15. После добавления/изменения файла в локальный репозиторий Git состояние репозитория изменится на "изменено", что означает, что файл еще не был добавлен в индекс. После добавления нового/измененного файла под версионный контроль с помощью команды `git add` состояние репозитория изменится на "готово к коммиту", что означает, что файл был добавлен в индекс и готов к фиксации. После фиксации (коммита) изменений с помощью команды `git commit` состояние репозитория изменится на "готово к отправке", что означает, что изменения были зафиксированы в локальном репозитории. После отправки изменений на сервер с помощью команды `git push` состояние репозитория будет синхронизировано с удаленным репозиторием и будет отображаться как "последняя версия".

16. С помощью команды `git clone` клонировать репозиторий с GitHub на первый компьютер. Внести изменения в локальный репозиторий на первом компьютере и зафиксировать их с помощью команд `git add` и `git commit`. Отправить изменения на сервер с помощью команды `git push`. На втором компьютере выполнить команду `git pull`, чтобы получить изменения с сервера. Теперь оба локальных репозитория будут находиться в синхронизированном состоянии с удаленным репозиторием на GitHub.

17. Один из основных конкурентов GitHub - это GitLab. Оба сервиса предоставляют возможности для хранения и управления репозиториями Git, а также инструменты для совместной работы над проектами. Однако есть несколько отличий между ними. GitLab более ориентирован на корпоративные клиенты, чем GitHub. Он предоставляет больше инструментов для управления процессами разработки, таких как возможность создания задач и отслеживания их выполнения. Это делает GitLab более подходящим для крупных команд разработчиков и компаний.

18. Некоторые известные программные средства с графическим интерфейсом для работы с Git включают в себя: GitHub Desktop - официальный клиент GitHub для Windows и macOS; GitKraken - кроссплатформенный клиент Git с поддержкой GitHub, GitLab и Bitbucket; SourceTree - бесплатный клиент Git для Windows и macOS, разработанный компанией Atlassian; TortoiseGit - бесплатный клиент Git для Windows с интеграцией в проводник операционной системы; GitExtensions - бесплатный клиент Git для Windows с поддержкой GitHub и GitLab.

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