

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 №2
дисциплины «Основы программной инженерии»

Выполнила:
Ламская Ксения Вячеславовна
2 курс, группа ПИЖ-б-о-22-1,
09.03.04 «Программная инженерия»,
направленность (профиль) «Разработка и
сопровождение программного обеспечения»,
очная форма обучения

(подпись)

Доцент кафедры инфокоммуникаций
Воронкин Роман Александрович

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Тема: Исследование возможностей Git для работы с локальными репозиториями.

Цель работы: исследовать базовые возможности системы контроля версий Git для работы с локальными репозиториями.

Порядок выполнения работы

1. Создала репозиторий GitHub.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk ().*

Owner * / Repository name *

✔ 22 is available.

Great repository names are short and memorable. Need inspiration? How about [cautious-enigma](#) ?

Description (optional)

☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

☒ **Add a README file**
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

This will set **main** as the default branch. Change the default name in your [settings](#).

You are creating a public repository in your personal account.

[Create repository](#)

Рисунок 1- Создание репозитория


```

C:\Ksen\chatting>git log --stat
commit e6bbb6e9bce565173500411826493c4c444b0d3 (HEAD -> main, origin/main, origin/HEAD)
Author: Ksenia <klamskaa@gmail.com>
Date:   Wed Sep 13 12:32:00 2023 +0300

    changee

...0\321\217\320\232\320\222_\320\233\320\2401.1.pdf" | Bin 0 -> 1481045 bytes
1 file changed, 0 insertions(+), 0 deletions(-)

commit ee8422c4a37915b4aef51f4a640b8fc9d43fd981
Author: Ksenia <klamskaa@gmail.com>
Date:   Wed Sep 13 12:11:34 2023 +0300

    new README

README.md | 1 +
1 file changed, 1 insertion(+)

commit 6dbbef85191423075147abb0456e6add12e1d31a
Author: Ksenia <klamskaa@gmail.com>
Date:   Wed Sep 13 12:05:10 2023 +0300

    six

chat.py | 5 +++++
1 file changed, 5 insertions(+)

commit 6624a82f7c241e682d524f631f56d8ad84c219ce
Author: Ksenia <klamskaa@gmail.com>

```

Рисунок 2.3 – Сокращенная статистика для каждого коммита

```

C:\Ksen\chatting>git log --pretty=oneline
e6bbb6e9bce565173500411826493c4c444b0d3 (HEAD -> main, origin/main, origin/HEAD) changee
ee8422c4a37915b4aef51f4a640b8fc9d43fd981 new README
6dbbef85191423075147abb0456e6add12e1d31a six
6624a82f7c241e682d524f631f56d8ad84c219ce five
385545bf76191b38f2ac9add52027cd3cbfc912 fours
81d2591fd53223a4374114975eb8a7187527416a third
81cfa7cf382fd306cf5c5256c78ac63ed6547883 second
2c745e959e67987331b6299a4d0a3dc1a8011c27 second
ea362e7aec28e78764aa395b010c8b4053f2ab4d first
6030841df1489d04f599968c85b3ce85913cf95f change
4e8fe196396b2b267e6c2b5674e0273c9785474a Update README.md
630245a40e2f12c3d85209eb319a6e5ff5fbd533 Update README.md
60a3b93c7842084f290b9e33b6b37624cee14111 Create README.md
1420024c372c3fbbcb7376d2f1d2c81a0530afc1 Initial commit

```

Рисунок 2.4 – Использование команды git log с аргументом --pretty

```

C:\Ksen\chatting>git remote -v
origin  https://github.com/ksenia-lamskaya/chatting.git (fetch)
origin  https://github.com/ksenia-lamskaya/chatting.git (push)

```

Рисунок 2.5 – Просмотр удаленных репозиторийев

```
C:\Ksen\chatting>git remote add pb https://github.com/paulboone/ticgit  
  
C:\Ksen\chatting>git remote -v  
origin  https://github.com/ksenia-lamskaya/chatting.git (fetch)  
origin  https://github.com/ksenia-lamskaya/chatting.git (push)  
pb      https://github.com/paulboone/ticgit (fetch)  
pb      https://github.com/paulboone/ticgit (push)
```

Рисунок 2.6 – Добавление удаленного репозитория

```
C:\Ksen\chatting>git tag  
v1.0
```

Рисунок 2.7 – Просмотр списка тэгов

```
C:\Ksen\chatting>git tag -a v1.4 -m "my version 1.4"  
  
C:\Ksen\chatting>git tag  
v1.0  
v1.4
```

Рисунок 2.8 – Добавление тэга

3. Выполнила клонирование репозитория на рабочий компьютер.

```
C:\Ksen>git clone https://github.com/ksenia-lamskaya/2.git  
Cloning into '2'...  
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  
Receiving objects: 100% (5/5), done.
```

Рисунок 3 – Клонирование репозитория

4. Добавила академические данные в файл README.

```
# 2
Основы программной инженерии
Ламская Ксения Вячеславовна ПИЖ-6-о-22-1
```

Рисунок 4 – Измененный файл README

5. Написала небольшую программу на языке Python.

```
C:\Ksen > 2 > 2.py > ...
1 my_list = ["leaf", "cherry", "fish"]
2
3 my_list1 = ["D", "C", "B", "A"]
4 my_list2 = [1, 2, 3, 4, 5]
5
6 my_list.sort() # ['cherry', 'fish', 'leaf']
7 my_list1.sort() # ['A', 'B', 'C', 'D']
8
9 print(sorted(my_list2, reverse=True)) # [5, 4, 3, 2, 1]
```

Рисунок 5 – Написанная программа

6. Просмотрела историю коммитов.

```
C:\Ksen\2>git log
commit 705dc7c30e6b7cd665d7980ee7972c036b3c29d4 (HEAD -> main, origin/main, origin/HEAD)
Author: Ksenia <klamskaa@gmail.com>
Date: Tue Sep 19 21:00:18 2023 +0300

    sevens

commit 790f2002545808de10b9966f2db4af54cfaaa806
Author: Ksenia <klamskaa@gmail.com>
Date: Tue Sep 19 20:58:41 2023 +0300

    sixs

commit b087b51a24bb73aa594f3b30e434792a4b029abb
Author: Ksenia <klamskaa@gmail.com>
Date: Tue Sep 19 20:57:36 2023 +0300

    fives

commit 2cba8b3c8e0aae03a56234c1553f4a91f2bdd9c6
Author: Ksenia <klamskaa@gmail.com>
Date: Tue Sep 19 20:56:04 2023 +0300

    fours

commit 8c1f146c70bd42d681b9b37e68494cff27e4f338
Author: Ksenia <klamskaa@gmail.com>
Date: Tue Sep 19 20:54:45 2023 +0300

    third
```

Рисунок 6 – История коммитов

7. Просмотрела содержимое коммитов командой git show.

```
C:\Ksen\2>git show HEAD
commit 705dc7c30e6b7cd665d7980ee7972c036b3c29d4 (HEAD -> main, origin/main, origin/HEAD)
Author: Ksenia <klamskaa@gmail.com>
Date: Tue Sep 19 21:00:18 2023 +0300

    sevens

diff --git a/2.py b/2.py
index df05b0e..c9d971e 100644
--- a/2.py
+++ b/2.py
@@ -2,6 +2,7 @@ my_list = ["leaf", "cherry", "fish"]

my_list1 = ["D","C","B","A"]
my_list2 = [1,2,3,4,5]
+
my_list.sort() # ['cherry', 'fish', 'leaf']
my_list1.sort() # ['A', 'B', 'C', 'D']
```

Рисунок 7.1 – Содержимое последнего коммита

```
C:\Ksen\2>git show HEAD~1
commit 790f2002545808de10b9966f2db4af54cf8aaa806
Author: Ksenia <klamskaa@gmail.com>
Date: Tue Sep 19 20:58:41 2023 +0300

    sixs

diff --git a/2.py b/2.py
index 235f4ef..df05b0e 100644
--- a/2.py
+++ b/2.py
@@ -1,8 +1,5 @@
my_list = ["leaf", "cherry", "fish"]

-
-
-
my_list1 = ["D","C","B","A"]
my_list2 = [1,2,3,4,5]
my_list.sort() # ['cherry', 'fish', 'leaf']
```

Рисунок 7.2 – Содержимое предпоследнего коммита

```

C:\Ksen\2>git show b087b51a24bb73aa594f3b30e434792a4b029abb
commit b087b51a24bb73aa594f3b30e434792a4b029abb
Author: Ksenia <klamskaa@gmail.com>
Date:   Tue Sep 19 20:57:36 2023 +0300

    fives

diff --git a/2.py b/2.py
index c5ac2f6..235f4ef 100644
--- a/2.py
+++ b/2.py
@@ -1,4 +1,8 @@
 my_list = ["leaf", "cherry", "fish"]
+
+
+
+
 my_list1 = ["D","C","B","A"]
 my_list2 = [1,2,3,4,5]
 my_list.sort() # ['cherry', 'fish', 'leaf']

```

Рисунок 7.3 – Содержимое коммита с указанным хэшем

8. Удалил весь код из файла index.py репозитория и сохранил изменения. Далее удалил все несохраненные изменения в файле командой: «git checkout – 2.py». Снова удалил код из файла index.py и сохранил изменения(рис.8.1-8.2).

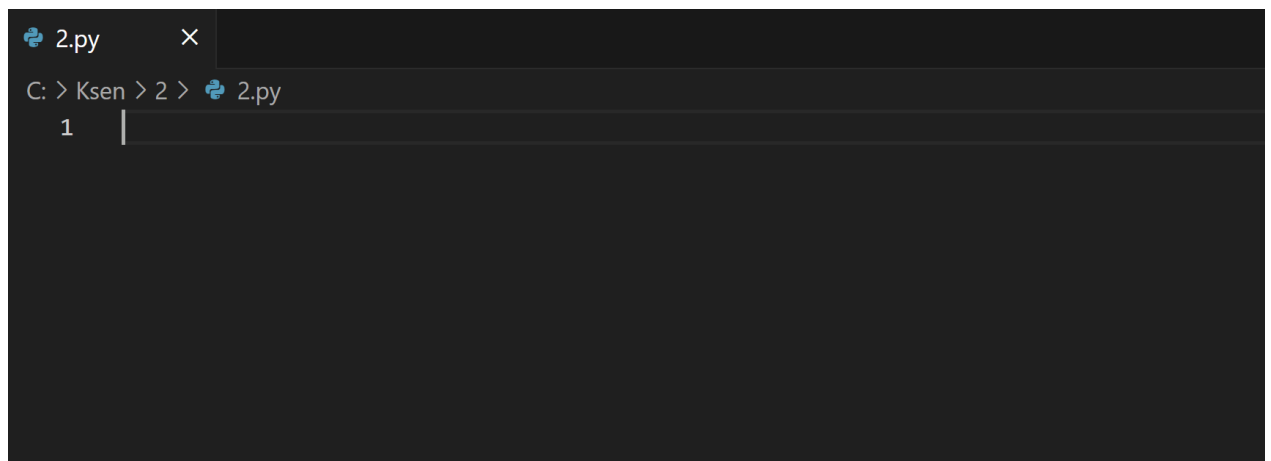


Рисунок 8.1 - Результат изменений файла 2.py

Далее откатился состояние хранилища к предыдущей версии командой: «git reset --hard HEAD~1».



```
2.py
C: > Ksen > 2 > 2.py > ...
1 my_list = ["leaf", "cherry", "fish"]
2
3 my_list1 = ["D", "C", "B", "A"]
4 my_list2 = [1, 2, 3, 4, 5]
5
6 my_list.sort() # ['cherry', 'fish', 'leaf']
7 my_list1.sort() # ['A', 'B', 'C', 'D']
8
9 print(sorted(my_list2, reverse=True)) # [5, 4, 3, 2, 1]
```

Рисунок 8.2 — Результат изменений файла 2.py

Ответы на контрольные вопросы

1. Как выполнить историю коммитов в Git? Какие существуют дополнительные опции для просмотра истории коммитов?

С помощью команды `git log`. Существуют множество дополнительных опций вот некоторые из них:

- `-p` или `-patch` – показывает разницу, внесенную в каждый коммит;
- `--stat` – позволяет увидеть сокращенную статистику;
- `--pretty` – эта опция меняет формат вывода;
- `--since` и `--until` – опции для ограничения вывода по времени

2. Как ограничить вывод при просмотре истории коммитов?

С помощью команды `git log` и его аргументов, например `--since`.

3. Как внести изменения в уже сделанный коммит?

С помощью команды `git commit --amend`. Отмена может потребоваться, если коммит сделан слишком рано, например, забыли добавить какие-то файлы или комментарий к коммиту.

4. Как отменить индексацию файла в Git?

С помощью команды `git reset HEAD`.

5. Как отменить изменения в файле?

С помощью команды `git checkout -- <file>`.

6. Что такое удаленный репозиторий Git?

Удалённые репозитории представляют собой версии вашего проекта, сохранённые в интернете или ещё где-то в сети.

7. Как выполнить просмотр удаленных репозиториях данного локального репозитория?

С помощью команды `git remote -v`.

8. Как добавить удаленный репозиторий для данного локального репозитория?

Для того, чтобы добавить удалённый репозиторий и присвоить ему имя (shortname), просто выполните команду `git remote add <shortname> <url>`.

9. Как выполнить отправку/получение изменений с удаленного репозитория?

Получение изменение – `git fetch [remote-name]`, отправка изменений `git push <remote-name> <branch-name>`.

10. Как выполнить просмотр удаленного репозитория?

Если нужно получить побольше информации об одном из удалённых репозиториях, вы можете использовать команду `git remote show <remote>`. Выполнив эту команду с некоторым именем, например, `origin`.

11. Каково назначение тэгов Git?

Как и большинство СКВ, Git имеет возможность пометить определённые моменты в истории как важные. Как правило, эта функциональность используется для отметки моментов выпуска версий (v1.0, и т. п.). Такие пометки в Git называются тэгами.

12. Как осуществляется работа с тэгами Git?

Для просмотра тэгов – `git tag`, для создания аннотированного тэга – `git tag -a v1.4 -m “сообщение”`, для отправки тэга на удаленный сервер – `git push origin <tagname>`, для удаления тэгов – `git tag -d <tagname>`.

13. . Самостоятельно изучите назначение флага --prune в командах git fetch и git push . Каково назначение этого флага?

Флаг `--prune` в командах `git fetch` и `git push` используется для удаления удаленных веток или тэгов, которые больше не существуют на удаленном репозитории. Если выполнить команду `git fetch --prune`, Git удалит локальные ссылки на удаленные ветки и тэги, которые были удалены на удаленном репозитории. Если выполнить команду `git push --prune`, Git удалит удаленные ветки и тэги на удаленном репозитории, которых больше нет в вашем локальном репозитории.