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

Институт перспективной инженерии
Департамент цифровых, робототехнических систем и электроники

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №2
дисциплины
«Основы кроссплатформенного программирования»

Выполнил:
Хохлачев Вадим Александрович
2 курс, группа ИТС-б-о-23-1,
11.03.02«Инфокоммуникационные технологии и системы связи», очная форма обучения

_____
(подпись)

Проверил:
Доцент департамента цифровых, робототехнических систем и электроники Воронкин Р.А.

_____
(подпись)

Отчет защищен с оценкой _____ Дата защиты_____

Ставрополь, 2024 г.

Тема: Исследование возможностей Git для работы с локальными репозиториями.

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

Порядок выполнения работы:

1. Изучил теоретический материал.

2. Приступил к выполнению заданий.

3. Создал общедоступны репозиторий.

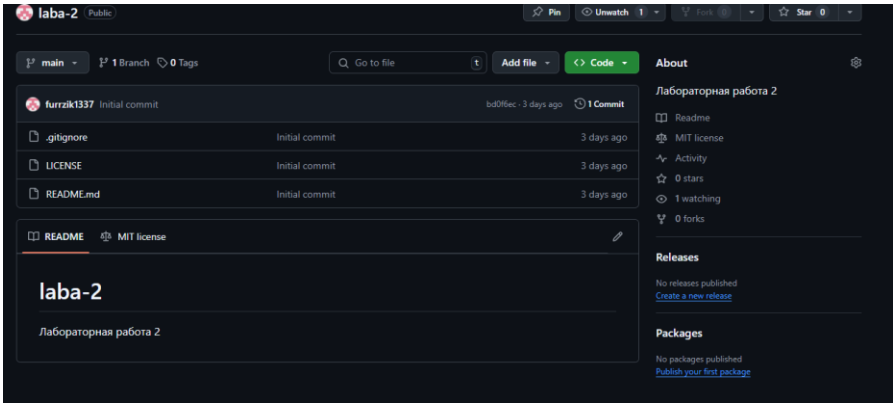

Рисунок 1. Репозиторий

4. Проработал примеры лабораторной работы.

5. Выполнил клонирование репозитория на компьютер.

Рисунок 2. Клонирование

6. Дополнил файл .gitignore необходимыми правилами.

7. Добавил в файл README.md дополнительную информацию.

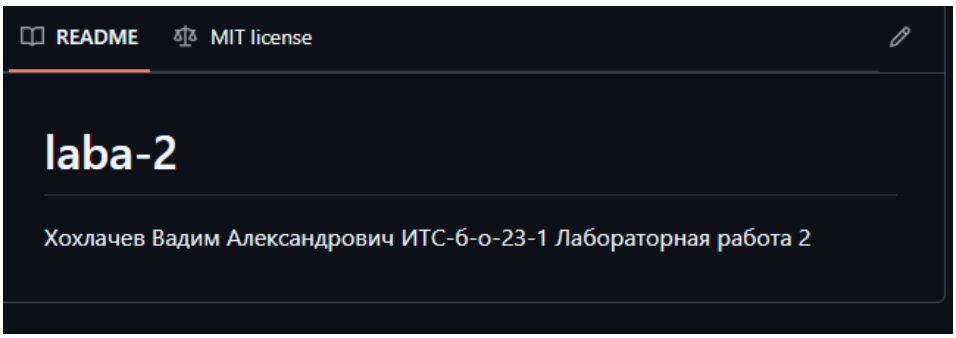

Рисунок 3. Дополнительная информация

8. Написал небольшую программу и фиксировал изменения.

Рисунок 4. Python2.py является программой

9. Добавил тег.

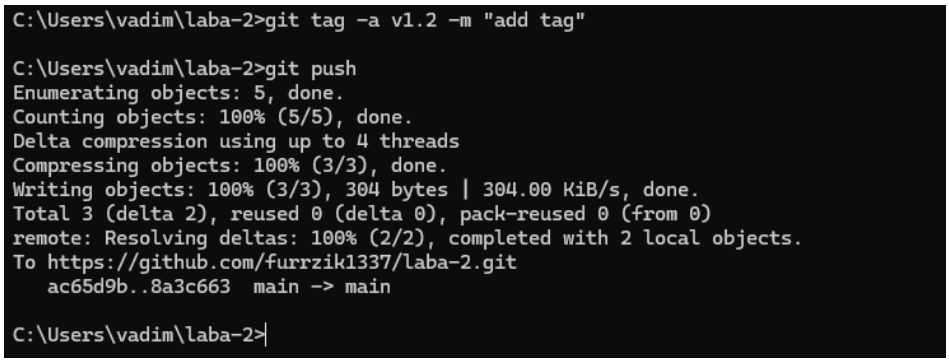

Рисунок 5. Добавление тега

10. Сделал не менее 7 коммитов.

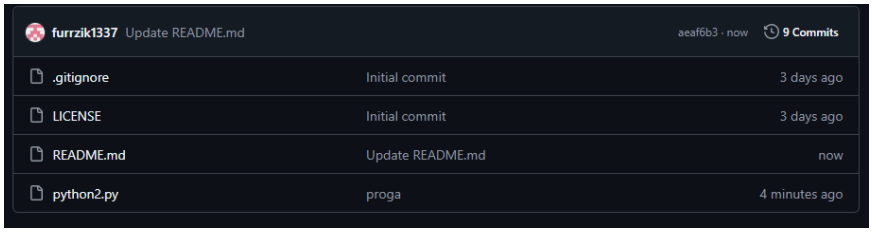

Рисунок 6. Коммиты

11. Посмотрел историю хранилища командой git log

```
C:\Users\vadim\laba-2>git log --graph --pretty=oneline --abbrev-commit
* 8a3c663 (HEAD -> main, tag: v1.2, origin/main, origin/HEAD) proga
* ac65d9b (tag: v1.1) proga
* 8da1fe4 bio
* 97b4992 da
* bd0f6ec Initial commit

C:\Users\vadim\laba-2>
```

Рисунок 7. История хранилища

12. Посмотрел содержимое коммитов командой git show head

```
C:\Users\vadim\laba-2>git show head
commit 8a3c6636ef0d9a4d765845b064e723277bcde8e3 (HEAD -> main, tag: v1.2, origin/main,
 origin/HEAD)
Author: furrzik <vadimhohlacev@gmail.com>
Date:   Mon Oct 7 20:37:03 2024 +0300

    proga

diff --git a/python2.py b/python2.py
index c47564a..6335a0e 100644
--- a/python2.py
+++ b/python2.py
@@ -5,7 +5,7 @@ y = 3+sqrt(9-x^2) -3<=x<3

 y =kx+b

-прямая после окружности -2x - 9 ; 3<=x<6
+вторая прямая  -2x - 9 ; 3<=x<6


 последняя прямая y =kx+b

C:\Users\vadim\laba-2>
```

Рисунок 8. Последний коммит

```
C:\Users\vadim\laba-2>git show head~1
commit ac65d9b9da720ed80406e9627d2e36b68abb71c7 (tag: v1.1)
Author: furrzik <vadimhohlacev@gmail.com>
Date:   Mon Oct 7 20:32:29 2024 +0300

    proga

diff --git a/python2.py b/python2.py
new file mode 100644
index 0000000..c47564a
--- /dev/null
+++ b/python2.py
@@ -0,0 +1,26 @@
+'''прямая y = 3 (x<-3)
+
+окружность (x-0)^2+(y-3)^2=9
+y = 3+sqrt(9-x^2) -3<=x<3
+
+y =kx+b
+
+прямая после окружности -2x - 9 ; 3<=x<6
+
+
+последняя прямая y =kx+b
+
+x-9; 6<=x<=11'''
+
+from math import *
+x = float (input('x='))
:
commit ac65d9b9da720ed80406e9627d2e36b68abb71c7 (tag: v1.1)
Author: furrzik <vadimhohlacev@gmail.com>
Date:   Mon Oct 7 20:32:29 2024 +0300

    proga
```

Рисунок 9. Предпоследний коммит

Рисунок 10. Коммит с указанным хэшем

13. Освоил возможность отката к заданной версии. Удалил весь код из одного из файлов программы репозитория и сохранил этот файл.

14. Удалил все несохраненные изменения в файле командой git checkout.

15. Откатил состояние хранилища к предыдущей версии.

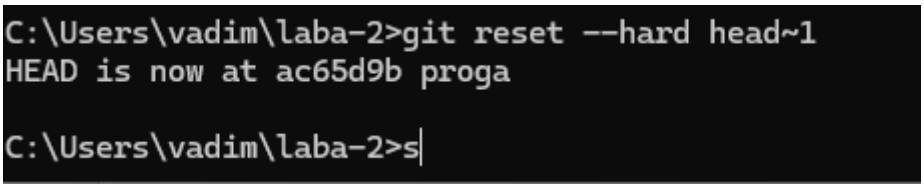

Рисунок 11. Откат

Вывод: в результате использования команды git reset –hard отменяется последние коммиты и сбрасывается рабочая копия до определенного состояния.

Ответы на контрольные вопросы:

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

– Для просмотра истории коммитов в Git используется команда git log. Она отображает список коммитов в хронологическом порядке, начиная с самого последнего.

2. Как ограничить вывод при просмотре истории коммитов?

– Для ограничения вывода при просмотре истории коммитов в Git можно использовать несколько опций:

- Ограничение количества коммитов;
- Фильтрация по времени;
- Фильтрация по автору;
- Фильтрация по сообщению коммита;
- Фильтрация по файлам;
- Ограничение вывода по формату.

3. Как внести изменения в уже сделанный коммит?

– В Git есть несколько способов внести изменения в уже сделанный коммит:

- git commit –amend;
- git rebase -i <имя_коммита>;
- git revert <имя_коммита>;
- git reset <имя_коммита>.

4. Как отменить индексацию файла в Git?

– git checkout. Эта команда отменит все изменения, внесенные в файлы, вернув их к состоянию в последнем коммите.

– git reset HEAD <имя_файла>. Эта команда удалит файл из индекса, но оставит его в рабочей области.

5. Как отменить изменения в файле?

– В Git есть несколько способов отменить изменения в файле:

- git checkout -- <имя_файла>;
- git reset HEAD <имя_файла>;
- git restore <имя_файла>;
- git revert <имя_коммита>;
- git stash.

6. Что такое удаленный репозиторий Git?

– Удаленный репозиторий Git – это хранилище кода, доступное с разных компьютеров через сеть, например, через интернет.

7. Как выполнить просмотр удаленных репозиториев данного локального репозитория?

– Для просмотра удаленных репозиториев, связанных с вашим локальным репозиторием Git, используйте команду git remote. Эта команда выведет список имен удаленных репозиториев, которые вы добавили к своему локальному репозиторию.

8. Как добавить удаленный репозиторий для данного локального репозитория?

– Чтобы добавить удаленный репозиторий к вашему локальному репозиторию Git, используйте команду git remote add: git remote add <имя_удаленного_репозитория> <URL_удаленного_репозитория>

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

– Отправка изменений (push): git push <имя_удаленного_репозитория> <ветвь>: Эта команда отправляет ваши локальные изменения в указанную ветвь на удаленный репозиторий.

– Получение изменений (pull):git pull <имя_удаленного_репозитория> <ветвь>: Эта команда получает изменения из указанной ветви на удаленном репозитории и объединяет их с вашей локальной веткой.

10. Как выполнить просмотр удаленного репозитория?

– С помощью команды git log;

– С помощью команды git fetch;

– С помощью команды git remote show;

– С помощью графических интерфейсов Git.

11. Каково назначение тэгов Git?

– Тэги Git – это метки, которые позволяют помечать определенные коммиты в репозитории. Они служат для идентификации важных моментов в

истории проекта, таких как релизы, версии или отправные точки для ветвления.

12. Как осуществляется работа с тэгами Git?

– Создание тега;

– Просмотр тегов;

– Перемещение тега;

– Удаление тега;

– Отправка тегов на удаленный репозиторий;

– Просмотр информации о теге;

– Переключение на коммит, помеченный тегом.

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, которые уже не существуют на удаленном сервере.

Вывод: в ходе работы исследовал базовые возможности системы контроля версий Git для работы с локальными репозиториями.