

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»

Выполнил:

Звездин Алексей Сергеевич
2 курс, группа ПИЖ-б-о-22-1,
09.03.04 «Программная инженерия»,
направленность (профиль) «Разработка
и сопровождение программного
обеспечения», очная форма обучения

(подпись)

Руководитель практики:

Воронкин Р. А., доцент кафедры
инфокоммуникаций

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Ход работы

1. Я изучил теоретический материал работы

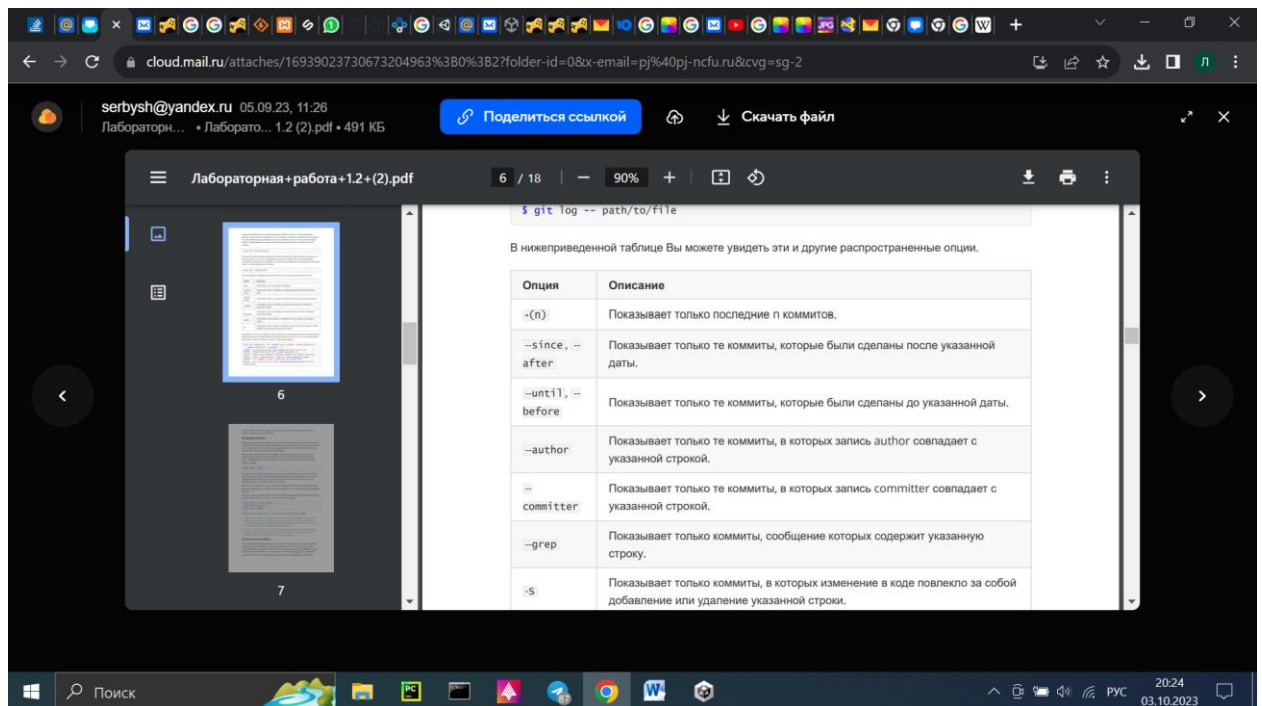


Рисунок 1.1 – Изучение материала для лабораторной работы

2. Создал общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT и выбранный мною язык программирования

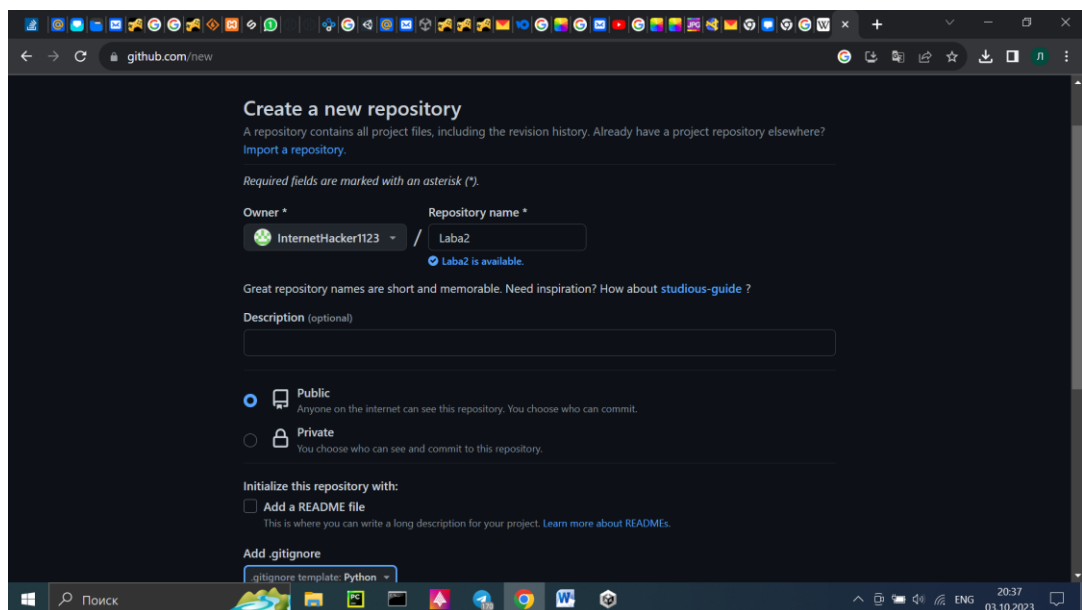


Рисунок 2.1 – Настройка репозитория

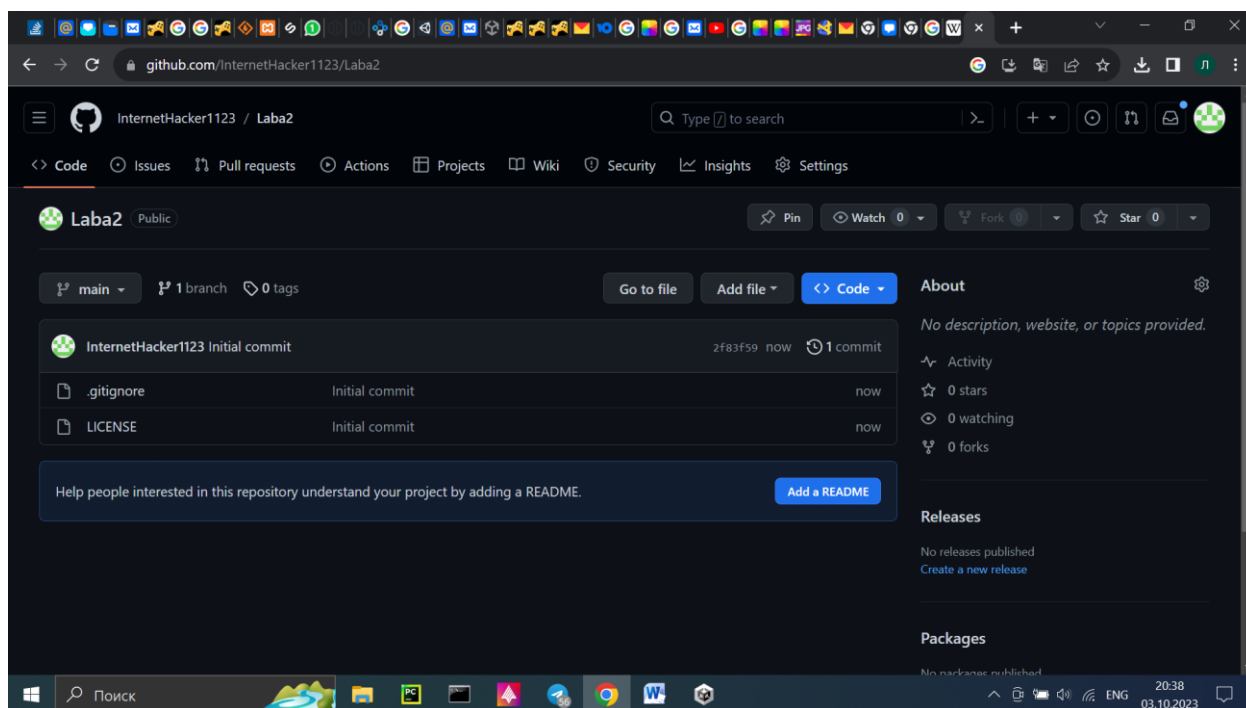


Рисунок 2.2 – Готовый репозиторий

3. Проработал примеры лабораторной работы

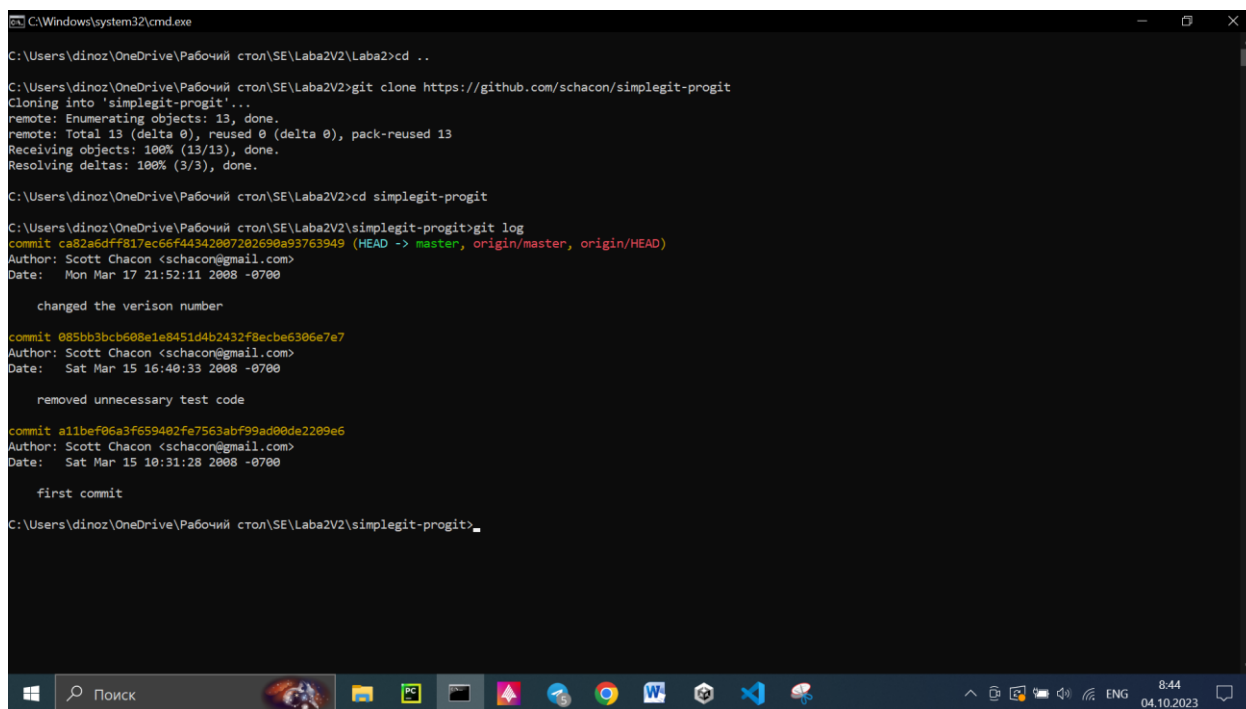


Рисунок 3.1 – Проработка примеров

4. Выполнил клонирование созданного репозитория на рабочий компьютер

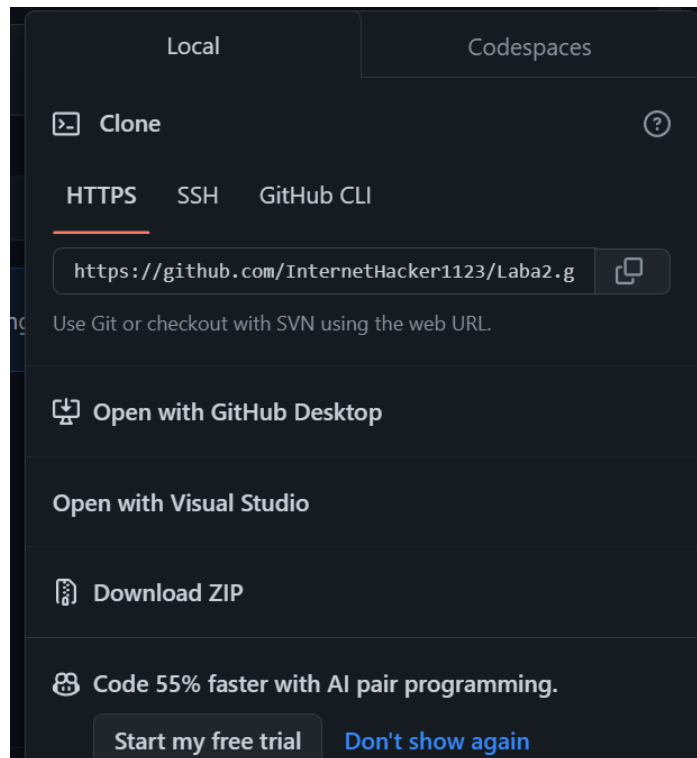


Рисунок 4.1 – Копирование ссылки репозитория

```
C:\Windows\system32\cmd.exe

C:\Users\dinoz\OneDrive\Рабочий стол>git clone https://github.com/InternetHacker1123/Laba2.git
Cloning into 'Laba2'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (4/4), done.

C:\Users\dinoz\OneDrive\Рабочий стол>dir
Том в устройстве C имеет метку Windows-SSD
Серийный номер тома: A243-A740

Содержимое папки C:\Users\dinoz\OneDrive\Рабочий стол\SE

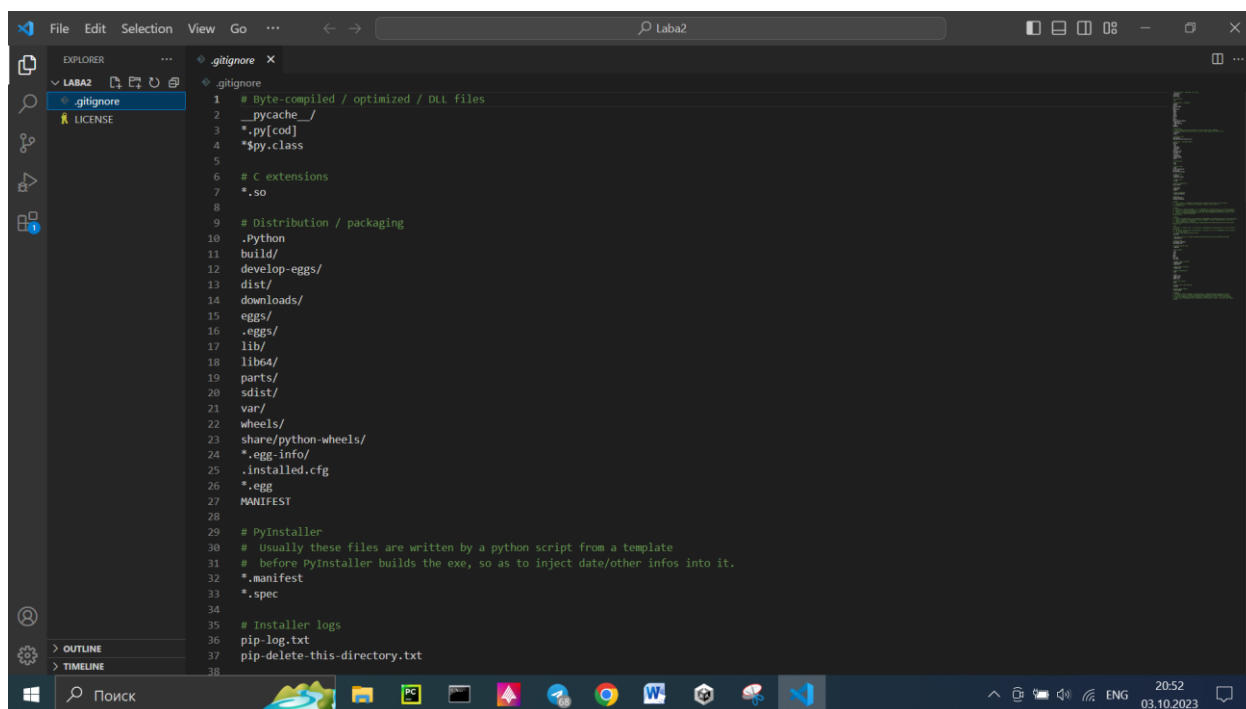
03.10.2023  20:45    <DIR>        .
03.10.2023  20:45    <DIR>        ..
03.10.2023  20:45    <DIR>        Laba2
06.09.2023  12:16    <DIR>        SETest
              0 файлов              0 байт
              4 папок  285 854 535 680 байт свободно

C:\Users\dinoz\OneDrive\Рабочий стол>cd Laba2
C:\Users\dinoz\OneDrive\Рабочий стол\SE\Laba2>dir
Том в устройстве C имеет метку Windows-SSD
Серийный номер тома: A243-A740

Содержимое папки C:\Users\dinoz\OneDrive\Рабочий стол\SE\Laba2
```

Рисунок 4.2 – Копирование репозитория на рабочий компьютер с помощью командной строки

5. Дополнил файл .gitignore необходимыми правилами для выбранного языка программирования и интегрированной среды разработки



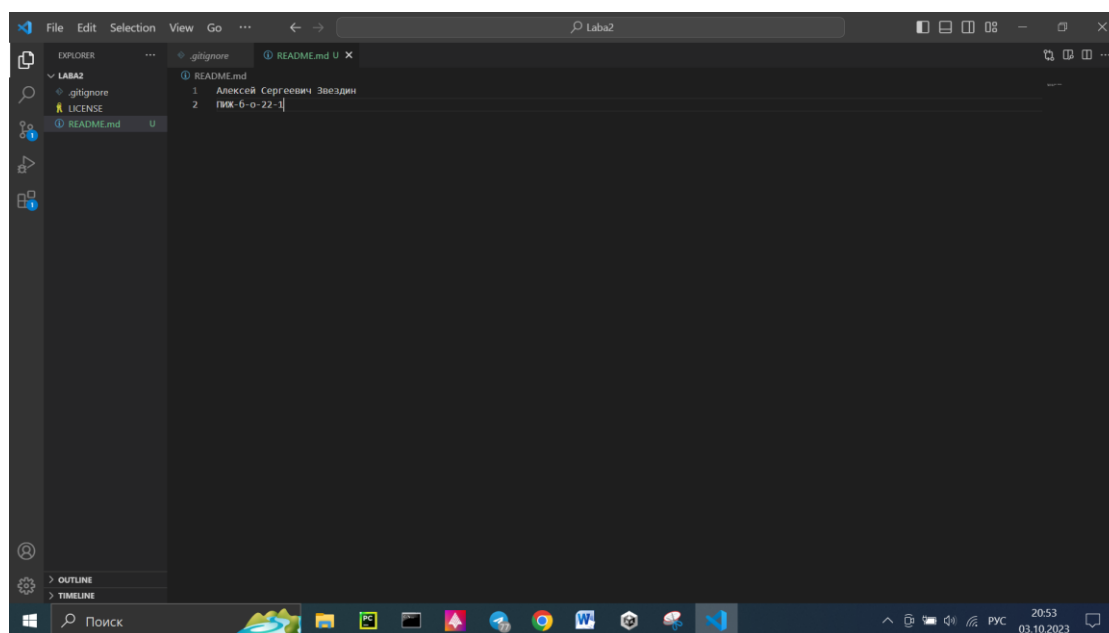
The screenshot shows the Visual Studio Code editor with the .gitignore file open. The Explorer sidebar on the left shows the project structure with LABA2, .gitignore, and LICENSE. The main editor area displays the following content in the .gitignore file:

```
1 # Byte-compiled / optimized / DLL files
2 __pycache__
3 *.py[co]
4 *.py.class
5
6 # C extensions
7 *.so
8
9 # Distribution / packaging
10 .Python
11 build/
12 develop-eggs/
13 dist/
14 downloads/
15 eggs/
16 .eggs/
17 lib/
18 lib64/
19 parts/
20 sdist/
21 var/
22 wheels/
23 share/python-wheels/
24 *.egg-info/
25 .installed.cfg
26 *.egg
27 MANIFEST
28
29 # PyInstaller
30 # Usually these files are written by a python script from a template
31 # before PyInstaller builds the exe, so as to inject date/other infos into it.
32 *.manifest
33 *.spec
34
35 # Installer logs
36 pip-log.txt
37 pip-delete-this-directory.txt
38
```

The Windows taskbar at the bottom shows the time as 20:52 on 03.10.2023.

Рисунок 5.1 – Файл с необходимыми правилами для языка Python

6. Добавил в файл README.md информацию о своей группе и моём ФИО



The screenshot shows the Visual Studio Code editor with the README.md file open. The Explorer sidebar on the left shows the project structure with LABA2, .gitignore, LICENSE, and README.md. The main editor area displays the following content in the README.md file:

```
1 Алексей Сергеевич Звездин
2 ПИК-6-0-22-11
```

The Windows taskbar at the bottom shows the time as 20:53 on 03.10.2023.

Рисунок 6.1 – Локально создаю файл README.md

```
C:\Windows\system32\cmd.exe
fatal: Authentication failed for 'https://github.com/InternetHacker1123/Laba2.git/'

C:\Users\dinoz\OneDrive\Рабочий стол\SE\Laba2V2\Laba2>git push
Username for 'https://github.com': InternetHacker1123
Password for 'https://InternetHacker1123@github.com':
remote: Support for password authentication was removed on August 13, 2021.
remote: Please see https://docs.github.com/en/get-started/getting-started-with-git/about-remote-repositories#cloning-with-https-urls for information on currently recommended modes of authentication.
fatal: Authentication failed for 'https://github.com/InternetHacker1123/Laba2.git/'

C:\Users\dinoz\OneDrive\Рабочий стол\SE\Laba2V2\Laba2>git config --global credential.helper manager

C:\Users\dinoz\OneDrive\Рабочий стол\SE\Laba2V2\Laba2>git push
C:\Users\dinoz\OneDrive\Рабочий стол\SE\Laba2V2\Laba2>
C:\Users\dinoz\OneDrive\Рабочий стол\SE\Laba2V2\Laba2>
C:\Users\dinoz\OneDrive\Рабочий стол\SE\Laba2V2\Laba2>
C:\Users\dinoz\OneDrive\Рабочий стол\SE\Laba2V2\Laba2>git add .
C:\Users\dinoz\OneDrive\Рабочий стол\SE\Laba2V2\Laba2>git commit -m "Readme add"
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)

nothing to commit, working tree clean

C:\Users\dinoz\OneDrive\Рабочий стол\SE\Laba2V2\Laba2>git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 383 bytes | 383.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/InternetHacker1123/Laba2.git
2f83f59..2f55b5c main -> main

C:\Users\dinoz\OneDrive\Рабочий стол\SE\Laba2V2\Laba2>
```

Рисунок 6.2 – С помощью командной строки пересылаю обновленный репозиторий на GitHub

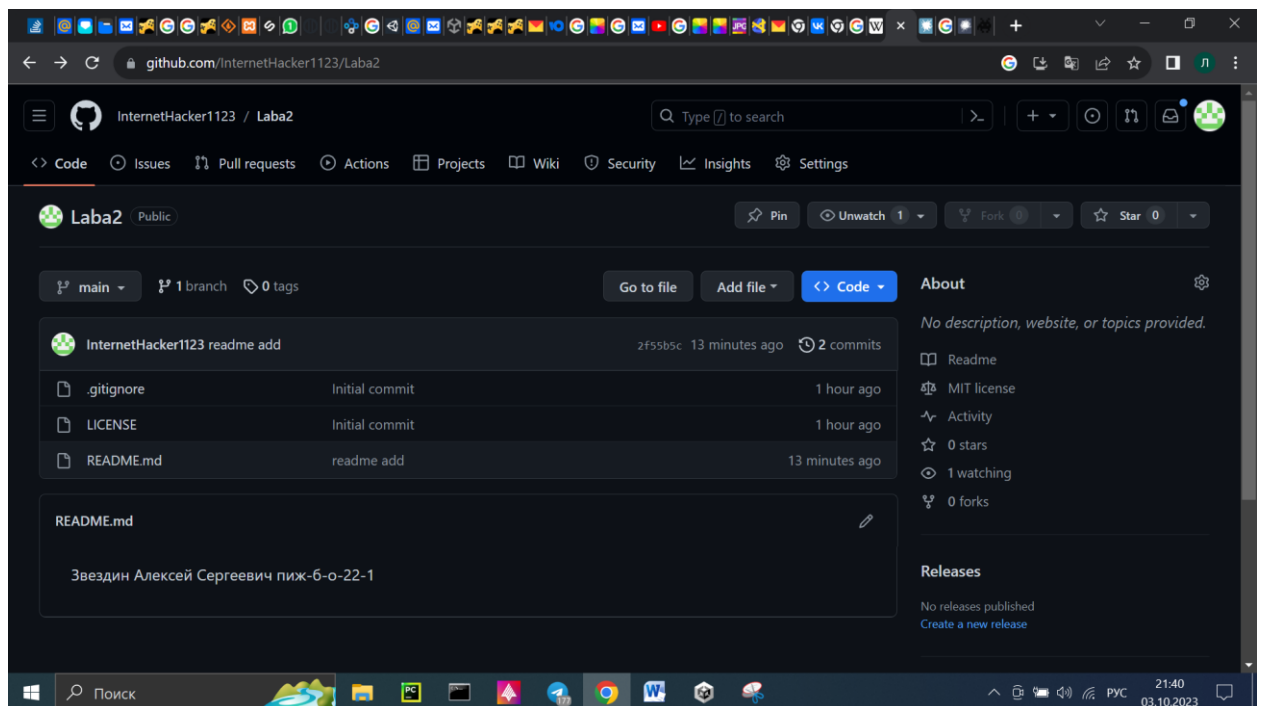


Рисунок 6.3 – Мой файл README.md на GitHub

7. Написал небольшую программу на выбранном мною языке программирования. Зафиксировал изменения при написании программы в

локальном репозитории. Сделала не менее 7 коммитов, отмеченных не менее 3 тэгами

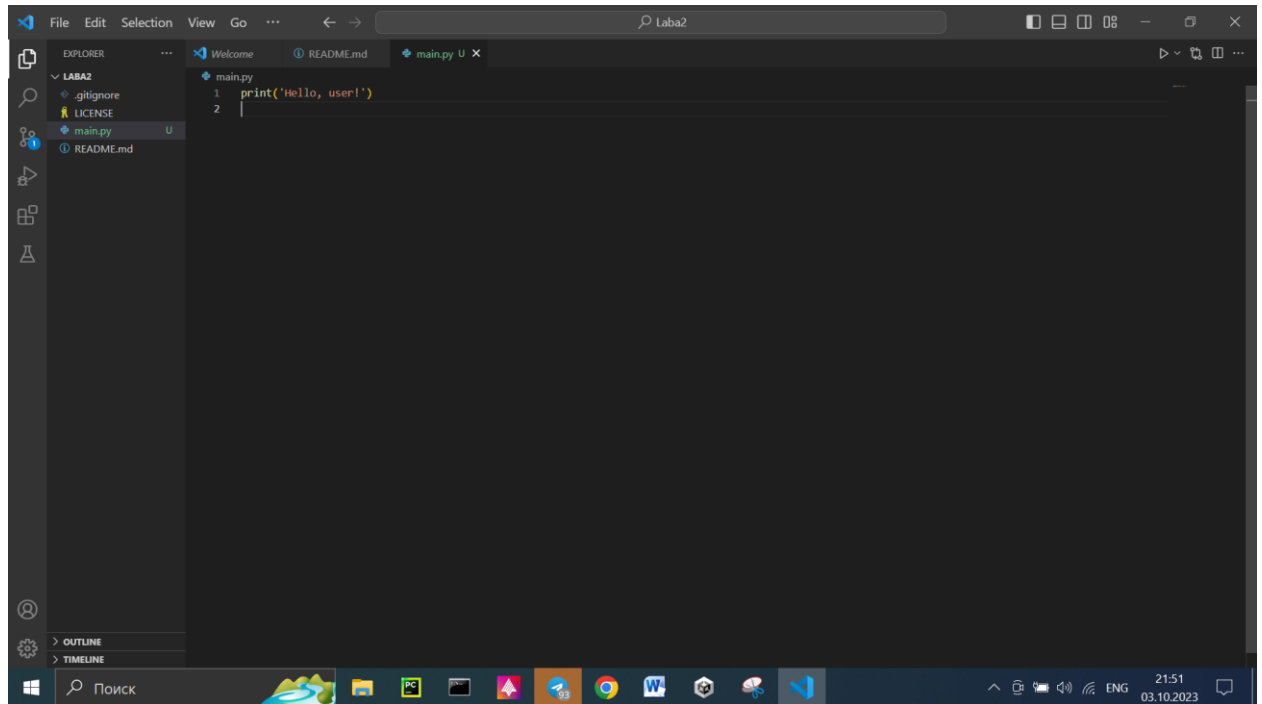


Рисунок 7.1 – Моя программа на Python

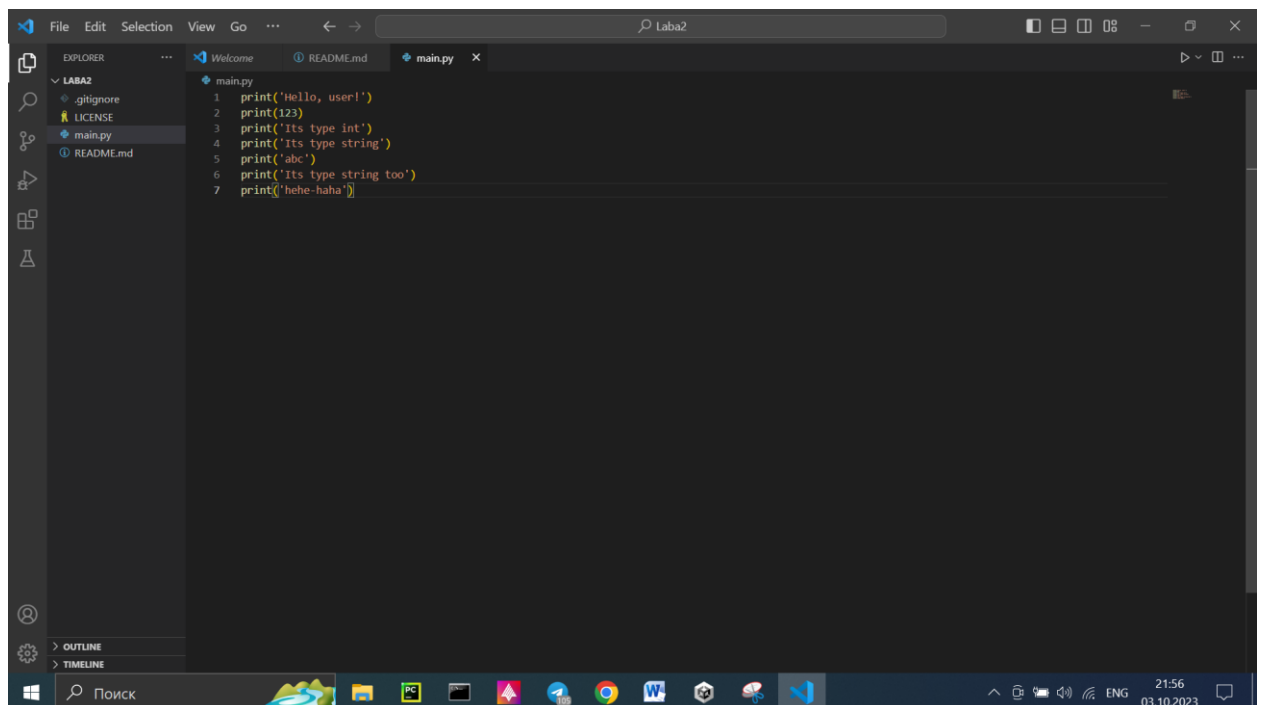


Рисунок 7.2 – Изменяю программу

```
C:\Windows\system32\cmd.exe
Writing objects: 100% (3/3), 299 bytes | 299.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/InternetHacker1123/Laba2.git
703cce9..754b31f main -> main

C:\Users\dinoz\OneDrive\Рабочий стол\SE\Laba2V2\Laba2>git add .

C:\Users\dinoz\OneDrive\Рабочий стол\SE\Laba2V2\Laba2>git commit -m "main .py"
[main bcf5682] main .py
1 file changed, 2 insertions(+), 1 deletion(-)

C:\Users\dinoz\OneDrive\Рабочий стол\SE\Laba2V2\Laba2>git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 315 bytes | 315.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/InternetHacker1123/Laba2.git
754b31f..bcf5682 main -> main

C:\Users\dinoz\OneDrive\Рабочий стол\SE\Laba2V2\Laba2>git add .

C:\Users\dinoz\OneDrive\Рабочий стол\SE\Laba2V2\Laba2>git commit -m "main .py"
[main 688ce26] main .py
1 file changed, 2 insertions(+), 1 deletion(-)

C:\Users\dinoz\OneDrive\Рабочий стол\SE\Laba2V2\Laba2>git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 305 bytes | 305.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/InternetHacker1123/Laba2.git
bcf5682..688ce26 main -> main

C:\Users\dinoz\OneDrive\Рабочий стол\SE\Laba2V2\Laba2>
```

Рисунок 7.3 – Делаю не менее 7 коммитов

```
C:\Windows\system32\cmd.exe
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 305 bytes | 305.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/InternetHacker1123/Laba2.git
bcf5682..688ce26 main -> main

C:\Users\dinoz\OneDrive\Рабочий стол\SE\Laba2V2\Laba2>git tag -a v1.1 "version 1.1"
fatal: Failed to resolve 'version 1.1' as a valid ref.

C:\Users\dinoz\OneDrive\Рабочий стол\SE\Laba2V2\Laba2>git tag v1.1 master
fatal: Failed to resolve 'master' as a valid ref.

C:\Users\dinoz\OneDrive\Рабочий стол\SE\Laba2V2\Laba2>git tag v1.1 origin

C:\Users\dinoz\OneDrive\Рабочий стол\SE\Laba2V2\Laba2>git tag
v1.1

C:\Users\dinoz\OneDrive\Рабочий стол\SE\Laba2V2\Laba2>git v1.2 origin
git: 'v1.2' is not a git command. See 'git --help'.

C:\Users\dinoz\OneDrive\Рабочий стол\SE\Laba2V2\Laba2>git tag v1.2 origin

C:\Users\dinoz\OneDrive\Рабочий стол\SE\Laba2V2\Laba2>git tag v1.3 origin

C:\Users\dinoz\OneDrive\Рабочий стол\SE\Laba2V2\Laba2>git tag
v1.1
v1.2
v1.3

C:\Users\dinoz\OneDrive\Рабочий стол\SE\Laba2V2\Laba2>git push origin --tags
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/InternetHacker1123/Laba2.git
* [new tag] v1.1 -> v1.1
* [new tag] v1.2 -> v1.2
* [new tag] v1.3 -> v1.3

C:\Users\dinoz\OneDrive\Рабочий стол\SE\Laba2V2\Laba2>
```

Рисунок 7.4 – Отмечаю не менее 3 тэгов

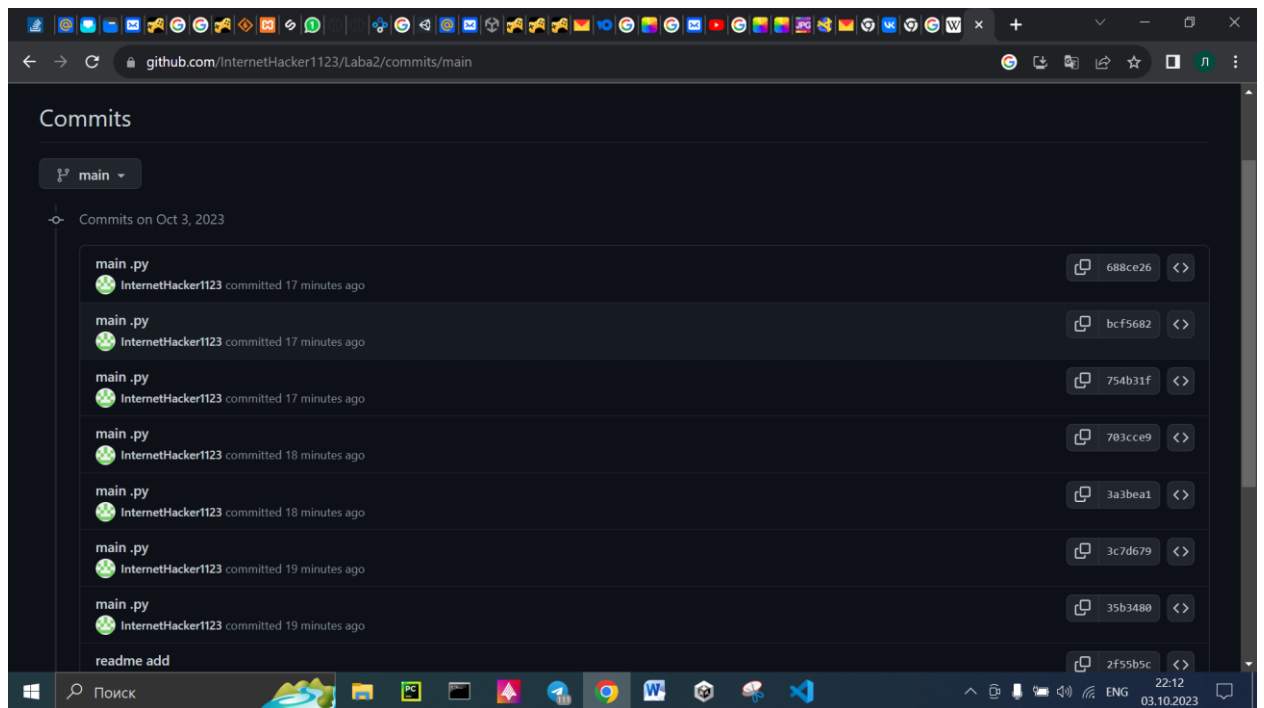


Рисунок 7.5 – Изменения, отображённые на GitHub

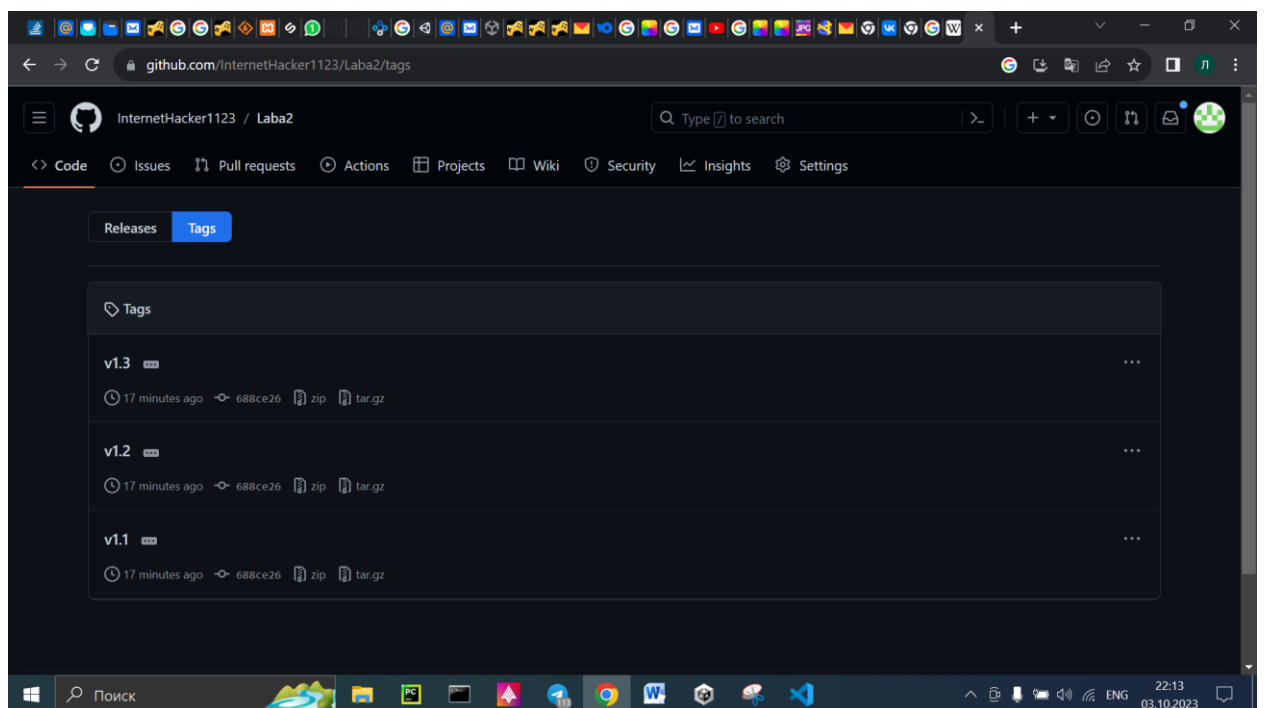


Рисунок 7.6 – Отмеченные тэги, отображённые на GitHub

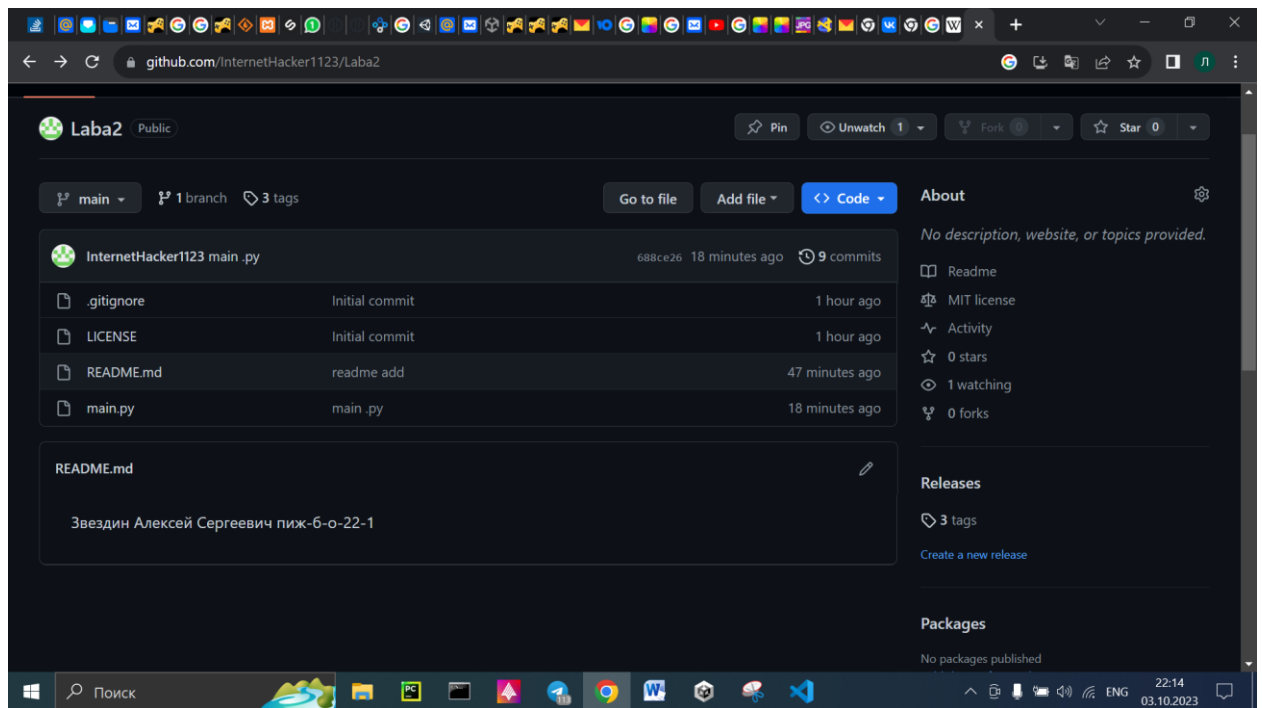


Рисунок 7.7 – Мой репозиторий

8. Просмотрел историю (журнал) хранилища командой `git log` с помощью команды `git log --graph --pretty=oneline --abbrev-commit`

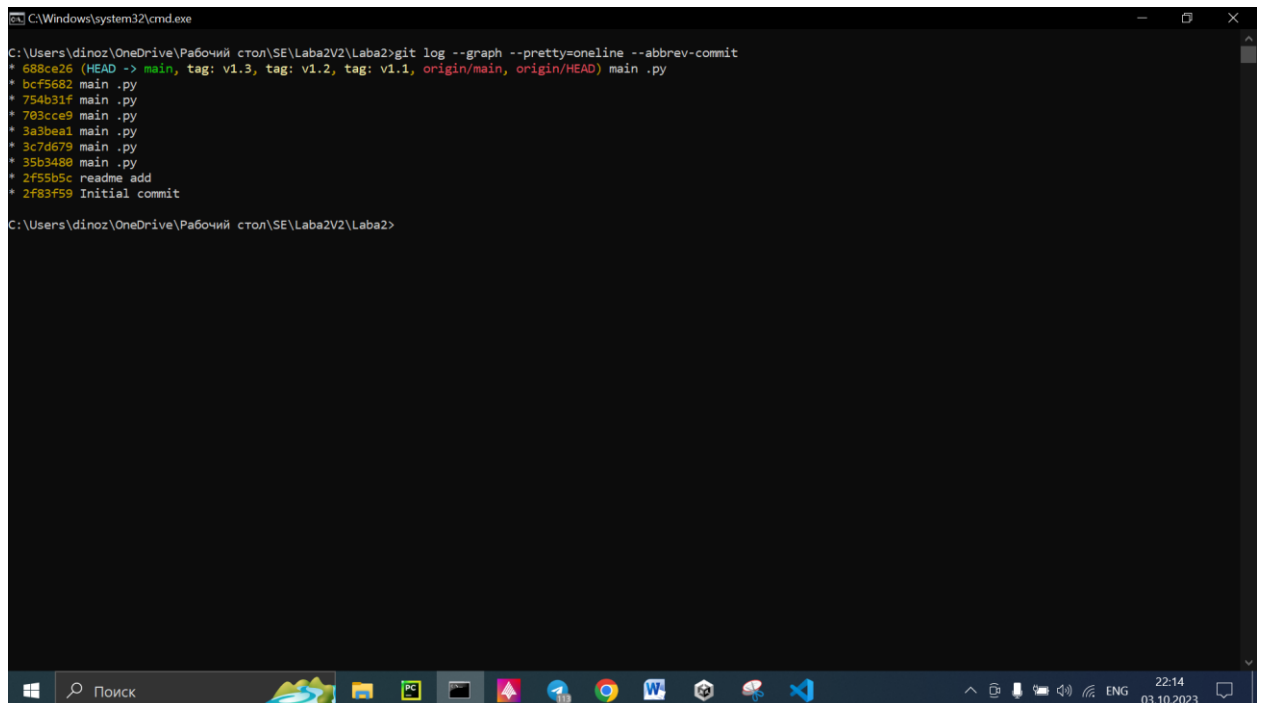


Рисунок 8.1 – История хранилища

9. Просмотреть содержимое коммитов командой `git show HEAD`, `git show HEAD~1`.

```
C:\Windows\system32\cmd.exe
C:\Users\dinoz\OneDrive\Рабочий стол\SE\Laba2V2\Laba2>git show HEAD
commit 688ce26345f86b7fab8444d22949e9013efb77dc (HEAD -> main, tag: v1.3, tag: v1.2, tag: v1.1, origin/main, origin/HEAD)
Author: InternetHacker1123 <dinozavrik9911@mail.ru>
Date: Tue Oct 3 21:55:57 2023 +0300

    main .py

diff --git a/main.py b/main.py
index 61e745b..ae75cef 100644
--- a/main.py
+++ b/main.py
@@ -3,4 +3,5 @@ print(123)
 print('Its type int')
 print('Its type string')
 print('abc')
-print('Its type string too')
+print('Its type string too')
\ No newline at end of file
-print('Its type string too')
+print('hehe-haha')
\ No newline at end of file

C:\Users\dinoz\OneDrive\Рабочий стол\SE\Laba2V2\Laba2>
```

Рисунок 9.1 – Вывод последнего коммита с помощью команды git show HEAD

```
C:\Windows\system32\cmd.exe
C:\Users\dinoz\OneDrive\Рабочий стол\SE\Laba2V2\Laba2>git show HEAD~1
commit bc5682522c083f0538eeb0a7a8ac31c6d13da59
Author: InternetHacker1123 <dinozavrik9911@mail.ru>
Date: Tue Oct 3 21:55:37 2023 +0300

    main .py

diff --git a/main.py b/main.py
index 46a5b3a..61e745b 100644
--- a/main.py
+++ b/main.py
@@ -2,4 +2,5 @@ print('Hello, user!')
 print(123)
 print('Its type int')
 print('Its type string')
-print('abc')
+print('abc')
\ No newline at end of file
-print('abc')
+print('Its type string too')
\ No newline at end of file

C:\Users\dinoz\OneDrive\Рабочий стол\SE\Laba2V2\Laba2>
```

Рисунок 9.2 – Вывод предпоследнего коммита с помощью команды git show HEAD~1

```
C:\Windows\system32\cmd.exe
C:\Users\dinoz\OneDrive\Рабочий стол\SE\Laba2V2\Laba2>git show v1.1
commit 688ce26345f86b7fab8444d229d0e9023efb77dc (HEAD -> main, tag: v1.3, tag: v1.2, tag: v1.1, origin/main, origin/HEAD)
Author: InternetHacker1123 <dinozavr1k9911@mail.ru>
Date:   Tue Oct 3 21:55:57 2023 +0300

    main .py

diff --git a/main.py b/main.py
index 61e745b..ae75cef 100644
--- a/main.py
+++ b/main.py
@@ -3,4 +3,5 @@ print(123)
 print('Its type int')
 print('Its type string')
 print('abc')
+print('Its type string too')
\ No newline at end of file
+print('Its type string too')
+print('hehe-haha')
\ No newline at end of file

C:\Users\dinoz\OneDrive\Рабочий стол\SE\Laba2V2\Laba2>
```

Рисунок 9.3 – Вывод первого тега с помощью команды git show v1.1

10. Я освоил возможность отката к заданной версии

10.1. Удалил весь код из одного из файлов программы репозитория, в моём случае main.py, и сохранила этот файл.

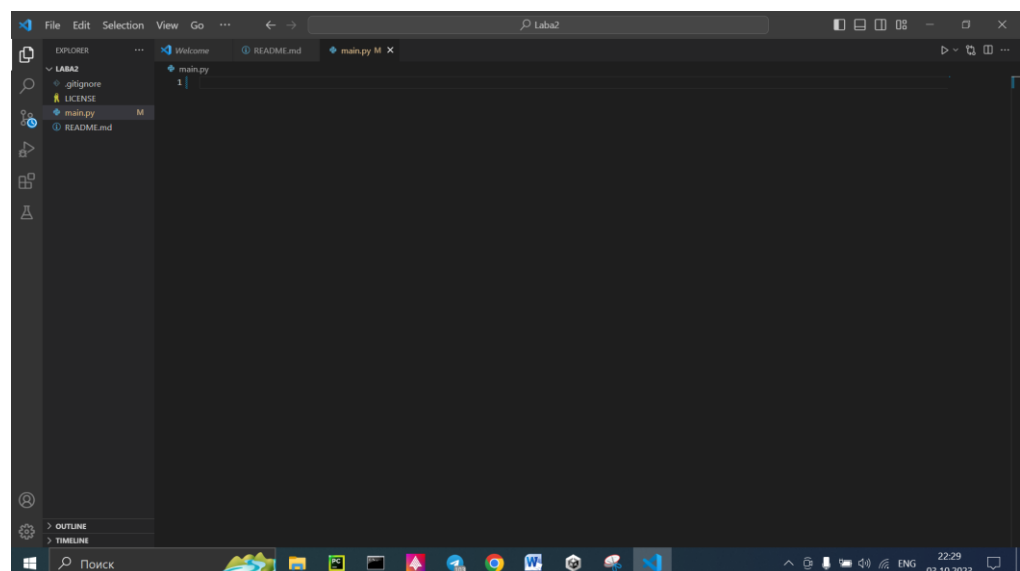


Рисунок 10.1.1 – Удаление кода и сохранение файла

10.2. Удалил все несохраненные изменения в файле командой: git checkout -- main.py

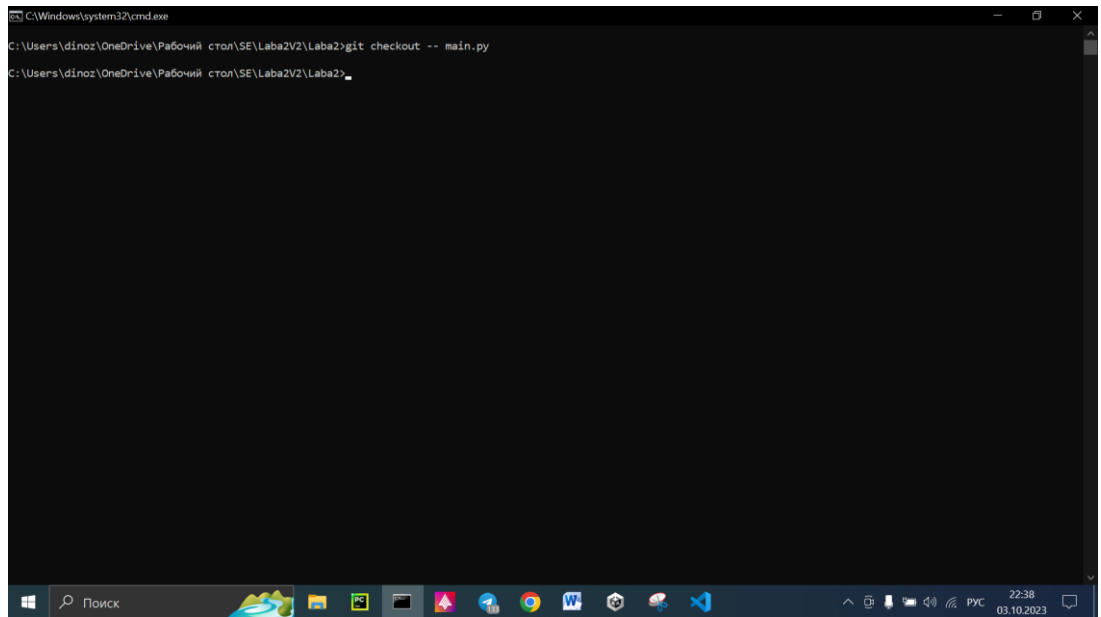


Рисунок 10.2.1 – Восстановление последнего коммита с помощью команды
`git checkout -- main.py`

10.3. Повторяю пункт 10.1 и делаю коммит

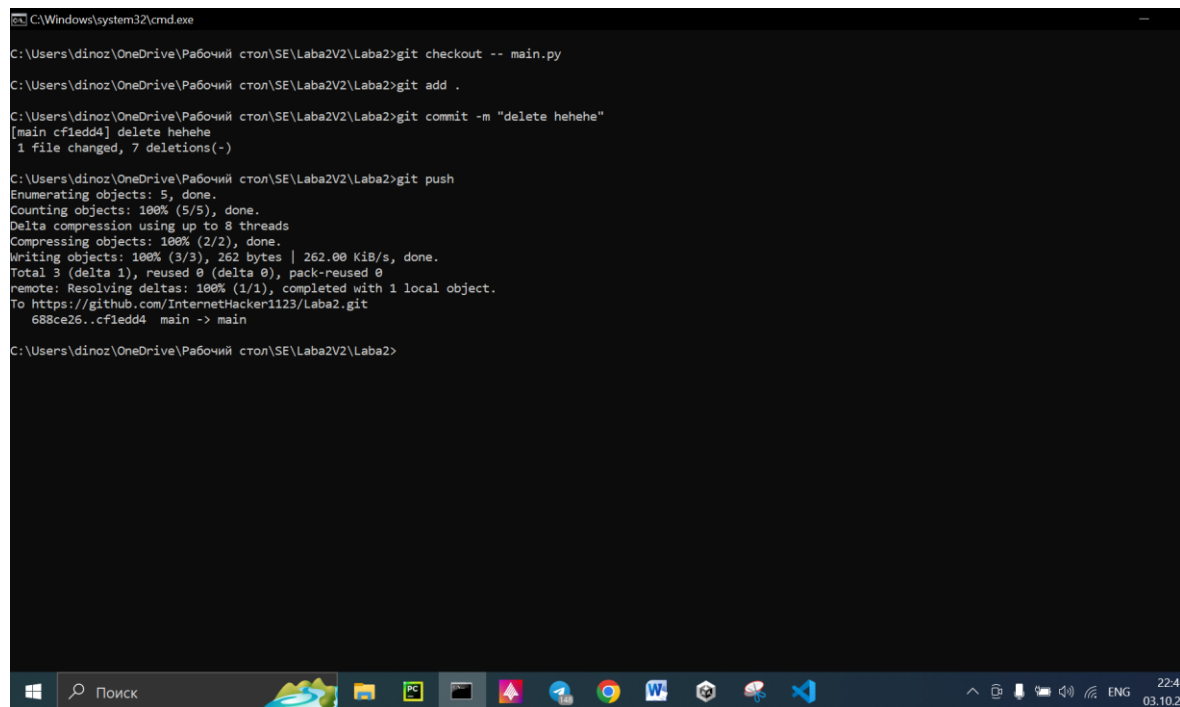


Рисунок 10.3.1 – Отправляю удалённый файл на GitHub

10.4. Откатываю состояние хранилища к предыдущей версии командой: `git reset --hard HEAD~1`

```
C:\Users\dinoz\OneDrive\Рабочий стол\SE\Laba2V2\Laba2>git reset --hard HEAD~1
HEAD is now at 688ce26 main .py

C:\Users\dinoz\OneDrive\Рабочий стол\SE\Laba2V2\Laba2>
```

Рисунок 10.4.1 – Возвращаем предпоследний коммит

Вывод: Git предоставляет возможность отката к предыдущей версии. Это позволяет исправить ошибки или проблемы, возникшие после обновления кода, а также сохранять историю изменений, что облегчает восстановление предыдущих версий проекта при необходимости. С помощью команды `git checkout -- <имя файла>` можно вернуть файл к последнему фиксированному состоянию, а с помощью команды `git reset --hard HEAD~1` откатить состояние хранилища к предыдущей версии. То есть даже после выполнения коммита можно вернуть предыдущую версию программы.

11. Зафиксировал сделанные изменения

```
Администратор: Командная строка
Указано недопустимое время.
Введите новое время:

C:\TestGit\WHATIAM\Whatiam>git add.
git: 'add.' is not a git command. See 'git --help'.

The most similar command is
    add

C:\TestGit\WHATIAM\Whatiam>git add .

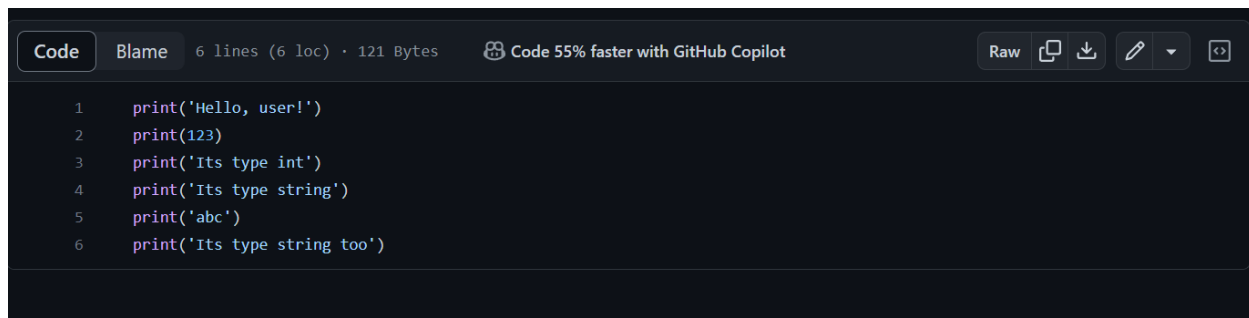
C:\TestGit\WHATIAM\Whatiam>git commit -m"I return the deleted file"
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

C:\TestGit\WHATIAM\Whatiam>git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 755 bytes | 755.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/MrFinicheck/Whatiam.git
   e088e3b..4ec4253  main -> main

C:\TestGit\WHATIAM\Whatiam>
```

Рисунок 11.1 – Отправляю изменённый файл на GitHub



```
Code Blame 6 lines (6 loc) · 121 Bytes Code 55% faster with GitHub Copilot Raw Copy Download Edit View
```

```
1 print('Hello, user!')
2 print(123)
3 print('Its type int')
4 print('Its type string')
5 print('abc')
6 print('Its type string too')
```

Рисунок 11.2 – Моя программа на GitHub

Контрольные вопросы

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

Одним из основных и наиболее мощных инструментов для просмотра истории коммитов является команда `git log`. Команда `git log` имеет очень большое количество опций для поиска коммитов по разным критериям. Одним из самых полезных аргументов является `-p` или `--patch`, который показывает разницу, внесенную в каждый коммит. Если вы хотите увидеть сокращенную статистику для каждого коммита, вы можете использовать опцию `--stat`. Наиболее интересной опцией является `format`, которая позволяет указать формат для вывода информации

2. Как ограничить вывод при просмотре истории коммитов?

В дополнение к опциям форматирования вывода команда `git log` принимает несколько опций для ограничения вывода — опций, с помощью которых можно увидеть определенное подмножество коммитов. Вы можете использовать `--<n>`, где `n` — это любое натуральное число и представляет собой `n` последних коммитов. Опции для ограничения вывода по времени, такие как `--since` и `--until`, являются очень удобными. Опция `--author` дает возможность фильтровать по автору коммита, а опция `--grep` искать по ключевым словам в сообщении коммита.

3. Как внести изменения в уже сделанный коммит?

Если вы хотите переделать коммит — внесите необходимые изменения, добавьте их в индекс и сделайте коммит ещё раз, указав параметр `--amend`

4. Как отменить индексацию файла в Git?

Используйте `git reset HEAD <имя_файла>...` для отмены индексации файла

5. Как отменить изменения в файле?

С помощью команды `git checkout -- <имя_файла>`. Она выполнит откат изменений

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?

Для того, чтобы просмотреть список настроенных удалённых репозиториях, вы можете запустить команду `git remote`. Она выведет названия доступных удалённых репозиториях.

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?

Отправка изменений с удалённого репозитория осуществляется с помощью команды `git push`, а получение изменений с удаленного репозитория – с помощью команды `git fetch` или `git pull`

10. Как выполнить просмотр удаленного репозитория?

Если хотите получить побольше информации об одном из удалённых репозиториях, вы можете использовать команду `git remote show <remote>`

11. Каково назначение тэгов Git?

Эта функциональность используется для отметки моментов выпуска версий

12. Как осуществляется работа с тэгами Git?

Git использует два основных типа тегов: легковесные и аннотированные. Легковесный тег — это что-то очень похожее на ветку, которая не изменяется — просто указатель на определённый коммит. А вот аннотированные теги хранятся в базе данных Git как полноценные объекты. Создание аннотированного тега в Git выполняется легко. Самый простой способ — это указать -a при выполнении команды tag. Опция -m задаёт сообщение, которое будет храниться вместе с тегом. По умолчанию, команда git push не отправляет теги на удалённые сервера. После создания теги нужно отправлять явно на удалённый сервер. Процесс аналогичен отправке веток — достаточно выполнить команду git push origin <tagname>.

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

Команда --prune вводится для очистки любых устаревших или «зависших» ссылок в локальной репозитории. Сюда входят объекты, которые больше недоступны из какой-либо ветви или тега, а также ветви удалённого отслеживания, которые больше не существуют на удалённом компьютере.