

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 «Программирование на Python»

Выполнил:
Кулешов Олег Иванович
2 курс, группа ИВТ-6-о-22-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной техники и
автоматизированных систем», очная
форма обучения

(подпись)

Руководитель практики:
Воронкин Р. А.

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023

Исследование возможностей Git для работы с локальными репозиториями

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

```
commit 334f738605497337f432cb6f8a4054083095c776 (HEAD -> master)
Author: kulesh00v <oleg.kuleshov.1261@mail.ru>
Date: Tue Oct 10 22:29:49 2023 +0300

    дополнил гитигнор

commit bd2621f517d7ad2966e2c3ad1d96d8f9c12c0085 (tag: v3.0)
Author: kulesh00v <oleg.kuleshov.1261@mail.ru>
Date: Tue Oct 10 17:51:43 2023 +0300

    Добавлены все тонкости как в условии, так и в самом коде

commit 137a617f477b9d8884a7d3d4f5be65655492b8b7 (tag: v2.0)
Author: kulesh00v <oleg.kuleshov.1261@mail.ru>
Date: Tue Oct 10 17:39:06 2023 +0300

    Оптимизировал часть условного оператора

commit b6aaea055e43e3d2e679c978c342d64f2007d22e (tag: v1.0)
Author: kulesh00v <oleg.kuleshov.1261@mail.ru>
Date: Tue Oct 10 17:19:09 2023 +0300

    Написал условие задачи и прикинул решение

commit bfc7df11fdd5b619a5f05ad22159e40b20a08e10
Author: kulesh00v <oleg.kuleshov.1261@mail.ru>
Date: Mon Oct 9 22:48:07 2023 +0300

    Add file

~
(END)
```

Рисунок 1. Выполнение команды “git log”

```

(venv) PS C:\Users\User\PycharmProjects\Python_laba_2> git log -p -2
commit 334f738605497337f432cb6f8a4054083095c776 (HEAD -> master)
Author: kulesh00v <oleg.kuleshov.1261@mail.ru>
Date: Tue Oct 10 22:29:49 2023 +0300

    дополнил гитигнор

diff --git a/venv/.gitignore b/venv/.gitignore
new file mode 100644
index 0000000..fc90a84
--- /dev/null
+++ b/venv/.gitignore
@@ -0,0 +1,266 @@
+.idea/
+# Создано https://www.toptal.com/developers/gitignore/api/python , pycharm
+# Редактировать по адресу https://www.toptal.com/developers/gitignore?templates=python,pycharm
+

```

Рисунок 2. Выполнение команды “git log -p -2”

```

(venv) PS C:\Users\User\PycharmProjects\Python_laba_2> git log --stat
commit 334f738605497337f432cb6f8a4054083095c776 (HEAD -> master)
Author: kulesh00v <oleg.kuleshov.1261@mail.ru>
Date: Tue Oct 10 22:29:49 2023 +0300

    дополнил гитигнор

venv/.gitignore | 266 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
1 file changed, 266 insertions(+)

commit bd2621f517d7ad2966e2c3ad1d96d8f9c12c0085 (tag: v3.0)
Author: kulesh00v <oleg.kuleshov.1261@mail.ru>
Date: Tue Oct 10 17:51:43 2023 +0300

    Добавлены все тонкости как в условии, так и в самом коде

main.py | 9 ++++++--
1 file changed, 7 insertions(+), 2 deletions(-)

commit 137a617f477b9d8884a7d3d4f5be65655492b8b7 (tag: v2.0)
Author: kulesh00v <oleg.kuleshov.1261@mail.ru>

```

Рисунок 3. Выполнение команды “git log -stat”

```
(venv) PS C:\Users\User\PycharmProjects\Python_laba_2> git log --pretty=oneline
334f738605497337f432cb6f8a4054083095c776 (HEAD -> master) дополнил гитигнор
bd2621f517d7ad2966e2c3ad1d96d8f9c12c0085 (tag: v3.0) Добавлены все тонкости как в условии, так и в самом коде
137a617f477b9d8884a7d3d4f5be65655492b8b7 (tag: v2.0) Оптимизировал часть условного оператора
b6aaea055e43e3d2e679c978c342d64f2007d22e (tag: v1.0) Написал условие задачи и прикинул решение
bfc7df11fdd5b619a5f05ad22159e40b20a08e10 Add file
```

Рисунок 4. Выполнение команды “git log –pretty=oneline”

```
(venv) PS C:\Users\User\PycharmProjects\Python_laba_2> git log --pretty=format:"%h - %an, %ar : %s"
334f738 - kulesh00v, 2 hours ago : дополнил гитигнор
bd2621f - kulesh00v, 7 hours ago : Добавлены все тонкости как в условии, так и в самом коде
137a617 - kulesh00v, 7 hours ago : Оптимизировал часть условного оператора
b6aaea0 - kulesh00v, 8 hours ago : Написал условие задачи и прикинул решение
bfc7df1 - kulesh00v, 26 hours ago : Add file
```

Рисунок 5. Выполнение команды “git log –pretty=oneline”

```
(venv) PS C:\Users\User\PycharmProjects\Python_laba_2> git log --pretty=format:"%h %s" --graph
* 334f738 дополнил гитигнор
* bd2621f Добавлены все тонкости как в условии, так и в самом коде
* 137a617 Оптимизировал часть условного оператора
* b6aaea0 Написал условие задачи и прикинул решение
* bfc7df1 Add file
```

Рисунок 6. Выполнение команды “git log --pretty=format:"%h %s" –graph”

```
commit bd2621f517d7ad2966e2c3ad1d96d8f9c12c0085 (tag: v3.0)
Author: kulesh00v <oleg.kuleshov.1261@mail.ru>
Date: Tue Oct 10 17:51:43 2023 +0300

    Добавлены все тонкости как в условии, так и в самом коде

commit 137a617f477b9d8884a7d3d4f5be65655492b8b7 (tag: v2.0)
Author: kulesh00v <oleg.kuleshov.1261@mail.ru>
Date: Tue Oct 10 17:39:06 2023 +0300

    Оптимизировал часть условного оператора

commit b6aaea055e43e3d2e679c978c342d64f2007d22e (tag: v1.0)
Author: kulesh00v <oleg.kuleshov.1261@mail.ru>
Date: Tue Oct 10 17:19:09 2023 +0300

    Написал условие задачи и прикинул решение

commit bfc7df11fdd5b619a5f05ad22159e40b20a08e10
Author: kulesh00v <oleg.kuleshov.1261@mail.ru>
Date: Mon Oct 9 22:48:07 2023 +0300

    Add file
```

Рисунок 7. Выполнение команды “git log --since=2.weeks”

```
(venv) PS C:\Users\User\PycharmProjects\Python_laba_2> git remote -v
origin  https://github.com/ku1esh00v/Python_laba_2.git (fetch)
origin  https://github.com/ku1esh00v/Python_laba_2.git (push)
```

Рисунок 8. Выполнение программы “git remote –v”

```
(venv) PS C:\Users\User\PycharmProjects\Python_laba_2> git tag v1.0 b6aaea055e43e3d2e679c978c342d64f2007d22e
(venv) PS C:\Users\User\PycharmProjects\Python_laba_2> git show
Author: ku1esh00v <oleg.kuleshov.1261@mail.ru>
Date: Tue Oct 10 17:19:09 2023 +0300

    Написал условие задачи и прикинул решение

diff --git a/main.py b/main.py
index bc63dc6..33f81a0 100644
--- a/main.py
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 816 bytes | 816.00 KiB/s, done.
To https://github.com/ku1esh00v/Python_laba_2.git
   bfc7df1..b6aaea0  master -> master
(venv) PS C:\Users\User\PycharmProjects\Python_laba_2>
(venv) PS C:\Users\User\PycharmProjects\Python_laba_2> git status
On branch master

nothing to commit, working tree clean
(venv) PS C:\Users\User\PycharmProjects\Python_laba_2> git push v1.0
```

Рисунок 9. Создание и пуш тега v1.0

```
(venv) PS C:\Users\User\PycharmProjects\Python_laba_2> git tag v2.0 137a617f477b9d8884a7d3d4f5be65655492b8b7
(venv) PS C:\Users\User\PycharmProjects\Python_laba_2> git tag
v1.0
v2.0
(venv) PS C:\Users\User\PycharmProjects\Python_laba_2> git push
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 397 bytes | 397.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
To https://github.com/ku1esh00v/Python_laba_2.git
   b6aaea0..137a617  master -> master
(venv) PS C:\Users\User\PycharmProjects\Python_laba_2>
(venv) PS C:\Users\User\PycharmProjects\Python_laba_2>
(venv) PS C:\Users\User\PycharmProjects\Python_laba_2> git push origin v2.0
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/ku1esh00v/Python_laba_2.git
 * [new tag]           v2.0 -> v2.0
```

Рисунок 10. Создание и пуш тега v2.0

```

(venv) PS C:\Users\User\PycharmProjects\Python_laba_2> git tag v3.0 bd2621f517d7ad2966e2c3ad1d96d8f9c12c0085
(venv) PS C:\Users\User\PycharmProjects\Python_laba_2> git push origin v3.0
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), 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/ku1esh00v/Python\_laba\_2.git
 * [new tag]          v3.0 -> v3.0
(venv) PS C:\Users\User\PycharmProjects\Python_laba_2> git push
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/ku1esh00v/Python\_laba\_2.git
 137a617..bd2621f  master -> master

```

Рисунок 11. Создание и пуш тега v3.0

```

(venv) PS C:\Users\User\PycharmProjects\Python_laba_2> git log --graph
* commit 334f738605497337f432cb6f8a4054083095c776 (HEAD -> master)
| Author: ku1esh00v <oleg.kuleshov.1261@mail.ru>
| Date: Tue Oct 10 22:29:49 2023 +0300
|
|    дополнил гитигнор
|
* commit bd2621f517d7ad2966e2c3ad1d96d8f9c12c0085 (tag: v3.0)
| Author: ku1esh00v <oleg.kuleshov.1261@mail.ru>
| Date: Tue Oct 10 17:51:43 2023 +0300
|
|    Добавлены все тонкости как в условии, так и в самом коде
|
* commit 137a617f477b9d8884a7d3d4f5be65655492b8b7 (tag: v2.0)
| Author: ku1esh00v <oleg.kuleshov.1261@mail.ru>
| Date: Tue Oct 10 17:39:06 2023 +0300
|
|    Оптимизировал часть условного оператора
|
* commit b6aaea055e43e3d2e679c978c342d64f2007d22e (tag: v1.0)
| Author: ku1esh00v <oleg.kuleshov.1261@mail.ru>
| Date: Tue Oct 10 17:19:09 2023 +0300
|
|    Написал условие задачи и прикинул решение
|
* commit bfc7df11fdd5b619a5f05ad22159e40b20a08e10
| Author: ku1esh00v <oleg.kuleshov.1261@mail.ru>
| Date: Mon Oct 9 22:48:07 2023 +0300

```

Рисунок 12. Выполнение команды “git log –graph”

```
(venv) PS C:\Users\User\PycharmProjects\Python_laba_2> git log --pretty=oneline
334f738605497337f432cb6f8a4054083095c776 (HEAD -> master) дополнил гитигнор
bd2621f517d7ad2966e2c3ad1d96d8f9c12c0085 (tag: v3.0) Добавлены все тонкости как в условии, так и в самом коде
137a617f477b9d8884a7d3d4f5be65655492b8b7 (tag: v2.0) Оптимизировал часть условного оператора
b6aaaa055e43e3d2e679c978c342d64f2007d22e (tag: v1.0) Написал условие задачи и прикинул решение
bfc7df11fdd5b619a5f05ad22159e40b20a08e10 Add file
```

Рисунок 13. Выполнение команды “--pretty=oneline”

```
(venv) PS C:\Users\User\PycharmProjects\Python_laba_2> git log --abbrev-commit
commit 334f738 (HEAD -> master)
Author: ku1esh00v <oleg.kuleshov.1261@mail.ru>
Date: Tue Oct 10 22:29:49 2023 +0300

    дополнил гитигнор

commit bd2621f (tag: v3.0)
Author: ku1esh00v <oleg.kuleshov.1261@mail.ru>
Date: Tue Oct 10 17:51:43 2023 +0300

    Добавлены все тонкости как в условии, так и в самом коде

commit 137a617 (tag: v2.0)
Author: ku1esh00v <oleg.kuleshov.1261@mail.ru>
Date: Tue Oct 10 17:39:06 2023 +0300

    Оптимизировал часть условного оператора

commit b6aaaa0 (tag: v1.0)
Author: ku1esh00v <oleg.kuleshov.1261@mail.ru>
Date: Tue Oct 10 17:19:09 2023 +0300

    Написал условие задачи и прикинул решение

commit bfc7df1
Author: ku1esh00v <oleg.kuleshov.1261@mail.ru>
Date: Mon Oct 9 22:48:07 2023 +0300

    Add file

(venv) PS C:\Users\User\PycharmProjects\Python_laba_2>
```

Рисунок 14. Выполнение команды “--abbrev-commit”

```

commit 334f738605497337f432cb6f8a4054083095c776 (HEAD -> master)
Author: kulesh00v <oleg.kuleshov.1261@mail.ru>
Date: Tue Oct 10 22:29:49 2023 +0300

    дополнил гитигнор

diff --git a/venv/.gitignore b/venv/.gitignore
new file mode 100644
index 0000000..fc90a84
--- /dev/null
+++ b/venv/.gitignore
@@ -0,0 +1,266 @@
+.idea/
+# Создано https://www.toptal.com/developers/gitignore/api/python , pycharm
+# Редактировать по адресу https://www.toptal.com/developers/gitignore?templates=python,pycharm
+
:

```

Рисунок 15. Выполнение команды “git show head”

```

(venv) PS C:\Users\User\PycharmProjects\Python_laba_2> git show head~2
commit 137a617f477b9d8884a7d3d4f5be65655492b8b7 (tag: v2.0)
Author: kulesh00v <oleg.kuleshov.1261@mail.ru>
Date: Tue Oct 10 17:39:06 2023 +0300

    Оптимизировал часть условного оператора

diff --git a/main.py b/main.py
index 33f81a0..221475c 100644
--- a/main.py
+++ b/main.py
@@ -13,5 +13,5 @@ m = int(input())
    n = int(input())
    if m <= n:
        for i in range(m, n + 1):
-           if i == 9 or (i + 1) % 10 == 0 or i % 17 == 0 or (i % 3 == 0 and i % 5 == 0):
+           if (i % 10 == 9) or (i % 17 == 0) or (i % 3 == 0 and i % 5 == 0):
                print(i)
\ No newline at end of file
(venv) PS C:\Users\User\PycharmProjects\Python_laba_2>

```

Рисунок 16. Выполнение команды “git show head~2”


```
(venv) PS C:\Users\User\PycharmProjects\Python_laba_2> git show bd2621f517d7ad2966e2c3ad1d96d8f9c12c0085
commit bd2621f517d7ad2966e2c3ad1d96d8f9c12c0085 (tag: v3.0, origin/master)
Author: kuleshov00v <oleg.kuleshov.1261@mail.ru>
Date: Tue Oct 10 17:51:43 2023 +0300

    Добавлены все тонкости как в условии, так и в самом коде

diff --git a/main.py b/main.py
index 221475c..de05abc 100644
--- a/main.py
+++ b/main.py
@@ -8,10 +8,15 @@
```

Рисунок 17. Обращение к коммиту по хэш-сумме

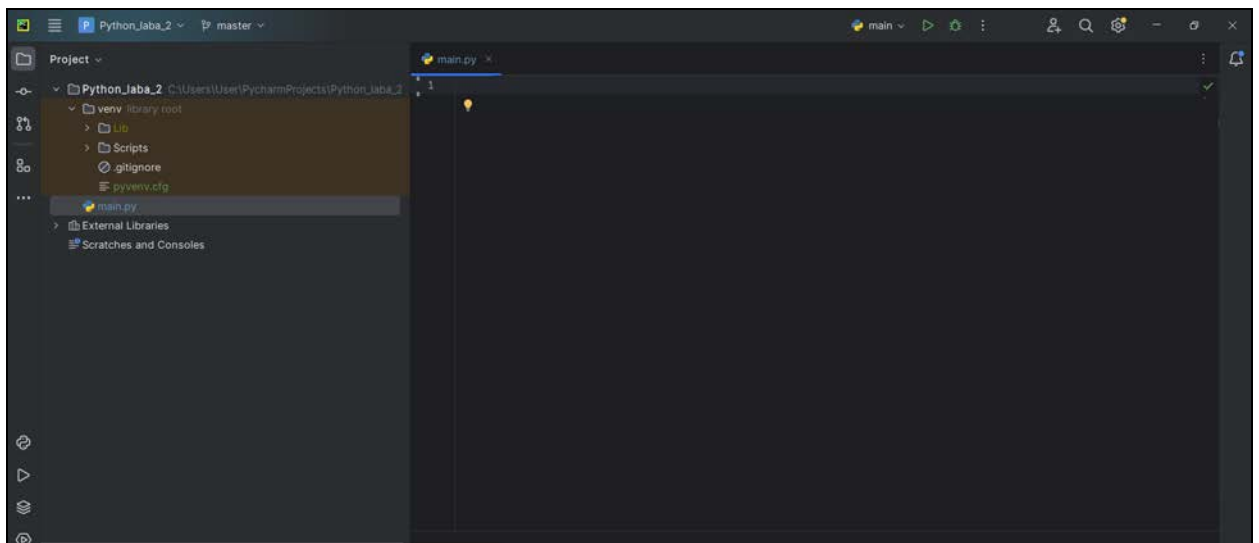


Рисунок 18. Удаление кода

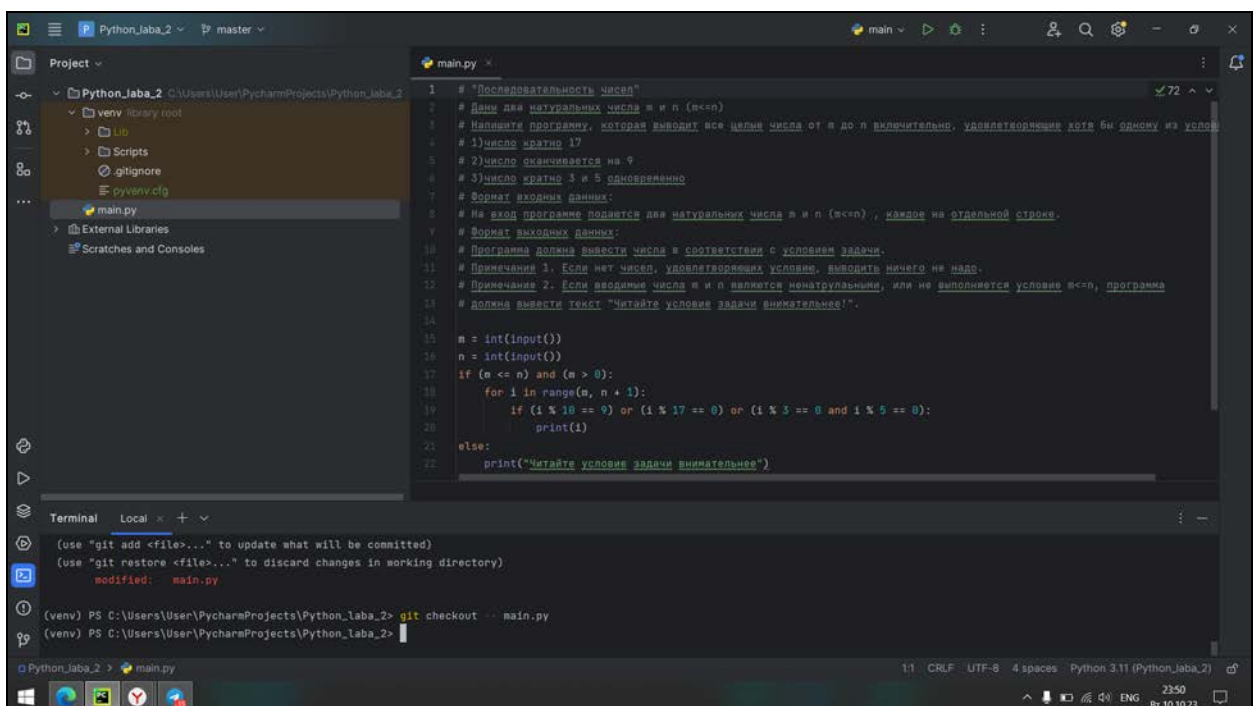


Рисунок 19. Работа команды checkout – main.py (код вернулся)

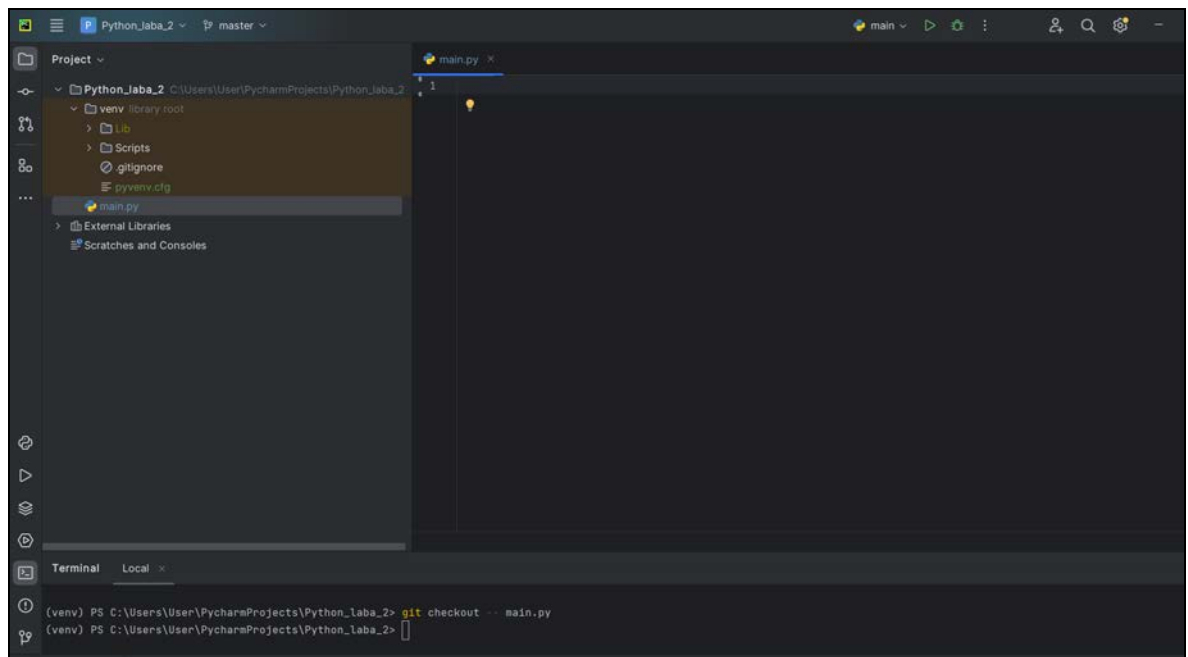


Рисунок 20. Удаление кода во второй раз

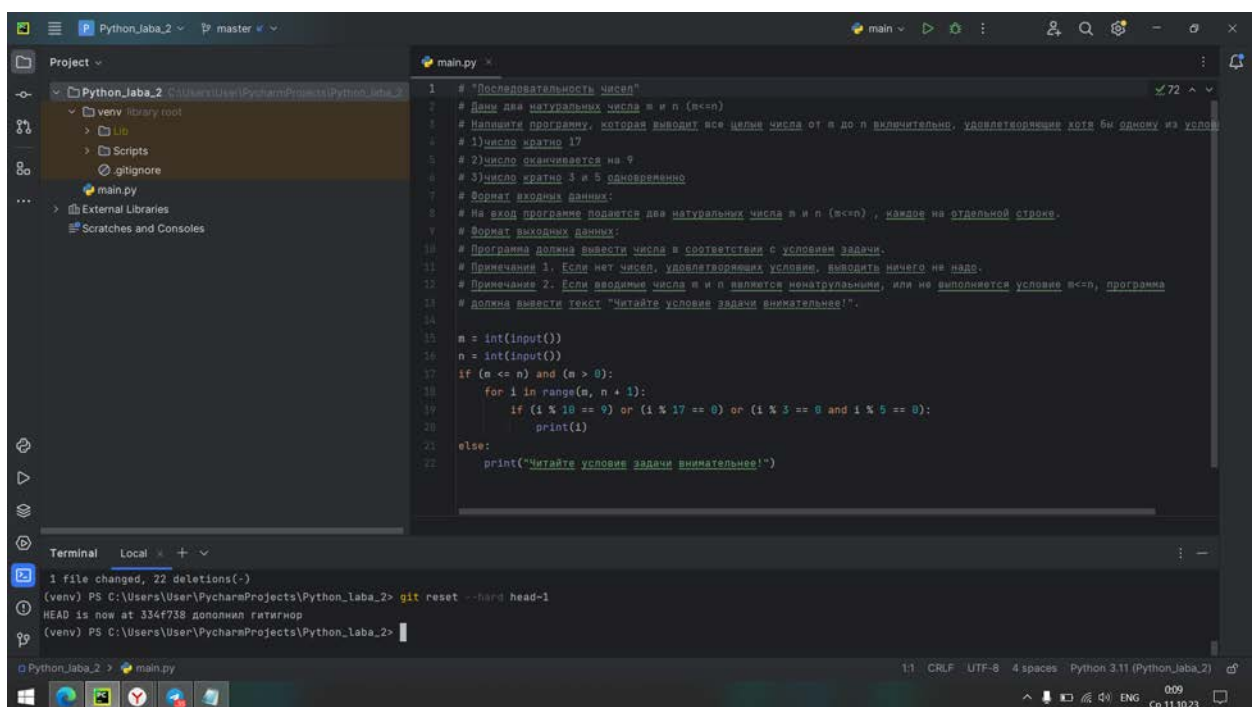


Рисунок 20. Результат команды “git reset –hard head~1” после коммита

Вывод по пунктам 10.1 – 10.4: содержимое файла в двух случаях никак не изменилось (то есть код успешно вернулся). Разница лишь в самих ситуациях. В первом случае я не производил коммит, а во втором коммит был произведён. Следовательно, когда изменения закоммичены, но необходим откат, то командой checkout проблему не решить: необходимо использовать git reset –hard head~n)

Ответы на контрольные вопросы:

1) Для просмотра истории коммитов в Git можно использовать команду "git log". Эта команда показывает список всех коммитов в репозитории, начиная с самого нового и заканчивая самым старым.

Некоторые дополнительные опции для просмотра истории коммитов:

- "--oneline": отображает каждый коммит в одной строке, содержащей хэш коммита и его сообщение.
- "--graph": отображает историю коммитов в виде графа, показывая связи между коммитами и ветками.
- "--author": фильтрует историю коммитов по автору.
- "--since" и "--until": фильтруют историю коммитов по дате, позволяя указать интервал времени.

Например, команда "git log --oneline --author=John --since=2021-01-01" покажет только те коммиты, которые были сделаны пользователем с именем "John" после 1 января 2021 года, в однострочном формате.

2) Команда git log принимает несколько опций для ограничения вывода — опций, с помощью которых можно увидеть определенное подмножество коммитов. В действительности можно использовать -n, где n — это любое натуральное число и представляет собой n последних коммитов. Есть опции для ограничения вывода по времени, такие как --since и --until. Опция --author дает возможность фильтровать по автору коммита, а опция --grep искать по ключевым словам в сообщении коммита.

3) Если вы хотите переделать коммит — внесите необходимые изменения, добавьте их в индекс и сделайте коммит ещё раз, указав параметр --amend .
Пример:

```
$ git commit -m 'Initial commit'
```

```
$ git add forgotten_file
```

```
$ git commit --amend
```

В итоге получится единый коммит — второй коммит заменит результаты первого.

4) Чтобы отменить индексацию файла в Git, можно использовать команду "git reset HEAD <имя_файла>". Например, если вы хотите отменить индексацию файла "file.txt", выполните следующую команду:

```
git reset HEAD file.txt
```

После этого файл будет снова находиться в состоянии "неотслеживаемых" изменений.

5) Для отмены изменений в файле можно использовать команду "git checkout -- <название файла>". Эта команда отменит все несохраненные изменения в указанном файле и восстановит его до последней версии в репозитории.

6) Удаленный репозиторий Git - это репозиторий, который расположен на удаленном сервере или хостинг-платформе, а не на локальном компьютере разработчика. Он используется для хранения и совместной работы над кодом с другими разработчиками. Удаленный репозиторий содержит все версии файлов и историю изменений, а также предоставляет возможность синхронизировать изменения между локальными и удаленными репозиториями.

7) Для просмотра удаленных репозиториях данного локального репозитория в Git можно использовать команду "git remote -v". Эта команда покажет список

всех удаленных репозиториях, связанных с текущим локальным репозиторием, а также их URL-адреса.

8) Чтобы добавить удаленный репозиторий для текущего локального репозитория в Git, можно использовать команду "git remote add". Ниже приведен синтаксис этой команды:

```
git remote add <имя_удаленного_репозитория>  
<URL_адрес_удаленного_репозитория>
```

где:

- <имя_удаленного_репозитория> - имя, которое вы хотите присвоить удаленному репозиторию (обычно используется "origin");
- <URL_адрес_удаленного_репозитория> - URL-адрес удаленного репозитория, к которому вы хотите добавить доступ.

9) Чтобы отправить изменения на удаленный репозиторий, вы можете использовать команду "git push". Ниже приведен синтаксис этой команды:

```
git push <имя_удаленного_репозитория> <ветка>
```

где:

- <имя_удаленного_репозитория> - имя удаленного репозитория, на который вы хотите отправить изменения (например, "origin");
- <ветка> - имя ветки, содержащей изменения, которые вы хотите отправить.

Например, если вы хотите отправить изменения из текущей ветки на удаленный репозиторий с именем "origin", вы можете выполнить следующую команду:

```
git push origin master
```

Чтобы получить изменения с удаленного репозитория, вы можете использовать команду "git pull". Ниже приведен синтаксис этой команды:

```
git pull <имя_удаленного_репозитория> <ветка>
```

где:

- <имя_удаленного_репозитория> - имя удаленного репозитория, с которого вы хотите получить изменения (например, "origin");
- <ветка> - имя ветки, содержащей изменения, которые вы хотите получить.

10) Если хотите получить побольше информации об одном из удалённых репозиториях, вы можете использовать команду `git remote show <remote>`. Она выдаёт URL удалённого репозитория, а также информацию об отслеживаемых ветках. Эта команда любезно сообщает вам, что если вы, находясь на ветке master, выполните `git pull`, ветка master с удалённого сервера будет автоматически влита в вашу сразу после получения всех необходимых данных. Она также выдаёт список всех полученных ею ссылок.

11) 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. Они имеют контрольную сумму, содержат имя автора, его e-mail и дату создания, имеют комментарий и могут быть подписаны и проверены с помощью GNU Privacy Guard (GPG). Обычно рекомендуется создавать аннотированные теги, чтобы иметь всю перечисленную информацию; но если вы хотите сделать временную метку или по какой-то причине не хотите сохранять остальную информацию, то для этого годятся и легковесные.

12) Работа с тегами в Git довольно проста. Вот некоторые основные команды:

1. Создание тега: Чтобы создать тег, используйте команду `git tag` с указанием имени тега и коммита, к которому вы хотите привязать тег. Например:

```
git tag <имя тега> <идентификатор коммита>
```

2. Просмотр списка тегов: Чтобы просмотреть список доступных тегов, вы можете использовать команду `git tag` без аргументов:

```
git tag
```

3. Просмотр информации о теге: Чтобы посмотреть дополнительную информацию о конкретном теге, вы можете использовать команду `git show` с указанием имени тега:

```
git show <имя тега>
```

4. Удаление тега: Чтобы удалить тег, используйте команду `git tag` с опцией `-d` и указанием имени тега:

```
git tag -d <имя тега>
```

5. Отправка тегов на удаленный репозиторий: По умолчанию, теги не отправляются на удаленный репозиторий при выполнении команды `git push`. Чтобы отправить один или все теги на удаленный репозиторий, используйте команду `git push` с опцией `--tags`:

```
git push origin --tags
```

13) Флаг `--prune` в командах `git fetch` и `git push` используется для удаления удаленных веток и тегов, которые больше не существуют на удаленном репозитории.

Вот как работает флаг `--prune` в каждой из этих команд:

1. `git fetch --prune`: Команда `git fetch` используется для получения последних изменений из удаленного репозитория. При использовании флага `--prune`, Git будет проверять, какие ветки и теги удалены на удаленном репозитории, и затем удалит соответствующие ссылки на эти удаленные ветки и теги на вашем локальном репозитории. Это полезно, когда ветки или теги удалены на удаленном репозитории, но они все еще присутствуют в вашем локальном репозитории.

2. `git push --prune`: Команда `git push` используется для отправки изменений на удаленный репозиторий. При использовании флага `--prune`, Git будет проверять, какие ветки и теги удалены на вашем локальном репозитории, и

затем удалит соответствующие удаленные ветки и теги на удаленном репозитории. Это полезно, когда вы удалите ветку или тег на вашем локальном репозитории, но они все еще присутствуют на удаленном репозитории.

Использование флага `--prune` позволяет поддерживать синхронизацию между вашим локальным репозиторием и удаленным репозиторием, удаляя ссылки на удаленные ветки и теги.