

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ
ФЕДЕРАЦИИ**
**Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРОКАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Кафедра инфокоммуникаций

Институт цифрового развития

ОТЧЁТ

по лабораторной работе №1.3

Дисциплина: «Программирование на Python» Тема: «Основы
ветвления Git»

Выполнил: студент 2 курса

группы ИВТ-б-о-22-1

Болуров Ислам Расулович

Ставрополь 2023


1. Создал репозиторий с использованием лицензии MIT. Затем клонировал его на ПК:

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 *

 IslamBolurov / Lab-1.3pyt

✓ Lab-1.3pyt is available.

Great repository names are short and memorable. Need inspiration? How about **fictional-doodle** ?

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

.gitignore template: Python ▾

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

Choose a license

License: MIT License ▾

Рисунок 1.1 Создание репозитория

```
IslamBolurov@Islam666 MINGW64 ~/Desktop/lab333 (main)
$ git clone https://github.com/IslamBolurov/Lab-1.3pyt.git
Cloning into 'Lab-1.3pyt'...
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 8 (delta 1), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (8/8), done.
Resolving deltas: 100% (1/1), done.
```

Рисунок 1.2 Клонирование репозитория

2. Создал три файла, проиндексировал 1-й и сделал коммит “add 1.txt file”. Затем проиндексировал 2-й и 3-й и перезаписал коммит с новым комментарием “add 2.txt and 3.txt”:

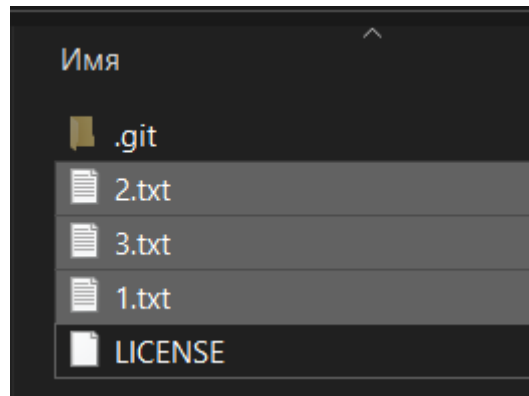


Рисунок 2.1 Созданные файлы в репозитории

```
IslamBolurov@Islam666 MINGW64 ~/Desktop/lab333/Lab-1.3pyt (main)
$ git add 1.txt.txt

IslamBolurov@Islam666 MINGW64 ~/Desktop/lab333/Lab-1.3pyt (main)
$ git commit -m "add 1.txt file"
[main 8879905] add 1.txt file
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 1.txt.txt
```

Рисунок 2.2 1-й коммит

```
IslamBolurov@Islam666 MINGW64 ~/Desktop/lab333/Lab-1.3pyt (main)
$ git add .

IslamBolurov@Islam666 MINGW64 ~/Desktop/lab333/Lab-1.3pyt (main)
$ git commit --amend -m"add 2.txt and 3txt"
[main 66af840] add 2.txt and 3txt
Date: Tue Feb 20 21:34:49 2024 +0300
3 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 1.txt.txt
create mode 100644 2.txt.txt
create mode 100644 3.txt.txt
```

Рисунок 2.3 Изменение 1-го коммита

3. Создал новую ветку my_first_branch, перешел на нее и создал в ней новый файл in_branch.txt, закоммитил изменения. После этого вернулся на ветку master, создал и сразу перешел на ветку new_branch, сделал изменения в файле 1.txt и закоммитил изменения.

```
IslamBolurov@Islam666 MINGW64 ~/Desktop/lab333/Lab-1.3pyt (main)
$ git branch my_first_branch

IslamBolurov@Islam666 MINGW64 ~/Desktop/lab333/Lab-1.3pyt (main)
$ git checkout my_first_branch
Switched to branch 'my_first_branch'

IslamBolurov@Islam666 MINGW64 ~/Desktop/lab333/Lab-1.3pyt (my_first_branch)
$ git branch
  main
* my_first_branch
```

Рисунок 3.1 Создание и переход на новую ветку

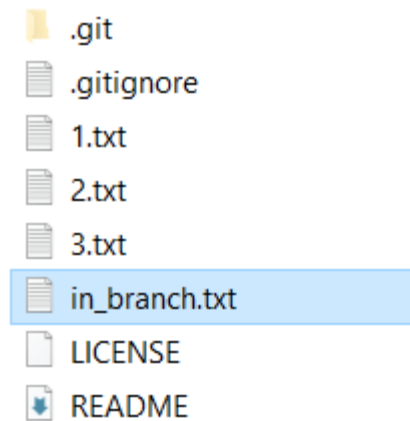


Рисунок 3.2 Добавление файла в репозиторий

```
IslamBolurov@Islam666 MINGW64 ~/Desktop/lab333/Lab-1.3pyt (my_first_branch)
$ git add .

IslamBolurov@Islam666 MINGW64 ~/Desktop/lab333/Lab-1.3pyt (my_first_branch)
$ git commit -m "add in_branch.txt"
[my_first_branch c8249be] add in_branch.txt
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 in_branch.txt.txt
```

Рисунок 3.3 Коммит изменений

```
IslamBolurov@Islam666 MINGW64 ~/Desktop/lab333/Lab-1.3pyt (my_first_branch)
$ git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)

IslamBolurov@Islam666 MINGW64 ~/Desktop/lab333/Lab-1.3pyt (main)
$ git checkout -b new_branch
Switched to a new branch 'new_branch'

IslamBolurov@Islam666 MINGW64 ~/Desktop/lab333/Lab-1.3pyt (new_branch)
$ git branch
main
my_first_branch
* new_branch
```

Рисунок 3.4 Переход на ветку main и создание + переход на новую ветку

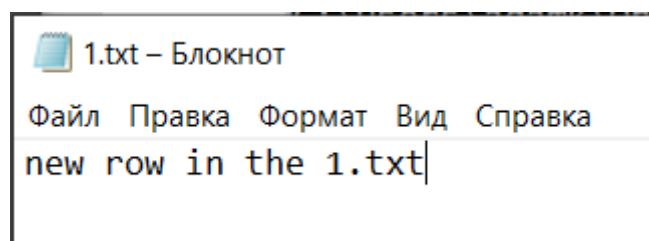


Рисунок 3.5 Изменения в файле 1.txt

```
IslamBolurov@Islam666 MINGW64 ~/Desktop/lab333/Lab-1.3pyt (new_branch)
$ git add 1.txt.txt

IslamBolurov@Islam666 MINGW64 ~/Desktop/lab333/Lab-1.3pyt (new_branch)
$ git commit -am "fix 1.txt"
[new_branch 571b166] fix 1.txt
1 file changed, 1 insertion(+)
```

Рисунок 3.6 Коммит изменений

4. Перешел на ветку main и слил ветки main и my_first_branch, после этого слил ветки main и new_branch. Удалил все ветки кроме main:

```
IslamBolurov@Islam666 MINGW64 ~/Desktop/lab333/Lab-1.3pyt (new_branch)
$ git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)

IslamBolurov@Islam666 MINGW64 ~/Desktop/lab333/Lab-1.3pyt (main)
$ git merge my_first_branch
Updating eafc623..49578ce
Fast-forward
 in_branch.txt.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 in_branch.txt.txt

IslamBolurov@Islam666 MINGW64 ~/Desktop/lab333/Lab-1.3pyt (main)
$ git merge new_branch
Merge made by the 'ort' strategy.
 1.txt.txt | 1 +
 1 file changed, 1 insertion(+)

IslamBolurov@Islam666 MINGW64 ~/Desktop/lab333/Lab-1.3pyt (main)
$ git branch -d my_first_branch
Deleted branch my_first_branch (was 49578ce).

IslamBolurov@Islam666 MINGW64 ~/Desktop/lab333/Lab-1.3pyt (main)
$ git branch -d new_branch
Deleted branch new_branch (was 8708a0d).
```

Рисунок 4.1 Слияние и удаления веток

5. Создал ветки branch_1 и branch_2. Перешел на ветку branch_1 и изменил файлы 1.txt, 3.txt, после чего закоммитил изменения. Перешел на ветку branch_2, изменил файлы 1.txt, 3.txt, после чего закоммитил изменения. После этого слил изменения ветки branch_2 в ветку branch_1:

```
IslamBolurov@Islam666 MINGW64 ~/Desktop/lab333/Lab-1.3pyt (main)
$ git branch branch_1

IslamBolurov@Islam666 MINGW64 ~/Desktop/lab333/Lab-1.3pyt (main)
$ git branch branch_2

IslamBolurov@Islam666 MINGW64 ~/Desktop/lab333/Lab-1.3pyt (main)
$ git branch
  branch_1
  branch_2
* main
```

Рисунок 5.1 Создание веток branch_1 и branch_2

```
IslamBolurov@Islam666 MINGW64 ~/Desktop/lab333/Lab-1.3pyt
$ git checkout branch_2
Switched to branch 'branch_2'
```

Рисунок 5.2 Переход на ветку branch_1

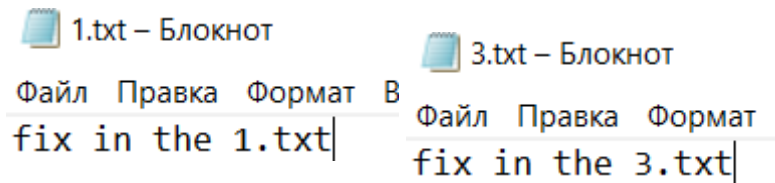


Рисунок 5.3 Изменения в файлах 1.txt и 3.txt

```
IslamBolurov@Islam666 MINGW64 ~/Desktop/lab333/Lab-1.3pyt (branch_2)
$ git commit -am "fix 1.txt & 3.txt in branch_2"
[branch_2 1394956] fix 1.txt & 3.txt in branch_2
2 files changed, 2 insertions(+), 1 deletion(-)
```

Рисунок 5.4 Коммит изменений

```
IslamBolurov@Islam666 MINGW64 ~/Desktop/lab333/Lab-1.3pyt (branch_2)
$ git checkout branch_2
Switched to branch 'branch_2'
```

Рисунок 5.5 Переход на ветку branch_2

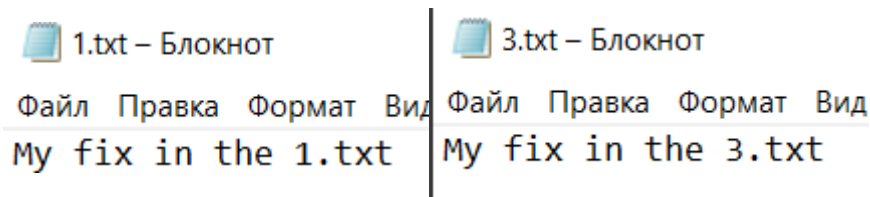


Рисунок 5.6 Изменения в файлах 1.txt и 3.txt

```
IslamBolurov@Islam666 MINGW64 ~/Desktop/lab333/Lab-1.3pyt (branch_2)
$ git commit -am "fix 1.txt & 3.txt in branch_2"
[branch_2 1394956] fix 1.txt & 3.txt in branch_2
2 files changed, 2 insertions(+), 1 deletion(-)
```

Рисунок 5.7 Коммит изменений

6. Выполнил слияние новых веток и решил конфликт файла 1.txt в ручном режиме, а 3.txt в автоматическом инструментом vimdiff, после этого отправил ветку branch_1 в GitHub:

```
IslamBolurov@Islam666 MINGW64 ~/Desktop/lab333/Lab-1.3pyt (branch_2)
$ git checkout branch_1
Switched to branch 'branch_1'

IslamBolurov@Islam666 MINGW64 ~/Desktop/lab333/Lab-1.3pyt (branch_1)
$ git merge branch_2
Updating 0ecea6..1394956
Fast-forward
 1.txt.txt | 2 +-
 3.txt.txt | 1 +
2 files changed, 2 insertions(+), 1 deletion(-)
```

Рисунок 6.1 Слияние веток

1.txt – Блокнот

Файл Правка Формат В
My fix in the 1.txt

Рисунок 6.2 Решение конфликта в ручном режиме

```
IslamBo1urov@Islam666 MINGW64 ~/Desktop/lab333/Lab-1.3pyt (branch_1)
$ git add 1.txt.txt

IslamBo1urov@Islam666 MINGW64 ~/Desktop/lab333/Lab-1.3pyt (branch_1)
$ git status
On branch branch_1
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   1.txt.txt
```

Рисунок 6.3 Добавление 1.txt в индекс, так как конфликт решен

```
IslamBo1urov@Islam666 MINGW64 ~/Desktop/lab333/Lab-1.3pyt (branch_1)
$ git mergetool

This message is displayed because 'merge.tool' is not configured.
See 'git mergetool --tool-help' or 'git help config' for more details.
'git mergetool' will now attempt to use one of the following tools:
opendiff kdiff3 tkdiff xxdiff meld tortoisemerge gvimdiff diffuse diffmerge ecme
rge p4merge araxis bc codecompare smerge emerge vimdiff nvimdiff
No files need merging
```

Рисунок 6.4 Команда mergetool

fix in the 3.txt

My fix in the 3.txt

My fix in the 3.txt

<2.txt [unix] (01:07 06/05/2022)1,1 All <.txt [unix] (01:07 06/05/2022)0,0-1 All <2.txt [unix] (01:07 06/05/2022)1,1 All

<<<<<< HEAD

My fix in the 3.txt

3.txt [dos] (01:07 06/05/2022) 1,1 All :diffg LO

Рисунок 6.5 Команда :diffg LO

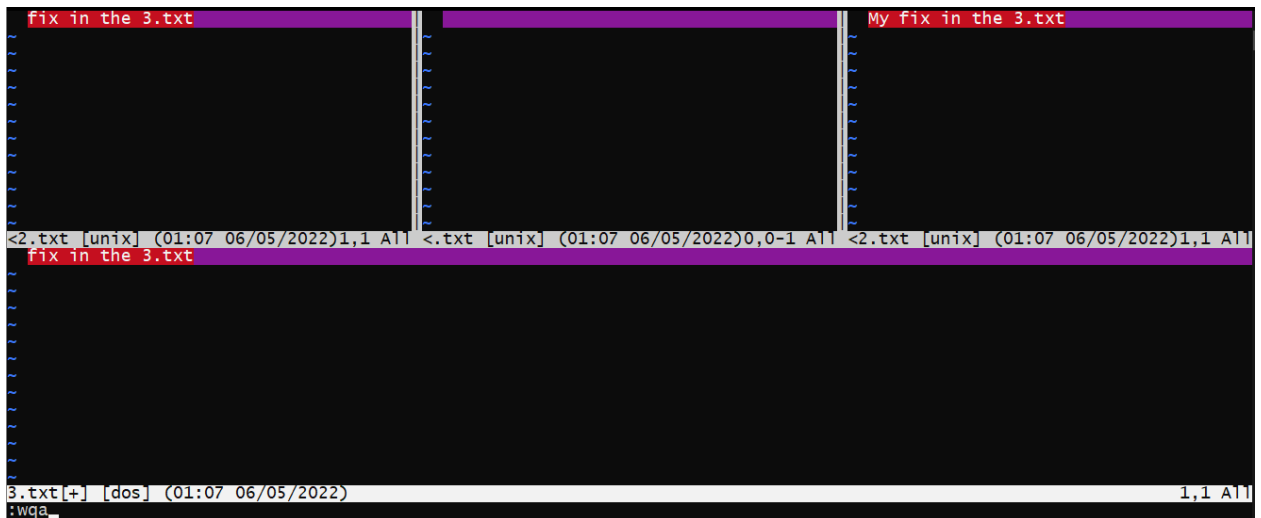


Рисунок 6.6 Команда :wga

```
IslamBolurov@Islam666 MINGW64 ~/Desktop/lab333/Lab-1.3pyt (branch_1)
$ git push --set-upstream origin branch_1
Enumerating objects: 15, done.
Counting objects: 100% (15/15), done.
Delta compression using up to 4 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (14/14), 1.21 KiB | 1.21 MiB/s, done.
Total 14 (delta 5), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (5/5), done.
remote:
remote: Create a pull request for 'branch_1' on GitHub by visiting:
remote:   https://github.com/IslamBolurov/Lab-1.3pyt/pull/new/branch_1
remote:
To https://github.com/IslamBolurov/Lab-1.3pyt.git
 * [new branch]      branch_1 -> branch_1
branch 'branch_1' set up to track 'origin/branch_1'.
```

Рисунок 6.7 Пуш ветки на уд. репозиторий

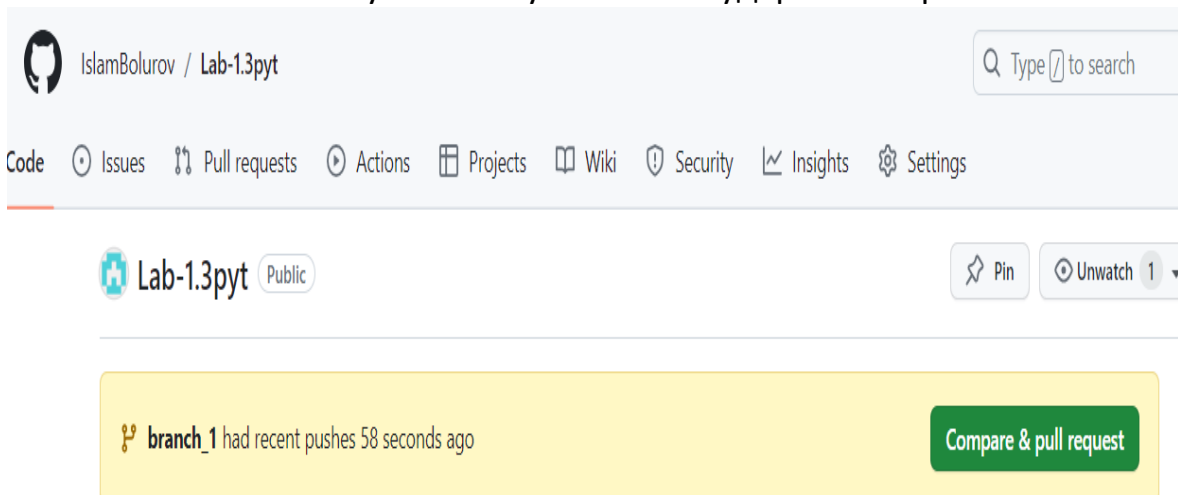
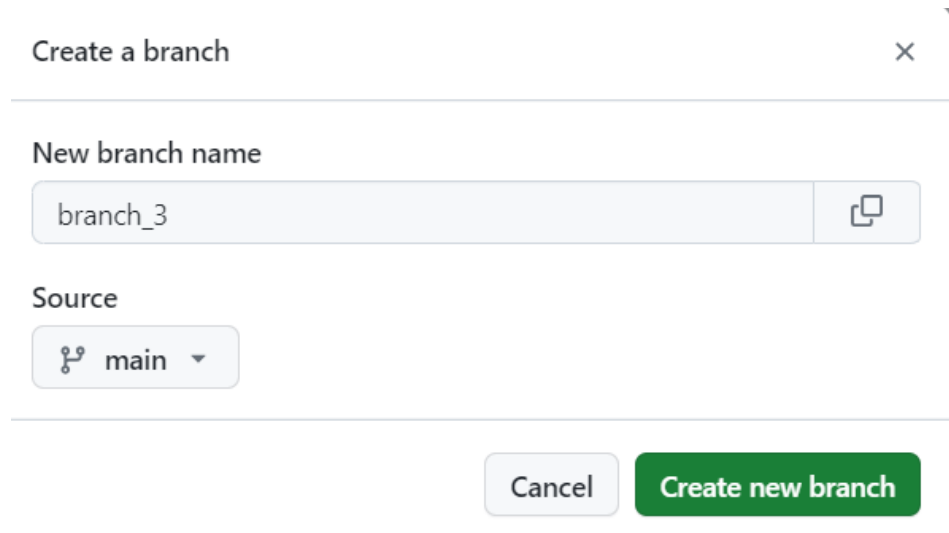


Рисунок 6.8 Изменения на уд. репозитории

7. Создал средствами GitHub удаленную ветку branch_3, создал в лок. репозитории ветку отслеживания удаленной ветки branch_3. Перешел на ветку branch_3 и добавил в файл 2.txt строку “the final fantasy in the 4.txt file”:



Create a branch

New branch name

branch_3

Source

main

Cancel Create new branch

Рисунок 7.1 Создание уд. ветки branch_3

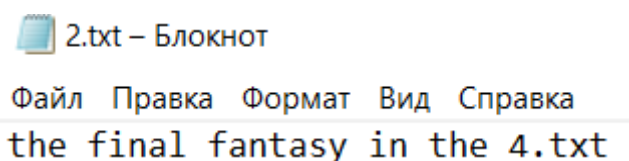
```
IslamBolurov@Islam666 MINGW64 ~/Desktop/lab333/Lab-1.3pyt (branch_1)
$ git fetch --all
From https://github.com/IslamBolurov/Lab-1.3pyt
* [new branch]      branch_3    -> origin/branch_3
```

Рисунок 7.2 Загрузка содержимого из уд. репозитория перед отслеживанием

```
IslamBolurov@Islam666 MINGW64 ~/Desktop/lab333/Lab-1.3pyt (branch_1)
$ git checkout --track origin/branch_3
Switched to a new branch 'branch_3'
branch 'branch_3' set up to track 'origin/branch_3'.

IslamBolurov@Islam666 MINGW64 ~/Desktop/lab333/Lab-1.3pyt (branch_3)
```

Рисунок 7.3 Создание ветки отслеживания branch_3



2.txt – Блокнот

Файл Правка Формат Вид Справка

the final fantasy in the 4.txt

Рисунок 7.4 Добавление в 2.txt строку “the final fantasy in the 4.txt file” на ветке branch_3

8. Выполнил перемещение ветки main на ветку branch_2 и отправил изменения веток main и branch_2 на GitHub:

```

IslamBolurov@Islam666 MINGW64 ~/Desktop/lab333/Lab-1.3pyt (branch_
$ git checkout branch_2
Switched to branch 'branch_2'

IslamBolurov@Islam666 MINGW64 ~/Desktop/lab333/Lab-1.3pyt (branch_
$ git rebase main
Current branch branch_2 is up to date.

IslamBolurov@Islam666 MINGW64 ~/Desktop/lab333/Lab-1.3pyt (branch_
$ git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 4 commits.
  (use "git push" to publish your local commits)

IslamBolurov@Islam666 MINGW64 ~/Desktop/lab333/Lab-1.3pyt (main)
$ git merge branch_2
Updating 0ecea6..1394956
Fast-forward
 1.txt.txt | 2 +-
 3.txt.txt | 1 +
 2 files changed, 2 insertions(+), 1 deletion(-)

```

Рисунок 8.1 Результат выполнения команд

```

IslamBolurov@Islam666 MINGW64 ~/Desktop/lab333/Lab-1.3pyt (main)
$ git push
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/IslamBolurov/Lab-1.3pyt.git
 67c5ac8..1394956  main -> main

IslamBolurov@Islam666 MINGW64 ~/Desktop/lab333/Lab-1.3pyt (main)
$ git checkout main
Already on 'main'
Your branch is up to date with 'origin/main'.

IslamBolurov@Islam666 MINGW64 ~/Desktop/lab333/Lab-1.3pyt (main)
$ git push
Everything up-to-date

```

Рисунок 8.2 Пуш веток

Ответы на контрольные вопросы:

1) Что такое ветка?

Это последовательность коммитов, отклоняющаяся от основной линии разработки.

2) Что такое HEAD?

Это указатель, задача которого ссылаться на определённый коммит в репозитории, если точнее: на коммит, который станет родителем для следующего коммита.

3) Способы создания веток?

Новую ветку можно создать командой `git branch <название_ветки>` или на удалённом репозитории `git hub`.

4) Как узнать текущую ветку?

При помощи команды `git branch`.

5) Как переключаться между ветками?

При помощи команды `git checkout <название_ветки>`.

6) Что такое удаленная ветка?

Это ветка, находящаяся на удаленном репозитории. Или ссылка на состояние ветки на удаленном репозитории.

7) Что такое ветка отслеживания?

Это ветка в локальном репозитории, которая напрямую связана с удаленной веткой на удаленном репозитории.

8) Как создать ветку отслеживания?

Командой `git checkout --track origin/<название_ветки>`.

9) Как отправить изменения из локальной ветки в удаленную ветку?

Командой `git push origin <название_ветки>`.

10) В чем отличие команд `git fetch` и `git pull`?

`Git pull` – это сочетание команд `git fetch` (получение изменений с удаленного репозитория) и `git merge` (объединение веток).

11) Как удалить локальную и удаленную ветки?

Используя команду `git branch -d <название_ветки>`. Для удаление удаленной ветки существует команда `git push origin -d <название_ветки>`.