

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»  
ИНСТИТУТ ЦИФРОВОГО РАЗВИТИЯ**

**Отчет о лабораторной работе №1.3 по дисциплине основы  
программной инженерии**

Выполнил:

Кожухов Филипп Денисович,  
2 курс, группа ПИЖ-б-о-20-1,

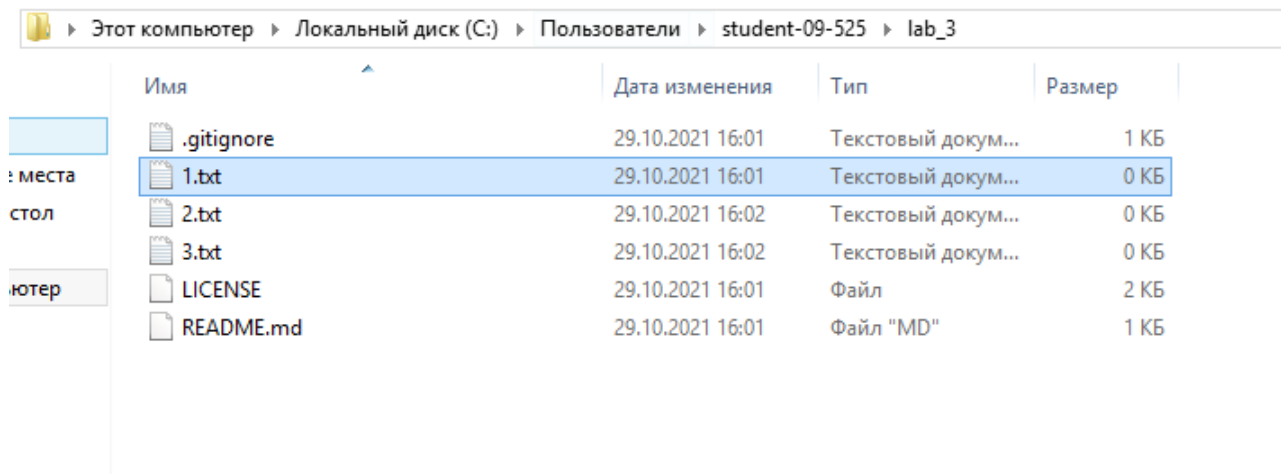
Проверил:

Доцент кафедры  
прикладной математики и  
компьютерной  
безопасности, Воронкин Р.А.

Отчет защищен с оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_

Ставрополь, 2021 г.

## ВЫПОЛНЕНИЕ:



Этот компьютер	Локальный диск (C:)	Пользователи	student-09-525	lab_3
Имя	Дата изменения	Тип	Размер	
.gitignore	29.10.2021 16:01	Текстовый докум...	1 КБ	
1.txt	29.10.2021 16:01	Текстовый докум...	0 КБ	
2.txt	29.10.2021 16:02	Текстовый докум...	0 КБ	
3.txt	29.10.2021 16:02	Текстовый докум...	0 КБ	
LICENSE	29.10.2021 16:01	Файл	2 КБ	
README.md	29.10.2021 16:01	Файл "MD"	1 КБ	

Рисунок 3.1 - Создание трёх файлов

```
C:\Users\student-09-525\lab_3>git add 1.txt

C:\Users\student-09-525\lab_3>git commit -m "add 1.txt file"
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

Untracked files:
  (use "git add <file>..." to include in what will be committed)
      2.txt
      3.txt

nothing added to commit but untracked files present (use "git add" to track)

C:\Users\student-09-525\lab_3>git push
info: please complete authentication in your browser...
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 2 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 336 bytes | 336.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/SatomiRen/lab_3.git
   bf60d57..621e8ae  main -> main

C:\Users\student-09-525\lab_3>
```

Рисунок 3.2 - Коммит первого файла

```
C:\Users\student-09-525\lab_3>git add 2.txt
C:\Users\student-09-525\lab_3>git add 3.txt
C:\Users\student-09-525\lab_3>
```

Рисунок 3.3 - Индексация 2-го и 3-го файлов

```

C:\Users\student-09-525\lab_3>git commit --amend -m "add 2.txt and 3.txt files"
[main 1e34d57] add 2.txt and 3.txt files
Author: student-09-525 <student-09-525@ncfu.net>
Date: Fri Oct 29 16:15:25 2021 +0300
Committer: SatomiRen <student-09-525@ncfu.net>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly:

    git config --global user.name "Your Name"
    git config --global user.email you@example.com

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

3 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 1.txt
create mode 100644 2.txt
create mode 100644 3.txt

```

Рисунок 3.4 - Перезапись предыдущего коммита

```

C:\Users\student-09-525\lab_3>git checkout -b "my_first_branch"
Switched to a new branch 'my_first_branch'

C:\Users\student-09-525\lab_3>git status
On branch my_first_branch
nothing to commit, working tree clean

C:\Users\student-09-525\lab_3>

```

Рисунок 3.5 - Создание ветки

```

C:\Users\CMDR Inferion\lab_3>git checkout my_first_branch
Already on 'my_first_branch'
Your branch is up to date with 'origin/my_first_branch'.

C:\Users\CMDR Inferion\lab_3>

```

Рисунок 3.6 - Переключение на созданную ветку

```

C:\Users\CMDR Inferion\lab_3>git add in_branch.txt

C:\Users\CMDR Inferion\lab_3>

```

Рисунок 3.7 - Создание файла in\_branch.txt

```

C:\Users\CMDR Inferion\lab_3>git commit -m "add my_first_branch and in_branch.txt"
[my_first_branch 6b91020] add my_first_branch and in_branch.txt
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 in_branch.txt

```

Рисунок 3.8 - Коммит созданной ветки

```
C:\Users\CMDR Inferion\lab_3>git checkout master  
Switched to branch 'master'  
Your branch is up to date with 'origin/main'.
```

Рисунок 3.9 - Переключение на ветку master

```
C:\Users\CMDR Inferion\lab_3>git checkout -b "new_branch"  
Switched to a new branch 'new_branch'
```

Рисунок 3.10 - Создание и переход на ветку new\_branch

#### ОТВЕТЫ НА ВОПРОСЫ:

1. Ветка в Git — это последовательность коммитов. С технической точки зрения ветка — это указатель или ссылка на последний коммит в этой ветке.
2. HEAD — это указатель на коммит в вашем репозитории, который станет родителем следующего коммита.
3. Способы создания веток: `git branch <имя ветки>`; `git checkout -b <имя ветки>`; `git remote add` (для создания веток в удаленном репозитории).
4. С помощью команды "`git branch`".
5. С помощью команды "`git checkout <имя ветки>`".
6. Удалённые ветки — это ссылки (указатели) в ваших удалённых репозиториях, включая ветки, теги и так далее.
7. Ветка отслеживания — это ссылки на определённое состояние удалённых веток.
8. Нужно использовать команду "`git checkout --track origin/имя_ветки`".
9. Выполните команду "`git push <remote> <branch>`".
10. `git pull` получает изменения с сервера и сливает коммиты в текущую ветку, а `git fetch` не сливает их.
11. Удаление локальной ветки: `git branch -d <имя ветки>` ; удаление удаленной ветки: `git push origin --delete <имя ветки>`.
12. Типы веток: `main`, `develop`, `feature`, `release`, `hotfix`. В `git-flow` используются функциональные ветки и несколько основных веток. `Git-flow` не слишком удобно применять в непрерывной разработке ПО (CI/CD).
- 13.