

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:  
Яблоновский Дмитрий Николаевич  
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:  
Богданов С.С., ассистент кафедры  
инфокоммуникаций

---

(подпись)

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

Ставрополь, 2023 г.

**Тема:** Исследование возможностей Git для работы с локальными репозиториями.

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

### Порядок выполнения работы

1. Создал репозиторий GitHub.

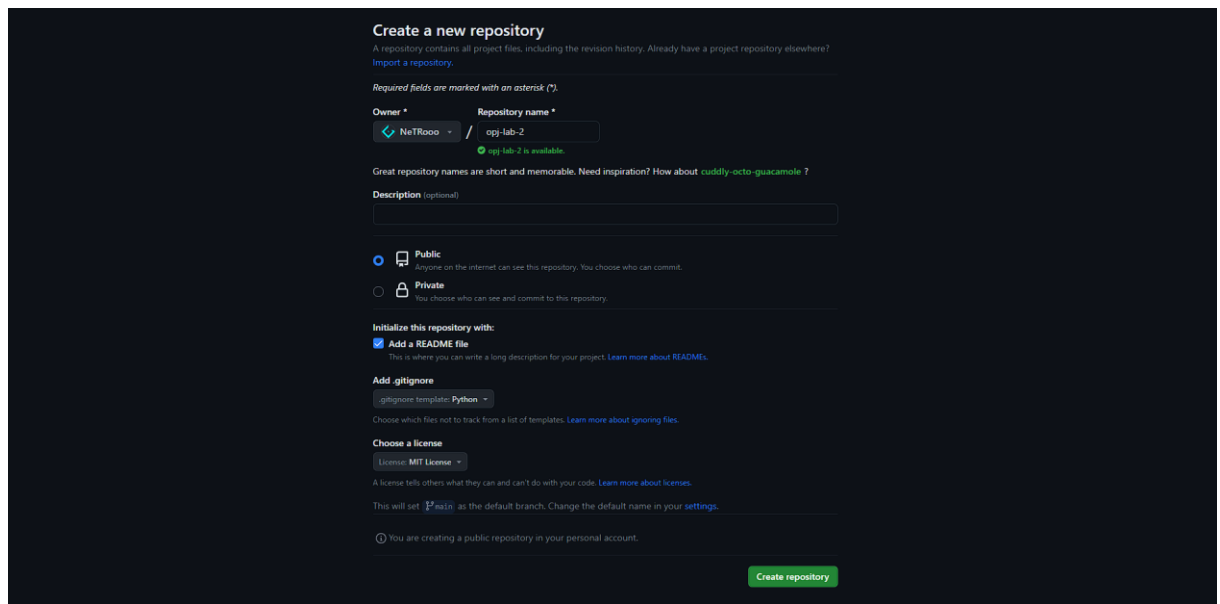


Рисунок 1- Создание репозитория

2. Проработал примеры из лабораторной работы.

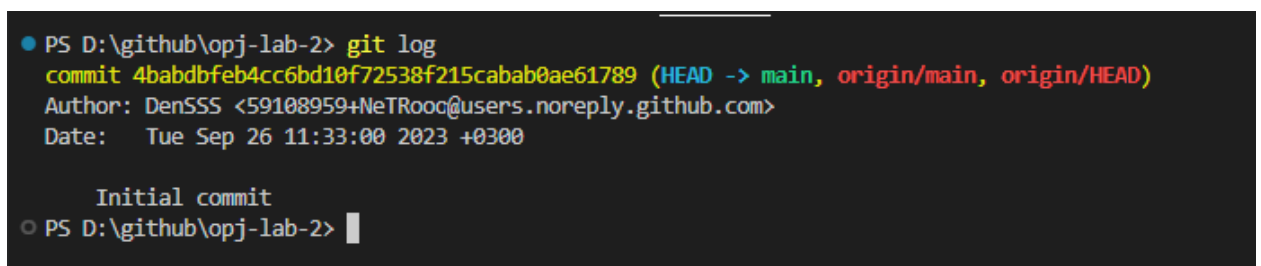


Рисунок 2.1 – История коммитов

```

PS D:\github\opj-lab-2> git log -p -2
commit 4babdbfeb4cc6bd10f72538f215cabab0ae61789 (HEAD -> main, origin/main, origin/HEAD)
Author: DenSSS <59108959+NeTRooq@users.noreply.github.com>
Date: Tue Sep 26 11:33:00 2023 +0300

    Initial commit

diff --git a/.gitignore b/.gitignore
new file mode 100644
index 0000000..68bc17f
--- /dev/null
+++ b/.gitignore
@@ -0,0 +1,160 @@
+# Byte-compiled / optimized / DLL files
+__pycache__/
+*.py[cod]
+*$py.class
+
+# C extensions
+*.so
+
+# Distribution / packaging
+*.Python
+build/
+develop-eggs/
+dist/
+downloads/
+eggs/
+*.eggs/
+lib/
+lib64/
+parts/
+sdist/
+var/
+wheels/
+share/python-wheels/
+*.egg-info/
+.installed.cfg
+*.egg
+MANIFEST
+
+# PyInstaller
+# Usually these files are written by a python script from a template
+# before PyInstaller builds the exe, so as to inject date/other infos into it.
+*.manifest
+*.spec
+
+# Installer logs
+pip-log.txt
+pip-delete-this-directory.txt
+
+# Unit test / coverage reports
+htmlcov/
+.tox/
+.nox/

```

Рисунок 2.2 – Использование команды git log с аргументов -p

```

● PS D:\github\opj-lab-2> git log --stat
commit 4babdbfeb4cc6bd10f72538f215cabab0ae61789 (HEAD -> main, origin/main, origin/HEAD)
Author: DenSSS <59108959+NeTRooq@users.noreply.github.com>
Date: Tue Sep 26 11:33:00 2023 +0300

    Initial commit

 .gitignore | 160 +++++
 LICENSE    | 21 +
 README.md  |  1 +
 3 files changed, 182 insertions(+)
○ PS D:\github\opj-lab-2>

```

Рисунок 2.3 – Сокращенная статистика

```

● PS D:\github\opj-lab-2> git log --pretty=oneline
4babdbfeb4cc6bd10f72538f215cabab0ae61789 (HEAD -> main, origin/main, origin/HEAD) Initial commit
○ PS D:\github\opj-lab-2>

```

Рисунок 2.4 – Использование команды git log с аргументом --pretty

```

PS D:\github\opj-lab-2> git remote -v
origin https://github.com/NeTRooo/opj-lab-2.git (fetch)
origin https://github.com/NeTRooo/opj-lab-2.git (push)
PS D:\github\opj-lab-2>

```

Рисунок 2.5 – Просмотр удаленных репозиториев

```

PS D:\github\opj-lab-2> git log --pretty=format:"%h - %an, %ar : %s"
4babdbf - DenSSS, 5 minutes ago : Initial commit
PS D:\github\opj-lab-2>

```

Рисунок 2.6 – Использование команды git log с аргументом –pretty с опцией format

```

PS D:\github\opj-lab-2> git remote add pb https://github.com/NeTRooo/opj-lab-2.git
PS D:\github\opj-lab-2> git remote -v
git: 'remove' is not a git command. See 'git --help'.

The most similar command is
  remote
PS D:\github\opj-lab-2> git remove -v
git: 'remove' is not a git command. See 'git --help'.

The most similar command is
  remote
PS D:\github\opj-lab-2> git remote -v
origin https://github.com/NeTRooo/opj-lab-2.git (fetch)
origin https://github.com/NeTRooo/opj-lab-2.git (push)
pb      https://github.com/NeTRooo/opj-lab-2.git (fetch)
pb      https://github.com/NeTRooo/opj-lab-2.git (push)
PS D:\github\opj-lab-2>

```

Рисунок 2.7 – Добавление удаленного репозитория

```

PS D:\github\opj-lab-2> git tag -a v1.5 -m "my love version 1.5"
PS D:\github\opj-lab-2> git tag
v1.5
PS D:\github\opj-lab-2>

```

Рисунок 2.8 – Добавление тэга

3. Выполнил клонирование репозитория на рабочий компьютер.

```

PS C:\Users\maxik\OneDrive\Desktop\hdf> git clone https://github.com/NeTRooo/opj-lab-2.git
Cloning into 'opj-lab-2'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.
PS C:\Users\maxik\OneDrive\Desktop\hdf>

```

Рисунок 3 – Клонирование репозитория

4. Добавил академические данные в файл README.

## opj-lab-2

Лабораторная работа №1 по основам программной инженерии

Выполнил: Яблоновский Дмитрий Николаевич 2 курс ПИЖ-б-о-22-1

Рисунок 4 – Измененный файл README

### 5. Написал небольшую программу на языке Python.

```

main.py x
main.py > ...
1  from flask import Flask
2  from flask import request as r
3  from flask_cors import CORS, cross_origin
4  import sqlite3
5  import json
6
7  app = Flask(__name__)
8  cors = CORS(app)
9  app.config['CORS_HEADERS'] = 'Content-Type'
10
11 def get_user_info(uid):
12     db = sqlite3.connect('heylon_api_sqlite3.sqlite3')
13     sql = db.cursor()
14     user_info = {'uid': uid}
15
16     sql.execute(f"SELECT uid FROM users WHERE uid = {uid}")
17     if sql.fetchone() is None:
18         user_info['on_server'] = "false"
19         user_info['server_nick'] = "no"
20         user_info['role'] = "no"
21     else:
22         for on_server in sql.execute(f"SELECT on_server FROM users WHERE uid = {uid}"):
23             on_server = on_server[0]
24             user_info['on_server'] = on_server
25         for server_nick in sql.execute(f"SELECT server_nick FROM users WHERE uid = {uid}"):
26             server_nick = server_nick[0]
27             user_info['server_nick'] = server_nick
28         for role in sql.execute(f"SELECT role FROM users WHERE uid = {uid}"):
29             role = role[0]
30             user_info['role'] = role
31     return user_info
32
33 @cross_origin()
34 @app.route('/api/check', methods=['GET'])
35 def save():
36     try:
37         uid = r.args.get("uid")
38         if uid:
39             try:
40                 uid = int(uid)
41             except ValueError:
42                 return 'uid must be integer'
43             user_info = get_user_info(uid)
44             return user_info
45         else:
46             return 'no "uid" param was given'
47     except Exception as e:
48         print(e)
49         return 'error'
50
51 if __name__ == '__main__':
52     app.run(host='0.0.0.0', port=8892, debug=True)

```

Рисунок 5 – Написанная программа

### 6. Просмотрел историю коммитов.

```
PS D:\github\opj-lab-2> git log
commit 4babdbfeb4cc6bd10f72538f215cabab0ae61789 (HEAD -> main, tag: v1.5)
Author: DenSSS <59108959+NeTRoooc@users.noreply.github.com>
Date: Tue Sep 26 11:33:00 2023 +0300

    Initial commit
PS D:\github\opj-lab-2>
```

Рисунок 6 – История коммитов

7. Просмотрел содержимое коммитов командой git show.

```
PS D:\github\opj-lab-2> git show HEAD
commit 76187d9536a8e98258764ee83ac3ed21cbf15249 (HEAD -> main, origin/main, origin/HEAD)
Merge: b06c1a3 7c98da2
Author: DenSSS <dimon.yablonovskiyy@mail.ru>
Date: Tue Sep 26 11:52:07 2023 +0300

    Merge branch 'main' of https://github.com/NeTRoooc/opj-lab-2
PS D:\github\opj-lab-2>
```

Рисунок 7.1 – Содержимое последнего коммита

```

PS D:\github\opj-lab-2> git show HEAD~1
commit b06c1a393d146c681376249ad95f73abb09df29d
Author: DenSSS <dimon.yablonovskiyy@mail.ru>
Date: Tue Sep 26 11:52:04 2023 +0300

    create flask app

diff --git a/main.py b/main.py
new file mode 100644
index 0000000..725371d
--- /dev/null
+++ b/main.py
@@ -0,0 +1,52 @@
+from flask import Flask
+from flask import request as r
+from flask_cors import CORS, cross_origin
+import sqlite3
+import json
+
+
+app = Flask(__name__)
+cors = CORS(app)
+app.config['CORS_HEADERS'] = 'Content-Type'
+
+def get_user_info(uid):
+    db = sqlite3.connect('heylon_api_sqlite3.sqlite3')
+    sql = db.cursor()
+    user_info = {'uid': uid}
+
+    sql.execute(f"SELECT uid FROM users WHERE uid = {uid}")
+    if sql.fetchone() is None:
+        user_info['on_server'] = "false"
+        user_info['server_nick'] = "no"
+        user_info['role'] = "no"
+    else:
+        for on_server in sql.execute(f"SELECT on_server FROM users WHERE uid = {uid}"):
+            on_server = on_server[0]
+            user_info['on_server'] = on_server
+        for server_nick in sql.execute(f"SELECT server_nick FROM users WHERE uid = {uid}"):
+            server_nick = server_nick[0]
+            user_info['server_nick'] = server_nick
+        for role in sql.execute(f"SELECT role FROM users WHERE uid = {uid}"):
+            role = role[0]
+            user_info['role'] = role
+    return user_info
+
+@cross_origin()
+@app.route('/api/check', methods=['GET'])
+def save():
+    try:
+        uid = r.args.get("uid")
+        if uid:
+            try:
+                uid = int(uid)
+            except ValueError:
+                return 'uid must be integer'
+            user_info = get_user_info(uid)
+            return user_info
+        else:
+            return 'no "uid" param was given'
+    except Exception as e:
+        print(e)

```

Рисунок 7.2 – Содержимое предпоследнего коммита

```

PS D:\github\opj-lab-2> git show 7c98da20c2f7eb0922ff67e060e1596e675ab96a
commit 7c98da20c2f7eb0922ff67e060e1596e675ab96a
Author: DenSSS <59108959+NeTrooc@users.noreply.github.com>
Date: Tue Sep 26 11:46:00 2023 +0300

    Update README.md

diff --git a/README.md b/README.md
index ed38e66..b6bbf8a 100644
--- a/README.md
+++ b/README.md
@@ -1,7 @@
-# opj-lab-2
\ No newline at end of file
+# opj-lab-2
+
+**Лабораторная работа №1 по основам программной инженерии**
+
+Выполнил:
+Яблоновский Дмитрий Николаевич
+2 курс ПИИ-6-0-22-1
PS D:\github\opj-lab-2>

```

Рисунок 7.3 – Содержимое коммита с указанным хэшем

### Ответы на контрольные вопросы

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

С помощью команды `git log`. Существуют множество дополнительных опций вот некоторые из них:

- `-p` или `-patch` – показывает разницу, внесенную в каждый коммит;
- `--stat` – позволяет увидеть сокращенную статистику;
- `--pretty` – эта опция меняет формат вывода;
- `--since` и `--until` – опции для ограничения вывода по времени

2. Как ограничить вывод при просмотре истории коммитов?

С помощью команды `git log` и его аргументов, например `--since`.

3. Как внести изменения в уже сделанный коммит?

С помощью команды `git commit --amend`.

4. Как отменить индексацию файла в Git?

С помощью команды `git reset HEAD`.

5. Как отменить изменения в файле?

С помощью команды `git checkout -- <file>`.

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?

С помощью команды `git remote -v`.

8. Как добавить удаленный репозиторий для данного локального репозитория?

С помощью команды `git remote add <shortname> <url>`.

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

Получение изменений – `git fetch [remote-name]`, отправка изменений `git push <remote-name> <branch-name>`.

10. Как выполнить просмотр удаленного репозитория?  
С помощью команды `git remote -v`.

11. Каково назначение тэгов Git?

Тэги Git - это ссылки на определенные коммиты в истории разработки. Они используются для пометки определенных версий или моментов в вашем проекте. Тэги обычно используются для обозначения релизов или важных этапов в разработке.

12. Как осуществляется работа с тэгами Git?

Для просмотра тэгов – `git tag`, для создания аннотированного тэга – `git tag -a v1.4 -m "сообщение"`, для отправки тэга на удаленный сервер – `git push origin <tagname>`, для удаления тэгов – `git tag -d <tagname>`.

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 или тэгов, которые больше не существуют на удаленном репозитории. Если выполнить команду `git fetch --prune`, Git удалит локальные ссылки на удаленные ветки и тэги, которые были удалены на удаленном репозитории. Если выполнить команду `git push --prune`, Git удалит удаленные ветки и тэги на удаленном репозитории, которых больше нет в вашем локальном репозитории.