МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт Информационных технологий (ИТ)

Кафедра Математического обеспечения и стандартизации информационных технологий (МОСИТ)

ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 1
по дисциплине
«Технология разработки программных приложений»

Тема: «Git»

Выполнил студент группы: ИКБО-50-23 Павлов Н.С.

Принял Степанов П.В.

Практическая работа выполнена «__»_____2025г. *(подпись студента)*

«Зачтено» «__»_____2025 г. *(подпись руководителя)*

Москва 2025

СОДЕРЖАНИЕ

1. ОСНОВНЫЕ КОМАНДЫ GIT

1.1 ПОСТАНОВКА ЗАДАЧИ

1. Установите и настройте клиент git на своей рабочей станции.
2. Создайте локальный репозиторий и добавьте в него несколько файлов.
3. Внесите изменения в один из файлов.
4. Проиндексируйте изменения и проверьте состояние.
5. Сделайте коммит того, что было проиндексировано в репозиторий. Добавьте к коммиту комментарий.
6. Измените еще один файл. Добавьте это изменение в индекс git. Измените файл еще раз. Проверьте состояние и произведите коммит проиндексированного изменения. Теперь добавьте второе изменение в индекс, а затем проверьте состояние с помощью команды git status. Сделайте коммит второго изменения.
7. Просмотрите историю коммитов с помощью команды git log. Ознакомьтесь с параметрами команды и используйте некоторые из них для различного формата отображения истории коммитов.
8. Верните рабочий каталог к одному из предыдущих состояний.
9. Изучите, как создавать теги для коммитов для использования в будущем.
10. Отмените некоторые изменения в рабочем каталоге (до и после индексирования).
11. Отмените один из коммитов в локальном репозитории.

1.2 ВЫПОЛНЕНИЕ ПОСТАВЛЕННЫХ ЗАДАЧ

```
C:\Users\nikni>git -v
git version 2.47.0.windows.2

C:\Users\nikni>git config --global user.name "n1kpavlov"

C:\Users\nikni>git config --global user.email "nikniknik388@gmail.com"

C:\Users\nikni>git config --global core.autocrlf true

C:\Users\nikni>git config --global core.safecrlf warn

C:\Users\nikni>git config --global core.quotepath off
```

Рисунок 1 – Настройка git

```
C:\Users\nikni>git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/etc/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.name=n1kpavlov
user.email=nikniknik388@gmail.com
core.autocrlf=true
core.safecrlf=warn
core.quotepath=off
```

Рисунок 2 – Проверка конфигурации

```
D:\Учеба\ТРПП\1_Git\repositorii>echo Hello > file1.txt

D:\Учеба\ТРПП\1_Git\repositorii>echo World!> file2.txt

D:\Учеба\ТРПП\1_Git\repositorii>git init
Initialized empty Git repository in D:/Учеба/ТРПП/1_Git/repositorii/.git/

D:\Учеба\ТРПП\1_Git\repositorii>git add file1.txt

D:\Учеба\ТРПП\1_Git\repositorii>git add file2.txt

D:\Учеба\ТРПП\1_Git\repositorii>git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   file1.txt
        new file:   file2.txt
```

Рисунок 3 – Создание локального репозитория и добавление в него файлов

```
D:\Учеба\ТРПП\1_Git\repositorii>echo Nikita> file2.txt

D:\Учеба\ТРПП\1_Git\repositorii>git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   file1.txt
        new file:   file2.txt

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   file2.txt
```

Рисунок 4 – Внесение изменений в файл

```
D:\Учеба\ТРПП\1_Git\repositorii>git add file2.txt

D:\Учеба\ТРПП\1_Git\repositorii>git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   file1.txt
        new file:   file2.txt
```

Рисунок 5 – Индексация изменения файла

```
D:\Учеба\ТРПП\1_Git\repositorii>git commit -m "Changes for file2"
[master (root-commit) 5fb92d6] Changes for file2
 2 files changed, 2 insertions(+)
 create mode 100644 file1.txt
 create mode 100644 file2.txt
```

Рисунок 6 – Создание коммита к изменению

```
D:\Учеба\ТРПП\1_Git\repositorii>echo My name is> file1.txt

D:\Учеба\ТРПП\1_Git\repositorii>git add file1.txt

D:\Учеба\ТРПП\1_Git\repositorii>echo My name is > file1.txt

D:\Учеба\ТРПП\1_Git\repositorii>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   file1.txt

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   file1.txt


D:\Учеба\ТРПП\1_Git\repositorii>git commit -m "Change file1"
[master f7f6fe7] Change file1
 1 file changed, 1 insertion(+), 1 deletion(-)
```

Рисунок 7 – Повторное изменение файла для наблюдения поведения индексации и коммитов

```
D:\Учеба\ТРПП\1_Git\repositorii>git add file1.txt

D:\Учеба\ТРПП\1_Git\repositorii>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   file1.txt


D:\Учеба\ТРПП\1_Git\repositorii>git commit -m "Change file1 again"
[master 03f7b11] Change file1 again
 1 file changed, 1 insertion(+), 1 deletion(-)
```

Рисунок 8 – Индексирование и коммит второго изменения

```
D:\Учеба\ТРПП\1_Git\repositorii>git log
commit 03f7b115549e6981030feb5cdf43e48f55774bb8 (HEAD -> master)
Author: n1kpavlov <nikniknik388@gmail.com>
Date:   Tue Feb 11 21:16:37 2025 +0300

    Change file1 again

commit f7f6fe738057e68e290bd838951003e3ae44a39e
Author: n1kpavlov <nikniknik388@gmail.com>
Date:   Tue Feb 11 21:15:23 2025 +0300

    Change file1

commit 5fb92d66cee8ff43aecbef498d0d5b7688393641
Author: n1kpavlov <nikniknik388@gmail.com>
Date:   Tue Feb 11 21:08:21 2025 +0300

    Changes for file2

D:\Учеба\ТРПП\1_Git\repositorii>git log --pretty=format:"%h %ad | %s%d [%an]" --graph --date=short
* 03f7b11 2025-02-11 | Change file1 again (HEAD -> master) [n1kpavlov]
* f7f6fe7 2025-02-11 | Change file1 [n1kpavlov]
* 5fb92d6 2025-02-11 | Changes for file2 [n1kpavlov]
```

Рисунок 9 – Просмотр истории коммитов

```
D:\Учеба\ТРПП\1_Git\repositorii>git checkout 5fb92d6
Note: switching to '5fb92d6'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

  git switch -c <new-branch-name>

Or undo this operation with:

  git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at 5fb92d6 Changes for file2

D:\Учеба\ТРПП\1_Git\repositorii>type file2.txt
Nikita

D:\Учеба\ТРПП\1_Git\repositorii>type file1.txt
Hello
```

Рисунок 10 – Возврат к одному из предыдущих состояний

```
D:\Учеба\ТРПП\1_Git\repositorii>git checkout master
Previous HEAD position was 5fb92d6 Changes for file2
Switched to branch 'master'

D:\Учеба\ТРПП\1_Git\repositorii>echo Anton>file2.txt

D:\Учеба\ТРПП\1_Git\repositorii>git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   file2.txt

no changes added to commit (use "git add" and/or "git commit -a")

D:\Учеба\ТРПП\1_Git\repositorii>git checkout file2.txt
Updated 1 path from the index

D:\Учеба\ТРПП\1_Git\repositorii>git status
On branch master
nothing to commit, working tree clean

D:\Учеба\ТРПП\1_Git\repositorii>type file2.txt
Nikita
```

Рисунок 11 – Отмена изменений до индексирования

```
D:\Учеба\ТРПП\1_Git\repositorii>echo Anton>file2.txt

D:\Учеба\ТРПП\1_Git\repositorii>git add file2.txt

D:\Учеба\ТРПП\1_Git\repositorii>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   file2.txt


D:\Учеба\ТРПП\1_Git\repositorii>git reset HEAD file2.txt
Unstaged changes after reset:
M       file2.txt

D:\Учеба\ТРПП\1_Git\repositorii>git checkout file2.txt
Updated 1 path from the index

D:\Учеба\ТРПП\1_Git\repositorii>git status
On branch master
nothing to commit, working tree clean
```

Рисунок 12 – Отмена изменений после индексирования до коммита

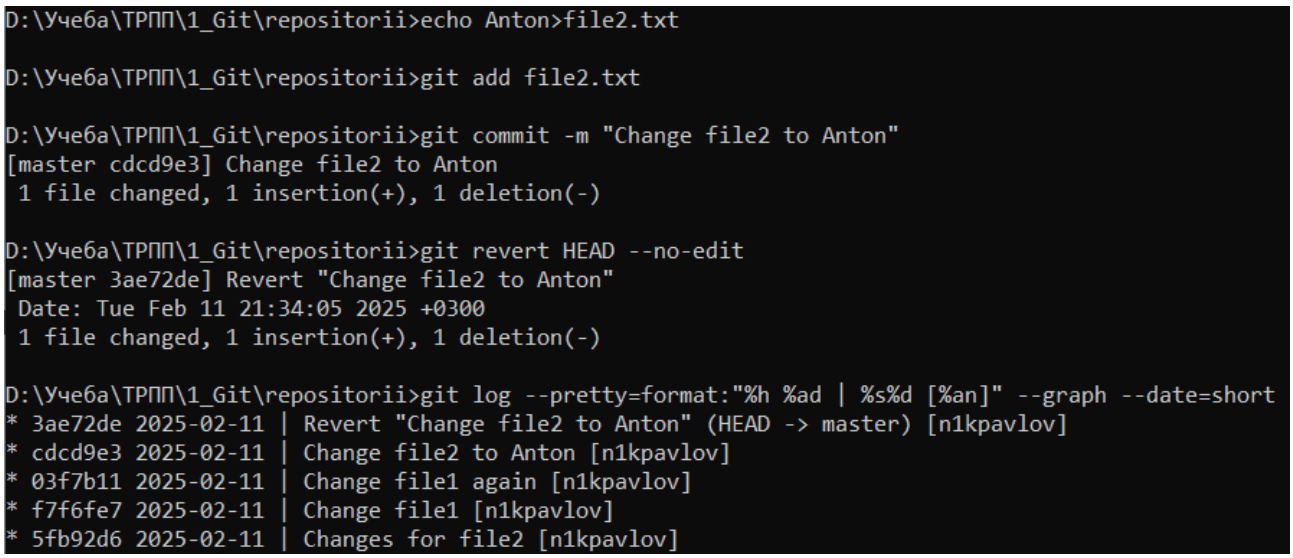

Рисунок 13 – Отмена последнего коммита

2 СИСТЕМЫ УПРАВЛЕНИЯ РЕПОЗИТОРЯИМИ

2.1 ПОСТАНОВКА ЗАДАЧИ

Персональный вариант: 2

1. Создайте аккаунт на GitHub (у кого нет),
2. Создайте репозиторий на GitHub и на локальной машине, согласно выбранной теме проекта,
3. Создайте несколько файлов на локальной машине при помощи консоли,
4. Создайте SSH-ключ для авторизации,
5. Свяжите репозиторий локальной машины с репозиторием на GitHub при помощи консоли,
6. Создайте новую ветку в репозитории с помощью команды, произведите в ней какие-нибудь изменения, а после слейте с веткой master,
7. Выполните цепочку действий в репозитории, согласно вариантам:
    - 7.1. Клонируйте непустой удаленный репозиторий на локальную машину
    - 7.2. Создайте новую ветку и выведите список всех веток
    - 7.3. Произведите коммит в ветке master
    - 7.4. Произведите 3 коммита в новой ветке в разные файлы
    - 7.5. Выгрузите изменения в удаленный репозиторий
    - 7.6. Откатите ветку обратно на 2 коммита (в том числе в удаленном репозитории)
    - 7.7. Выведите в консоли различия между веткой master и новой веткой
    - 7.8. Перебазируйте новую ветку на master

2.2 ВЫПОЛНЕНИЕ ПОСТАВЛЕННЫХ ЗАДАЧ

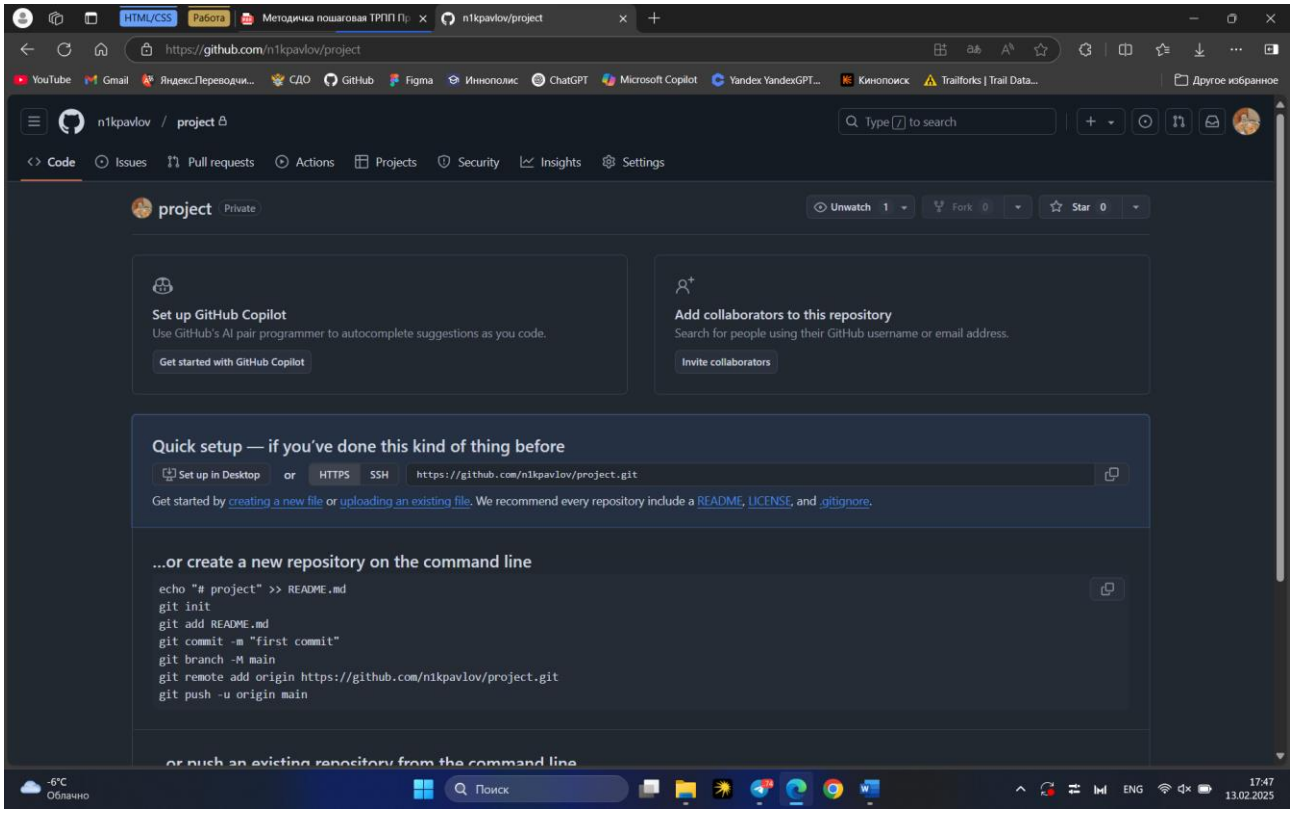

Рисунок 14 – Создание репозитория на GitHub

```
D:\Учеба\ТРПП\1_Git>mkdir project

D:\Учеба\ТРПП\1_Git>cd project

D:\Учеба\ТРПП\1_Git\project>git init
Initialized empty Git repository in D:/Учеба/ТРПП/1_Git/project/.git/
```

Рисунок 15 – Создание локального репозитория

```
D:\Учеба\ТРПП\1_Git\project>echo New file1> file1.txt

D:\Учеба\ТРПП\1_Git\project>echo New file2> file2.txt

D:\Учеба\ТРПП\1_Git\project>git add file1.txt

D:\Учеба\ТРПП\1_Git\project>git add file2.txt

D:\Учеба\ТРПП\1_Git\project>git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   file1.txt
        new file:   file2.txt


D:\Учеба\ТРПП\1_Git\project>git commit -m "Create files"
[master (root-commit) d000da2] Create files
 2 files changed, 2 insertions(+)
 create mode 100644 file1.txt
 create mode 100644 file2.txt
```

Рисунок 16 – Создание файлов в локальном репозитории

```
C:\Users\nikni>ssh-keygen -t rsa -b 4096 -C "nikniknik388@gmail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (C:\Users\nikni/.ssh/id_rsa):
Created directory 'C:\\Users\\nikni/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in C:\Users\nikni/.ssh/id_rsa
Your public key has been saved in C:\Users\nikni/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:c2O7jqDSu1A0Dv3qxjgkD/9THlZltxRafeL332B4rIU nikniknik388@gmail.com
The key's randomart image is:
+---[RSA 4096]----+
|            oo   |
|   .      ooo o .|
|  . +    o.o o o |
|   + o  .   . . .|
|    o ..S +  + ..|
|o .. .+  + oE * .|
| *.+.+..  .  = .o|
|  *o=... . ..   o|
|   =*+  ..o      |
+----[SHA256]-----+
```

Рисунок 17 – Генерация SSH ключей

```
C:\Users\nikni\.ssh>type id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQDasau1O391mmGD8ayN8hkp7alxkFGp9y0IqpZriIveWcIPeehZZo+LbEDBbhxgtkI27X2kTzOX4+oFnSHt
BdFMuVLlSYQwYVLVJ3dTbS5mYHL9qSkMHP8R3Px6TNzepV6jvH/heukyC639CXrD3k/10n+eUQEh7YIbWXUDEK1i7xR+z8pL1hSvqpRGyWyv7l9gxbh5Bl70
kn3on7GbBNOxr3ftqisgrQaEE9FNgPsuy5WfsFzN0MGdDhR4qulFuJEWXHxWoC6HaT3Ef31X7eXaBt6ovaJQseBFm2PFUfpJI7VKXNQJdoX/9xQx1YzARv1Y
0cFMTxtaSvPYn2cepQUiI93+GSyWp2gB5bdOts1CBZ9F67LrRr2skdxGY6e3zYB67ScMxNeU/BMEE4SBGKTvQApVR0PyotM+ojJ/aXfcFrtfbtmfQ4n2brzp
oTRGhesZZizY/h9RSWfNKrDqoad7IzY2o+e9H2HEHTgWqV6I6cEPoRRrf6oVt9WvA5FobhpupqaBG9sIXvNML1ui+JcphhOHfSYLL3J4qnKyyalWOFdIj8FO
poeg3glzR0LWrrL4w7m5utKHZ6LZ2jwJNN2745Z7yPrIM7pI4jSS1H5FNdZeFde3Eet05SkiiPUXX9WQGUqV9/3Is7M5ByCMyo5QtIgBbfrvm2dTGTzDYqkd
5w== nikniknik388@gmail.com
```

Рисунок 18 – Вывод в консоль публичного ключа

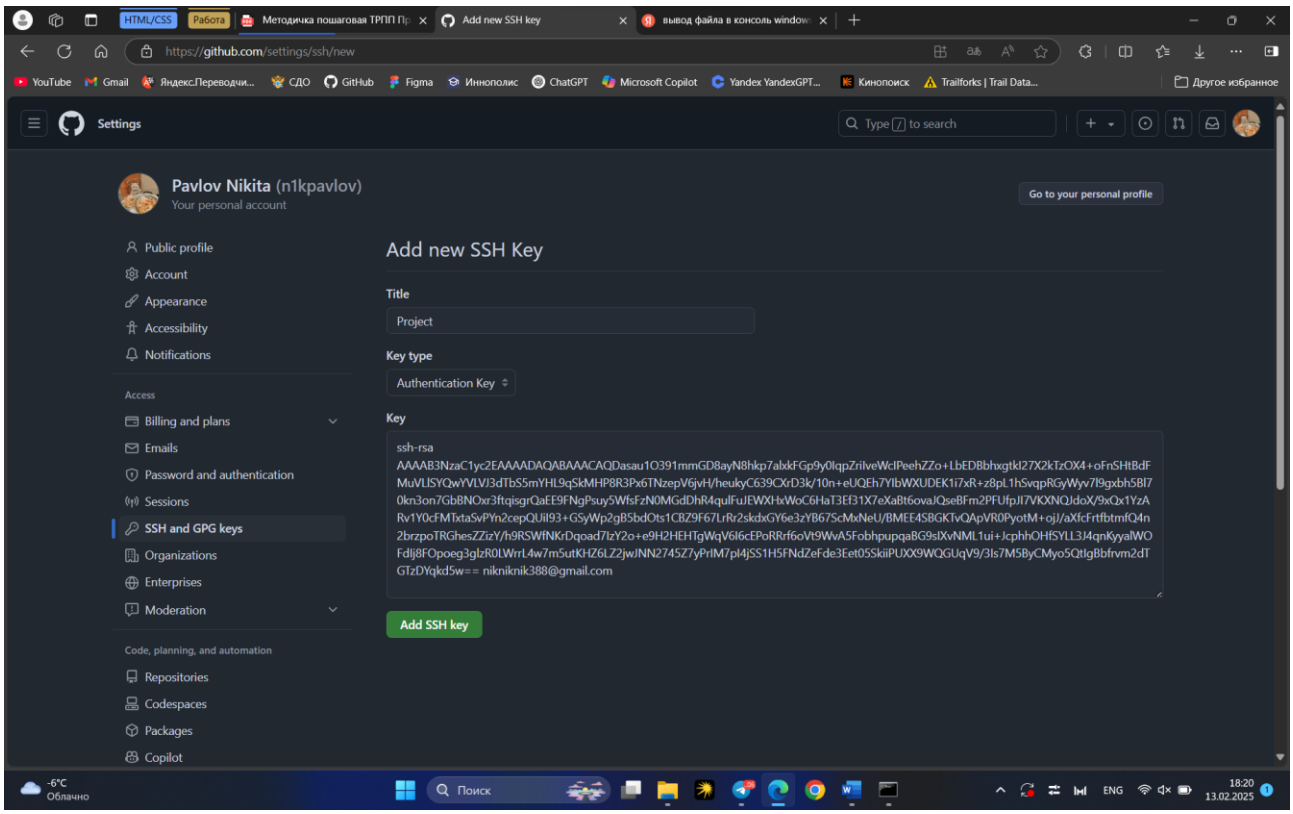

Рисунок 19 – Добавление SSH ключа на GitHub

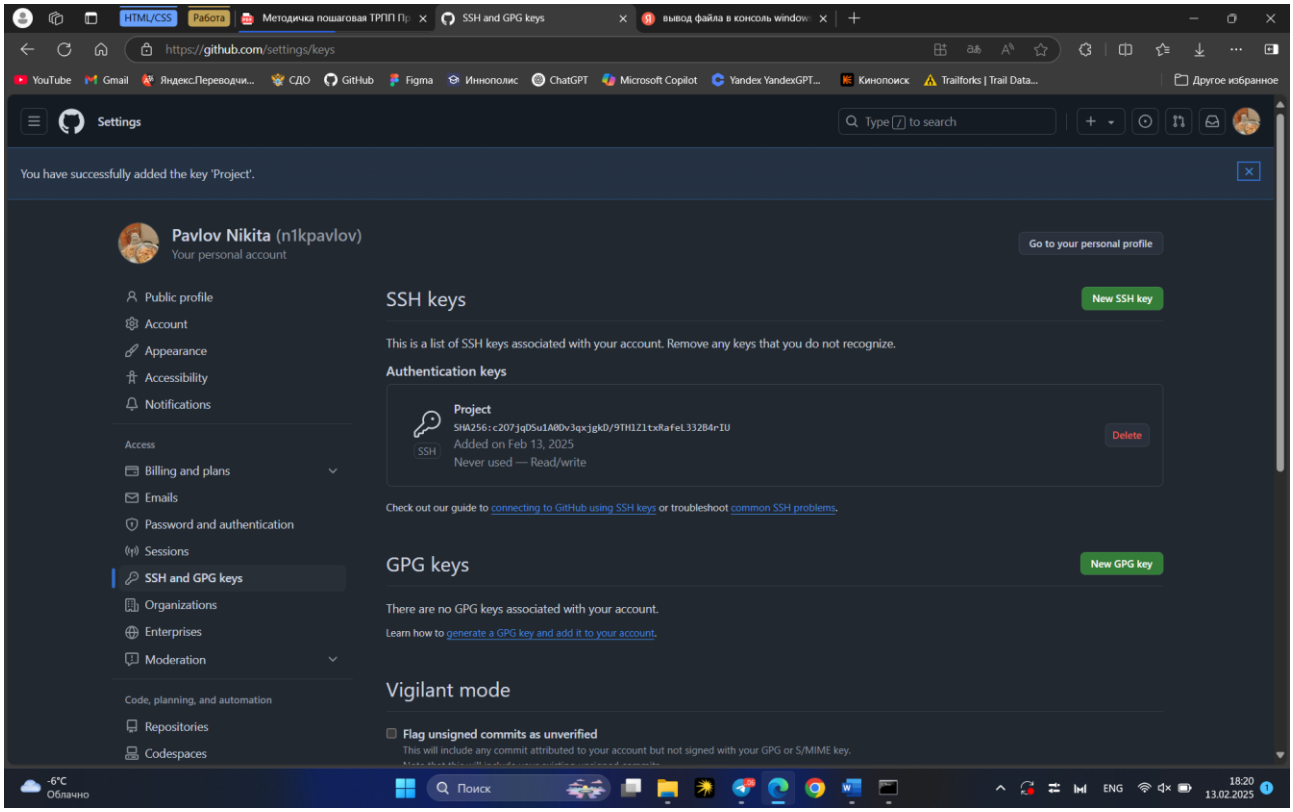

Рисунок 20 – Добавление SSH ключа на GitHub

```
D:\Учеба\ТРПП\1_Git\project>git remote add project git@github.com:n1kpavlov/project
```

Рисунок 21 – Связь локального и удаленного репозиториев

```
D:\Учеба\ТРПП\1_Git\project>git checkout -b main
Switched to a new branch 'main'

D:\Учеба\ТРПП\1_Git\project>echo Changed file1> file1.txt

D:\Учеба\ТРПП\1_Git\project>echo Changed file2 for main> file2.txt

D:\Учеба\ТРПП\1_Git\project>echo New file3> file3.txt

D:\Учеба\ТРПП\1_Git\project>git add file1.txt

D:\Учеба\ТРПП\1_Git\project>git add file2.txt

D:\Учеба\ТРПП\1_Git\project>git add file3.txt

D:\Учеба\ТРПП\1_Git\project>git commit -m "Ghanges for main"
[main d4b57dd] Ghanges for main
 3 files changed, 3 insertions(+), 2 deletions(-)
 create mode 100644 file3.txt
```

Рисунок 22 – Создание ветки main и изменения в ней

```
D:\Учеба\ТРПП\1_Git\project>git checkout master
Switched to branch 'master'

D:\Учеба\ТРПП\1_Git\project>git merge main
Updating d000da2..d4b57dd
Fast-forward
 file1.txt | 2 +-
 file2.txt | 2 +-
 file3.txt | 1 +
 3 files changed, 3 insertions(+), 2 deletions(-)
 create mode 100644 file3.txt
```

Рисунок 23 – Merge master с веткой main

```
D:\Учеба\ТРПП\1_Git>git clone https://github.com/n1kpavlov/config_converter
Cloning into 'config_converter'...
remote: Enumerating objects: 106, done.
remote: Counting objects: 100% (106/106), done.
remote: Compressing objects: 100% (103/103), done.
remote: Total 106 (delta 45), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (106/106), 41.82 KiB | 2.32 MiB/s, done.
Resolving deltas: 100% (45/45), done.

D:\Учеба\ТРПП\1_Git>cd config_converter

D:\Учеба\ТРПП\1_Git\config_converter>git remote add project git@github.com:n1kpavlov/config_converter
```

Рисунок 24 – Клонирование удаленного репозитория

```
D:\Учеба\ТРПП\1_Git\config_converter>git branch test

D:\Учеба\ТРПП\1_Git\config_converter>git branch
* main
  test
```

Рисунок 25 – Отделение новой ветки

```
D:\Учеба\ТРПП\1_Git\config_converter>echo pip install lark> script.sh

D:\Учеба\ТРПП\1_Git\config_converter>git add script.sh

D:\Учеба\ТРПП\1_Git\config_converter>git commit -m "Changed script.sh in main"
[main 44784e3] Changed script.sh in main
 1 file changed, 1 insertion(+), 1 deletion(-)
```

Рисунок 26 – Коммит в ветке main

```
D:\Учеба\ТРПП\1_Git\config_converter>git checkout test
Switched to branch 'test'

D:\Учеба\ТРПП\1_Git\config_converter>echo New file1> file1.txt

D:\Учеба\ТРПП\1_Git\config_converter>echo New file2> file2.txt

D:\Учеба\ТРПП\1_Git\config_converter>echo New file3> file3.txt

D:\Учеба\ТРПП\1_Git\config_converter>git add file1.txt

D:\Учеба\ТРПП\1_Git\config_converter>git add file2.txt

D:\Учеба\ТРПП\1_Git\config_converter>git add file3.txt

D:\Учеба\ТРПП\1_Git\config_converter>git commit -m "Create files"
[test 8e27d34] Create files
 3 files changed, 3 insertions(+)
 create mode 100644 file1.txt
 create mode 100644 file2.txt
 create mode 100644 file3.txt

D:\Учеба\ТРПП\1_Git\config_converter>echo Changed file2> file2.txt

D:\Учеба\ТРПП\1_Git\config_converter>git add file2.txt

D:\Учеба\ТРПП\1_Git\config_converter>git commit -m "Change file2"
[test bc48aca] Change file2
 1 file changed, 1 insertion(+), 1 deletion(-)

D:\Учеба\ТРПП\1_Git\config_converter>echo Changed file3> file3.txt

D:\Учеба\ТРПП\1_Git\config_converter>git add file3.txt

D:\Учеба\ТРПП\1_Git\config_converter>git commit -m "Change file3"
[test 4397a13] Change file3
 1 file changed, 1 insertion(+), 1 deletion(-)
```

Рисунок 27 – Коммиты в ветке test

```
D:\Учеба\ТРПП\1_Git\config_converter>git push origin main
info: please complete authentication in your browser...
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 283 bytes | 283.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/n1kpavlov/config_converter
   8b4f161..44784e3  main -> main

D:\Учеба\ТРПП\1_Git\config_converter>git push origin test
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 12 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (11/11), 838 bytes | 419.00 KiB/s, done.
Total 11 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 1 local object.
remote:
remote: Create a pull request for 'test' on GitHub by visiting:
remote:      https://github.com/n1kpavlov/config_converter/pull/new/test
remote:
To https://github.com/n1kpavlov/config_converter
 * [new branch]      test -> test
```

Рисунок 28 – Выгрузка изменений в удаленный репозиторий

```
D:\Учеба\ТРПП\1_Git\config_converter>git log --pretty=format:"%h %ad | %s%d [%an]" --graph --date=short
* 4397a13 2025-02-13 | Change file3 (HEAD -> test, origin/test) [n1kpavlov]
* bc48aca 2025-02-13 | Change file2 [n1kpavlov]
* 8e27d34 2025-02-13 | Create files [n1kpavlov]
* 8b4f161 2024-12-26 | Update README.md [Pavlov Nikita]
* 5374660 2024-12-17 | Update README.md [Pavlov Nikita]
* 52da3ff 2024-11-20 | Update README.md [Pavlov Nikita]
* 9e24de1 2024-11-20 | Add files via upload [Pavlov Nikita]
* 95bc072 2024-11-19 | Update README.md [Pavlov Nikita]
* bffaaaa 2024-11-19 | Update Web Application Configuration.txt [Pavlov Nikita]
* 2b67760 2024-11-19 | Update Configuring the database.txt [Pavlov Nikita]
* d7f6b71 2024-11-19 | Update Configuration of the monitoring system.txt [Pavlov Nikita]
* 18a7f92 2024-11-19 | Update Configuring the database.txt [Pavlov Nikita]
* 9e646b3 2024-11-19 | Update config converter test.py [Pavlov Nikita]
```

Рисунок 29 – Вывод списка коммитов

```
D:\Учеба\ТРПП\1_Git\config_converter>    git revert HEAD~2..HEAD~1 -n

D:\Учеба\ТРПП\1_Git\config_converter>
D:\Учеба\ТРПП\1_Git\config_converter>git commit -m "Revert последних двух коммитов"
[test d6bff4e] Revert последних двух коммитов
 1 file changed, 1 insertion(+), 1 deletion(-)

D:\Учеба\ТРПП\1_Git\config_converter>git push origin test
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 313 bytes | 156.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/n1kpavlov/config_converter
   4397a13..d6bff4e  test -> test
```

Рисунок 30 – Откат на 2 коммита назад

```
D:\Учеба\ТРПП\1_Git\config_converter>git diff main test
diff --git a/file1.txt b/file1.txt
new file mode 100644
index 0000000..f03cadd
--- /dev/null
+++ b/file1.txt
@@ -0,0 +1 @@
+New file1
diff --git a/file2.txt b/file2.txt
new file mode 100644
index 0000000..a7898f1
--- /dev/null
+++ b/file2.txt
@@ -0,0 +1 @@
+New file2
diff --git a/file3.txt b/file3.txt
new file mode 100644
index 0000000..a4afb05
--- /dev/null
+++ b/file3.txt
@@ -0,0 +1 @@
+Changed file3
diff --git a/script.sh b/script.sh
index 8f81d89..2ae681e 100644
--- a/script.sh
+++ b/script.sh
@@ -1 +1 @@
-pip install lark
+pip install lark
```

Рисунок 31 – Вывод различий веток main и test

```
D:\Учеба\ТРПП\1_Git\config_converter>git merge test
Already up to date.

D:\Учеба\ТРПП\1_Git\config_converter>git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 277 bytes | 277.00 KiB/s, done.
Total 2 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/n1kpavlov/config_converter
   44784e3..6ef13f9  main -> main
```

Рисунок 32 – Merge main с веткой test

3 РАБОТА С ВЕТВЛЕНИЕМ И ОФОРМЛЕНИЕМ КОДА

3.1 ПОСТАНОВКА ЗАДАЧИ

Персональный вариант: 4 (https://github.com/ripienaar/free-for-dev)

1. Сделайте форк репозитория в соответствии с вашим вариантом
2. Склонируйте его на локальную машину
3. Создайте две ветки branch1 и branch2 от последнего коммита в master'e
4. Проведите по 3 коммита в каждую из веток, которые меняют один и тот же кусочек файла
5. Выполните слияние ветки branch1 в ветку branch2, разрешив конфликты при этом
6. Выгрузите все изменения во всех ветках в удаленный репозиторий
7. Проведите еще 3 коммита в ветку branch1
8. Склонируйте репозиторий еще раз в другую директорию
9. В новом клоне репозитории сделайте 3 коммита в ветку branch1
10. Выгрузите все изменения из нового репозитория в удаленный репозиторий
11. Вернитесь в старый клон с репозиторием, выгрузите изменения с опцией --force
12. Получите все изменения в новом репозитории

3.2 ВЫПОЛНЕНИЕ ПОСТАВЛЕННЫХ ЗАДАЧ

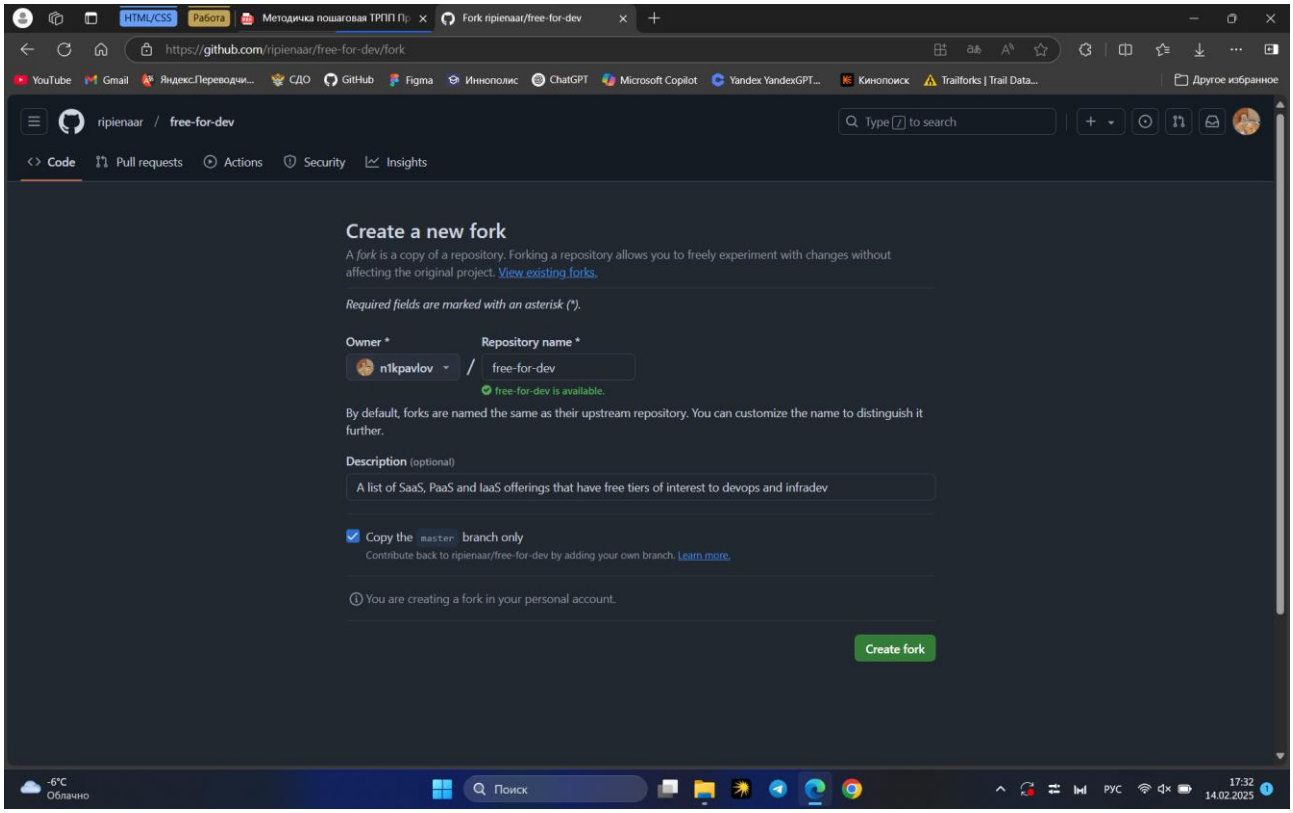

Рисунок 33 – Fork репозитория варианта

```
D:\Учеба\ТРПП\1_Git>git clone https://github.com/n1kpavlov/free-for-dev
Cloning into 'free-for-dev'...
remote: Enumerating objects: 14466, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 14466 (delta 0), reused 0 (delta 0), pack-reused 14463 (from 2)
Receiving objects: 100% (14466/14466), 8.11 MiB | 8.17 MiB/s, done.
Resolving deltas: 100% (7574/7574), done.
```

Рисунок 34 – Клонирование репозитория

```
D:\Учеба\ТРПП\1_Git>cd free-for-dev

D:\Учеба\ТРПП\1_Git\free-for-dev>git branch
* master

D:\Учеба\ТРПП\1_Git\free-for-dev>git checkout -b branch1
Switched to a new branch 'branch1'

D:\Учеба\ТРПП\1_Git\free-for-dev>git checkout -b branch2
Switched to a new branch 'branch2'
```

Рисунок 35 – Создание новых веток

```
D:\Учеба\ТРПП\1_Git\free-for-dev>git checkout branch1
Switched to branch 'branch1'

D:\Учеба\ТРПП\1_Git\free-for-dev>notepad index.html

D:\Учеба\ТРПП\1_Git\free-for-dev>git add index.html

D:\Учеба\ТРПП\1_Git\free-for-dev>git commit -m "First commit"
[branch1 aa89205] First commit
 1 file changed, 1 insertion(+), 1 deletion(-)

D:\Учеба\ТРПП\1_Git\free-for-dev>notepad index.html

D:\Учеба\ТРПП\1_Git\free-for-dev>git add index.html

D:\Учеба\ТРПП\1_Git\free-for-dev>git commit -m "Second commit"
[branch1 87235e5] Second commit
 1 file changed, 1 insertion(+), 1 deletion(-)

D:\Учеба\ТРПП\1_Git\free-for-dev>notepad index.html

D:\Учеба\ТРПП\1_Git\free-for-dev>git add index.html

D:\Учеба\ТРПП\1_Git\free-for-dev>git commit -m "Fird commit"
[branch1 6821d53] Fird commit
 1 file changed, 1 insertion(+), 1 deletion(-)
```

Рисунок 36 – 3 коммита в первой ветке

```
D:\Учеба\ТРПП\1_Git\free-for-dev>git checkout branch2
Switched to branch 'branch2'

D:\Учеба\ТРПП\1_Git\free-for-dev>notepad index.html

D:\Учеба\ТРПП\1_Git\free-for-dev>git add index.html

D:\Учеба\ТРПП\1_Git\free-for-dev>git commit -m "First commit"
[branch2 4999ac0] First commit
 1 file changed, 1 insertion(+), 1 deletion(-)

D:\Учеба\ТРПП\1_Git\free-for-dev>notepad index.html

D:\Учеба\ТРПП\1_Git\free-for-dev>git add index.html

D:\Учеба\ТРПП\1_Git\free-for-dev>git commit -m "Second commit"
[branch2 14b80aa] Second commit
 1 file changed, 1 insertion(+), 1 deletion(-)

D:\Учеба\ТРПП\1_Git\free-for-dev>notepad index.html

D:\Учеба\ТРПП\1_Git\free-for-dev>git add index.html

D:\Учеба\ТРПП\1_Git\free-for-dev>git commit -m "Fird commit"
[branch2 749d654] Fird commit
 1 file changed, 1 insertion(+), 1 deletion(-)
```

Рисунок 37 – 3 коммита во второй ветке

```
D:\Учеба\ТРПП\1_Git\free-for-dev>git merge branch1
error: Merging is not possible because you have unmerged files.
hint: Fix them up in the work tree, and then use 'git add/rm <file>'
hint: as appropriate to mark resolution and make a commit.
fatal: Exiting because of an unresolved conflict.

D:\Учеба\ТРПП\1_Git\free-for-dev>notepad index.html

D:\Учеба\ТРПП\1_Git\free-for-dev>git add index.html

D:\Учеба\ТРПП\1_Git\free-for-dev>git commit -m "merged branch2 and branch1"
[branch2 823acdf] merged branch2 and branch1
```

Рисунок 38 – Merge branch1 в branch2 и разрешение конфликтов

```
D:\Учеба\ТРПП\1_Git\free-for-dev>git push origin branch2
Enumerating objects: 21, done.
Counting objects: 100% (21/21), done.
Delta compression using up to 12 threads
Compressing objects: 100% (19/19), done.
Writing objects: 100% (19/19), 1.63 KiB | 416.00 KiB/s, done.
Total 19 (delta 14), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (14/14), completed with 2 local objects.
remote:
remote: Create a pull request for 'branch2' on GitHub by visiting:
remote:      https://github.com/n1kpavlov/free-for-dev/pull/new/branch2
remote:
To https://github.com/n1kpavlov/free-for-dev
 * [new branch]      branch2 -> branch2

D:\Учеба\ТРПП\1_Git\free-for-dev>git push origin branch1
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'branch1' on GitHub by visiting:
remote:      https://github.com/n1kpavlov/free-for-dev/pull/new/branch1
remote:
To https://github.com/n1kpavlov/free-for-dev
 * [new branch]      branch1 -> branch1
```

Рисунок 39 – Выгрузка изменений в удаленный репозиторий

```
D:\Учеба\ТРПП\1_Git\free-for-dev>git checkout branch1
Switched to branch 'branch1'

D:\Учеба\ТРПП\1_Git\free-for-dev>notepad index.html

D:\Учеба\ТРПП\1_Git\free-for-dev>git add index.html

D:\Учеба\ТРПП\1_Git\free-for-dev>git commit -m "New first commit"
[branch1 82e64e3] New first commit
 1 file changed, 1 insertion(+), 1 deletion(-)

D:\Учеба\ТРПП\1_Git\free-for-dev>notepad index.html

D:\Учеба\ТРПП\1_Git\free-for-dev>git add index.html

D:\Учеба\ТРПП\1_Git\free-for-dev>git commit -m "New second commit"
[branch1 291fd6c] New second commit
 1 file changed, 1 insertion(+), 1 deletion(-)

D:\Учеба\ТРПП\1_Git\free-for-dev>notepad index.html

D:\Учеба\ТРПП\1_Git\free-for-dev>git add index.html

D:\Учеба\ТРПП\1_Git\free-for-dev>git commit -m "New fird commit"
[branch1 b81ca0b] New fird commit
 1 file changed, 1 insertion(+), 1 deletion(-)
```

Рисунок 40 – Новый 3 коммита в первой ветке

```
D:\Учеба\ТРПП\1_Git>mkdir new_clone

D:\Учеба\ТРПП\1_Git>cd new_clone

D:\Учеба\ТРПП\1_Git\new_clone>git clone https://github.com/n1kpavlov/free-for-dev
Cloning into 'free-for-dev'...
remote: Enumerating objects: 14485, done.
remote: Counting objects: 100% (22/22), done.
remote: Compressing objects: 100% (14/14), done.
remote: Total 14485 (delta 13), reused 13 (delta 8), pack-reused 14463 (from 2)
Receiving objects: 100% (14485/14485), 8.11 MiB | 9.88 MiB/s, done.
Resolving deltas: 100% (7587/7587), done.
```

Рисунок 41 – Повторное клонирование репозитория

```
D:\Учеба\ТРПП\1_Git\new_clone\free-for-dev>git checkout -b branch1
Switched to a new branch 'branch1'

D:\Учеба\ТРПП\1_Git\new_clone\free-for-dev>notepad index.html

D:\Учеба\ТРПП\1_Git\new_clone\free-for-dev>git add index.html

D:\Учеба\ТРПП\1_Git\new_clone\free-for-dev>git commit -m "Clone first commit"
[branch1 f386aa6] Clone first commit
 1 file changed, 1 insertion(+), 1 deletion(-)

D:\Учеба\ТРПП\1_Git\new_clone\free-for-dev>notepad index.html

D:\Учеба\ТРПП\1_Git\new_clone\free-for-dev>git add index.html

D:\Учеба\ТРПП\1_Git\new_clone\free-for-dev>git commit -m "Clone second commit"
[branch1 4df8707] Clone second commit
 1 file changed, 1 insertion(+), 1 deletion(-)

D:\Учеба\ТРПП\1_Git\new_clone\free-for-dev>notepad index.html

D:\Учеба\ТРПП\1_Git\new_clone\free-for-dev>git add index.html

D:\Учеба\ТРПП\1_Git\new_clone\free-for-dev>git commit -m "Clone fird commit"
[branch1 d8a86fa] Clone fird commit
 1 file changed, 1 insertion(+), 1 deletion(-)
```

Рисунок 42 – 3 коммита в branch1

```
D:\Учеба\ТРПП\1_Git\new_clone\free-for-dev>git pull origin branch1
From https://github.com/n1kpavlov/free-for-dev
 * branch            branch1    -> FETCH_HEAD
Auto-merging index.html
CONFLICT (content): Merge conflict in index.html
Automatic merge failed; fix conflicts and then commit the result.
```

Рисунок 43 – Выгрузка изменений из нового клона

```
D:\Учеба\ТРПП\1_Git\free-for-dev>git push origin --force branch1
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 12 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (9/9), 824 bytes | 824.00 KiB/s, done.
Total 9 (delta 6), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (6/6), completed with 2 local objects.
To https://github.com/n1kpavlov/free-for-dev
   6821d53..b81ca0b  branch1 -> branch1
```

Рисунок 44 – Выгрузка изменений из старого клона

4 ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какие существуют типы систем контроля версий? Приведите примеры к каждому типу.

Существует два основных типа систем контроля версий:

* **Централизованные системы контроля версий (CVCS)**: В централизованных системах вся история проекта хранится на одном центральном сервере. Разработчики получают копии файлов с сервера и вносят изменения, которые затем отправляются обратно на сервер.

* Преимущества: Простота настройки и использования.

* Недостатки: Единственная точка отказа (сервер). Работа без сети ограничена.

* Примеры: Subversion (SVN), CVS, Perforce.

* **Распределенные системы контроля версий (DVCS):** В распределенных системах каждый разработчик имеет полную копию репозитория, включая всю историю изменений. Это позволяет работать автономно и делает систему более устойчивой к сбоям.

* Преимущества: Работа без сети. Более высокая скорость работы. Большая гибкость.

* Недостатки: Более сложная настройка. Большой размер локального репозитория (иногда).

* Примеры: Git, Mercurial, Bazaar.

2. К какому типу систем контроля версий относится Git?

Git относится к распределенным системам контроля версий (DVCS).

3. Что такое репозиторий Git?

Репозиторий Git — это хранилище всей информации о проекте, включая все файлы, историю изменений, ветки и теги. Он содержит:

* Рабочий каталог (Working Directory): Файлы, с которыми вы работаете непосредственно на своем компьютере.

* Область подготовки (Staging Area): Файлы, которые вы отметили для включения в следующий коммит.

* Git Directory (Repository): Собственно база данных Git, содержащая всю историю проекта.

4. Что такое коммит?

Коммит (commit) — это снимок состояния всех файлов проекта в определенный момент времени. Каждый коммит имеет уникальный идентификатор (хеш), автора, дату и сообщение, описывающее внесенные изменения. Коммиты образуют историю изменений проекта.

5. Что такое ветка в репозитории Git?

Ветка (branch) — это независимая линия разработки в репозитории Git. Она позволяет работать над новыми функциями, исправлять ошибки или проводить эксперименты, не затрагивая основную ветку разработки (обычно `main` или `master`).

6. Что такое тег в репозитории Git?

Тег (tag) — это метка, которую можно присвоить определенному коммиту. Теги обычно используются для обозначения важных версий проекта (например, `v1.0`, `v2.0`).

7. Что такое слияние двух веток?

Слияние (merge) — это процесс объединения изменений из одной ветки в другую. Например, после завершения работы над новой функцией в отдельной ветке, ее можно слить с основной веткой разработки.

8. Что такое конфликт в Git? Как его решить и почему они бывают?

Конфликт возникает, когда Git не может автоматически объединить изменения из разных веток. Это происходит, если в одной и той же строке одного и того же файла были внесены разные изменения в разных ветках.

Причины конфликтов:

* Параллельная разработка: Двое или более разработчиков изменяют один и тот же участок кода одновременно.

* Слияние веток с расходящимися изменениями: Если ветки разрабатывались независимо друг от друга в течение длительного времени, вероятность конфликтов увеличивается.

Решение конфликтов:

1. Открыть файл с конфликтом: Git пометит конфликтующие строки специальными маркерами (<<<<<<<, =======, >>>>>>>).

2. Редактировать файл: Вручную удалить маркеры конфликтов и выбрать нужную версию кода или объединить изменения.

3. Добавить измененный файл в область подготовки: `git add <имя_файла>`

4. Закоммитить изменения: `git commit -m "Resolved merge conflict"`

После решения конфликта слияние веток будет завершено.

5 ВЫВОДЫ

В ходе выполнения практической работы были получены навыки работы с командной строкой и Git. Изучены основные команды Git, система управления репозиториями и работа с ветвлениями и оформлением кода.