



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**

**«Технологии разработки программных приложений»**

**Тема: «Системы контроля версий»**

Выполнил студент группы ИНБО-12-23


Албахтин И.В.

Принял

Петренко А.А.

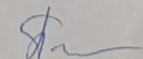
Практическая работа выполнена

✓ «26» 02 2025 г.

✓   
(подпись студента)

«Зачтено»

✓ «05» 03 2025 г.

  
(подпись руководителя)

Москва 2025

## Оглавление

Часть 1. Основные команды Git.....	4
1. Установите и настройте клиент Git на своей рабочей станции.....	4
2. Настройка Git.....	4
3. Начало работы – создание папки и файлов.....	5
4. Создание репозитория.....	6
5. Добавление файла в репозиторий и добавление первого коммита.....	6
6. Индексация изменений.....	6
7. Коммиты нескольких изменений.....	7
8. Просмотр истории коммитов.....	8
9. Получение старых версий.....	11
10. Отмена локальных изменений (до индексации).....	13
11. Отмена локальных изменений (после индексации и до коммита).....	15
12. Отмена коммита.....	15
Часть 2. Системы управления репозиториями.....	16
1. Создание аккаунта на GitHub.....	16
2. Создание SSH-ключа для авторизации.....	16
3. Создание нового репозитория для своего проекта.....	17
4. Связываем локальный и удаленный репозитории.....	18
5. Создание веток и переключение между ними.....	18
6. Слияние веток.....	19
7. Выполните цепочку действий в репозитории, согласно вариантам. Вариант 3. ....	20
2) Создайте новую ветку и выведите список всех веток.....	20
3) Произведите 5 коммитов в новой ветке в разные файлы.....	20
4) Выгрузите изменения в удаленный репозиторий.....	21
5) Произведите revert 2-го и 4-го коммита в новой ветке.....	21
6) Выведите в консоли различия между веткой master и новой веткой.....	22
7) Слейте новую ветку с master при помощи merge.....	22
Часть 3. Работа с ветвлением и оформление кода.....	23
1. Форк репозитория.....	23
2. Клонирование репозитория на локальную машину.....	23
3. Создание двух веток в deprecation-note.....	23
4. 3 коммита в каждую из веток, которые меняют один и тот же кусочек файла.....	24
5. Слияние ветки branch1 в ветку branch2 с разрешением конфликтов.....	26
6. Выгрузка всех изменений во всех ветках в удаленный репозиторий.....	28
7. Выполнение 3-х коммитов в branch1.....	29
8. Повторное клонирование репозитория в другую директорию.....	29
9. Выполнение 3-х коммитов в ветку branch1.....	30

10.	Выгрузка всех изменений в удаленный репозиторий.....	30
11.	Выгрузка изменения из старого репозитория с опцией –force .....	31
12.	Получение всех изменения в новом репозитории.....	31
	Ответы на 8 любых вопросов:.....	32
9.	Как отменить слияние веток, если произошел конфликт? .....	32
10.	Для чего нужен .gitignore? .....	32
11.	Что делает команда git status? .....	32
12.	Что делает команда git add? .....	32
15.	Что делает команда git show? .....	32
16.	Что делает команда git stash? .....	32
	Вывод.....	33

## Часть 1. Основные команды Git

### 1. Установите и настройте клиент Git на своей рабочей станции.






Имя	Дата изменения	Тип	Размер
 Git Bash	18.02.2025 19:47	Ярлык	2 КБ
 Git CMD	18.02.2025 19:47	Ярлык	2 КБ
 Git FAQs (Frequently Asked Questions)	18.02.2025 19:47	Ярлык Интернета	1 КБ
 Git GUI	18.02.2025 19:47	Ярлык	2 КБ
 Git Release Notes	18.02.2025 19:47	Ярлык	2 КБ

Рисунок 1 – установленный Git на рабочей станции

### 2. Настройка Git

```
ilalb@VeX MINGW64 ~  
$ git config --global user.name "VeXEveryOne"  
  
ilalb@VeX MINGW64 ~  
$ git config --global user.email "il.albahtin@gmail.com"  
  
ilalb@VeX MINGW64 ~  
$ git config --global core.autocrlf true  
  
ilalb@VeX MINGW64 ~  
$ git config --global core.safecrlf warn  
  
ilalb@VeX MINGW64 ~  
$ git config --global core.quotepath off
```

Рисунок 2 – Указание логина и почты для настройки Git

```
user.name=VeXEveryOne  
user.email=il.albahtin@gmail.com  
core.autocrlf=true  
core.safecrlf=warn  
core.quotepath=off
```

Рисунок 3 – Проверка настройки Git

### 3. Начало работы – создание папки и файлов

```
ilalb@VeX MINGW64 ~  
$ cd Documents  
  
ilalb@VeX MINGW64 ~/Documents  
$ |  
  
ilalb@VeX MINGW64 ~/Documents  
$ mkdir testrepositorii  
  
ilalb@VeX MINGW64 ~/Documents  
$  
  
ilalb@VeX MINGW64 ~/Documents  
$ cd testrepositorii  
  
ilalb@VeX MINGW64 ~/Documents/testrepositorii  
$ |  
  
ilalb@VeX MINGW64 ~/Documents/testrepositorii  
$ touch testsite.html  
  
ilalb@VeX MINGW64 ~/Documents/testrepositorii  
$ .....
```

Рисунки 4-7 – Создание директории и внесение изменений в файлы директории

#### 4. Создание репозитория

```
ilalb@veX MINGW64 ~/Documents/testrepositorii
$ git init
Initialized empty Git repository in C:/Users/ilalb/Documents/testrepositorii/.git/

ilalb@veX MINGW64 ~/Documents/testrepositorii (master)
$
```

Рисунок 8 – Инициализация репозитория

#### 5. Добавление файла в репозиторий и добавление первого коммита

```
ilalb@veX MINGW64 ~/Documents/testrepositorii (master)
$ git add testsite.html
warning: in the working copy of 'testsite.html', LF will be replaced by CRLF the
next time Git touches it

ilalb@veX MINGW64 ~/Documents/testrepositorii (master)
$ git commit -m "Первый коммит"
[master (root-commit) d4ff2b4] Первый коммит
1 file changed, 1 insertion(+)
create mode 100644 testsite.html

ilalb@veX MINGW64 ~/Documents/testrepositorii (master)
$
```

Рисунок 9 – Сделанный коммит с комментарием к нему

```
ilalb@veX MINGW64 ~/Documents/testrepositorii (master)
$ git status
On branch master
nothing to commit, working tree clean
```

Рисунок 10 - Проверка состояния ветки

#### 6. Индексация изменений

```
ilalb@veX MINGW64 ~/Documents/testrepositorii (master)
$ git add testsite.html
warning: in the working copy of 'testsite.html', LF will be replaced by CRLF the
next time Git touches it

ilalb@veX MINGW64 ~/Documents/testrepositorii (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   testsite.html
```

Рисунок 11 – Внесение изменений в файл

## 7. Коммиты нескольких изменений

```
ilalb@VeX MINGW64 ~/Documents/testrepositorii (master)
$ nano testsite.html

ilalb@VeX MINGW64 ~/Documents/testrepositorii (master)
$ git add testsite.html
warning: in the working copy of 'testsite.html', LF will be replaced by CRLF the
next time Git touches it

ilalb@VeX MINGW64 ~/Documents/testrepositorii (master)
$ nano testsite.html

ilalb@VeX MINGW64 ~/Documents/testrepositorii (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   testsite.html

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:   testsite.html
```

```
ilalb@VeX MINGW64 ~/Documents/testrepositorii (master)
$ git commit -m "Added standart HTML page tags"
[master 7a6885e] Added standart HTML page tags
1 file changed, 5 insertions(+), 2 deletions(-)

ilalb@VeX MINGW64 ~/Documents/testrepositorii (master)
$ 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:   testsite.html

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

```
ilalb@VeX MINGW64 ~/Documents/testrepositorii (master)
$ git add .
warning: in the working copy of 'testsite.html', LF will be replaced by CRLF the
next time Git touches it

ilalb@VeX MINGW64 ~/Documents/testrepositorii (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   testsite.html
```

```
ilalb@VeX MINGW64 ~/Documents/testrepositorii (master)
$ git commit -m "Added HTML header"
[master 7406a17] Added HTML header
1 file changed, 3 insertions(+)
```

Рисунок 12-15 – создание двух коммитов с изменениями файла proekt.html

## 8. Просмотр истории коммитов

```
ilalb@VeX MINGW64 ~/Documents/testrepositorii (master)
$ git log
commit 7406a17876f5feec7f2a1b689e7da77b24b1a773 (HEAD -> master)
Author: VeXEveryOne <il.albahtin@gmail.com>
Date: Tue Feb 18 20:31:53 2025 +0300
```

Added HTML header

```
commit 7a6885e89324f89ac545a91a8c45560b44dff566
Author: VeXEveryOne <il.albahtin@gmail.com>
Date: Tue Feb 18 20:30:23 2025 +0300
```

Added standart HTML page tags

```
commit 335dcfa76ed8287d864ee6ae77f4e31ea1b0f844
Author: VeXEveryOne <il.albahtin@gmail.com>
Date: Tue Feb 18 20:24:25 2025 +0300
```

Unrelated change to c

```
commit 558bfc64b5be0904fd8083cdcb5b6a3b2ad1e1e8
Author: VeXEveryOne <il.albahtin@gmail.com>
Date: Tue Feb 18 20:22:48 2025 +0300
```

Changes for a and b

```
commit d4ff2b4e6ef3a8b38000842815787c8124f2674e
Author: VeXEveryOne <il.albahtin@gmail.com>
Date: Tue Feb 18 20:14:54 2025 +0300
```

Первый КОММИТ

```
ilalb@VeX MINGW64 ~/Documents/testrepositorii (master)
$ git log --pretty=oneline
7406a17876f5feec7f2a1b689e7da77b24b1a773 (HEAD -> master) Added HTML header
7a6885e89324f89ac545a91a8c45560b44dff566 Added standart HTML page tags
335dcfa76ed8287d864ee6ae77f4e31ea1b0f844 Unrelated change to c
558bfc64b5be0904fd8083cdcb5b6a3b2ad1e1e8 Changes for a and b
d4ff2b4e6ef3a8b38000842815787c8124f2674e Первый КОММИТ
```



```

ilalb@veX MINGW64 ~/Documents/testrepositorii (master)
$ git log --pretty=oneline
7406a17876f5feec7f2a1b689e7da77b24b1a773 (HEAD -> master) Added HTML header
7a6885e89324f89ac545a91a8c45560b44dff566 Added standart HTML page tags
335dcfa76ed8287d864ee6ae77f4e31ea1b0f844 Unrelated change to c
558bfc64b5be0904fd8083cdcb5b6a3b2ad1e1e8 Changes for a and b
d4ff2b4e6ef3a8b38000842815787c8124f2674e Первый КОММИТ

ilalb@veX MINGW64 ~/Documents/testrepositorii (master)
$ git log --pretty=oneline --max-count=2
7406a17876f5feec7f2a1b689e7da77b24b1a773 (HEAD -> master) Added HTML header
7a6885e89324f89ac545a91a8c45560b44dff566 Added standart HTML page tags

ilalb@veX MINGW64 ~/Documents/testrepositorii (master)
$ git log --pretty=oneline --since='5 minutes ago'
7406a17876f5feec7f2a1b689e7da77b24b1a773 (HEAD -> master) Added HTML header
7a6885e89324f89ac545a91a8c45560b44dff566 Added standart HTML page tags

ilalb@veX MINGW64 ~/Documents/testrepositorii (master)
$ git log --pretty=oneline --until='5 minutes ago'
335dcfa76ed8287d864ee6ae77f4e31ea1b0f844 Unrelated change to c
558bfc64b5be0904fd8083cdcb5b6a3b2ad1e1e8 Changes for a and b
d4ff2b4e6ef3a8b38000842815787c8124f2674e Первый КОММИТ

ilalb@veX MINGW64 ~/Documents/testrepositorii (master)
$ git log --pretty=oneline --autor=VeXEveryOne
fatal: unrecognized argument: --autor=VeXEveryOne

ilalb@veX MINGW64 ~/Documents/testrepositorii (master)
$ git log --pretty=oneline --author=VeXEveryOne
7406a17876f5feec7f2a1b689e7da77b24b1a773 (HEAD -> master) Added HTML header
7a6885e89324f89ac545a91a8c45560b44dff566 Added standart HTML page tags
335dcfa76ed8287d864ee6ae77f4e31ea1b0f844 Unrelated change to c
558bfc64b5be0904fd8083cdcb5b6a3b2ad1e1e8 Changes for a and b
d4ff2b4e6ef3a8b38000842815787c8124f2674e Первый КОММИТ

ilalb@veX MINGW64 ~/Documents/testrepositorii (master)
$ git log --pretty=oneline --all
7406a17876f5feec7f2a1b689e7da77b24b1a773 (HEAD -> master) Added HTML header
7a6885e89324f89ac545a91a8c45560b44dff566 Added standart HTML page tags
335dcfa76ed8287d864ee6ae77f4e31ea1b0f844 Unrelated change to c
558bfc64b5be0904fd8083cdcb5b6a3b2ad1e1e8 Changes for a and b
d4ff2b4e6ef3a8b38000842815787c8124f2674e Первый КОММИТ

```

```
ila1b@VeX MINGW64 ~/Documents/testrepositorii (master)
$ git log --pretty=format:"%h %ad | %s [%an]" --graph --date=short
* 7406a17 2025-02-18 | Added HTML header (HEAD -> master) [VeXEveryOne]
* 7a6885e 2025-02-18 | Added standart HTML page tags [VeXEveryOne]
* 335dcfa 2025-02-18 | Unrelated change to c [VeXEveryOne]
* 558bfc6 2025-02-18 | Changes for a and b [VeXEveryOne]
* d4ff2b4 2025-02-18 | Первый коммит [VeXEveryOne]

ila1b@VeX MINGW64 ~/Documents/testrepositorii (master)
$ git log --pretty=format:"%h %ad | %s [%an]" --graph --date=short
* 7406a17 2025-02-18 | Added HTML header [VeXEveryOne]
* 7a6885e 2025-02-18 | Added standart HTML page tags [VeXEveryOne]
* 335dcfa 2025-02-18 | Unrelated change to c [VeXEveryOne]
* 558bfc6 2025-02-18 | Changes for a and b [VeXEveryOne]
* d4ff2b4 2025-02-18 | Первый коммит [VeXEveryOne]

ila1b@VeX MINGW64 ~/Documents/testrepositorii (master)
$ git log --pretty=format:"%h %ad | [%an]" --graph --date=short
* 7406a17 2025-02-18 | [VeXEveryOne]
* 7a6885e 2025-02-18 | [VeXEveryOne]
* 335dcfa 2025-02-18 | [VeXEveryOne]
* 558bfc6 2025-02-18 | [VeXEveryOne]
* d4ff2b4 2025-02-18 | [VeXEveryOne]
```

Рисунок 15-18 – Вывод истории коммитов с различными параметрами

## 9. Получение старых версий

```
Lyudmila@LAPTOP-GFGI5D5V MINGW64 ~/Desktop/practice1 (master)
$ git log --pretty=format:"%h %ad | %s%d [%an]" --graph --date=short
* b34a4b8 2025-02-18 | Added HTML header (HEAD -> master) [Ilya29]
* 57fed70 2025-02-18 | Added standard HTML page tags [Ilya29]
* e45f08c 2025-02-18 | First commit [Ilya29]

Lyudmila@LAPTOP-GFGI5D5V MINGW64 ~/Desktop/practice1 (master)
$ git checkout e45f08c
Note: switching to 'e45f08c'.

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 e45f08c First commit

Lyudmila@LAPTOP-GFGI5D5V MINGW64 ~/Desktop/practice1 ((e45f08c...))
$ cat project.html
Мы создали свое наполнение в данном файле!

Lyudmila@LAPTOP-GFGI5D5V MINGW64 ~/Desktop/practice1 ((e45f08c...))
$ git checkout master
Previous HEAD position was e45f08c First commit
Switched to branch 'master'

Lyudmila@LAPTOP-GFGI5D5V MINGW64 ~/Desktop/practice1 (master)
$ cat project.html
<html>

<head>
</head>
<body>
  <h1>Hello, World!</h1>
</body>
</html>
```

```
ilalb@VeX MINGW64 ~/Documents/testrepositorii (master)
$ git checkout 7406a17
Note: switching to '7406a17'.

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 7406a17 Added HTML header
```

```
ila1b@veX MINGW64 ~/Documents/testrepositorii ((7406a17...))
$ git checkout d4ff2b4
Previous HEAD position was 7406a17 Added HTML header
HEAD is now at d4ff2b4 Первый коммит

ila1b@veX MINGW64 ~/Documents/testrepositorii ((d4ff2b4...))
$ cat testsite.html
Тест редактора nano

ila1b@veX MINGW64 ~/Documents/testrepositorii ((d4ff2b4...))
$ git checkout master
Previous HEAD position was d4ff2b4 Первый коммит
Switched to branch 'master'

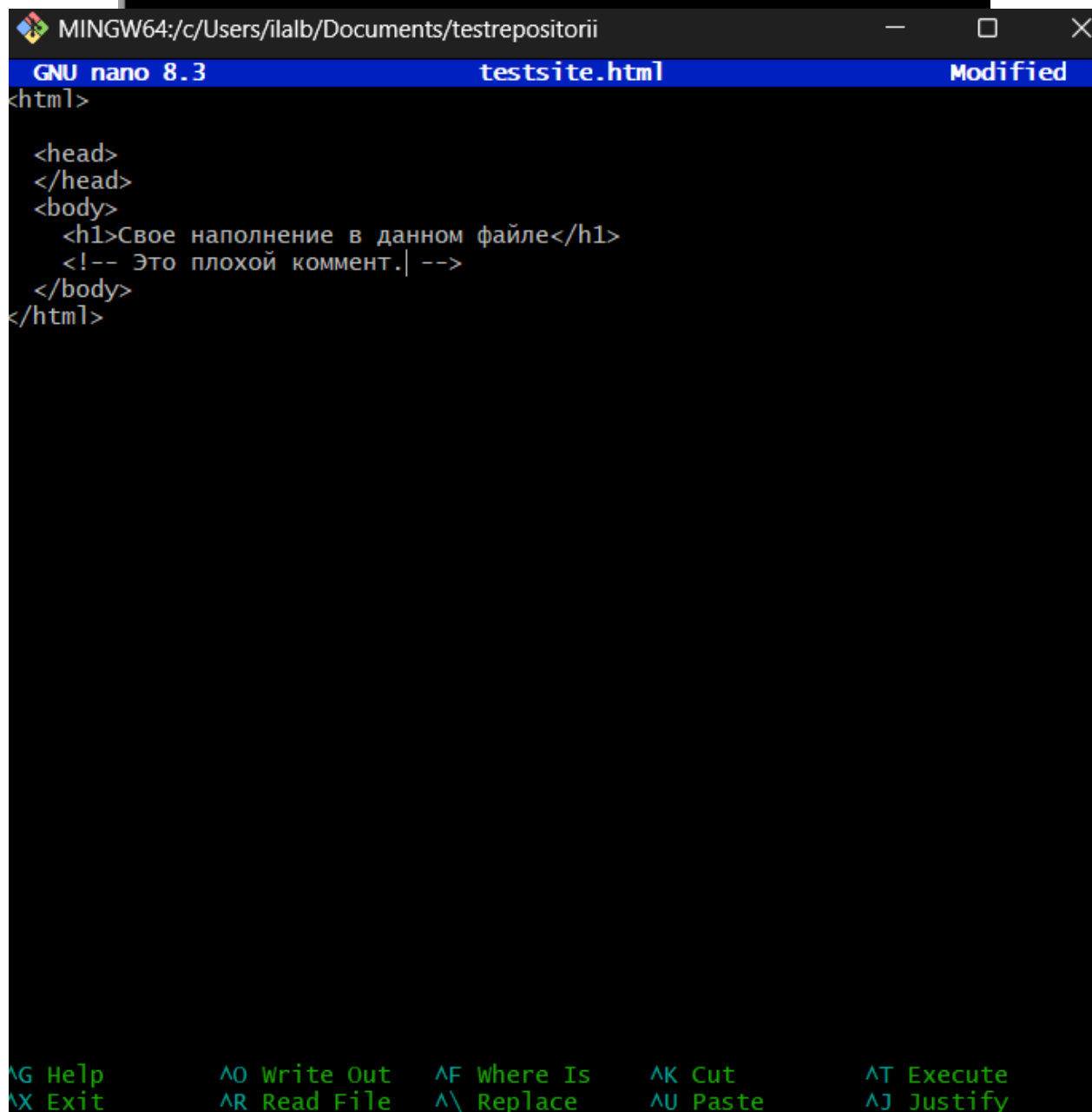
ila1b@veX MINGW64 ~/Documents/testrepositorii (master)
$ cat testsite.html
<html>

    <head>
    </head>
    <body>
        <h1>Hello, world!</h1>
    </body>
</html>
```

Рисунок 19-23 – получение хеша коммита с последующим переключением на него и чтением файла testsite.html. Затем переключение на ветку master и чтение файла из неё.

## 10. Отмена локальных изменений (до индексации)

```
ilalb@veX MINGW64 ~/Documents/testrepositorii (master)
$ git checkout master
Already on 'master'
```



```
GNU nano 8.3 testsite.html Modified
<html>

<head>
</head>
<body>
  <h1>Свое наполнение в данном файле</h1>
  <!-- Это плохой коммент.|-->
</body>
</html>
```

```
ilalb@veX MINGW64 ~/Documents/testrepositorii (master)
$ 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:   testsite.html

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

```
ilalb@veX MINGW64 ~/Documents/testrepositorii (master)
$ git checkout testsite.html
Updated 1 path from the index
```

```
ilalb@VeX MINGW64 ~/Documents/testrepositorii (master)
$ git status
On branch master
nothing to commit, working tree clean

ilalb@VeX MINGW64 ~/Documents/testrepositorii (master)
$ cat testsite.html
<html>

  <head>
</head>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>
```

Рисунок 24-28 – отмена локальных изменений до индексации

## 11. Отмена локальных изменений (после индексации и до коммита)

```
ila1b@veX MINGW64 ~/Documents/testrepositorii (master)
$ nano testsite.html

ila1b@veX MINGW64 ~/Documents/testrepositorii (master)
$ git add testsite.html

ila1b@veX MINGW64 ~/Documents/testrepositorii (master)
$ git commit -m "Удаляем этот коммит"
[master 577169b] Удаляем этот коммит
1 file changed, 1 insertion(+), 1 deletion(-)

ila1b@veX MINGW64 ~/Documents/testrepositorii (master)
$ git revert HEAD --no-edit
[master a5223eb] Revert "Удаляем этот коммит"
Date: Tue Feb 18 21:07:58 2025 +0300
1 file changed, 1 insertion(+), 1 deletion(-)
```

Рисунок 29-30 – Отмена локальных изменений после индексации и до коммита

## 12. Отмена коммита.

```
ila1b@veX MINGW64 ~/Documents/testrepositorii (master)
$ git log --pretty=format:"%h %ad | %s%d [%an]" --graph --date=short
* a5223eb 2025-02-18 | Revert "Удаляем этот коммит" (HEAD -> master) [VeXEveryOne]
* 577169b 2025-02-18 | Удаляем этот коммит [VeXEveryOne]
* f5924ed 2025-02-18 | Reapply "testsite.html" [VeXEveryOne]
* c2a9c9d 2025-02-18 | Revert "testsite.html" [VeXEveryOne]
* bd3444c 2025-02-18 | testsite.html [VeXEveryOne]
* 7406a17 2025-02-18 | Added HTML header [VeXEveryOne]
* 7a6885e 2025-02-18 | Added standart HTML page tags [VeXEveryOne]
* 335dcfa 2025-02-18 | Unrelated change to c [VeXEveryOne]
* 558bfc6 2025-02-18 | Changes for a and b [VeXEveryOne]
* d4ff2b4 2025-02-18 | Первый коммит [VeXEveryOne]
```

Рисунок 31 – Отмена коммита

## Часть 2. Системы управления репозиториями

Расчет варианта:  $V = \text{Ост}((3-1)/11)+1 = 3$

### 1. Создание аккаунта на GitHub

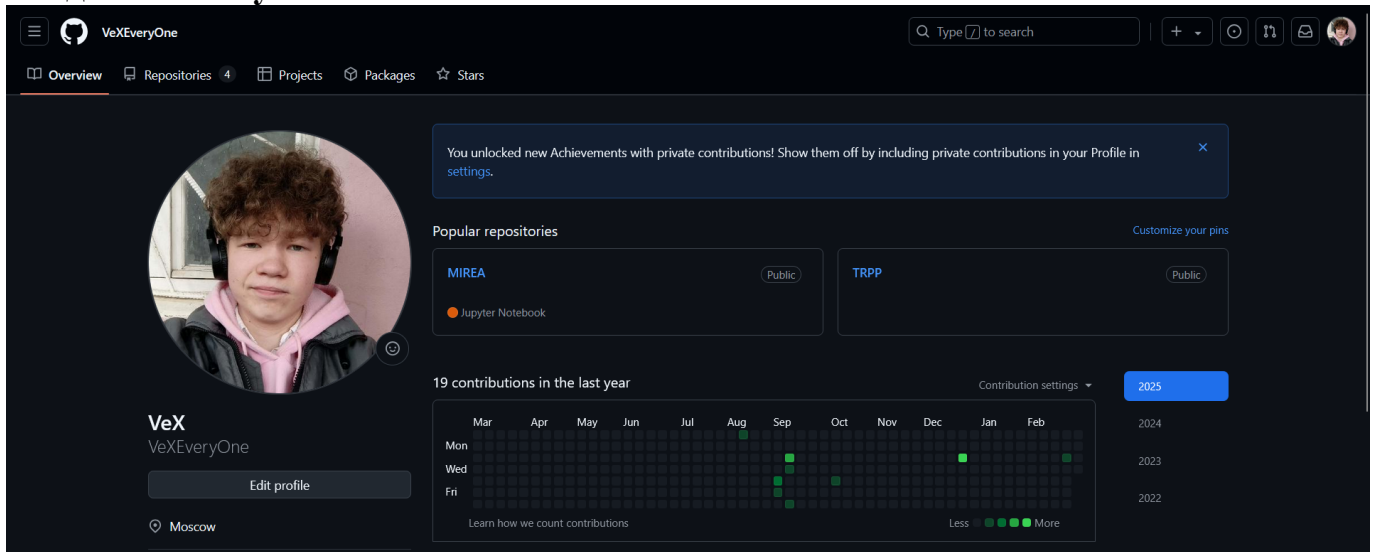


Рисунок 32 – Созданный профиль на GitHub

### 2. Создание SSH-ключа для авторизации

```
ilalb@veX MINGW64 /
$ ~/.ssh
bash: /c/Users/ilalb/.ssh: Is a directory

ilalb@veX MINGW64 /
$ ssh-keygen -t rsa -b 4096 -C "il.albahtin@gmail.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/ilalb/.ssh/id_rsa):
/c/Users/ilalb/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase for "/c/Users/ilalb/.ssh/id_rsa" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/ilalb/.ssh/id_rsa
Your public key has been saved in /c/Users/ilalb/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:40v7g82AKKrbXOLEHa/FKUOYBS8u4jgsE8Lf2MC7/98 il.albahtin@gmail.com
The key's randomart image is:
+---[RSA 4096]-----+
|  .                    |
|  . o .               |
| ...= . .            |
|=.++o.. S            |
|B+ooB+.o             |
|*+++=o*=             |
|oB o.*. + .          |
|+.+.o.++oo E         |
+---[SHA256]-----+

ilalb@veX MINGW64 /
$ ssh-agent
SSH_AUTH_SOCK=/tmp/ssh-OuLmBrA6VFc2/agent.511; export SSH_AUTH_SOCK;
SSH_AGENT_PID=512; export SSH_AGENT_PID;
echo Agent pid 512;
```

Рисунок 33 – генерация SSH-ключа на рабочей машине



```

ilalb@veX MINGW64 /
$ ssh-agent
SSH_AUTH_SOCK=/tmp/ssh-OuLmBrA6VFc2/agent.511; export SSH_AUTH_SOCK;
SSH_AGENT_PID=512; export SSH_AGENT_PID;
echo Agent pid 512;

```

Рисунок 34 – добавление ключа в ssh-agent

```

ilalb@veX MINGW64 /
$ eval "$(ssh-agent -s)"
Agent pid 517

ilalb@veX MINGW64 /
$ ssh-add ~/.ssh/id_rsa
Enter passphrase for /c/Users/ilalb/.ssh/id_rsa:
Identity added: /c/Users/ilalb/.ssh/id_rsa (il.albahtin@gmail.com)

```

Рисунок 35 – Проверка доступности ключа и добавление ключа в систему

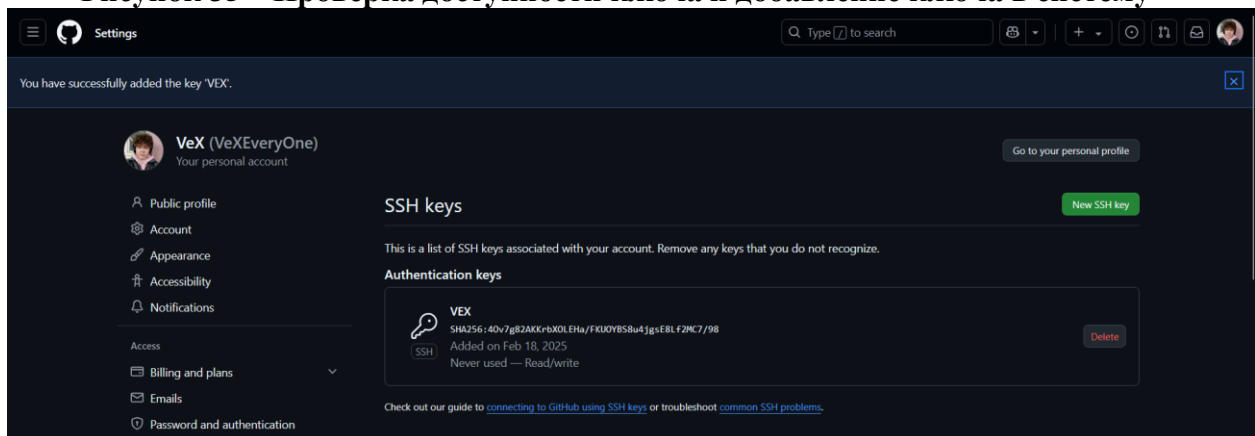


Рисунок 36 – Добавление ключа на Github

### 3. Создание нового репозитория для своего проекта

```

ilalb@veX MINGW64 /
$ cd Documents
bash: cd: Documents: No such file or directory

ilalb@veX MINGW64 /
$ cd /Documents/testrepositorii
bash: cd: /Documents/testrepositorii: No such file or directory

ilalb@veX MINGW64 /
$ cd ~/Documents/testrepositorii

ilalb@veX MINGW64 ~/Documents/testrepositorii (master)
$ mkdir task2

ilalb@veX MINGW64 ~/Documents/testrepositorii (master)
$ cd ~/Documents/testrepositorii/task1
bash: cd: /c/Users/ilalb/Documents/testrepositorii/task1: No such file or directory

ilalb@veX MINGW64 ~/Documents/testrepositorii (master)
$ cd task2

ilalb@veX MINGW64 ~/Documents/testrepositorii/task2 (master)
$ touch readme.txt

ilalb@veX MINGW64 ~/Documents/testrepositorii/task2 (master)
$ touch task1.cpp

ilalb@veX MINGW64 ~/Documents/testrepositorii/task2 (master)
$ touch task2.cpp

```

Рисунок 37 – создание репозитория с несколькими файлами

#### 4. Связываем локальный и удаленный репозитории

```

ilalb@veX MINGW64 ~/Documents/testrepositorii/task2 (master)
$ git init
Initialized empty Git repository in C:/Users/ilalb/Documents/testrepositorii/task2/.git/

ilalb@veX MINGW64 ~/Documents/testrepositorii/task2 (master)
$ git remote add project git@github.com:VeXEveryOne/TRPP.git

```

Рисунок 38 – связывание локального репозитория с удалённым

#### 5. Создание веток и переключение между ними

```

ilalb@veX MINGW64 ~/Documents/testrepositorii (dev)
$ git checkout -b main
Switched to a new branch 'main'

```

```

ilalb@VeX MINGW64 ~/Documents/testrepositorii (main)
$ nano a.html

ilalb@VeX MINGW64 ~/Documents/testrepositorii (main)
$ git add a.html
warning: in the working copy of 'a.html', LF will be replaced

ilalb@VeX MINGW64 ~/Documents/testrepositorii (main)
$ git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   a.html

ilalb@VeX MINGW64 ~/Documents/testrepositorii (main)
$ git commit -m "Change a.html"
[main 56dbc44] Change a.html
1 file changed, 2 insertions(+)

```

Рисунок 39-40 –Создание новой ветки и работа в ней

## 6. Слияние веток

```

ilalb@VeX MINGW64 ~/Documents/testrepositorii (main)
$ git checkout dev
Switched to branch 'dev'

ilalb@VeX MINGW64 ~/Documents/testrepositorii (dev)
$ git merge main
Updating a5223eb..56dbc44
Fast-forward
 a.html | 2 ++
1 file changed, 2 insertions(+)

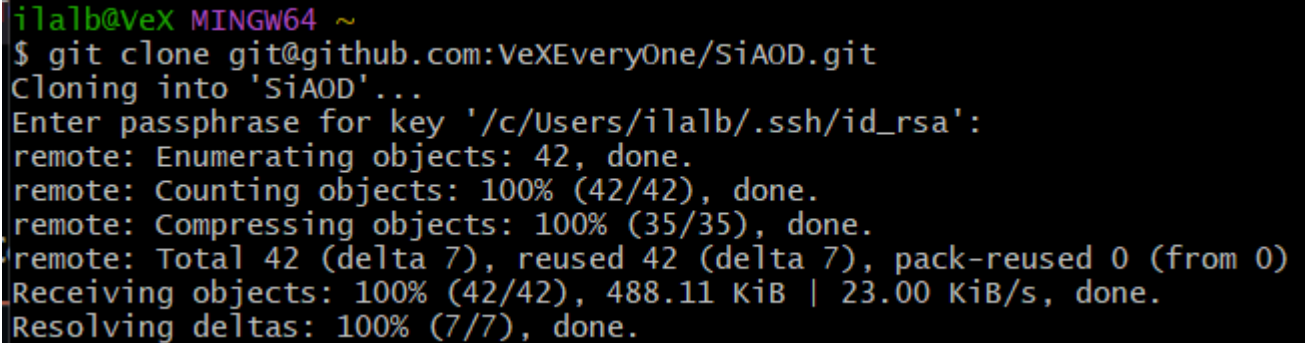
```

Рисунок 41 – слияние веток main и dev

7. Выполните цепочку действий в репозитории, согласно вариантам.

**Вариант 3.**

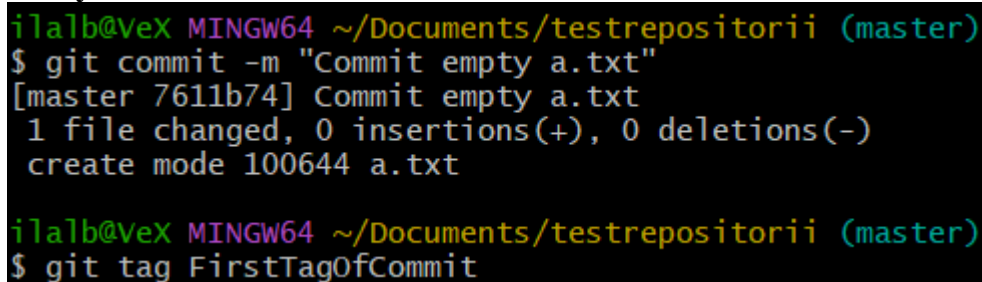
1) Клонировать непустой удаленный репозиторий на локальную машину



```
ilalb@VeX MINGW64 ~  
$ git clone git@github.com:VeXEveryOne/SiAOD.git  
Cloning into 'SiAOD'...  
Enter passphrase for key '/c/Users/ilalb/.ssh/id_rsa':  
remote: Enumerating objects: 42, done.  
remote: Counting objects: 100% (42/42), done.  
remote: Compressing objects: 100% (35/35), done.  
remote: Total 42 (delta 7), reused 42 (delta 7), pack-reused 0 (from 0)  
Receiving objects: 100% (42/42), 488.11 KiB | 23.00 KiB/s, done.  
Resolving deltas: 100% (7/7), done.
```

Рисунок 42 – клонирование непустого удаленного репозитория

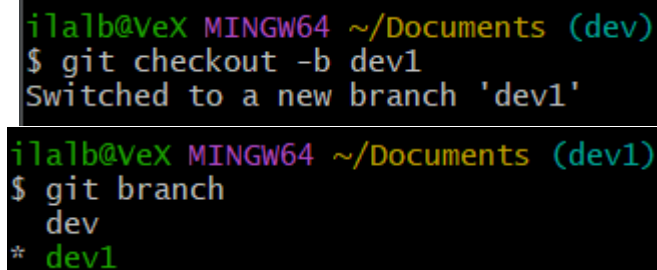
2) Создайте тег указывающий на последний коммит в ветке master



```
ilalb@VeX MINGW64 ~/Documents/testrepositorii (master)  
$ git commit -m "Commit empty a.txt"  
[master 7611b74] Commit empty a.txt  
1 file changed, 0 insertions(+), 0 deletions(-)  
create mode 100644 a.txt  
  
ilalb@VeX MINGW64 ~/Documents/testrepositorii (master)  
$ git tag FirstTagOfCommit
```

Рисунок 43 – Создание тега на последний коммит ветки master

3) Создайте новую ветку и выведите список всех веток



```
ilalb@VeX MINGW64 ~/Documents (dev)  
$ git checkout -b dev1  
Switched to a new branch 'dev1'  
  
ilalb@VeX MINGW64 ~/Documents (dev1)  
$ git branch  
dev  
* dev1
```

Рисунок 44-45 – создание новой ветки и вывод списка всех веток

4) Произведите 3 коммита в новой ветке в разные файлы

```

ilalb@VeX MINGW64 ~/Documents (dev1)
$ git add script.py

ilalb@VeX MINGW64 ~/Documents (dev1)
$ git commit -m "New Script"
[dev1 a6e017c] New Script
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 script.py

ilalb@VeX MINGW64 ~/Documents (dev1)
$ git add PythonProject.py

ilalb@VeX MINGW64 ~/Documents (dev1)
$ git commit -m "For script.py"
[dev1 65c5209] For script.py
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 PythonProject.py

ilalb@VeX MINGW64 ~/Documents (dev1)
$ git add PythonMap.py

ilalb@VeX MINGW64 ~/Documents (dev1)
$ git commit -m "Map for script.py"
[dev1 b28370f] Map for script.py
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 PythonMap.py

```

Рисунок 46 – 3 коммита в новой ветке в разные файлы

## 5) Выгрузите изменения в удаленный репозиторий

```

ilalb@VeX MINGW64 ~/Documents (dev1)
$ git push git@github.com:VeXEveryone/TRPP.git dev1
Enter passphrase for key '/c/Users/ilalb/.ssh/id_rsa':
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 12 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (9/9), 858 bytes | 286.00 KiB/s, done.
Total 9 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), done.
To github.com:VeXEveryone/TRPP.git
 * [new branch]      dev1 -> dev1

```

Рисунок 47 – выгрузка изменений в удалённый репозиторий

6) Откатите ветку к созданному тегу (в том числе в удаленном репозитории)

```
ila1b@VeX MINGW64 ~/Documents/testrepositorii (master)
$ git tag
FirstTagOfCommit

ila1b@VeX MINGW64 ~/Documents/testrepositorii (master)
$ git reset --hard FirstTagOfCommit
HEAD is now at 7611b74 Commit empty a.txt
```

Рисунок 48 – Откат к тегу

7) Выведите в консоли различия между веткой master и новой веткой

```
ila1b@VeX MINGW64 ~/Documents/testrepositorii (master)
$ git diff master..dev
diff --git a/a.html b/a.html
index e69de29..955c544 100644
--- a/a.html
+++ b/a.html
@@ -0,0 +1,2 @@
+<html>
+</html>
diff --git a/a.txt b/a.txt
deleted file mode 100644
index e69de29..0000000
```

Рисунок 49 – вывод различий между ветками master и dev

## Часть 3. Работа с ветвлением и оформление кода

Расчет варианта:  $V = \text{Ост}((3-1)/11)+1 = 3$

### 1. Форк репозитория

3	<a href="https://github.com/gto76/python-cheatsheet">https://github.com/gto76/python-cheatsheet</a>
---	---

Рисунок 50 – персональный вариант

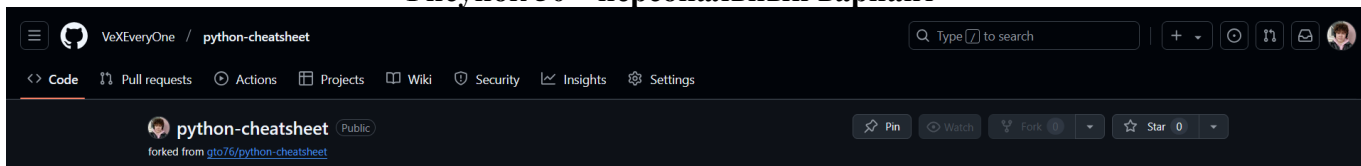


Рисунок 51 – сделанный форк репозитория

### 2. Клонирование репозитория на локальную машину

```
ilalb@VeX MINGW64 ~/Documents (dev1)
$ cd ~/Documents/Fork

ilalb@VeX MINGW64 ~/Documents/Fork (dev1)
$ git clone git@github.com:VeXEveryOne/python-cheatsheet.git
Cloning into 'python-cheatsheet'...
Enter passphrase for key '/c/Users/ilalb/.ssh/id_rsa':
remote: Enumerating objects: 10225, done.
remote: Counting objects: 100% (118/118), done.
remote: Compressing objects: 100% (62/62), done.
remote: Total 10225 (delta 67), reused 58 (delta 56), pack-reused 10107 (from 2)
Receiving objects: 100% (10225/10225), 12.11 MiB | 2.44 MiB/s, done.
Resolving deltas: 100% (6077/6077), done.
```

Рисунок 52 – клонирование репозитория на локальную машину

### 3. Создание двух веток в deprecation-note

```
ilalb@VeX MINGW64 ~/Documents/Fork (dev1)
$ git checkout -b branch1
Switched to a new branch 'branch1'

ilalb@VeX MINGW64 ~/Documents/Fork (branch1)
$ git checkout deprecation-note
error: pathspec 'deprecation-note' did not match any file(s) known to git

ilalb@VeX MINGW64 ~/Documents/Fork (branch1)
$ git checkout -b branch2
Switched to a new branch 'branch2'
```

Рисунок 53 – создание двух веток branch1 и branch2

#### 4. 3 коммита в каждую из веток, которые меняют один и тот же кусочек файла

```
ila1b@VeX MINGW64 ~/Documents/Fork (branch1)
$ touch README.md; nano README.md

ila1b@VeX MINGW64 ~/Documents/Fork (branch1)
$ nano README.md

ila1b@VeX MINGW64 ~/Documents/Fork (branch1)
$ git add README.md; git commit -m "Edited README.md"
warning: in the working copy of 'Fork/README.md', LF will be replaced by CRLF the next time Git touches it
[branch1 d98a3bb] Edited README.md
1 file changed, 1 insertion(+)
create mode 100644 Fork/README.md

ila1b@VeX MINGW64 ~/Documents/Fork (branch1)
$ touch commitlint.config; nano README.md

ila1b@VeX MINGW64 ~/Documents/Fork (branch1)
$ touch commitlint.config; nano commitlint.config

ila1b@VeX MINGW64 ~/Documents/Fork (branch1)
$ git add commitlint.config; git commit -m "Edited commitlint.config"
warning: in the working copy of 'Fork/commitlint.config', LF will be replaced by CRLF the next time Git touches it
[branch1 8aa13f6] Edited commitlint.config
1 file changed, 1 insertion(+)
create mode 100644 Fork/commitlint.config

ila1b@VeX MINGW64 ~/Documents/Fork (branch1)
$ touch package.json; nano package.json

ila1b@VeX MINGW64 ~/Documents/Fork (branch1)
$ git add package.json; git commit -m "Edited package.json"
warning: in the working copy of 'Fork/package.json', LF will be replaced by CRLF the next time Git touches it
[branch1 c323e02] Edited package.json
1 file changed, 2 insertions(+)
create mode 100644 Fork/package.json
```



```
ilalb@VeX MINGW64 ~/Documents/Fork (branch1)
$ git checkout branch2
M       testrepositorii
Switched to branch 'branch2'

ilalb@VeX MINGW64 ~/Documents/Fork (branch2)
$ nano README.md; git add README.md; git commit -m "Edited README.md"
warning: in the working copy of 'Fork/README.md', LF will be replaced by CRLF th
e next time Git touches it
[branch2 dcbb94f] Edited README.md
 1 file changed, 2 insertions(+)
 create mode 100644 Fork/README.md

ilalb@VeX MINGW64 ~/Documents/Fork (branch2)
$ nano commitlint.config; git add commitlint.config; git commit -m "Edited commi
tlint.config"
warning: in the working copy of 'Fork/commitlint.config', LF will be replaced by
CRLF the next time Git touches it
[branch2 9acf4df] Edited commitlint.config
 1 file changed, 1 insertion(+)
 create mode 100644 Fork/commitlint.config

ilalb@VeX MINGW64 ~/Documents/Fork (branch2)
$ nano package.json; git add package.json; git commit -m "Edited package.json"
warning: in the working copy of 'Fork/package.json', LF will be replaced by CRLF
the next time Git touches it
[branch2 4ccb8c1] Edited package.json
 1 file changed, 1 insertion(+)
 create mode 100644 Fork/package.json
```

Рисунок 54-55 – создание 3-х коммитов в каждую из веток

## 5. Слияние ветки branch1 в ветку branch2 с разрешением конфликтов

```
ila1b@VeX MINGW64 ~/Documents/Fork (branch2)
$ git checkout deprecation-note
error: pathspec 'deprecation-note' did not match any file(s) known to git
```

```
ila1b@VeX MINGW64 ~/Documents/Fork (branch2)
$ git checkout -b deprecation-note
Switched to a new branch 'deprecation-note'
```

```
ila1b@VeX MINGW64 ~/Documents/Fork (deprecation-note)
$ git merge branch1
Auto-merging Fork/README.md
CONFLICT (add/add): Merge conflict in Fork/README.md
Auto-merging Fork/commitlint.config
CONFLICT (add/add): Merge conflict in Fork/commitlint.config
Auto-merging Fork/package.json
CONFLICT (add/add): Merge conflict in Fork/package.json
Automatic merge failed; fix conflicts and then commit the result.
```

```
ila1b@VeX MINGW64 ~/Documents/Fork (deprecation-note)
$ git merge branch2
Already up to date.
```

```
ila1b@VeX MINGW64 ~/Documents/Fork (deprecation-note)
$ git add -A; git commit -m "added theirs"
warning: could not open directory 'Мои видеозаписи/': Permission denied
warning: could not open directory 'мои рисунки/': Permission denied
warning: could not open directory 'Моя музыка/': Permission denied
warning: adding embedded git repository: Fork/python-cheatsheet
hint: You've added another git repository inside your current repository.
hint: Clones of the outer repository will not contain the contents of
hint: the embedded repository and will not know how to obtain it.
hint: If you meant to add a submodule, use:
hint:
hint:   git submodule add <url> Fork/python-cheatsheet
hint:
hint: If you added this path by mistake, you can remove it from the
hint: index with:
hint:
hint:   git rm --cached Fork/python-cheatsheet
hint:
hint: See "git help submodule" for more information.
hint: Disable this message with "git config set advice.addEmbeddedRepo false"
warning: adding embedded git repository: python-cheatsheet
[deprecation-note c4288ac] added theirs
3 files changed, 3 insertions(+), 1 deletion(-)
create mode 160000 Fork/python-cheatsheet
create mode 160000 python-cheatsheet

ila1b@VeX MINGW64 ~/Documents/Fork (deprecation-note)
$ git commit -m "added theirs"
warning: could not open directory 'Мои видеозаписи/': Permission denied
warning: could not open directory 'мои рисунки/': Permission denied
warning: could not open directory 'Моя музыка/': Permission denied
On branch deprecation-note
nothing to commit, working tree clean
```

**Рисунок 56-58 – слияние веток branch1 и branch2 с разрешением конфликтов**

## 6. Выгрузка всех изменений во всех ветках в удаленный репозиторий

```
ilalb@VeX MINGW64 ~/Documents/Fork (deprecation-note)
$ git push --all project
Enter passphrase for key '/c/Users/ilalb/.ssh/id_rsa':
Enumerating objects: 34, done.
Counting objects: 100% (34/34), done.
Delta compression using up to 12 threads
Compressing objects: 100% (25/25), done.
Writing objects: 100% (33/33), 3.10 KiB | 352.00 KiB/s, done.
Total 33 (delta 12), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (12/12), completed with 1 local object.
To github.com:VeXEveryOne/TRPP.git
* [new branch]      branch1 -> branch1
* [new branch]      branch2 -> branch2
* [new branch]      deprecation-note -> deprecation-note
* [new branch]      dev -> dev

ilalb@VeX MINGW64 ~/Documents/Fork (deprecation-note)
$ git checkout branch1
warning: unable to rmdir 'Fork/python-cheatsheet': Directory not empty
warning: unable to rmdir 'python-cheatsheet': Directory not empty
M       testrepositorii
Switched to branch 'branch1'

ilalb@VeX MINGW64 ~/Documents/Fork (branch1)
$ git push project branch1
Enter passphrase for key '/c/Users/ilalb/.ssh/id_rsa':
Everything up-to-date

ilalb@VeX MINGW64 ~/Documents/Fork (branch1)
$ git push project branch2
Enter passphrase for key '/c/Users/ilalb/.ssh/id_rsa':
Everything up-to-date

ilalb@VeX MINGW64 ~/Documents/Fork (branch1)
$ git checkout branch2
M       testrepositorii
Switched to branch 'branch2'

ilalb@VeX MINGW64 ~/Documents/Fork (branch2)
$ git push project branch2
Enter passphrase for key '/c/Users/ilalb/.ssh/id_rsa':
Everything up-to-date
```

Рисунок 59 – выгрузка изменений во всех ветках в удаленный репозиторий

## 7. Выполнение 3-х коммитов в branch1

```
ilalb@VeX MINGW64 ~/Documents/Fork (branch1)
$ touch yarn.lock; nano yarn.lock; git add yarn.lock; git commit -m "Edited yarn.lock"
warning: in the working copy of 'Fork/yarn.lock', LF will be replaced by CRLF the next time Git touches it
[branch1 0ed740d] Edited yarn.lock
1 file changed, 1 insertion(+)
create mode 100644 Fork/yarn.lock

ilalb@VeX MINGW64 ~/Documents/Fork (branch1)
$ touch LICENSES; nano LICENSES; git add LICENSES; git commit -m "Edited LICENSES"
warning: in the working copy of 'Fork/LICENSES', LF will be replaced by CRLF the next time Git touches it
[branch1 044d62a] Edited LICENSES
1 file changed, 1 insertion(+)
create mode 100644 Fork/LICENSES

ilalb@VeX MINGW64 ~/Documents/Fork (branch1)
$ touch .gitignor; nano .gitignor; git add .gitignor; git commit -m "Edited .gitignor"
warning: in the working copy of 'Fork/.gitignor', LF will be replaced by CRLF the next time Git touches it
[branch1 4d1fc13] Edited .gitignor
1 file changed, 1 insertion(+)
create mode 100644 Fork/.gitignor
```

Рисунок 60 – Выполнение 3-х коммитов в branch1

## 8. Повторное клонирование репозитория в другую директорию

```
ilalb@VeX MINGW64 ~/Documents/Fork (branch1)
$ cd ~/Documents/123

ilalb@VeX MINGW64 ~/Documents/123 (branch1)
$ git clone git@github.com:VeXEveryOne/TRPP.git
Cloning into 'TRPP'...
Enter passphrase for key '/c/Users/ilalb/.ssh/id_rsa':
remote: Enumerating objects: 70, done.
remote: Counting objects: 100% (70/70), done.
remote: Compressing objects: 100% (28/28), done.
remote: Total 70 (delta 30), reused 70 (delta 30), pack-reused 0 (from 0)
Receiving objects: 100% (70/70), 6.08 KiB | 566.00 KiB/s, done.
Resolving deltas: 100% (30/30), done.
```

Рисунок 61 – повторное клонирование репозитория в другую директорию

## 9. Выполнение 3-х коммитов в ветку branch1

```
ilalb@VeX MINGW64 ~/Documents/Fork (branch1)
$ nano README.md; git add README.md; git commit -m "Edited README.md"
[branch1 3d5a884] Edited README.md
1 file changed, 1 insertion(+), 1 deletion(-)

ilalb@VeX MINGW64 ~/Documents/Fork (branch1)
$ nano commitlint.config; git add commitlint.config; git commit -m "Edited commitlint.config"
[branch1 f86496d] Edited commitlint.config
1 file changed, 1 insertion(+), 1 deletion(-)

ilalb@VeX MINGW64 ~/Documents/Fork (branch1)
$ nano package.json; git add package.json; git commit -m "Edited package.json"
[branch1 3782551] Edited package.json
1 file changed, 1 insertion(+), 1 deletion(-)
```

Рисунок 62 – три коммита в ветке branch1

## 10. Выгрузка всех изменений в удаленный репозиторий

```
ilalb@VeX MINGW64 ~/Documents/Fork (branch1)
$ git push project branch1
Enter passphrase for key '/c/Users/ilalb/.ssh/id_rsa':
Enumerating objects: 33, done.
Counting objects: 100% (33/33), done.
Delta compression using up to 12 threads
Compressing objects: 100% (25/25), done.
Writing objects: 100% (28/28), 2.48 KiB | 317.00 KiB/s, done.
Total 28 (delta 15), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (15/15), completed with 1 local object.
To github.com:VeXEveryOne/TRPP.git
c323e02..3782551 branch1 -> branch1
```

Рисунок 63 – выгрузка всех изменений из локального репозитория в удаленный

## 11. Выгрузка изменения из старого репозитория с опцией –force

```
ilalb@VeX MINGW64 ~/Documents/123 (branch1)
$ cd ~/Documents/Fork

ilalb@VeX MINGW64 ~/Documents/Fork (branch1)
$ git push project branch1 --force
Enter passphrase for key '/c/Users/ilalb/.ssh/id_rsa':
Everything up-to-date
```

Рисунок 64 – вернулись в старый клон с репозиторием и выгрузили изменения с опцией – force

## 12. Получение всех изменения в новом репозитории

```
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (1/1), done.
remote: Total 8 (delta 1), reused 1 (delta 1), pack-reused 7 (from 1)
Unpacking objects: 100% (8/8), 766 bytes | 30.00 KiB/s, done.
From github.com:ilyunchio/react-native-camera
+ 18fb2aa...350289e branch1 -> origin/branch1 (forced update)
Merge made by the 'ort' strategy.
.gitignore | 1 +
LICENSE    | 1 +
yarn.lock  | 1 +
3 files changed, 3 insertions(+)
create mode 100644 .gitignore
create mode 100644 LICENSE
create mode 100644 yarn.lock
```

Рисунок 65 – перешли в новый репозиторий и получили в нём все изменения

## Ответы на 8 любых вопросов:

### 9. Как отменить слияние веток, если произошел конфликт?

Отменить слияние веток при появлении конфликта можно командой **git merge -abort**.

### 10. Для чего нужен .gitignore?

**.gitignore** используется для указания файлов и директорий, которые не должны учитываться системой контроля версий Git. Это позволяет игнорировать временные файлы, конфигурационные файлы, и прочее, чтобы они не попадали в репозиторий.

### 11. Что делает команда git status?

Команда **git status** показывает текущий статус репозитория, включая измененные, добавленные и неотслеживаемые файлы.

### 12. Что делает команда git add?

Команда **git add** добавляет изменения файлов в индекс, подготавливая их к следующему коммиту.

### 13. Что делает команда git log?

Команда **git log** показывает историю коммитов. Она перечисляет коммиты, которые можно достичь, следуя связям родительских коммитов от указанных коммитов.

### 14. Что делает команда git diff?

Команда **git diff** показывает различия между рабочим деревом и индексом, а также между индексом и последним коммитом.

### 15. Что делает команда git show?

Команда **git show** отображает информацию о конкретном коммите, включая изменения, внесенные этим коммитом.

### 16. Что делает команда git stash?

Команда **git stash** временно сохраняет изменения в рабочей директории, чтобы можно было переключиться на другую ветку или выполнить другие операции.



## **Вывод**

В ходе работы были изучены основные команды Git и принципы работы с GitHub. Выполнены задачи по настройке Git, созданию и управлению локальным и удаленным репозиториями, работе с ветками, слиянию и разрешению конфликтов. Также рассмотрены команды для отмены изменений, просмотра истории коммитов и управления изменениями. Полученные навыки позволяют эффективно управлять версиями проектов и работать в команде.