Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

«Основы ветвления Git»

ОТЧЕТ
по лабораторной работе №3
дисциплины
«Основы программной инженерия»

Выполнил:
Зиёдуллаев Жавохир Эркин угли
2 курс, группа ПИЖ-б-о-21-1,
09.03.04 «Программная инженерия»,
направленность (профиль) «Разработка
и сопровождение программного
обеспечения», очная форма обучения

_____
(подпись)

Проверил:
_____
(подпись)

Отчет защищен с оценкой _____ Дата защиты_____

Ставрополь, 2022 г.

Лабораторная работа 1.3 Основы ветвления GitHub

Цель работы: исследование базовых возможностей по работе с локальными и удаленными ветками Git

```
C:\Users\work\laba-head>color a

C:\Users\work\laba-head>git add 1.txt

C:\Users\work\laba-head>git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   1.txt
```

Рисунок 1 – индексация и коммит первого файла

```
C:\Users\work\laba-head>git add 2.txt 3.txt

C:\Users\work\laba-head>git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   2.txt
        new file:   3.txt
```

Рисунок 2 – индексация первого коммита

```
C:\Users\work\laba-head>git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   2.txt
        new file:   3.txt


C:\Users\work\laba-head>git commit -m "add 2.txt 3.txt"
[main ba81388] add 2.txt 3.txt
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 2.txt
 create mode 100644 3.txt

C:\Users\work\laba-head>git status
On branch main
Your branch is ahead of 'origin/main' by 2 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

Рисунок 3 – перезапись первого коммита

```
C:\Users\work\laba-head\.git>git branch develop

C:\Users\work\laba-head\.git>git log --oneline --decorate
f579c8a (HEAD, origin/main, origin/HEAD, develop) added file git.txt
5b51354 change README.md
2c5c2cb Initial commit

C:\Users\work\laba-head\.git>git checkout develop
fatal: this operation must be run in a work tree
```

Рисунок 4 – создание и переход на новой ветке

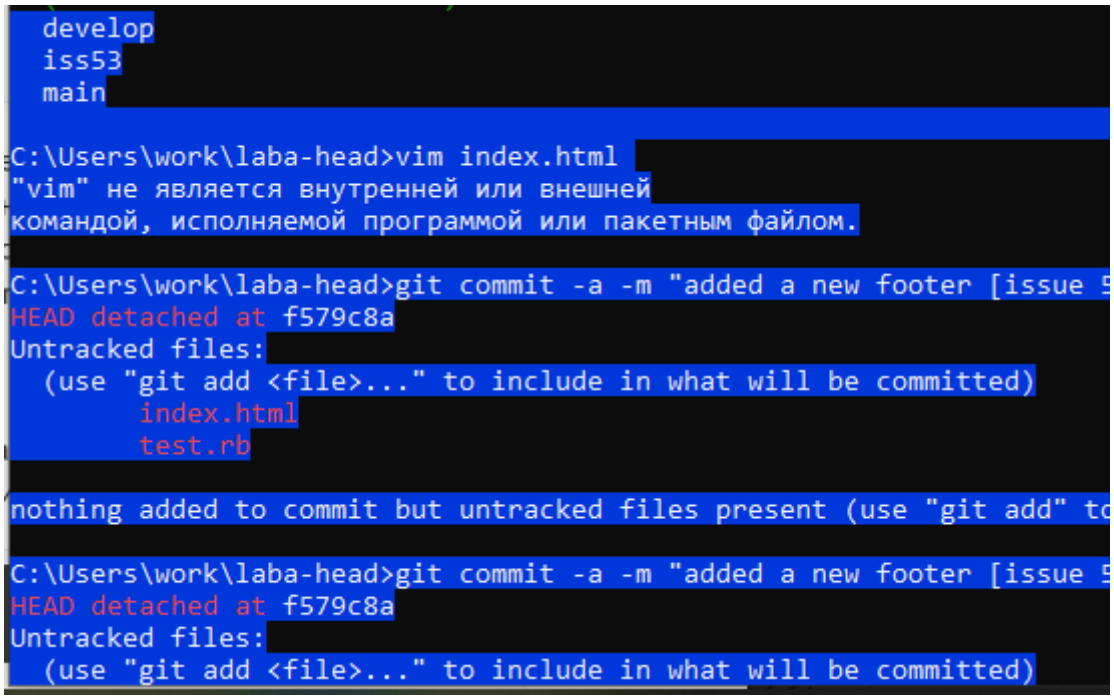

Рисунок 5 – создание новой ветке ISS53

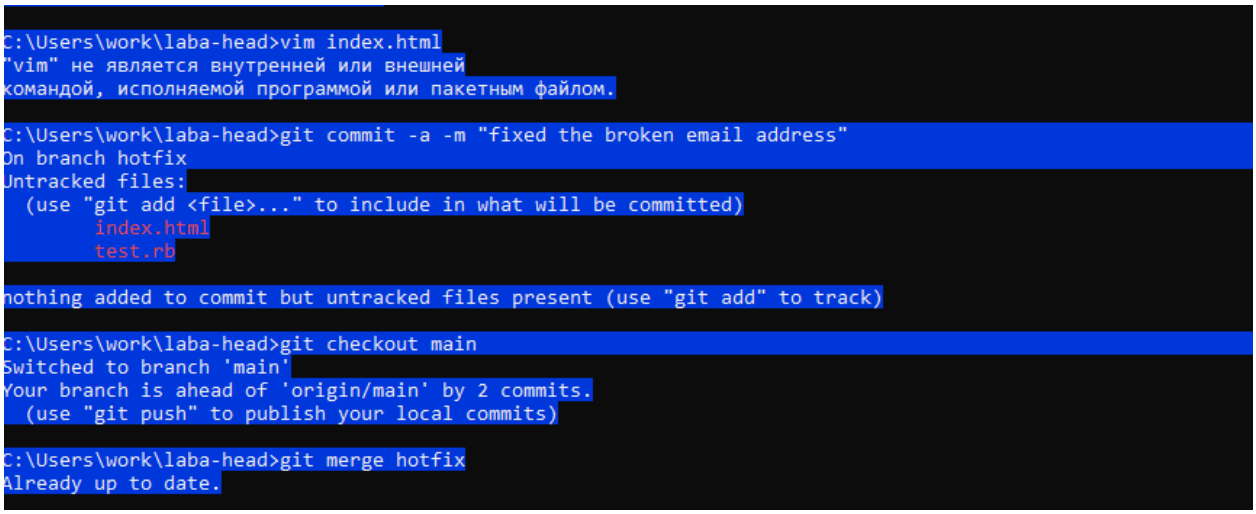

Рисунок 6 – слияние и удаление веток

Рисунок 7 – утилита Meld для решения конфликтов файла

```
C:\Users\work\laba-head>git push
Enumerating objects: 22, done.
Counting objects: 100% (22/22), done.
Delta compression using up to 4 threads
Compressing objects: 100% (16/16), done.
Writing objects: 100% (20/20), 2.07 KiB | 423.00 KiB/s, done.
Total 20 (delta 8), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (8/8), done.
To https://github.com/javoxir21/laba-head.git
   f579c8a..9d7b99a  main -> main

C:\Users\work\laba-head>branch
"branch" не является внутренней или внешней
командой, исполняемой программой или пакетным файлом.

C:\Users\work\laba-head>git branch
  iss53
* main
  testing

C:\Users\work\laba-head>git merge testing
Already up to date.

C:\Users\work\laba-head>git merge testing
```

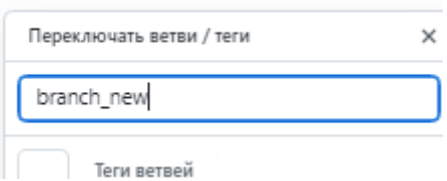

Рисунок 9 – Создание ветки с помощью GitHub

```
C:\Users\work\laba-head>git pull
From https://github.com/javoxir21/laba-head
 * [new branch]      branch_new -> origin/branch_new
Already up to date.

C:\Users\work\laba-head>
```

Рисунок 10 – Переход на ветку branch_new

Ответы на вопросы:

1. Ветка – это простой перемещаемый указатель на коммит

2. HEAD – это указатель, задача которого ссылаться на определенный коммит

3. Создать ветку можно через консоль командой git branch , либо через интерфейс GitHub

4. Чтобы узнать текущую ветку, необходимо ввести git status

5. Для переключения между ветками необходимо ввести команду git checkout

6. Удалённые ветки – это ветки в удалённых репозиториях

7. Ветки слежения – это локальные ветки, напрямую связанные с удаленными

8. Для создания ветки отслеживания нужно ввести git checkout –track origin/name, или ввести сокращение git checkout name

9. Для отправки изменений из локальной ветки нужно ввести git push

10. Git pull берет все новые коммиты и сливает их в одну ветку. Git fetch же только берет изменения с сервера и сохраняет их в локальном репозитории

11. Для удаления удаленной ветки нужно ввести git push origin –delete , для удаления локальной ветки нужно ввести git branch -d

12. Основные типы веток git-flow: develop, release, feature, hotfix. Работа с ветками организована следующим образом: из ветки main формируется ветка develop, из неё создается ветка release и feature. Когда работа над feature завершается, она сливается в ветку develop. Когда работа над веткой release завершается, она сливается с ветками develop и main. Если в main обнаруживается проблема, из неё создается hotfix. Когда работа с hotfix завершается, она сливается с ветками develop и main. Недостатки git-flow: сложно делать часто релизы, большие функции могут потратить много времени на мёрж, история имеет кучу мёржей и затрудняет просмотр истории проектов

13. В GitKraken есть инструменты для слияния веток, перебазирования, отслеживани