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 №1.3**

**дисциплины**

**«Основы кроссплатформенного программирования»**

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Выполнил:  Пинаева Диана Олеговна  1 курс, группа ИТС-б-о-22-1,  11.03.02 «Инфокоммуникационные технологии и системы связи», направленность (профиль) «Инфокоммуникационные системы и сети», заочная форма обучения  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | |
|  | | Проверил:  Воронкин Р. А., доцент кафедры инфокоммуникаций  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | |
|  | |  | |

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

Ставрополь, 2023 г.

**Тема:** Основы ветвления Git.

**Цель:** Исследование базовых возможностей по работе с локальными и удаленными ветками Git.

**Ход работы:**

1. **HEAD**

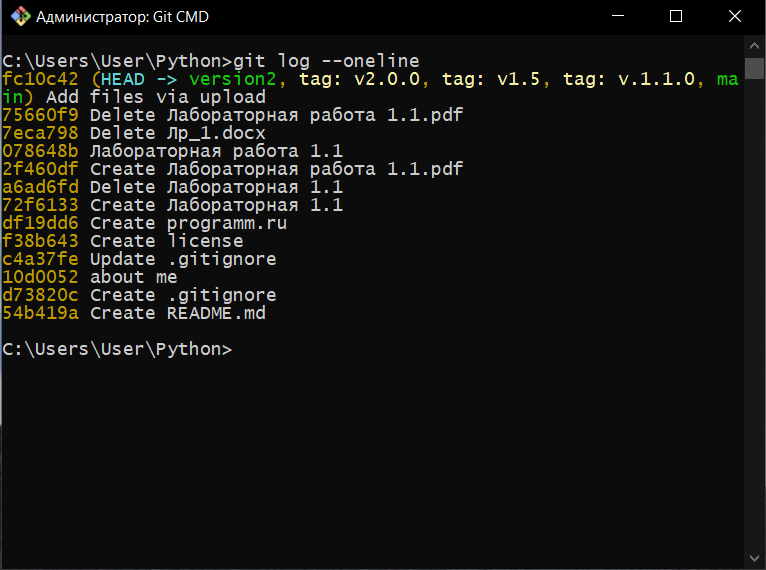


Рисунок 1

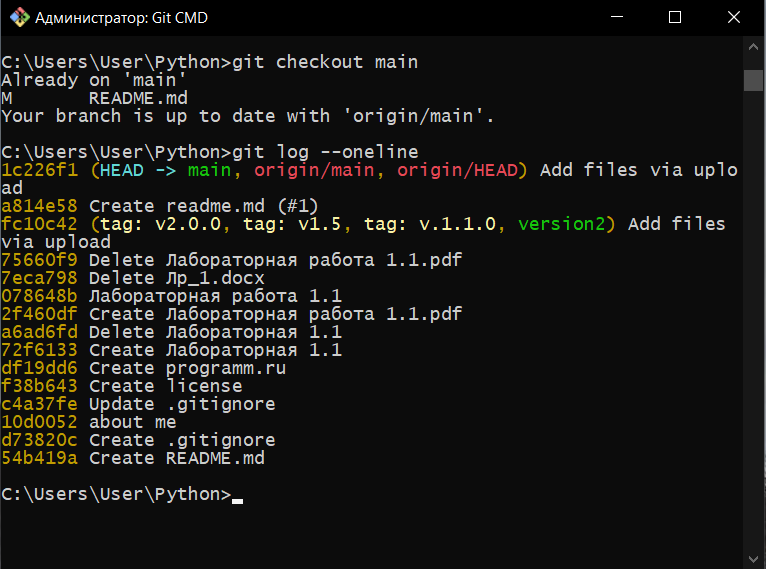


Рисунок 2

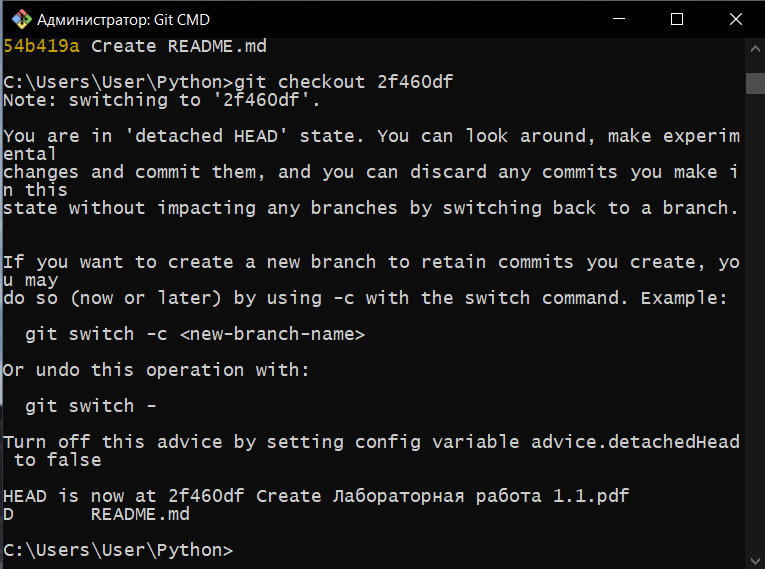


Рисунок 2

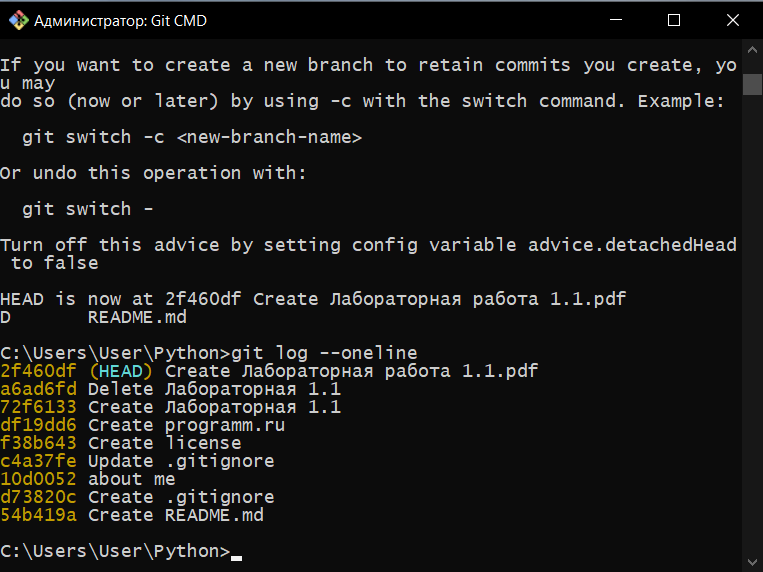


Рисунок 3

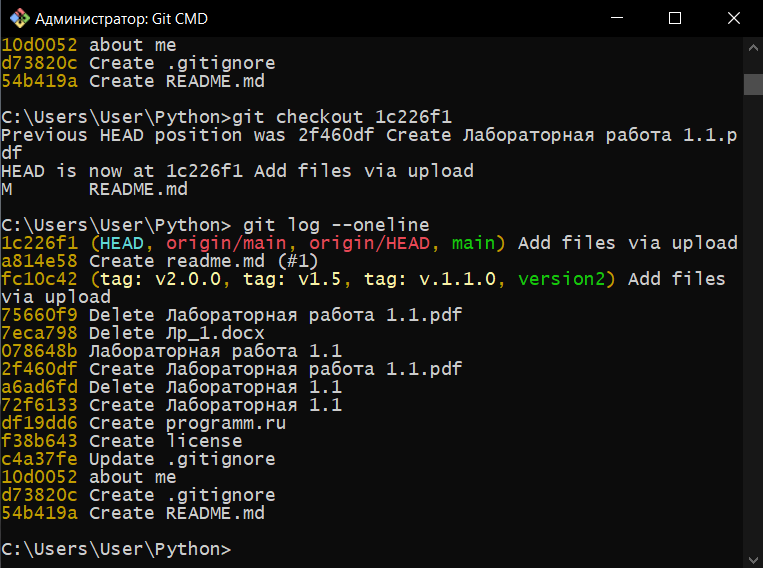


Рисунок 4

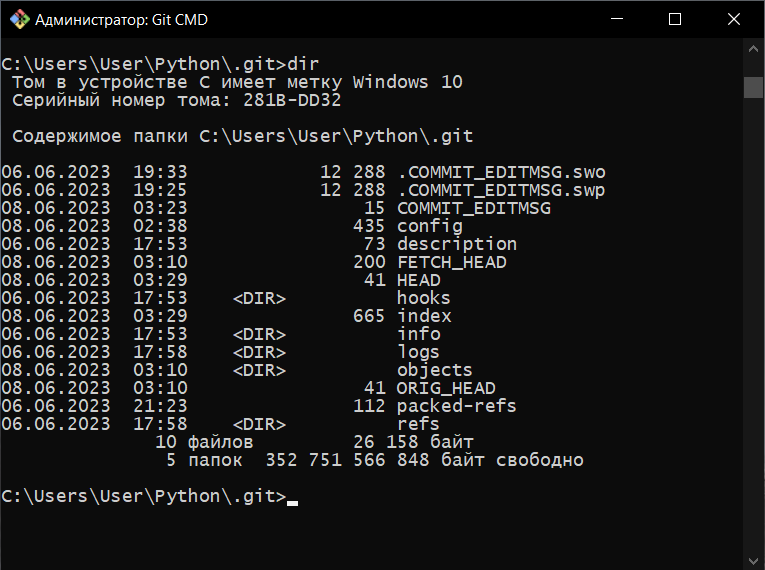


Рисунок 5

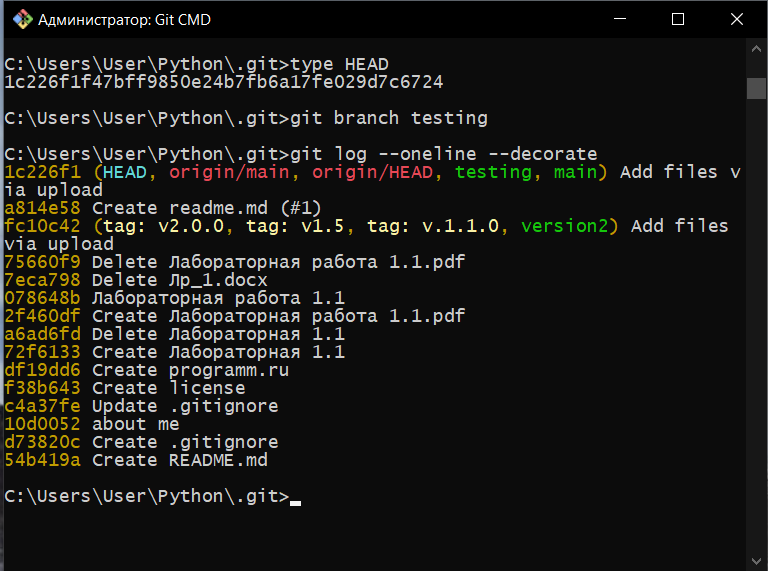


Рисунок 6

1. **Создание новой ветки.**

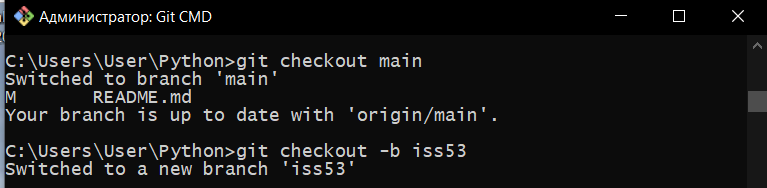


Рисунок 7

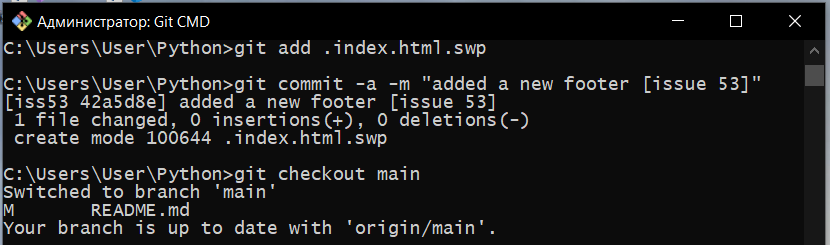


Рисунок 8

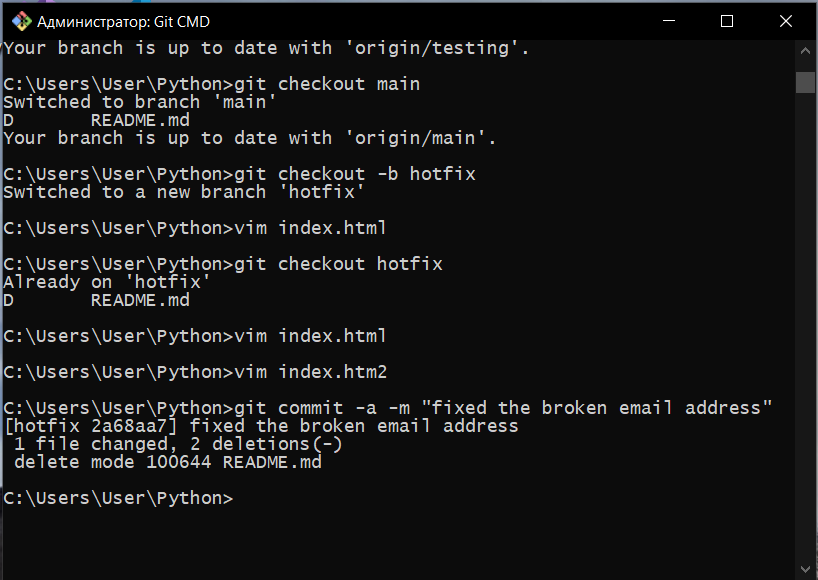


Рисунок 9

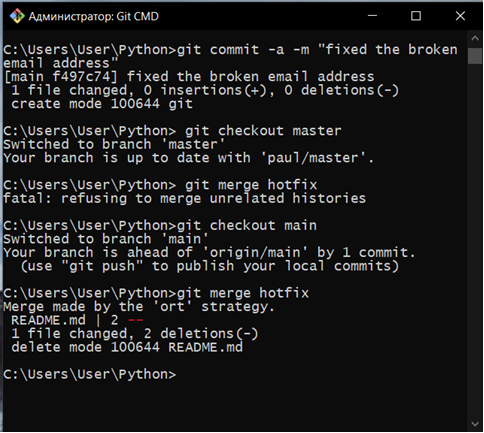
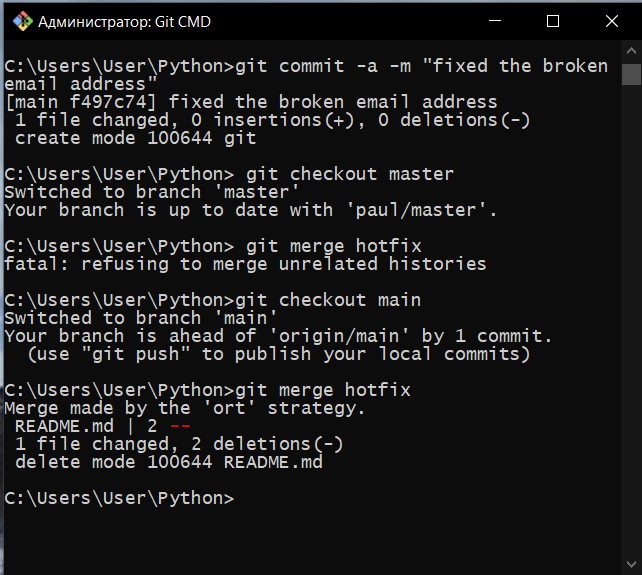


Рисунок 10

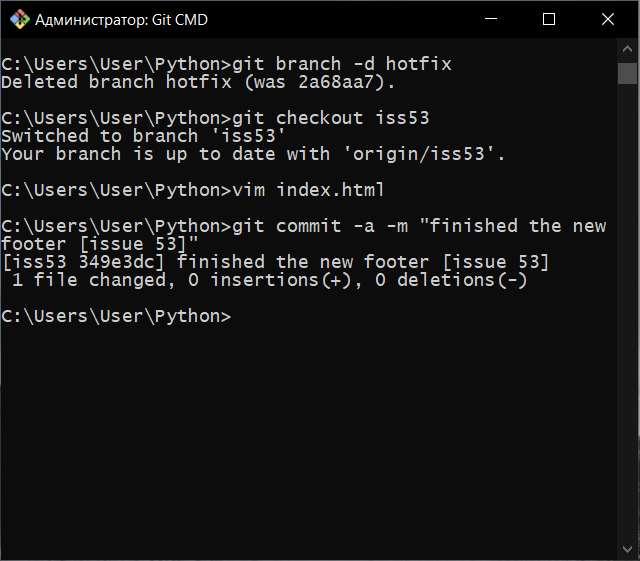


Рисунок 11

1. **Основы слияния.**

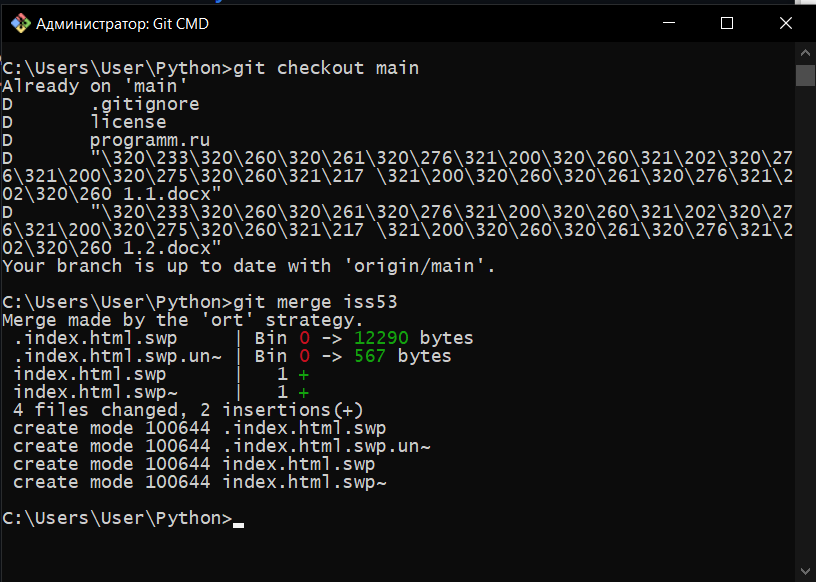


Рисунок 12

1. **Управление ветками**

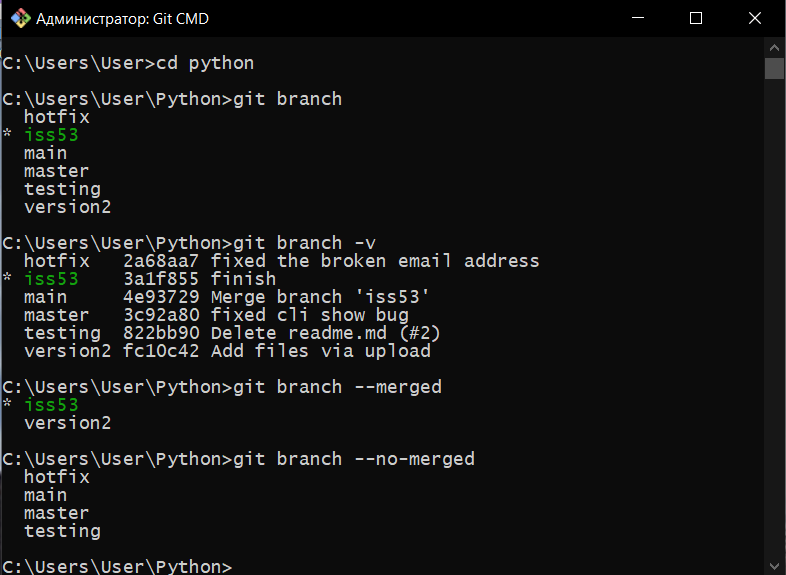


Рисунок 13

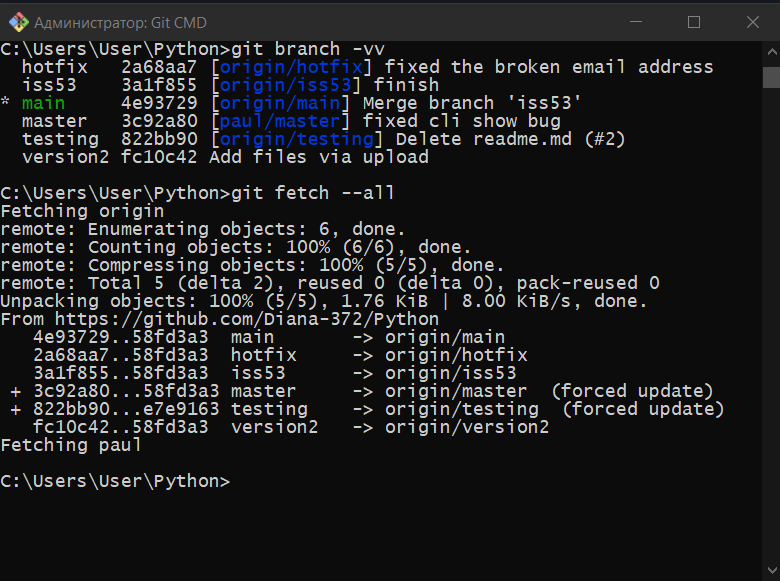


Рисунок 14

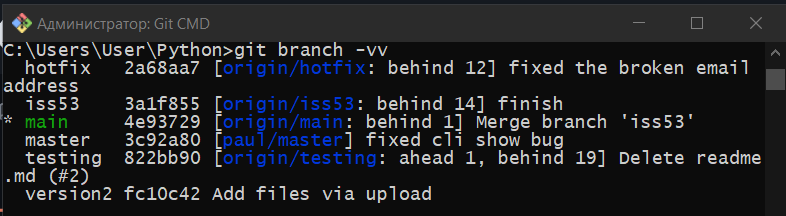


Рисунок 15

Ссылка: <https://github.com/Diana-372/Python.git>

**Контрольные вопросы:**

1. Ветка в Git это параллельная версия проекта, которая может быть использована для разработки новых фич, исправления багов или проведения экспериментов, не влияющих на основную версию проекта.

2. HEAD в Git это указатель на текущую ветку или коммит, на котором находится пользователь или репозиторий.

3. Способы создания веток в Git:

- Создание новой ветки с помощью команды git branch <имя\_ветки>

- Создание и переключение на новую ветку с помощью команды git checkout -b <имя\_ветки>

4. Текущую ветку можно узнать с помощью команды git branch, которая выведет список всех существующих веток, при этом звездочка (\*) будет указывать на текущую ветку.

5. Для переключения между ветками можно использовать команду git checkout <имя\_ветки>, где <имя\_ветки> - это имя ветки, на которую нужно переключиться.

6. Удаленная ветка в Git это ветка, которая находится на удаленном сервере (например, на Bitbucket), а не локально на компьютере пользователя.

7. Ветка отслеживания (tracking branch) в Git это локальная ветка, которая связана с удаленной веткой и служит для отслеживания изменений в удаленной ветке и синхронизации их с локальной веткой.

8. Создать ветку отслеживания можно с помощью команды git checkout -b <имя\_ветки> <имя\_удаленной\_ветки>.

9. Чтобы отправить изменения из локальной ветки в удаленную ветку, нужно выполнить команду git push <имя\_удаленного\_репозитория> <имя\_локальной\_ветки>:<имя\_удаленной\_ветки>.

10. Основное отличие между командами git fetch и git pull заключается в том, что git fetch загружает изменения из удаленного репозитория, но не объединяет их с локальной веткой, в то время как git pull загружает изменения и сразу же объединяет их с локальной веткой.

11. Удалить локальную ветку можно с помощью команды git branch -d <имя\_ветки>, а удаленную ветку - с помощью команды git push <имя\_удаленного\_репозитория> --delete <имя\_удаленной\_ветки>.

12. Git-flow это модель ветвления в Git, которая предлагает определенный набор стандартов и практик, которые помогают организовать процесс совместной разработки. Она состоит из двух основных веток: master и develop, а также из дополнительных веток типа feature, release и hotfix.

13. Основные типы веток в git-flow:

- master - главная ветка, в которой находится последняя стабильная версия проекта;

- develop - ветка, в которой происходит разработка новых фич;

- feature - ветка, создаваемая для разработки новой функциональности;

- release - ветка, создаваемая для подготовки выпуска новой версии проекта;

- hotfix - ветка, создаваемая для исправления критических ошибок в готовой версии проекта.

14. Инструменты для работы с ветками Git, предоставляемые средствами с GUI могут включать в себя: возможность создания, удаления и переключения между ветками с помощью графического интерфейса, просмотр списка всех доступных веток, отслеживание изменений и синхронизация с удаленным репозиторием. Некоторые популярные GUI-клиенты для работы с Git включают в себя GitHub Desktop, GitKraken, Sourcetree и другие.