

# Отчёт по 3 лабе, содержание проекта

October 2022

- 1 Структура проекта
- 2 Описание использованных команд
- 3 История изменения файлов и работы с проектом

# Структура проекта

Проект содержит 2 ветки - main(основная) и резервная(Brovkin)

В основной ветке содержатся описание проекта и его версии

В резервной ветке хранится README с описанием проекта и первая лабораторная (Автоматизация сборки проекта)

## 1. config

В данном файле содержатся настройки Git репозитория. Например, здесь можно хранить email и имя пользователя.

## 2. description

Данный файл предназначен для GitWeb и содержит в себе информацию о проекте (название проекта и его описание). GitWeb - это веб интерфейс, написанный для просмотра Git репозитория используя веб-браузер. Если вы не пользуетесь GitWeb, то это не столь важно.

## 3. hooks

В этом каталоге Git предоставляет набор скриптов, которые могут автоматически запускаться во время выполнения git команд. В некоторых случаях это значительно упрощает разработку. Например, вы можете написать скрипт, который будет редактировать сообщение коммита согласно вашим требованиям.

## 4. info - exclude

Каталог info содержит файл exclude, в котором можно указывать любые файлы, и Git не станет добавлять их в свою историю. Это почти то же самое что и .gitignore (возможно вы сталкивались с ним. Его можно найти в корневом каталоге вашего проекта), за тем исключением, что exclude не сохраняется

в истории проекта, и вы не сможете им поделиться с другими.

#### 5. refs

Каталог refs хранит в себе копию ссылок на объекты коммитов в локальных и удаленных ветках.

#### 6. logs

Каталог logs хранит в себе историю проекта для всех веток в вашем проекте.

#### 7. objects

Каталог objects хранит в себе BLOB объекты, каждый из которых проиндексирован уникальным SHA.

#### 8. index

Промежуточная область с метаданными, такими как временные метки, имена файлов, а также SHA файлов, которые уже упакованы Git. В эту область попадают файлы, над которыми вы работали, при выполнении команды `git add`.

#### 9. HEAD

Файл содержит ссылку на текущую ветку, в которой вы работаете

#### 10. ORIGHEAD

Каждый раз во время слияния в этот файл попадает SHA ветки, с которой проводилось слияние

#### 11. FETCHHEAD

Файл хранит в себе ссылки в виде SHA на ветки, которые участвовали в `git fetch`

#### 12. MERGEHEAD

Файл хранит в себе ссылки в виде SHA на ветки, которые участвовали в git merge

### 13. COMMITEDITMSG

Файл содержит в себе последнее введенное вами сообщение коммита

## Описание использованных команд

Для создания ветки github'a и её обновлением использовался github desktop

Команды которые были выполнены с помощью github desktop

Команды которые были использованы

Добавить изменения в индекс(staging area) - `git add`

Закоммитить изменения - `git commit`

Запустить - `git push`

Команда `git status` отображает состояние директории и индекса(staging area). Это позволяет определить, какие файлы в проекте отслеживаются Git, а также какие изменения будут включены в следующий коммит.

`git remote add origin` - Эта команда сопоставит удаленное хранилище с ссылкой на локальный репозиторий. С этого момента можно обращаться к удаленному репозиторию через ссылку

`git push origin` - отправление результатов нашей работы в репозиторий.

`git branch "что-то"` - Создание ветки в проекте

`git checkout "что-то"` - Переключение на другую ветку проекта

Команды которые были прописаны  
работа с историей

`git log` - Данная команда предназначена для отображения всей нашей истории. Она может быть весьма удобна, если нам пона-

добилось узнать, какие изменения мы вносили ранее. Или если нам нужно откатиться до определенного места в истории, либо если есть нужда её отредактировать.

Команда `git show` используется для отображения полной информации о любом объекте в Git, будь то коммит или ветка. По умолчанию `git show` отображает информацию коммита, на который в данный момент времени указывает HEAD.

`git reflog` - Эта команда выводит упорядоченный список коммитов, на которые указывал HEAD. Грубо говоря, она отображает историю всех ваших перемещений по проекту. Основное преимущество этой команды заключается в том, что если мы вдруг случайно удалили часть истории или откатились назад, мы сможем проинспектировать момент утраты нужной вам информации и откатиться обратно.

Команды которые не использовались, но могут пригодиться при работе с историей

`git reset` - Она позволяет откатить проект до определенной точки. Эту команду можно использовать с тремя параметрами:

1. В случае с `-soft`, содержимое вашего индекса, а также рабочей директории, остается неизменным. Это значит, что если мы откатимся назад на пару коммитов, мы изменим ссылку указателя HEAD на указанный коммит и все изменения, которые были до этого внесены, окажутся в индексе.

2. При использовании параметра `-mixed`, мы опять-таки изменим ссылку указателя HEAD, но все предыдущие изменения в индекс не попадут, а будут отслеживаться как не занесенные в индекс. Это дает возможность внести в индекс только те изменения, которые нам необходимы, что довольно удобно!

3. Если использовать команду `git reset` с параметром `–hard`, мы снова изменим ссылку указателя **HEAD**, но все предыдущие изменения не попадут ни в индекс, ни в зону отслеживаемых файлов. Это значит, что мы полностью сотрем все изменения, которые вносили ранее. Это также удобно, если вы знаете, что вам больше не пригодится ваша предыдущая работа над проектом.

`git branch` - Посмотреть текущую ветку проекта

`git merge dev` - Слияние двух веток воедино

# История изменения файлов и работы с проектом

## История всех коммитов

commit 0241876b5ab45f93e98a2450f8a9e6f336d135ea (HEAD -> Brovkin, origin/Brovkin)

Author: BrovkinArtem <artembro2014@mail.ru>  
Date: Mon Nov 14 03:29:18 2022 +0300

**Delete - удаление файла питона из проекта (был лишним)**

commit 051b8e094b80a77ea89e0ffe7414c3e46e300915  
Author: BrovkinArtem <artembro2014@mail.ru>  
Date: Mon Nov 14 03:28:25 2022 +0300

**Update - добавление README файла в папку src 1 лабораторной**

commit e86b5efaec0ededb7744cc9669eb223c7aec251a  
Author: BrovkinArtem <artembro2014@mail.ru>  
Date: Mon Nov 14 02:41:06 2022 +0300

**Modify - Обновление main.c файла (добавление дополнительного описания)**

commit 81d1b9805b0494c3f7388982cd277733373dbb89  
Author: BrovkinArtem <artembro2014@mail.ru>  
Date: Mon Nov 14 02:14:16 2022 +0300

**add - Добавление папки первой лабораторной в резервную ветку**

commit 9ec63325e51777b2d49712d2fb57dd06ea6992b5



Author: BrovkinArtem <artembro2014@mail.ru>

Date: Mon Nov 14 02:03:36 2022 +0300

**first - Загрузка резервного проекта Casetools в репозиторий гитхаба netToolsLab3**

commit a50364470ce8636bbbcdcaee4597bef072c4d1e1

Author: BrovkinArtem <78733604+BrovkinArtem@users.noreply.github.com>

Date: Mon Nov 14 02:02:17 2022 +0300

**Initial commit - проинициализирование Git директории в своем проекте и добавление README файла  
(END)**