Отчет по лабораторной работе №1

Тегин Александр  
ПИН-34

1.1

**git init** - создает новый репозиторий Git

**git pull** - используется для извлечения и загрузки содержимого из удаленного репозитория и немедленного обновления локального репозитория этим содержимым

**git push** - используется для выгрузки содержимого локального репозитория в удаленный репозиторий

**git fetch** - загружает коммиты, файлы и ссылки из удаленного репозитория в ваш локальный репозиторий

**git add** - добавляет содержимое рабочего каталога в индекс (staging area) для последующего коммита

**git commit** - берет все подготовленные изменения и отправляет их в репозиторий как единое целое

**git status** - проверить индексацию изменений и увидеть файлы, которые не отслеживаются

**git clone** - команда клонирования репозитория - используется для первоначального копирования репозитория, т.е. для создания копии на вашей машине, когда у вас вообще ничего нет - скачает всю информацию

**git checkout** - используется для копирования файлов из истории или сцены в рабочую директорию

**git** **checkout -b** - команда создаст новую ветку, а затем переключится на нее

**git checkout –file** - копирует файлы со сцены в рабочую директорию

**git reset** - сообщаете к какой версии вы хотите вернуться, и она восстанавливает именно это состояние — отменяя все изменения, которые были осуществлены после этой временной точки

**git reset** **–hard** - команда, которую следует использовать с осторожностью. git reset --hard вернет дерево проекта и индекс в состояние, соответствующее указанному коммиту, удалив изменения последующих коммитов

**git reset** **–hard** - оставит нетронутыми ваши индекс и все дерево файлов и директорий проекта, вернется к работе с указанным коммитом

**git branch** - создание, перечисление и удаление веток

**git log** - получить информацию об истории коммитов; коммитах, изменивших отдельный файл; коммитах за определенный отрезок времени и так далее

**git cherry-pick** - применение к дереву проекта изменений, внесенных отдельным коммитом

**git merge** - слияние веток, разрешение возможных конфликтов

**git config** – удобная функция, которая используется для настройки значений конфигурации Git на глобальном и локальном уровнях проекта. Эти уровни конфигурации соответствуют текстовым файлам .gitconfig . При выполнении команды git config происходит изменение текстового файла конфигурации

* **--global**

Конфигурация глобального уровня зависит от пользователя, то есть применяется к пользователю операционной системы. Значения глобальной конфигурации хранятся в файле, который находится в домашнем каталоге пользователя. Это ~ /.gitconfig в Unix-системах и C:\<имя\_пользователя>\.gitconfig в системах Windows.

1.3

