

*МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ*

*НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ*

*„КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. І. СІКОРСЬКОГО”*

*НАВЧАЛЬНО-НАУКОВИЙ КОМПЛЕКС*

*„ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ”*

# *Лабораторна робота №1*

## *з курсу «ПРОЕКТУВАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ*»

## *Тема:* «***Системи контролю версій SVN, GIT***»

*Варіант №21*

Виконав

студент IV курсу

групи ДА-61

*Саядов Ільмар*

***Київ – 2019***

**Мета роботи:** за допомогою системи контролю версій завантажити коди програми у репозиторій. Відтворити типовий цикл розробки програмного забезпечення з використанням системи контролю версій.

1. **Обрати безкоштовну систему репозиторія для системи контролю версіями, наприклад projectlocker, або інш.**

Для виконання лабораторних робіт було вибрано систему контролю версій Github.

Було створено репозиторій для половини студентів группи, у якому є дві гілки: dev and master. В гільці dev буде проводись вся робота.

1. **Встановити клієнтське безкоштовне програмне забезпечення для роботи с системою контролю версій (GIT, SVN clients).**

Було встановлено системою контролю версій – GIT.

1. **Опис команд, які використовувалися протягом виконання роботи з системою контролю версіями.**



**git init** — создает репозиторий в существующей директории.

**git clone** — клонирует Git-репозиторий из существующего репозитория. По сути, описать ее работу можно следующим образом: сначала она создает новый каталог, далее выполняет команду git init (создает пустой репозиторий), затем git remote add (добавляет новый удалённый репозиторий), после чего — git fetch (загружает данные с сервера) и напоследок git checkout (обновляет рабочию директорию до состояния последниго коммита).

**git add** — добавляет файлы из рабочей директории в индекс для последующего коммита (основные команды).

**git status** — показывает состояние рабочего дерева. Показывает какие файлы изменены, но недобавлены в индекс, а какие добавлены и ожидают коммита (основные команды).

**git diff** — показывает изменения между коммитами, меджу комиттом и рабочим деревом и т.д. Вобщем, эта команда вычисляет разницу между двумя деревьями (основные команды).

**git difftool** — показывает изменения между двумя Git деревьями, используя внешнюю утилиту сравнения (основные команды).

**git commit** — записывает изменения в репозиторий (которые добавлены в индекс) (основные команды).

**git reset** — изменяет текущий указатель HEAD на необходимое состояние. Эта команда в основном используется для отмены изменений (основные команды).

**git rm** — удаляет файл из рабочей директории и индекса (основные команды).

**git mv** — делает перемещение или переименования файл, каталог или символическую ссылку (основные команды).

**git clean** — удаляет мусор из рабочей директории (основные команды).

**git branch** — выводит список ветвей, создает новые или удаляет ветки. Общем, эта команда служит для управления ветками в Git.

**git checkout** — делает переключение между ветками и выгружает их содержимое в рабочую директорию.

**git merge** — делает слияние для одной или нескольких веток в текущую.

**git mergetool** — при возникновении конфликтов при слиянии веток, этой командой вызывается внешняя программа.

**git log** — показывает историю коммитов.

**git stash** — временно сохраняет все незакомиченные изменения для очистки рабочей директории.

**git tag** — команда задает постоянную метку на какой-либо коммит, а также, управляет этими метками. В основном, используется для релизов проектов (основные команды).

**git fetch** — забирает из удаленного репозитория данные, которых у вас нет и сохраняет их локально.

**git pull** — забирает из удаленного репозитория данные и сливает их с текущей веткой. Эта одна команда заменяет две git fetch и git merge.

**git push** — вносит на удаленный репозиторий данные, которых там нет, но есть локально у вас. Эта команда использует аутентификацию (основные команды).

**git remote** — управляет списком удаленных репозиториев.

**git archive** — упаковывает в архив весь репозиторий, или только указанные коммыты.

**git submodule** — управляет вложенными репозиториями (могут быть сторонние библиотеки).

**Висновок:** Система керування– це спеціальне програмне забезпечення для полегшення роботи з інформацією, яка часто змінюється. Система керування версіями дозволяє зберігати декілька версій одного і того ж документа, при необхідності повертатися до більш ранніх версій, визначати, хто і коли зробив ту чи іншу зміну, керувати гілками різноманітних версій програми.