**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**

**„КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”**

**НАВЧАЛЬНО-НАУКОВИЙ КОМПЛЕКС**

**„ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ”**

Лабораторна робота №1

з курсу «Проектування інформаційних систем» на тему:

“Системи контролю версій SVN, GIT.”

Виконав:

студент 4 курсу

групи ДА-62

Василевич Б.С.

Київ 2019

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

**Завдання:**

1. Обрати безкоштовну систему репозиторія для системи контролю версіями,

наприклад projectlocker, або інш.

2. Встановити клієнтське безкоштовне програмне забезпечення для роботи с системою

контролю версій (GIT, SVN clients).

3. Протягом роботи над лабораторними роботами 2-6 використовувати систему

контролю версіями.

4. Описати цикл розробки програмного забезпечення з використанням системи

контролю версій.

**Короткі теоретичні відомості:**

Git это распределенная система управления версиями. Это значит, что у каждого участника проекта есть своя копия репозитория, которая находиться в папке “.git”, которая расположена в корне проекта. Именно в этой папке хранятся все коммиты и другие объекты Git. Когда вы работаете с Git, он в свою очередь работает с этой папкой.

Завести новый репозиторий очень просто, это делается командой

git init

Таким образом у вас получиться новый пустой репозиторий. Если вы хотите присоединиться к разработке уже имеющегося проекта, то вам нужно будет скопировать этот репозиторий в свою локальную папку с удаленного репозитория. Делается это так:

git clone <url удаленного репозитория>

После чего в текущей папке появляется директория .git в которой и будет содержаться копия удаленного репозитория.

Существует несколько основных областей в которых находиться код.

* Рабочая директория – это файлы в корне проекта, тот код с которым вы работаете.
* Локальный репозиторий — она же директория “.git”. В ней хранятся коммиты и другие объекты.
* Удаленный репозиторий – тот самый репозиторий который считается общим, в который вы можете передать свои коммиты из локального репозитория, что бы остальные программисты могли их увидеть. Удаленных репозиториев может быть несколько, но обычно он бывает один.
* Есть еще одна область, с пониманием которой обычно бывают проблемы. Это так называемая область подготовленных изменений (staging area). Дело в том, что перед тем как включить какие-то изменения в коммит, нужно вначале отметить что именно вы хотите включить в этот коммит. В своей рабочей директории вы можете изменить несколько файлов, но в один конкретный коммит включать не все эти изменения.

В целом работа с гитом выглядит так: вы меняете файлы в своей рабочей директории, затем добавляете эти изменения в staging area используя команду

git add <имя/файла>

При этом можно использовать маски со звездочкой.

Потом вы делаете коммит в свой локальный репозиторий

git commit –m “Комментарий к коммиту”

Когда коммитов накопиться достаточно много, чтобы ими можно было поделиться, вы выполняете команду

git push

После чего ваши коммиты уходят в удаленный репозиторий.

Если нужно получить изменения из удаленного репозитория, то нужно выполнить команду

git pull

После этого, в вашем локальном репозитории появятся те изменения, которые были отправлены другими программистами.

Код в рабочей области проекта образуется применением тех изменений, которые содержаться в коммитах. У каждого коммита есть свое имя, которое представляет собой результат хеш функции sha-1 от содержимого самого коммита.

Просмотреть коммиты можно при помощи команды

git log

Формат ответа этой команды по дефолту не очень удобен. Вот такая команда выведет ответ в более читаемом виде

git log --pretty=format:"%H [%cd]: %an - %s" --graph --date=format:%c

Что бы закончить просмотр нужно нажать на клавишу q

Посмотреть, что находиться в рабочей директории и staging area можно командой

git status

Рабочую директорию можно переключить на предыдущее состояние выполнив команду

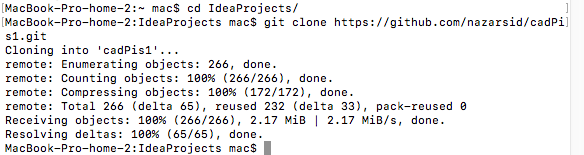
git checkout <hash коммита>

* Только перед тем как это делать выполните git status и убедитесь, что у вас нет никаких локальных и не зафиксированных изменений. Иначе Git не поймет, как ему переключаться. git status подскажет вам что можно сделать с локальными изменениями что бы можно было переключиться. Этого правила следует придерживаться и при всяких других переключениях рабочей области.

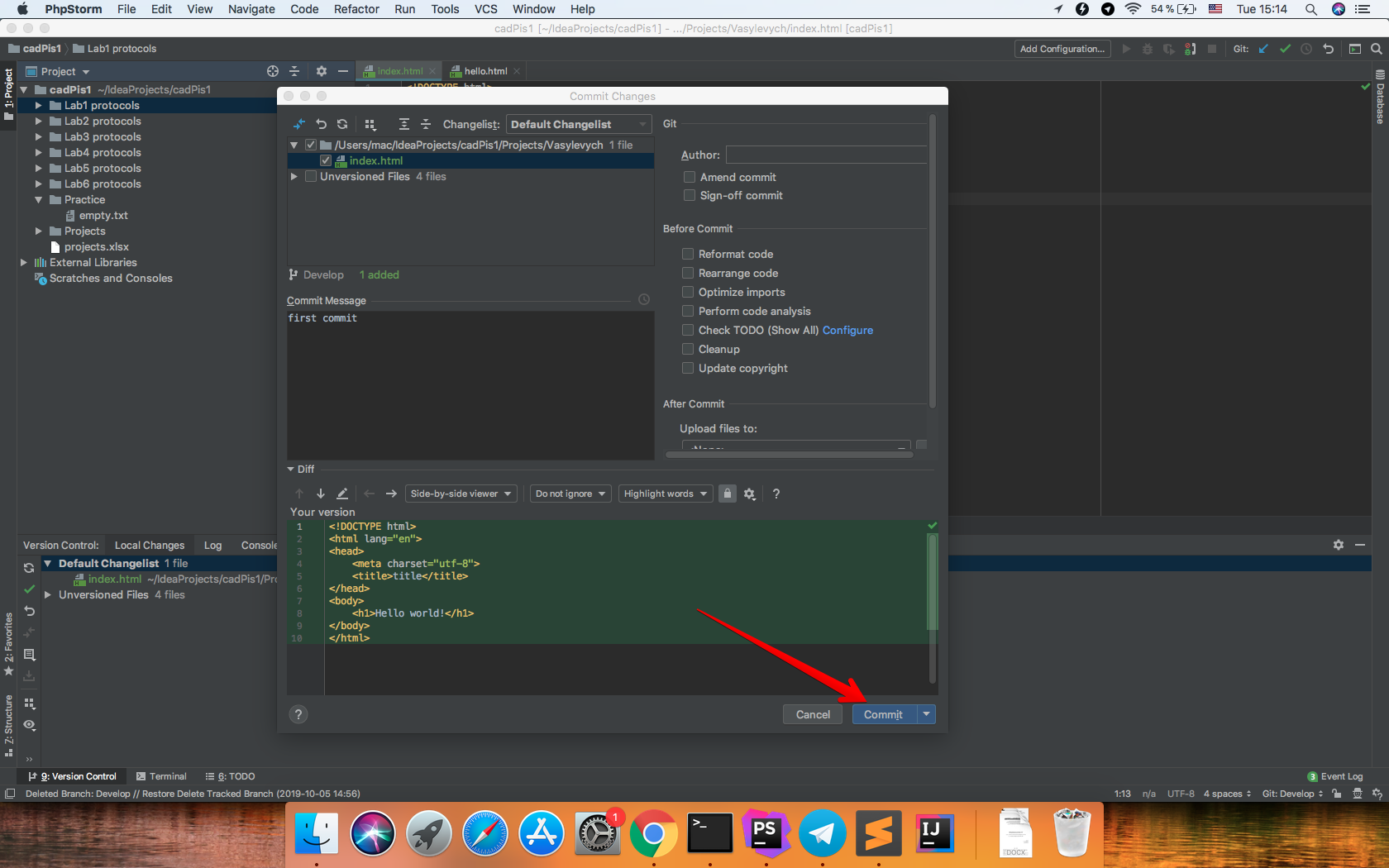
**Основні команди Git:**

1. Створення нового репозиторія - git init
2. Перевірка статусу – git status
3. Переміщення - git add
4. Внесення коміту – git commit
5. Підключення до дистанційного репозиторія – git remote add
6. Завантаження на сервер – git push
7. Клонування репозиторія – git clone
8. Отримання змін з серверу – git pull
9. Створення нової гілки – git branch
10. Перехід між гілками – git checkout
11. З'єднання гілок – git merge

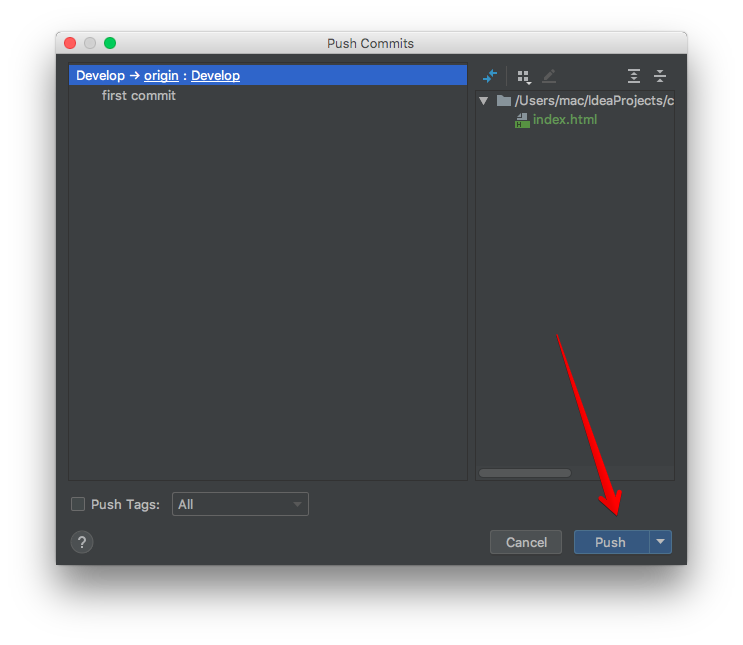
1. Клонуємо проект



2. Робимо перший комміт



3. Заливаємо на гітхаб



**Висновки:** у ході роботи було набуто навичок роботи з системою контролю версій - Git, опановано основні команди clone, branch, checkout, push, commit.