

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”

виконала: студентка IV курсу

групи ДА-71

Жук В. А.

Варіант - 5

КИЇВ 2020

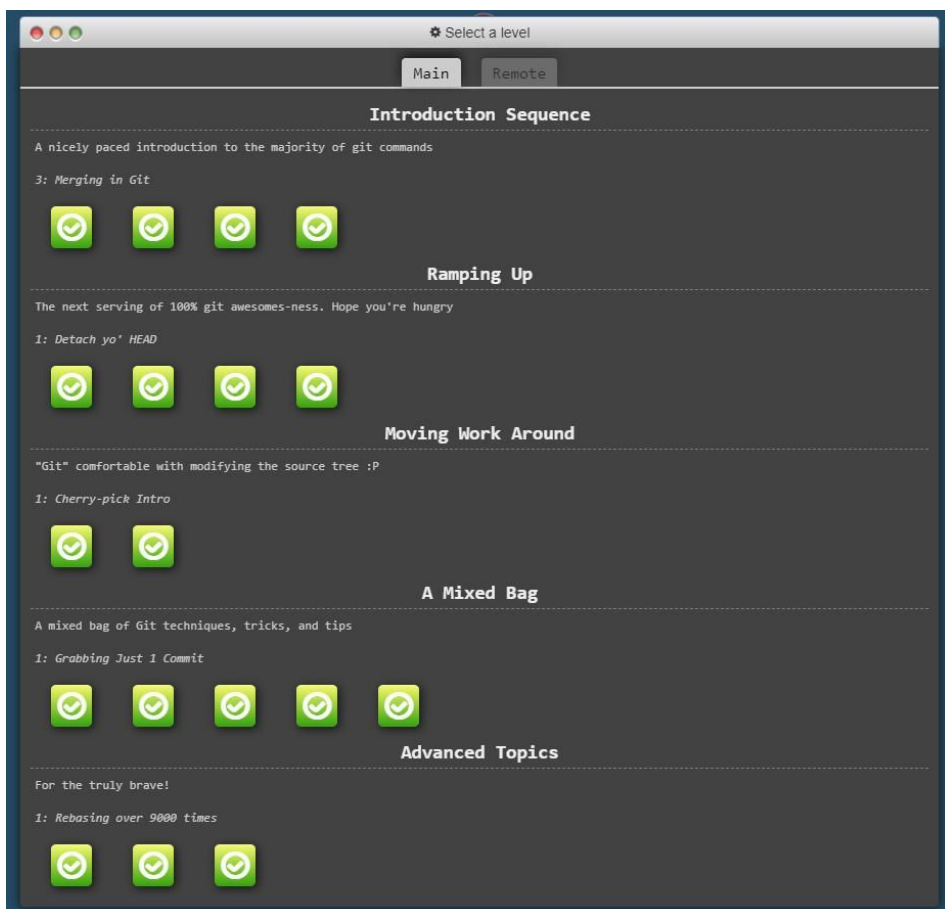
Мета роботи: за допомогою системи контролю версій завантажити коди програми у репозиторій. Відтворити типовий цикл розробки програмного забезпечення з використанням системи контролю версій.

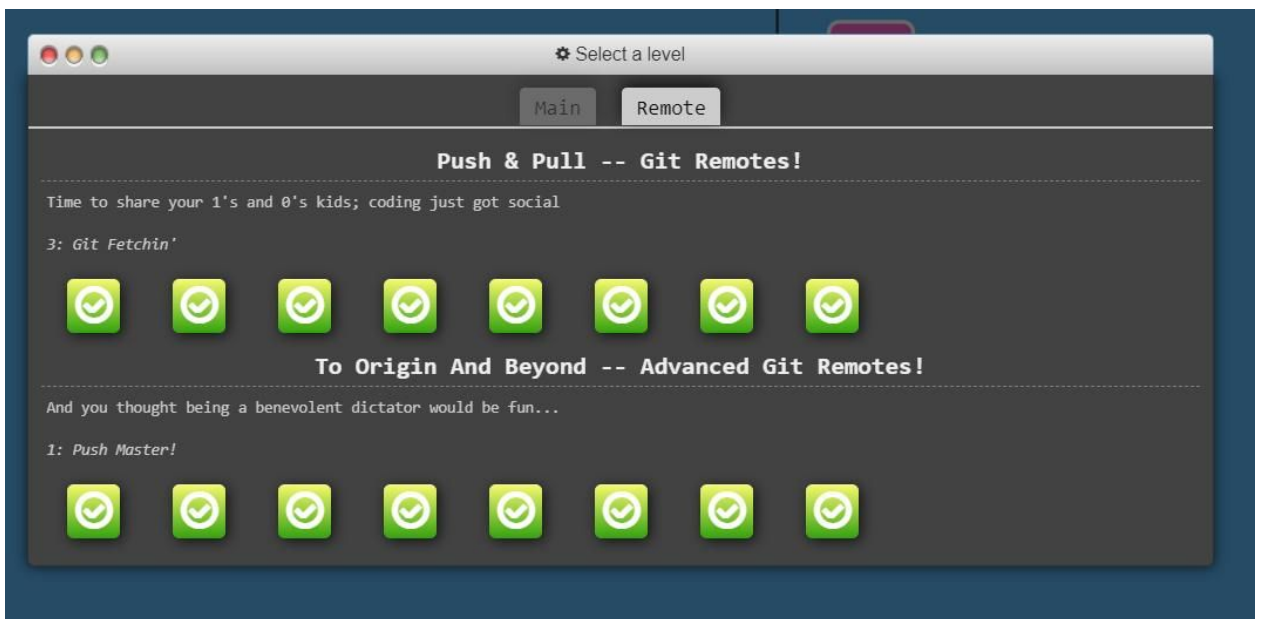
Завдання:

1. Обрати безкоштовну систему репозиторія для системи контролю версіями, наприклад projectlocker, або інш.
2. Встановити клієнтське безкоштовне програмне забезпечення для роботи з 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.

Хід роботи:

1. Результати проходження тренінгової системи:





2. Результати початку роботи у загальному репозиторії з описом унікальних команд:

```
valen@LAPTOP-KNN66D86 MINGW64 /d/PAIS
$ git clone https://github.com/TarasVolovodenko/da71.git
Cloning into 'da71'...
remote: Enumerating objects: 14, done.
remote: Counting objects: 100% (14/14), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 14 (delta 0), reused 8 (delta 0), pack-reused 0
Unpacking objects: 100% (14/14), 3.01 KiB | 3.00 KiB/s, done.
```

git clone - створення локальної копії віддаленого репозиторію

```
valen@LAPTOP-KNN66D86 MINGW64 /d/PAIS
$ cd da71

valen@LAPTOP-KNN66D86 MINGW64 /d/PAIS/da71 (master)
$ git fetch

valen@LAPTOP-KNN66D86 MINGW64 /d/PAIS/da71 (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
```

git fetch - збір всіх комітів цільової гілки, яких немає в поточній, та збереження їх в локальному репозиторії

git status - перевірка стану репозиторія, чи зараз є зміни, що очікують запису, чи зберігається поточний стан каталогу

```
valen@LAPTOP-KNN66D86 MINGW64 /d/PAIS/da71 (master)
$ git checkout -b valya_zhuk
Switched to a new branch 'valya_zhuk'
```

git checkout - перенесення вказівника HEAD (тобто куди дивиться зараз локальна копія)

git checkout -b - створення нової гілки і перенесення на неї вказівника HEAD

у локальному репозиторії створили папку Zhuk

```
valen@LAPTOP-KNN66D86 MINGW64 /d/PAIS/da71 (valya_zhuk)
$ git status
On branch valya_zhuk
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    zhuk/

nothing added to commit but untracked files present (use "git add" to track)
valen@LAPTOP-KNN66D86 MINGW64 /d/PAIS/da71 (valya_zhuk)
$ git add .
```

git add . - додає весь вміст робочої директорії в індекс (staging area - файл в каталозі git, що зберігає інформацію про те, що має увійти в наступний коміт) для подальшого коміта

```
valen@LAPTOP-KNN66D86 MINGW64 /d/PAIS/da71 (valya_zhuk)
$ git status
On branch valya_zhuk
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   zhuk/README.txt

valen@LAPTOP-KNN66D86 MINGW64 /d/PAIS/da71 (valya_zhuk)
$ git commit
[valya_zhuk 242a80f] initial commit
1 file changed, 1 insertion(+)
create mode 100644 zhuk/README.txt
```

git commit - бере усі дані, що додані в індекс за допомогою git add, і зберігає їх зліпок у внутрішній БД, і пересуває вказівник поточної гілки на це зліпок

```

valen@LAPTOP-KNN66D86 MINGW64 /d/PAIS/da71 (valya_zhuk)
$ git status
On branch valya_zhuk
nothing to commit, working tree clean

valen@LAPTOP-KNN66D86 MINGW64 /d/PAIS/da71 (valya_zhuk)
$ git push
fatal: The current branch valya_zhuk has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin valya_zhuk

```

git push - злиття локальних змін в віддалений репозиторій

```

valen@LAPTOP-KNN66D86 MINGW64 /d/PAIS/da71 (valya_zhuk)
$ git push --set-upstream origin valya_zhuk
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 312 bytes | 312.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'valya_zhuk' on GitHub by visiting:
remote:   https://github.com/TarasVolovodenko/da71/pull/new/valya_zhuk
remote:
To https://github.com/TarasVolovodenko/da71.git
 * [new branch]      valya_zhuk -> valya_zhuk
Branch 'valya_zhuk' set up to track remote branch 'valya_zhuk' from 'origin'.

```

git push --set--upstream origin - злиття локальних змін в віддалений репозиторій зі створенням моєї гілки на віддаленому репозиторії і відслідковуванням її моєю локальною гілкою

```

valen@LAPTOP-KNN66D86 MINGW64 /d/PAIS/da71 (valya_zhuk)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

valen@LAPTOP-KNN66D86 MINGW64 /d/PAIS/da71 (master)
$ git fetch

valen@LAPTOP-KNN66D86 MINGW64 /d/PAIS/da71 (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean

```

у локальному репозиторії в папці Zhuk створили файл README.txt


```

valen@LAPTOP-KNN66D86 MINGW64 /d/PAIS/da71 (master)
$ git merge valya_zhuk
Updating 7609b1c..242a80f
Fast-forward
 Zhuk/README.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 Zhuk/README.txt

valen@LAPTOP-KNN66D86 MINGW64 /d/PAIS/da71 (master)
$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

valen@LAPTOP-KNN66D86 MINGW64 /d/PAIS/da71 (master)
$ git push
Total 0 (delta 0), reused 0 (delta 0)
To https://github.com/TarasVolovodenko/da71.git
 7609b1c..242a80f master -> master

```

3. Опис циклу розробки програмного забезпечення з використанням системи контролю версій

Процес роботи з Git виглядає таким чином:

- Створення локальної копії віддаленого репозиторію (git clone)
- Зміни локальної копії проекту (додавання/зміна/видалення файлів)
- Оновлення локальної копії при необхідності (git pull)
- Підтвердження змін в локальному репозиторії (git commit)
- Додавання змін із локального репозиторія в центральний (git push)

Схема може виглядати таким чином:



Висновок: У цій лабораторній роботі ми розглянули роботу з системою контролю версій Git. Освоїли та повторили практичні навички з Git. Також була пройдена тренінгова поглиблена система для покращення знань та розглянуто ЖЦ розробки програмного забезпечення в розрізі роботи з Git.