

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 ”**

виконав: студент IV курсу  
групи ДА-71  
Циолковський Р.Р.

КИЇВ  
2020

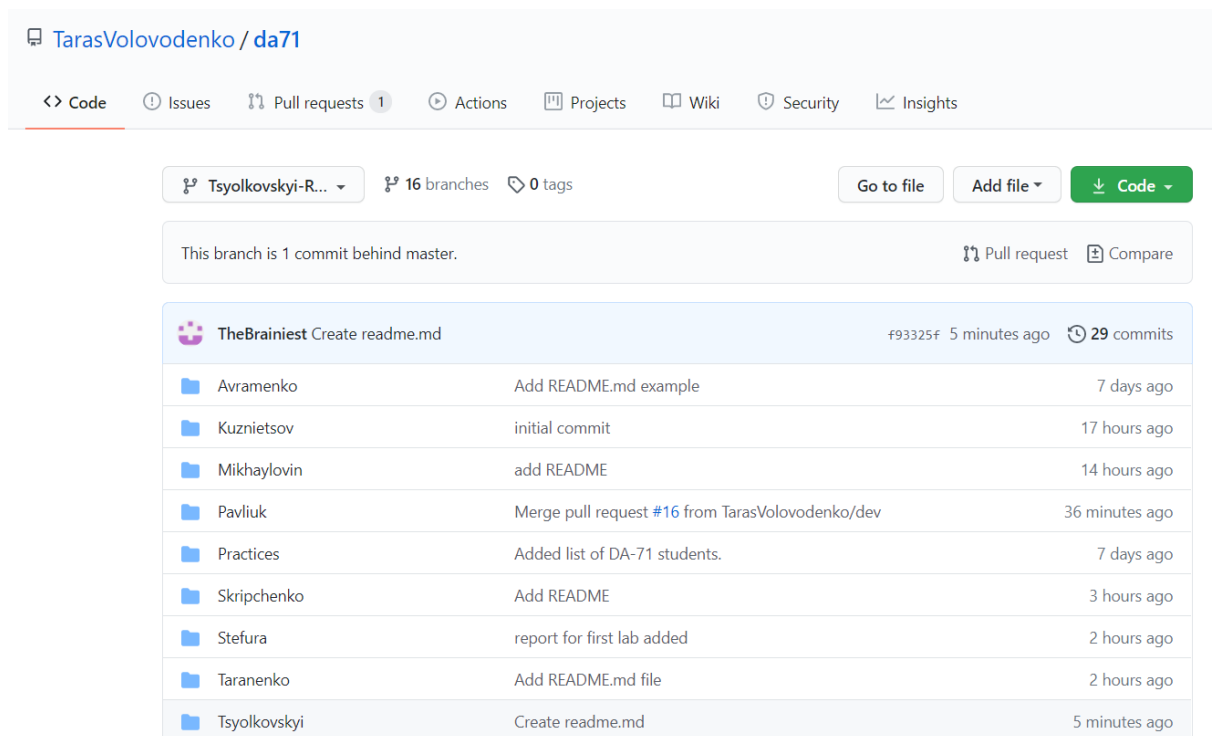
## Мета роботи:

За допомогою системи контролю версій завантажити коди програми у репозиторій. Відтворити типовий цикл розробки програмного забезпечення з використанням системи контролю версій.

## Хід роботи:

1. Оберемо безкоштовну систему репозиторія для системи контролю версіями.

Для виконання лабораторних робіт було вибрано систему контролю версій Github. Був створений репозиторій, у якому є дві гілки: dev and master. Вся робота буде проводитись у гілці dev з копіюванням стабільних версій у гілку master.



2. Встановимо клієнтське безкоштовне програмне забезпечення для роботи з системою контролю версій.

```
roma6@DESKTOP-5RE3QFB MINGW64 /e/3 курс/7 семестр/ПИС/da71 (master)
$ git version
git version 2.20.1.windows.1
```

3. Опишем команды, які ми використали для виконання лабораторної роботи.

#### **git add**

Команда `git add` добавляет содержимое рабочей директории в индекс (staging area) для последующего коммита. По умолчанию `git commit` использует лишь этот индекс, так что вы можете использовать `git add` для сборки слепок вашего следующего коммита.

#### **git status**

Команда `git status` показывает состояния файлов в рабочей директории и индексе: какие файлы изменены, но не добавлены в индекс; какие ожидают коммита в индексе. Вдобавок к этому выводятся подсказки о том, как изменить состояние файлов.

#### **git commit**

Команда `git commit` берёт все данные, добавленные в индекс с помощью `git add`, и сохраняет их слепок во внутренней базе данных, а затем сдвигает указатель текущей ветки на этот слепок.

#### **git branch**

Команда `git branch` — это своего рода “менеджер веток”. Она умеет перечислять ваши ветки, создавать новые, удалять и переименовывать их.

#### **git checkout**

Команда `git checkout` используется для переключения веток и выгрузки их содержимого в рабочую директорию. `git merge` Команда `git merge` используется для слияния одной или нескольких веток в текущую. Затем она устанавливает указатель текущей ветки на результирующий коммит.

#### **git pull**

Команда `git pull` работает как комбинация команд `git fetch` и `git merge`, т.е. Git вначале забирает изменения из указанного удалённого репозитория, а затем пытается слить их с текущей веткой.

#### **git push**

Команда `git push` используется для установления связи с удалённым репозиторием, вычисления локальных изменений отсутствующих в нём, и собственно их передачи в

вышеупомянутый репозиторий. Этой команде нужно право на запись в репозиторий, поэтому она использует аутентификацию.

**Висновок:**

У лабораторній роботі було продемонстрований алгоритм роботи з системою контролю версій Git та її найбільш характерні риси:

- Git – розподілена система контролю версій, взаємодія між різними учасниками проекту може зводитися тільки до вирішення конфліктів у коді.
- Проте Git має централізований сервер та репозитій з єдиним місцем збереження коду.
- Система Git має зручну систему роботи з гілками та створення своїх версій програмних продуктів (за допомогою процедури fork)