

**НТУУ КПІ ім. Ігоря Сікорського
Інститут прикладного системного аналізу
Кафедра Системного Проектування**

**Лабораторна робота №1
“Проектування інформаційних систем”**

Виконав студенти групи ДА-72

Самвелян Артур

Перевірила:

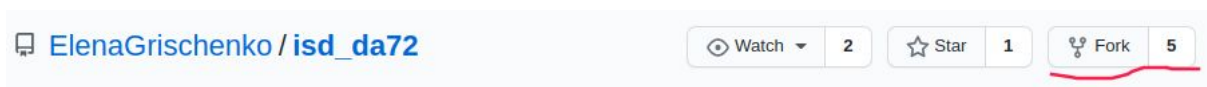
Науменко Т.О.

Київ - 2020

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

Хід роботи:

1. Робимо форк репозиторія:



2. Клонуємо репозиторій:

```
→ ~ git clone https://github.com/ElenaGrischenko/isd_da72.git
Cloning into 'isd_da72'...
remote: Enumerating objects: 211, done.
remote: Counting objects: 100% (211/211), done.
remote: Compressing objects: 100% (170/170), done.
remote: Total 211 (delta 36), reused 182 (delta 21), pack-reused 0
Receiving objects: 100% (211/211), 10.50 MiB | 856.00 KiB/s, done.
Resolving deltas: 100% (36/36), done.
→ ~ █
```

3. Створюємо власну гілку:

```

→ isd_da72 git:(master) git checkout dev
Branch 'dev' set up to track remote branch 'dev' from 'origin'.
Switched to a new branch 'dev'
→ isd_da72 git:(dev) git checkout -b samvelyan
Switched to a new branch 'samvelyan'
→ isd_da72 git:(samvelyan) ls
Avdienko_Vladyslav  Korol_Anton          Okhota_Denys
Bilorus_Ira         Koval_Serhiy         Pashkova_Yelizaveta
Chernomaz_Vadym     Kulyk_Vyacheslav     Ponomarov_Oleh
Diakonov_Denys      Kyrylchuk_Pavlo      Potapova_Sofiia
Hryshchenko_Olena   LICENSE               Practice
Kandel_Kirill       Lupyak_Anastasiya    README.md
Khomenko_Yevhen     Makarov_Ihnat        Samvelyan_Artur
Kohinova_Alisa       Marchenko_Iryna      Voichenko_Vladyslav
Kokidko_Bohdan      Mazurenko_Vladyslav  Zarichnyy_Yaroslav
Kondratiuk_Taras    Melkozerova_Olha     Zhuzha_Anhelina
→ isd_da72 git:(samvelyan) cd Samvelyan_Artur
→ Samvelyan_Artur git:(samvelyan) ls
README.md
→ Samvelyan_Artur git:(samvelyan) nvim README.md
→ Samvelyan_Artur git:(samvelyan) x

```

4. Зробимо зміни в якомусь файлі нашого проекту та зробимо коміт

```

→ Samvelyan_Artur git:(samvelyan) x git status
On branch samvelyan
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
→ Samvelyan_Artur git:(samvelyan) x
→ Samvelyan_Artur git:(samvelyan) x git add README.md
→ Samvelyan_Artur git:(samvelyan) x git status
On branch samvelyan
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   README.md

→ Samvelyan_Artur git:(samvelyan) x git commit -m "Augmented README"
[samvelyan 23787c4] Augmented README
1 file changed, 1 insertion(+), 1 deletion(-)
→ Samvelyan_Artur git:(samvelyan)

```

5. Робимо git push:

```
→ Samvelyan_Artur git:(samvelyan) git push origin samvelyan
Username for 'https://github.com': ElenaGrischenko
Password for 'https://ElenaGrischenko@github.com':
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 342 bytes | 342.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'samvelyan' on GitHub by visiting:
remote:   https://github.com/ElenaGrischenko/isd_da72/pull/new/samvelyan
remote:
To https://github.com/ElenaGrischenko/isd_da72.git
 * [new branch]      samvelyan -> samvelyan
→ Samvelyan_Artur git:(samvelyan) █
```

6. Заходимо на github та переходимо з гілки master на свою гілку та дивимося чи збереглися зміни:

 samvelyan ▾

Go to

Switch branches/tags ×

Branches

Tags

master	default
Kulik	
anastasiia_lupiak	
chernomaz	
dev	
hryshchenko	
kandel_kirill	
okhota	
✓ samvelyan	
View all branches	

ehind

...

addec

addec

addec

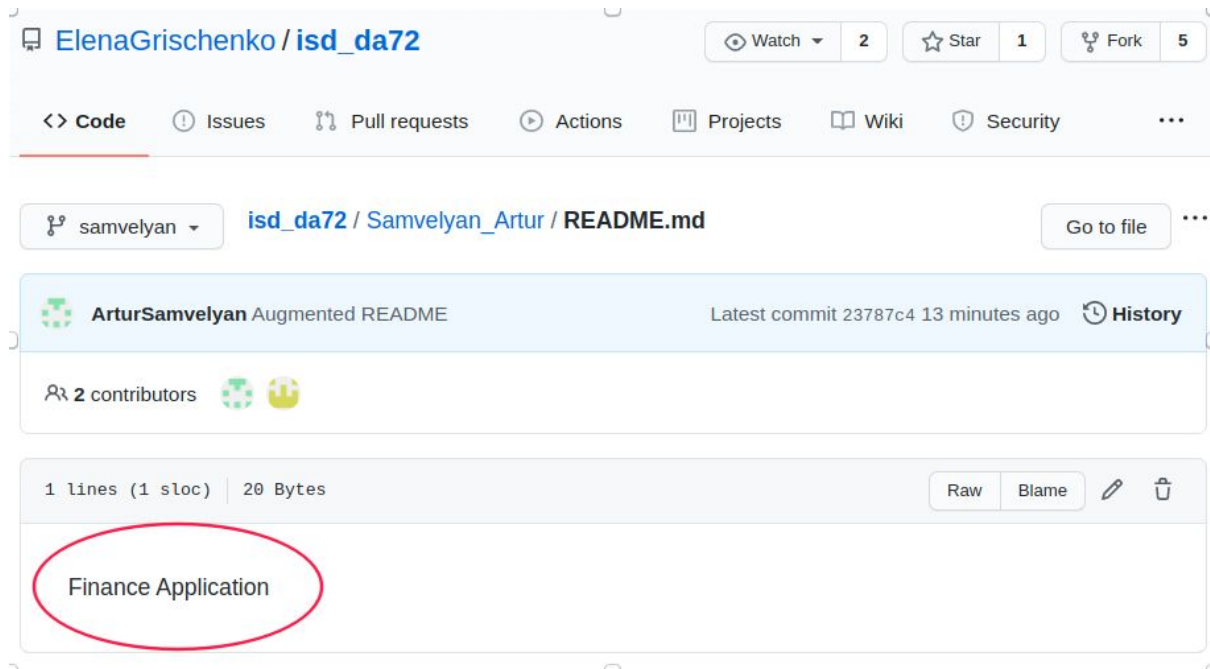
al

addec

addec

addec

addec



Висновки: В даній роботі я за допомогою системи контролю версій завантажив код програми у репозиторій та відтворив типовий цикл розробки програмного забезпечення з використанням системи контролю версій.

На своєму шляху використання git я зіткнувся з такими проблемами:

- 1) Не закомітивши зміни на одній гілці я зробив checkout на другу тим самим втративши ці зміни (не лініться лишній раз написати git status)
- 2) Одного разу забувши зробити зробити pull (оновлення репозиторію), я закомітив та зробив пуш чим самим визвав конфлікт, який я не виправив а залив, отримавши кашу замість коду.