

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ «КПІ»  
ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ  
КАФЕДРА СИСТЕМНОГО ПРОЕКТУВАННЯ**

**Лабораторна робота №1**

**3 курсу: «Проектування інформаційних систем»**

**Виконала:**

**Студентка 4 курсу**

**Групи ДА-72**

**Марченко Ірина**

**Київ – 2020**

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

## Теоретичні відомості

Система контролю версій - це система, що записує зміни у файл або набір файлів протягом деякого часу, так що ви зможете повернутися до певної версії пізніше.

Основною відмінністю від інших систем (таких як Subversion та подібних їй) є те, як Git сприймає дані. Концептуально, більшість СКВ зберігають інформацію як список файлових редагувань. Ці інші системи (CVS, Subversion, Perforce, Bazaar тощо) розглядають інформацію як список файлів та змін кожного з них протягом деякого часу (це зазвичай називають оснований на дельтах контроль версій).

Git не оброблює та не зберігає свої дані таким чином. Замість цього, Git сприймає свої дані радше як низку знімків мініатюрної файлової системи. У Git щоразу, як ви створюєте коміт, тобто зберігаєте стан вашого проекту, Git запам'ятовує як виглядають всі ваші файли в той момент і зберігає посилання на цей знімок. Для ефективності, якщо файли не змінилися, Git не зберігає файли знову, просто робить посилання на попередній ідентичний файл, котрий вже зберігається. Git вважає свої дані більш як **потік знімків**.

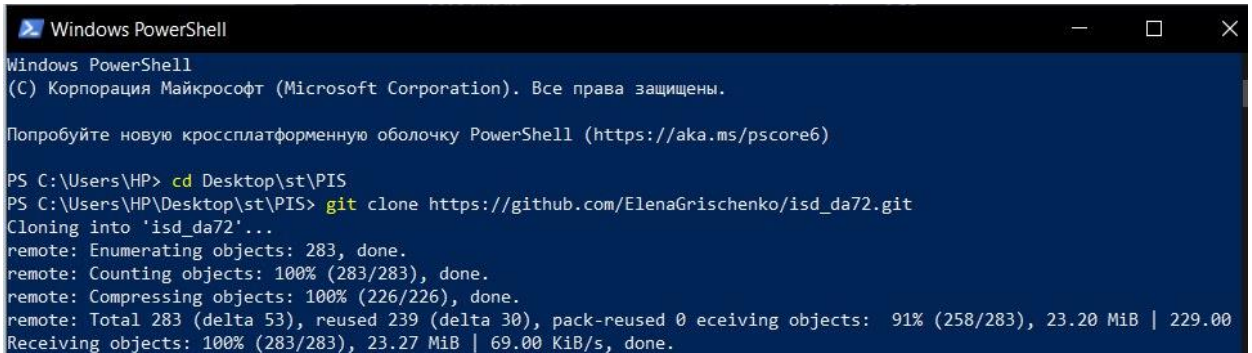
Це дуже важлива різниця між Git та майже всіма іншими СКВ. З цієї причини в Git було заново переосмислено майже кожен аспект контролю версій, які інші системи просто копіювали з попереднього покоління. Це зробило Git більш схожим на мініатюрну файлову систему з деякими неймовірно потужними вбудованими інструментами на додаток, а не просто СКВ.

Git цілісний. Будь-що в Git, перед збереженням, отримує контрольну суму, за якою потім і можна на нього посылатися. Таким чином, неможливо

змінити файл чи директорію так, щоб Git про це не дізнався. Цей функціонал вбудовано в систему на найнижчих рівнях і є невід'ємною частиною її філософії. Ви не можете втратити інформацію при передачі чи отримати пошкоджений файл без відома Git.

## Виконання роботи

### 1. Склонуємо репозиторій



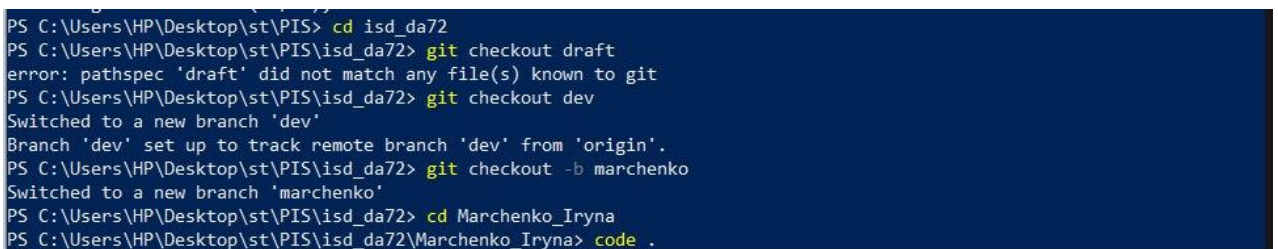
```
Windows PowerShell
(C) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

Попробуйте новую кроссплатформенную оболочку PowerShell (https://aka.ms/pscore6)

PS C:\Users\HP> cd Desktop\st\PIS
PS C:\Users\HP\Desktop\st\PIS> git clone https://github.com/ElenaGrischenko/isd_da72.git
Cloning into 'isd_da72'...
remote: Enumerating objects: 283, done.
remote: Counting objects: 100% (283/283), done.
remote: Compressing objects: 100% (226/226), done.
remote: Total 283 (delta 53), reused 239 (delta 30), pack-reused 0 eceiving objects: 91% (258/283), 23.20 MiB | 229.00
Receiving objects: 100% (283/283), 23.27 MiB | 69.00 KiB/s, done.
```

Рис.1

### 2. Перейдемо на гілку dev та створимо гілку marchenko, перейдемо у гілку marchenko



```
PS C:\Users\HP\Desktop\st\PIS> cd isd_da72
PS C:\Users\HP\Desktop\st\PIS\isd_da72> git checkout draft
error: pathspec 'draft' did not match any file(s) known to git
PS C:\Users\HP\Desktop\st\PIS\isd_da72> git checkout dev
Switched to a new branch 'dev'
Branch 'dev' set up to track remote branch 'dev' from 'origin'.
PS C:\Users\HP\Desktop\st\PIS\isd_da72> git checkout -b marchenko
Switched to a new branch 'marchenko'
PS C:\Users\HP\Desktop\st\PIS\isd_da72> cd Marchenko_Iryna
PS C:\Users\HP\Desktop\st\PIS\isd_da72\Marchenko_Iryna> code .
```

Рис.2

### 3. Відкриємо текстовий редактор, запишемо в нього назву проекту та додаю комміт

```

PS C:\Users\HP\Desktop\st\PIS\isd_da72\Marchenko_Iryna> git status
On branch marchenko
nothing to commit, working tree clean
PS C:\Users\HP\Desktop\st\PIS\isd_da72\Marchenko_Iryna> Notepad README.md
PS C:\Users\HP\Desktop\st\PIS\isd_da72\Marchenko_Iryna> git status
On branch marchenko
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")
PS C:\Users\HP\Desktop\st\PIS\isd_da72\Marchenko_Iryna> git commit -a
[marchenko b566642] feat: Add project name to README.md
 1 file changed, 1 insertion(+)
PS C:\Users\HP\Desktop\st\PIS\isd_da72\Marchenko_Iryna> git status
On branch marchenko
nothing to commit, working tree clean

```

Рис.5

4. Створюю текстовий файл під назвою project\_name.txt та індекую його:

```

PS C:\Users\HP\Desktop\st\PIS\isd_da72\Marchenko_Iryna> git status
Refresh index: 100% (125/125), done.
On branch marchenko
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

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        project_name.TXT

no changes added to commit (use "git add" and/or "git commit -a")
PS C:\Users\HP\Desktop\st\PIS\isd_da72\Marchenko_Iryna> git add .\
PS C:\Users\HP\Desktop\st\PIS\isd_da72\Marchenko_Iryna> git add .\project_name.txt
PS C:\Users\HP\Desktop\st\PIS\isd_da72\Marchenko_Iryna> git status
On branch marchenko
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   README.md
        new file:   project_name.TXT

```

Рис.6

5. Додаю гілку marchenko до репозиторію:

```

PS C:\Users\HP\Desktop\st\PIS\isd_da72\Marchenko_Iryna> git push --set-upstream origin marchenko
Enumerating objects: 13, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 4 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (9/9), 1.33 KiB | 679.00 KiB/s, done.
Total 9 (delta 2), reused 5 (delta 1), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
remote:
remote: Create a pull request for 'marchenko' on GitHub by visiting:

```

Рис.7

6. Додаємо комміт :

```

PS C:\Users\HP\Desktop\st\PIS\isd_da72\Marchenko_Iryna> git commit -m "feat: add project_name.txt"
[marchenko afa8f05] feat: add project_name.txt
 2 files changed, 1 insertion(+), 1 deletion(-)
 create mode 100644 Marchenko_Iryna/project_name.TXT
PS C:\Users\HP\Desktop\st\PIS\isd_da72\Marchenko_Iryna> git log
commit afa8f0588a9eb48e495b63a092b33bace2682719 (HEAD -> marchenko)
Author: marchenkoiryna1 <surgeonira@gmail.com>
Date:   Wed Sep 30 14:01:50 2020 +0300

```

Рис.8



## 7. Push:

```
Initial commit
PS C:\Users\HP\Desktop\st\PIS\isd_da72\Marchenko_Iryna> git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 351 bytes | 351.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/ElenaGrischenko/isd_da72.git
   b566642..afa8f05  marchenko -> marchenko
PS C:\Users\HP\Desktop\st\PIS\isd_da72\Marchenko_Iryna> git status
On branch marchenko
Your branch is up to date with 'origin/marchenko'.

It took 2.38 seconds to compute the branch ahead/behind values.
You can use '--no-ahead-behind' to avoid this.

nothing to commit, working tree clean
PS C:\Users\HP\Desktop\st\PIS\isd_da72\Marchenko_Iryna> git branch
dev
* marchenko
  master
PS C:\Users\HP\Desktop\st\PIS\isd_da72\Marchenko_Iryna>
```

Рис.9

## 8. Зробимо pull request та merge pull request :

Add project\_name.txt #30

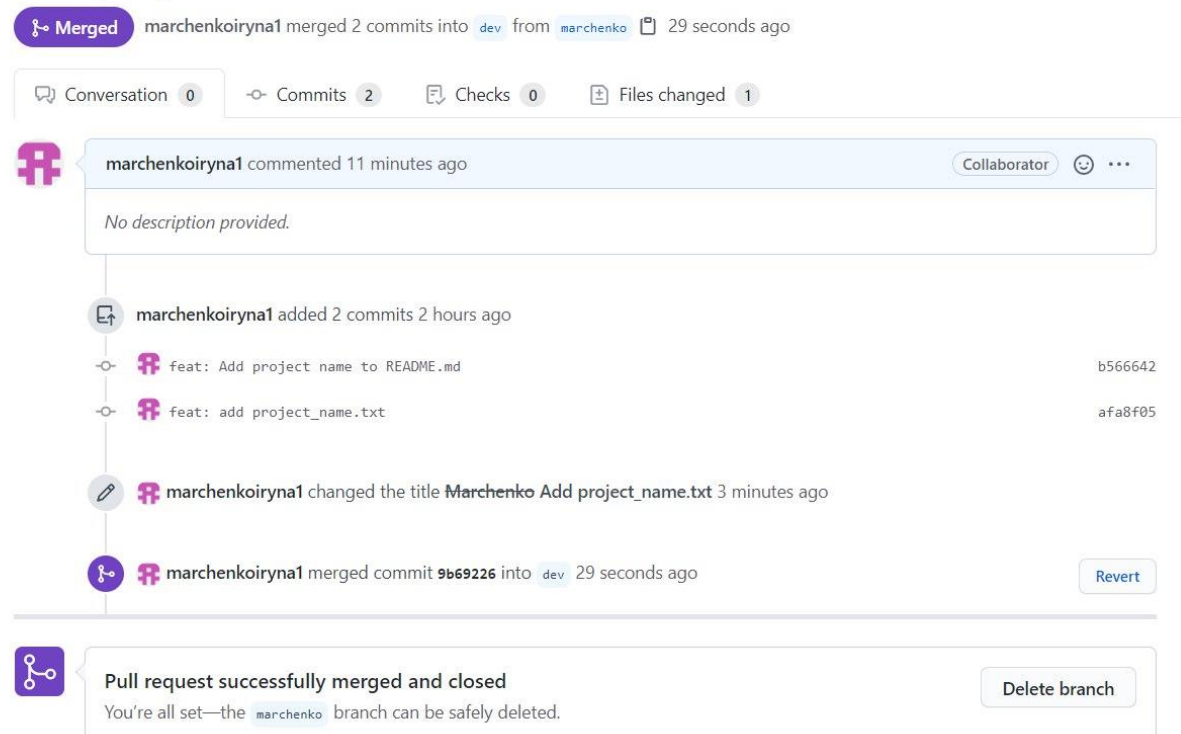


Рис.10

**Висновок:**

У даній лабораторній роботі отримала навички роботи з системою контролю версій GitHub. Склонувала локальний репозиторій, додала файл в який записала назву майбутнього проекту, змінила Readme.md, злила гілку marchenko з dev .