

**Міністерство освіти і науки України
Івано-Франківський національний технічний університет
нафти і газу**

**Кафедра Програмного Забезпечення
автоматизованих систем**

Лабораторна робота №2

Тема: Робота з системою керування версіями Git

Виконав:
студент групи ПІ-14-1
Йосипенко Михайло
Перевірила:
Бандура В. В.

**м. Івано-Франківськ
2017**

Теоретичні відомості:

GitHub — один з найбільших веб-сервісів для спільної розробки програмного забезпечення. Існують безкоштовні та платні тарифні плани користування сайтом. Базується на системі керування версіями Git і розроблений на Ruby on Rails і Erlang компанією GitHub, Inc (раніше Logical Awesome).

Сервіс безкоштовний для проектів з відкритим вихідним кодом, з наданням користувачам усіх своїх можливостей (включаючи SSL), а для окремих індивідуальних проектів пропонуються різні платні тарифні плани. 21 вересня 2011 року кількість користувачів стала більшою за мільйон.

Сайт має елементи соціальної мережі — крім розміщення коду розробники можуть спілкуватися, коментувати правки один одного, слідкувати за новинами знайомих. Також існує можливість об'єднувати різні репозиторії, в тому числі і такі, що мають різних власників.

Для проектів є персональні сторінки, Вікі та система відслідковування помилок. Прямо на сайті можна переглядати файли проектів, реалізовано підсвічування синтаксису для більшості мов програмування. На платних тарифних планах можна працювати з сервісом через SSL та показувати репозиторій тільки визначеним користувачам. Сам код проектів можна отримати не тільки через Git, а і завантажити звичайний архівний файл.

Команди git:

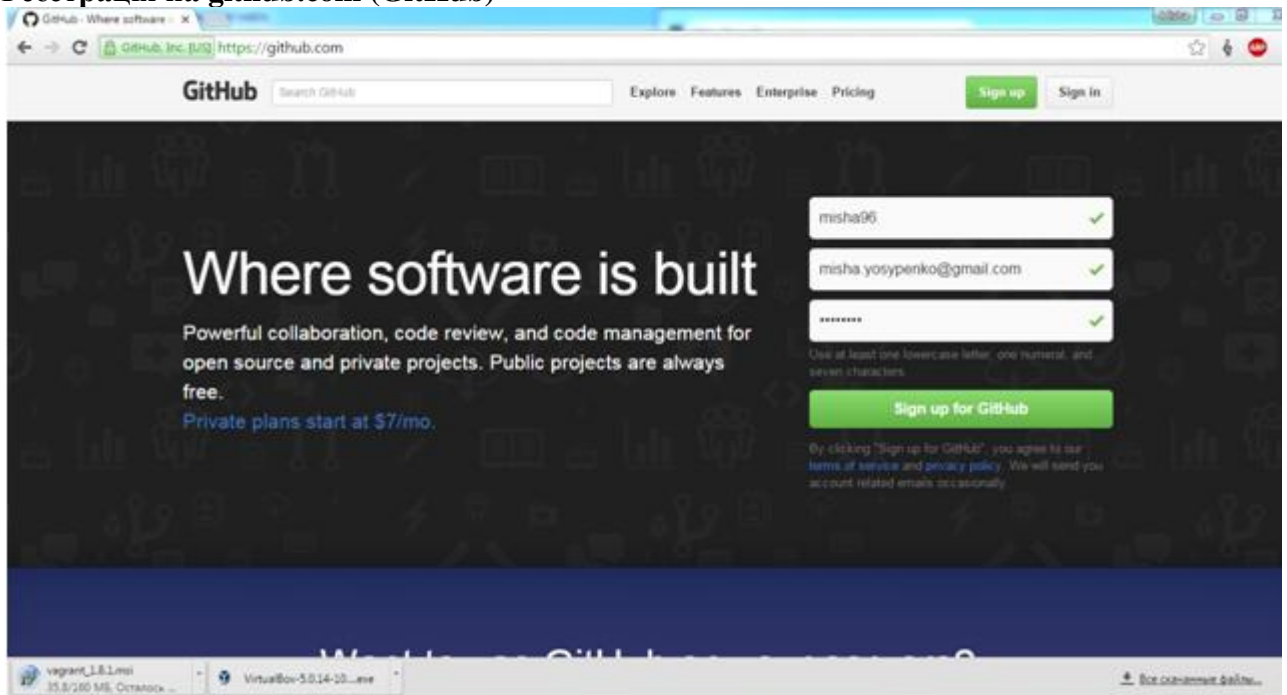
git init - створення git-репозитарія
git add <filename> - додати файл в буфер
git add . - додати усі файли в буфер
git commit -m "commit text" - коміт. відправляємо вміст буфера в репозиторій
git status
git log - найчастіша команда, показує логи (коміти)
git show <sha-1> - перегляд коміта
git rm --cached <filename> - видалення файлу із репозиторія (це приведе до того, що файл більше не буде відслідковуватися)
git commit --amend -m "commit text" - зміна останнього коміта
git revert <hash> - відміння коміт (виконує повну протилежність, те що було знищено вертає, те що було додано знищує)
git branch [-r,-a] - покаже список гілок (* - активний гілок) (-r покаже віддалені гілки, -a покаже локальні і віддалені гілки)
git branch <name_branch> - створення нового гілка
git branch --merged - покаже перелік гілок в поточному гілку
git branch -m <name_branch> <new_name_branch> - перейменування гілка
git checkout <name_branch> - переключення на гілку
git checkout -b <name_branch> - створення нового гілка та переключення на нього
git checkout -d <name_branch> - видалення гілка

Віддалені репозиторії

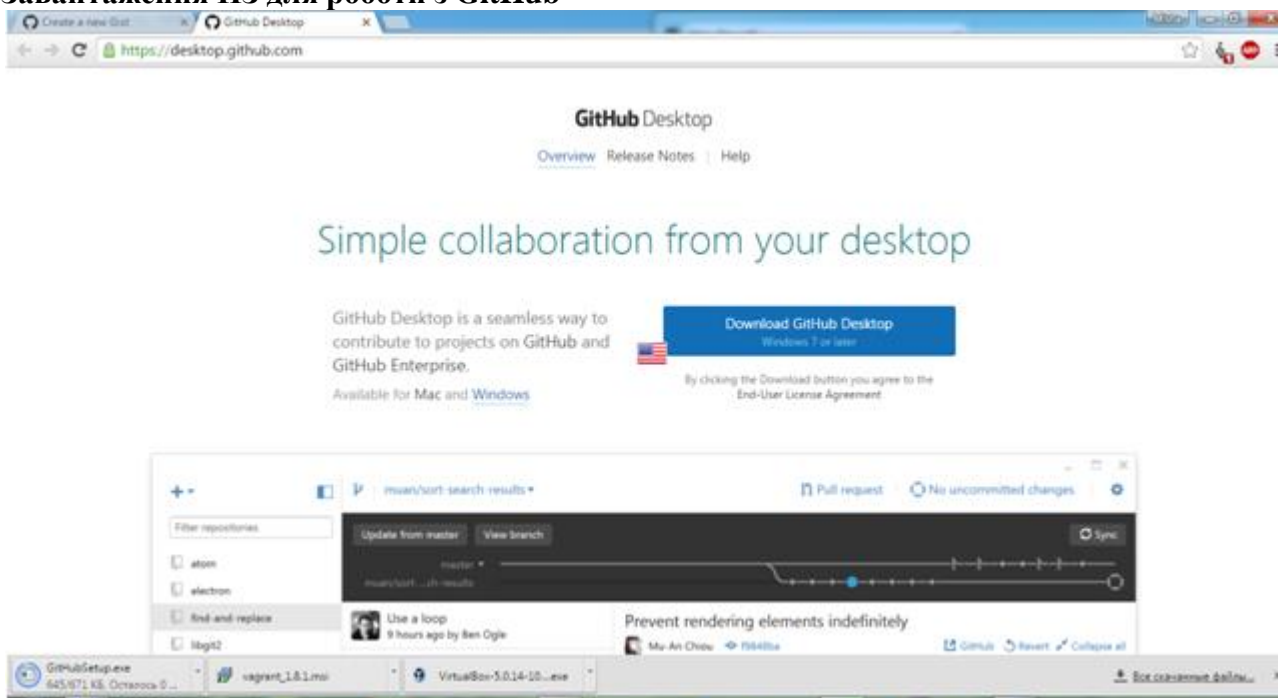
git remote [-v] - список усіх віддалених репозиторіїв, які відомі git'у
git remote add <alias> <url> - додання віддаленого серверу під назвою <alias>, який ми можемо знайти за адресою <url>
git remote rm <alias> - видалення запису про віддалений репозиторій
git push [-u] <alias> <branch_name> - відправити код на віддалений репозиторій (-u відслідковувати віддалений репозиторій)
git push - відправляє внесені зміни на віддалений репозиторій, який раніше був заданий як відслідковуваний
git push <alias> :<branch_name> - видалення гілки на віддаленому репозиторію
git push <alias> <local_branch_name>:<branch_name> - відправлення локального гілка <local_branch_name> у віддалений репозиторій <alias> у гілку <branch_name>
git push <alias> --delete <branch_name> - ще одна (більш зрозуміла команда) видалення гілки
git clone <url> [<folder>] - клонування віддаленого репозиторія

Виконання:

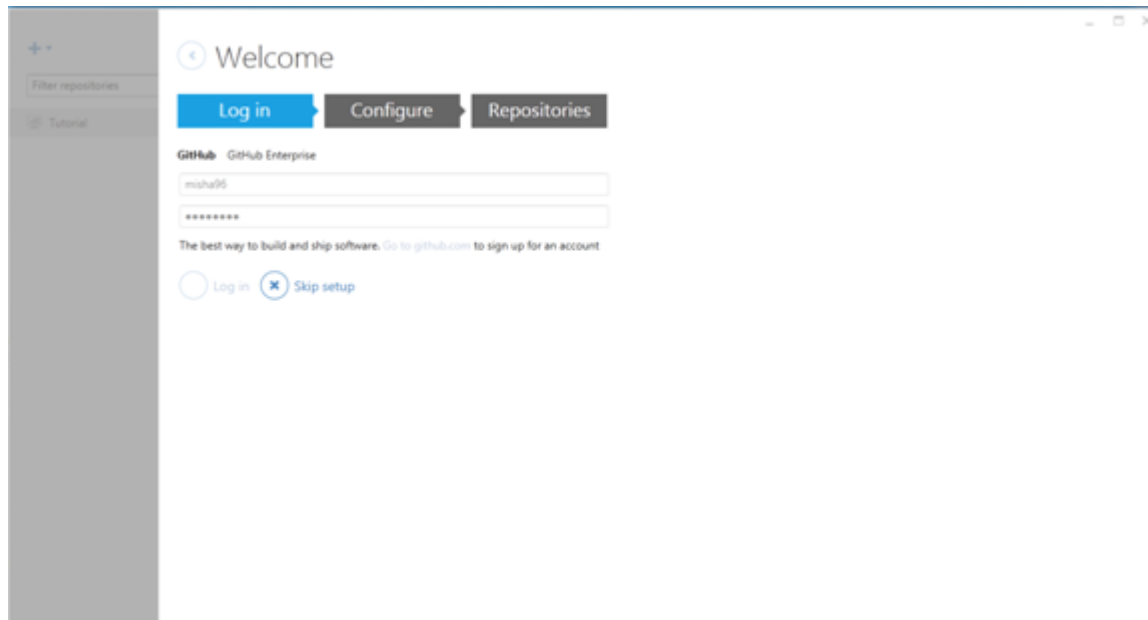
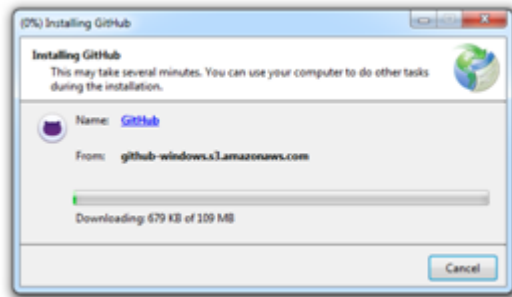
Рєєстрація на github.com (GitHub)



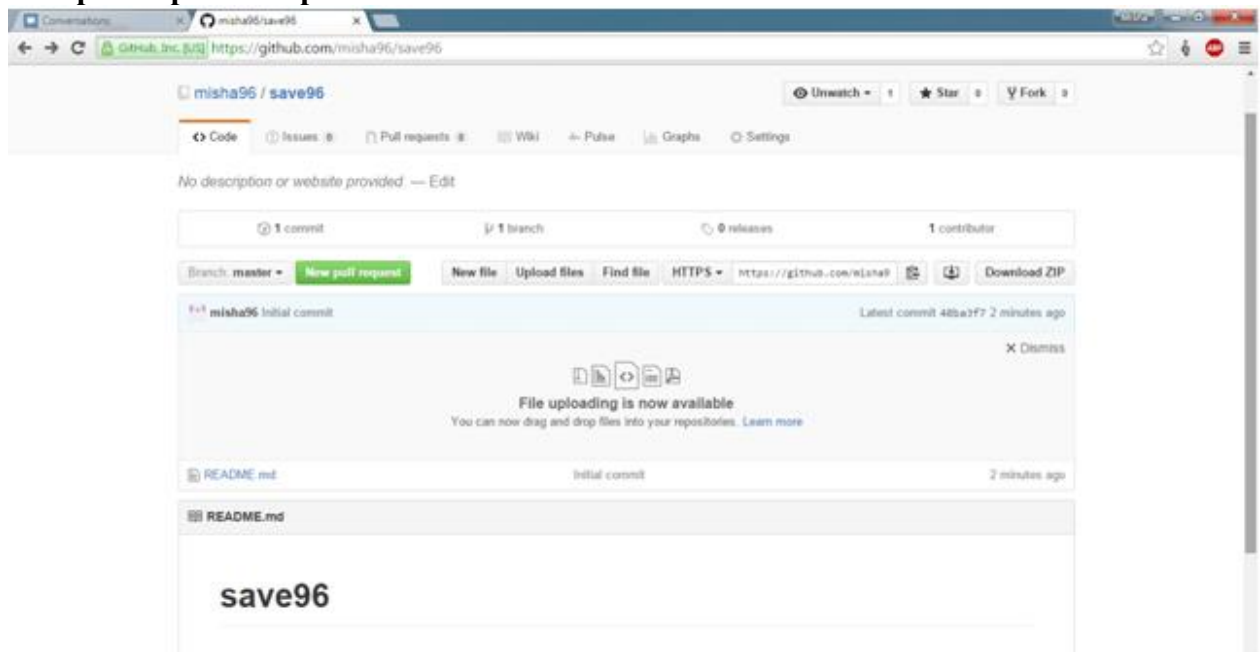
Завантаження ПЗ для роботи з GitHub



Встановлення на робочий комп'ютер



Створення репозиторія на GitHub



Створення локального репозиторія

```
posh~git ~ NewGit [master]
Windows PowerShell
(C) Корпорация Майкрософт, 2009. Все права защищены.

C:\Users\Misha\Documents\GitHub> cd NewGit
Set-Location : Не удается найти путь "C:\Users\Misha\Documents\GitHub\NewGit",
так как он не существует.
строка:1 знак:3
+ cd <<<< NewGit
    ~~~~~
    + CategoryInfo          : ObjectNotFound: (C:\Users\Misha\Documents\GitHub
    + FullyQualifiedErrorId : PathNotFound,Microsoft.PowerShell.Commands.SetLo
    cationCommand

C:\Users\Misha\Documents\GitHub> cd NewGit
C:\Users\Misha\Documents\GitHub\NewGit> git init
Initialized empty Git repository in C:/Users/Misha/Documents/GitHub/NewGit/.git/
C:\Users\Misha\Documents\GitHub\NewGit [master]>
```

Додавання файлу в локальний репозиторій

```
posh~git ~ NewGit [master]
C:\Users\Misha\Documents\GitHub\NewGit [master +1 ~0 -0 !]> git add one.txt
C:\Users\Misha\Documents\GitHub\NewGit [master +1 ~0 -0]> git status
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   one.txt

C:\Users\Misha\Documents\GitHub\NewGit [master +1 ~0 -0]> _
```

Закомічення файлу в локальному репозиторії

```
posh~git ~ NewGit [master]
C:\Users\Misha\Documents\GitHub\NewGit [master +1 ~0 -0]> git commit -m 'one'
[master (root-commit) 0a6a4b0] one
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 one.txt
C:\Users\Misha\Documents\GitHub\NewGit [master]>
```

Закидання змін з локального репозиторія в репозиторій створений в 4 пункті

```
posh~git ~ save96 [master]
C:\Users\Misha\Documents\GitHub\save96 [master +1 ~0 -0]> git commit -m 'first'
[master b02769f] first
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 101.txt
C:\Users\Misha\Documents\GitHub\save96 [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 directory clean
C:\Users\Misha\Documents\GitHub\save96 [master]> git push -u origin master
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 272 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/misha96/save96.git
 48ba3f7..b02769f  master -> master
Branch master set up to track remote branch master from origin.
C:\Users\Misha\Documents\GitHub\save96 [master]>
```

Додавання двох файлів із замічення та пушанням на GitHub

```
C:\Users\Misha\Documents\GitHub\save96 [master +2 ~0 -0 !]> git add .
C:\Users\Misha\Documents\GitHub\save96 [master +2 ~0 -0]> git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   102.txt
        new file:   103.txt

C:\Users\Misha\Documents\GitHub\save96 [master +2 ~0 -0]> git commit -m 'second'
[master 1e1f5f0] second
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 102.txt
 create mode 100644 103.txt
C:\Users\Misha\Documents\GitHub\save96 [master]> git push -u origin master
Counting objects: 2, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 270 bytes | 0 bytes/s, done.
Total 2 (delta 0), reused 0 (delta 0)
To https://github.com/misha96/save96.git
   b02769f..1e1f5f0  master -> master
Branch master set up to track remote branch master from origin.
C:\Users\Misha\Documents\GitHub\save96 [master]>
```

Відновлення до стану який був в пункті №8

```
posh~git ~ save96 [(b02769f...)]
C:\Users\Misha\Documents\GitHub\save96 [master]> git checkout b027
Note: checking out 'b027'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -b with the checkout command again. Example:

    git checkout -b <new-branch-name>

HEAD is now at b02769f... first
C:\Users\Misha\Documents\GitHub\save96 [(b02769f...)]>
```

Створення клона репозиторія з GitHub в іншій папці

```
C:\Users\Misha\Documents\GitHub> cd save966
C:\Users\Misha\Documents\GitHub\save966 [master]> git clone https://github.com/misha96/save96.git
Cloning into 'save96'...
remote: Counting objects: 8, done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 8 (delta 1), reused 4 (delta 0), pack-reused 0
Unpacking objects: 100% (8/8), done.
Checking connectivity... done.
C:\Users\Misha\Documents\GitHub\save966 [master +1 ~0 -0 !]>
```

Створення нової гілки і додавання в неї нового файлу

```
C:\Users\Misha\Documents\GitHub\save966 [master +1 ~0 -0 !]> git checkout -b tree
Switched to a new branch 'tree'
C:\Users\Misha\Documents\GitHub\save966 [tree +1 ~0 -0 !]> git add 105.txt
fatal: pathspec '105.txt' did not match any files
C:\Users\Misha\Documents\GitHub\save966 [tree +1 ~0 -0 !]> git add .
C:\Users\Misha\Documents\GitHub\save966 [tree +1 ~0 -0]>
```

Висновок: на цій лабораторній роботі я навчився працювати з системою керування версіями Git, також вивчив основні команди git та навчився ними користуватися.