

Facultatea Calculatoare, Informatica si Microelectronica

Universitatea Tehnica a Moldovei

Medii Interactive de Dezvoltare a Produselor Soft

Lucrarea de laborator#1

Version Control Systems si modul de setare a unui server

Autor:

Olga Golentova

TI-155

lector asistent:

Victor Gojin

lector superior:

Radu Melnic

Laboratory work #1

1 Цель лабораторной работы

Изучить использование системы контроля версий и способ установки сервера.

2 Задачи

Изучение системы контроля версий (git)

3 Выполнение лабораторной работы

3.1 Задания

Basic Level (nota 5 || 6):

- o initializeaza un nou repository
- o configureaza-ti VCS
- o crearea branch-urilor (creeaza cel putin 2 branches)
- o commit pe ambele branch-uri (cel putin 1 commit per branch)

Normal Level (nota 7 || 8):

- o seteaza un branch to track a remote origin pe care vei putea sa faci push (ex. Github, Bitbucket or custom server)
- o reseteaza un branch la commit-ul anterior
- o salvarea temporara a schimbarilor care nu se vor face commit imediat.
- o folosirea fisierului .gitignore

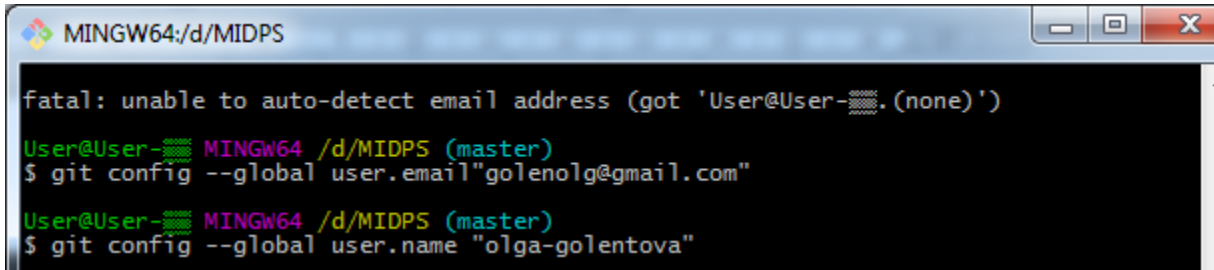
Advanced Level (nota 9 || 10):

- o merge 2 branches
- o rezolvarea conflictelor a 2 branches

Анализ лабораторной работы

Ссылка на репозиторий: <https://github.com/olga-golentova/MIDPS/>

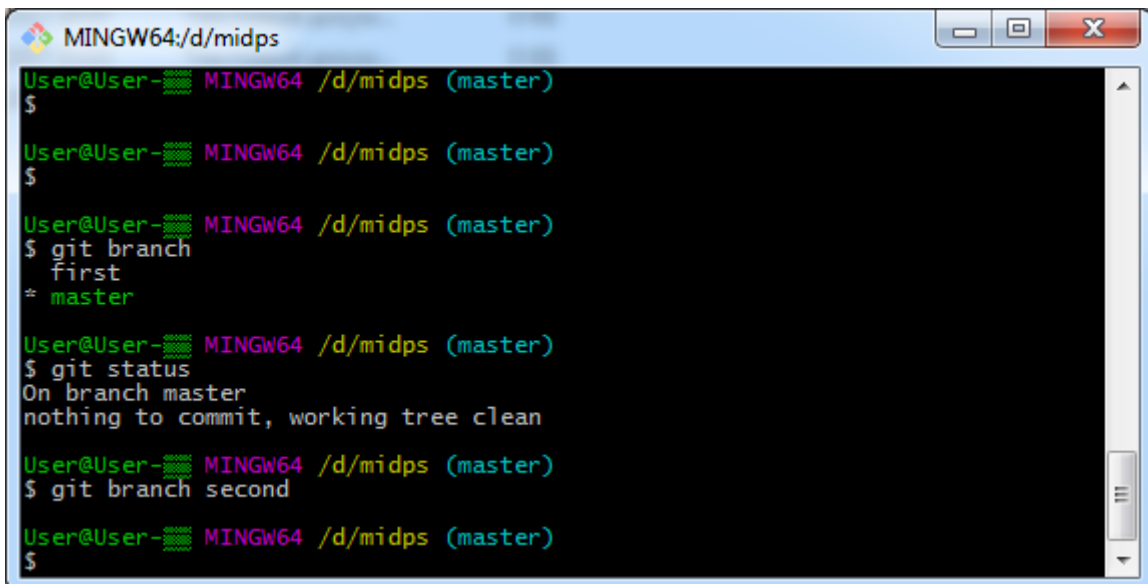
Установка имени и электронной почты

A terminal window titled 'MINGW64:/d/MIDPS' showing the process of setting up git configuration. It starts with an error message about not being able to auto-detect an email address. Then, the user runs 'git config --global user.email "golentol@gmail.com"' and 'git config --global user.name "olga-golentova"'.

```
MINGW64:/d/MIDPS
fatal: unable to auto-detect email address (got 'User@User-...(.none)')
User@User-... MINGW64 /d/MIDPS (master)
$ git config --global user.email "golentol@gmail.com"
User@User-... MINGW64 /d/MIDPS (master)
$ git config --global user.name "olga-golentova"
```

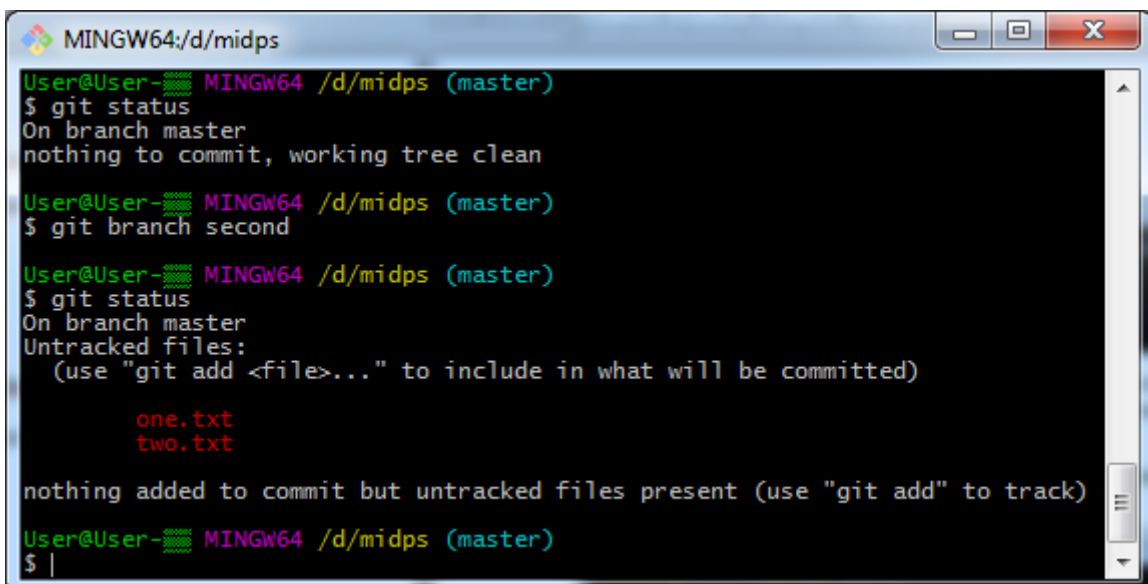
Команда `git status` – определяем какие файлы в каком состоянии находятся и на какой ветке `git branch` - список наших веток (я ранее создала `first`) зеленым цветом выделена ветка, на которой мы находимся в данный момент

Для создания новой ветки используем команды `git branch <имя новой ветки>`
создаю ветку `second`

A terminal window titled 'MINGW64:/d/midps' showing the creation of a new branch and checking the status. The user creates a branch named 'first', then 'second', and checks the status, which shows the working tree is clean.

```
MINGW64:/d/midps
User@User-... MINGW64 /d/midps (master)
$
User@User-... MINGW64 /d/midps (master)
$
User@User-... MINGW64 /d/midps (master)
$ git branch
first
* master
User@User-... MINGW64 /d/midps (master)
$ git status
On branch master
nothing to commit, working tree clean
User@User-... MINGW64 /d/midps (master)
$ git branch second
User@User-... MINGW64 /d/midps (master)
$
```

В папку на моем компьютере (`git init` использовала ранее) добавляю два новых текстовых файла, `one.txt` и `two.txt`, они пока пустые. Проверяем состояние

A terminal window titled 'MINGW64:/d/midps' showing the status of the repository after adding two new files. The status shows that the files are untracked and need to be added to the commit.

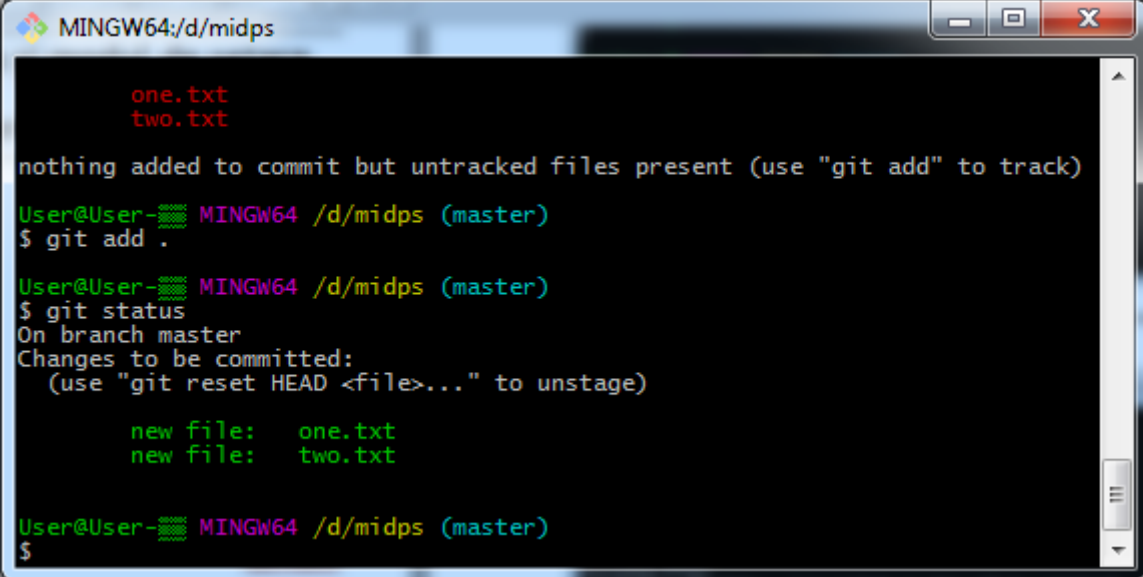
```
MINGW64:/d/midps
User@User-... MINGW64 /d/midps (master)
$ git status
On branch master
nothing to commit, working tree clean
User@User-... MINGW64 /d/midps (master)
$ git branch second
User@User-... MINGW64 /d/midps (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        one.txt
        two.txt

nothing added to commit but untracked files present (use "git add" to track)
User@User-... MINGW64 /d/midps (master)
$ |
```

Два не отслеживаемых файла, которые мы создали. Используем команду `add .` (добавить все untracked файлы). Файлы отслеживаются и ожидается коммит изменений.

Добавлю изменение в текстовый файл `one.txt`



```
MINGW64:/d/midps

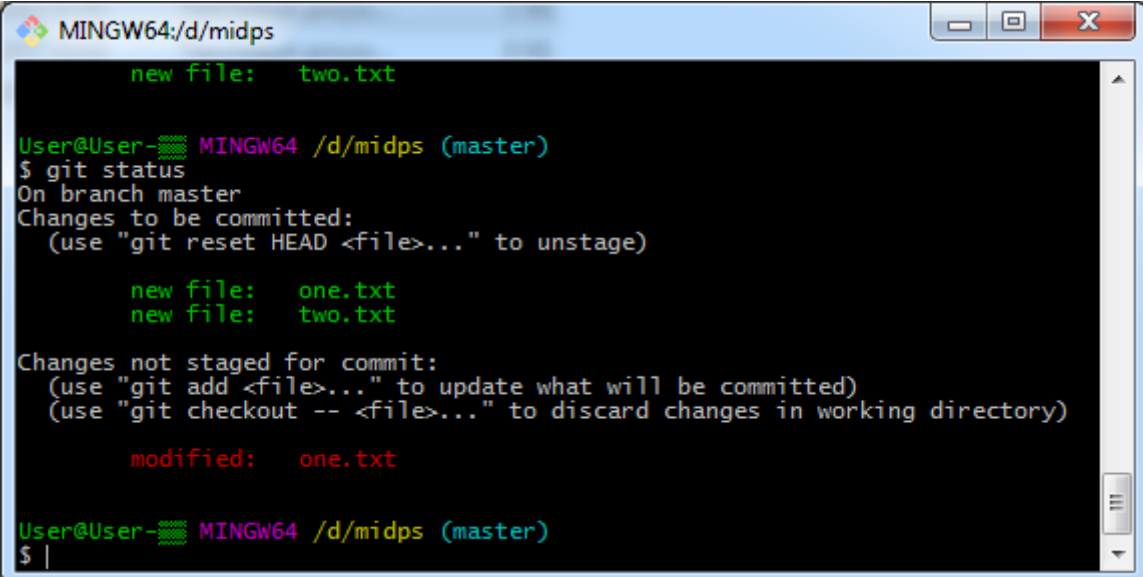
one.txt
two.txt

nothing added to commit but untracked files present (use "git add" to track)
User@User- MINGW64 /d/midps (master)
$ git add .
User@User- MINGW64 /d/midps (master)
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   one.txt
        new file:   two.txt

User@User- MINGW64 /d/midps (master)
$
```

Теперь измененный файл не проиндексирован и нам необходимо снова делать `git add` чтобы проиндексировать последнюю версию файла, иначе, если мы сделаем сейчас коммит, то файл попадет в коммит в том состоянии, в котором он был до последней команды `git add`



```
MINGW64:/d/midps

        new file:   two.txt

User@User- MINGW64 /d/midps (master)
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   one.txt
        new file:   two.txt

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   one.txt

User@User- MINGW64 /d/midps (master)
$ |
```

`git add` и коммитим

```
MINGW64:/d/midps
$ git add one.txt
User@User-MINGW64 /d/midps (master)
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   one.txt
        new file:   two.txt

User@User-MINGW64 /d/midps (master)
$ git commit -m "first commit for lab.1"
[master 6441d9c] first commit for lab.1
2 files changed, 1 insertion(+)
create mode 100644 one.txt
create mode 100644 two.txt

User@User-MINGW64 /d/midps (master)
$
```

Сменили ветку, создали файл и закоммитили

```
MINGW64:/d/midps
User@User-MINGW64 /d/midps (master)
$ git checkout second
Switched to branch 'second'

User@User-MINGW64 /d/midps (second)
$ git branch
  first
* master
  second

User@User-MINGW64 /d/midps (second)
$ git status
On branch second
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        one.txt

nothing added to commit but untracked files present (use "git add" to track)

User@User-MINGW64 /d/midps (second)
$ git add .

User@User-MINGW64 /d/midps (second)
$ git commit -m "creating file one.txt on second"
[second 798aa73] creating file one.txt on second
1 file changed, 1 insertion(+)
create mode 100644 one.txt

User@User-MINGW64 /d/midps (second)
$
```

Чтобы добавить новый удалённый git-репозиторий под именем-сокращением, к которому будет проще обращаться, выполните `git remote add [сокращение] [url]`:

посмотреть список существующих – `git remote -v` . Вместо полного URL –а я могу использовать сокращенное название `midps`

`fetch` – извлекаем всю информацию, что есть в репозитории `midps` но нет в моем

`git log` - информация о коммитах

```
MINGW64:/d/midps
User@User-MINGW64 /d/midps (second)
$ git remove -v
git: 'remove' is not a git command. See 'git --help'.

Did you mean this?
    remote

User@User-MINGW64 /d/midps (second)
$ git remote -v
midps  https://github.com/olga-golentova/MIDPS.git (fetch)
midps  https://github.com/olga-golentova/MIDPS.git (push)

User@User-MINGW64 /d/midps (second)
$ git fetch midps

User@User-MINGW64 /d/midps (second)
$ git log
commit 798aa730024ded8467e98c1b7bda7e3eaace0c4d
Author: olga-golentova <goleno1g@gmail.com>
Date:   Mon Mar 6 11:30:07 2017 +0200

    creating file one.txt on second

commit 0ca398dd6fe53f3359866f7edcc59ad40f98f1d5
Merge: f04983e 3601734
Author: olga-golentova <goleno1g@gmail.com>
Date:   Wed Mar 1 15:53:09 2017 +0200

    Merge branch 'master' of https://github.com/olga-golentova/MIDPS

commit f04983ec7ef10a69606ceb790cf16cce6d99d70e
Author: olga-golentova <goleno1g@gmail.com>
Date:   Wed Mar 1 15:39:49 2017 +0200

    ver.1_0

commit 36017348573cb42bde6805c5c8eafa777ba9eee6
Author: olga-golentova <goleno1g@gmail.com>
Date:   Mon Feb 13 10:35:25 2017 +0200

    i get started using git

User@User-MINGW64 /d/midps (second)
$ |
```

push – закидываем все изменения на удалённый репозиторий. Pull – извлекает с удаленного репозитория информацию и сливает с имеющейся у меня

```
MINGW64:/d/midps

git push <name>

User@User-MINGW64 /d/midps (master)
$ git push midps master

Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 318 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
o https://github.com/olga-golentova/MIDPS.git
0ca398d..6441d9c master -> master

User@User-MINGW64 /d/midps (master)
$

User@User-MINGW64 /d/midps (master)
$ git pull midps master
From https://github.com/olga-golentova/MIDPS
* branch master -> FETCH_HEAD
Already up-to-date.

User@User-MINGW64 /d/midps (master)
$ |
```

выполнила команду push с опцией --all, то, независимо от того, какая ветка была в этот момент текущей, программа git попыталась создать все недостающие в удалённом репозитории ветки и отправить в них локальные изменения

```
MINGW64:/d/midps

ing

git remote add <name> <url>

and then push using the remote name

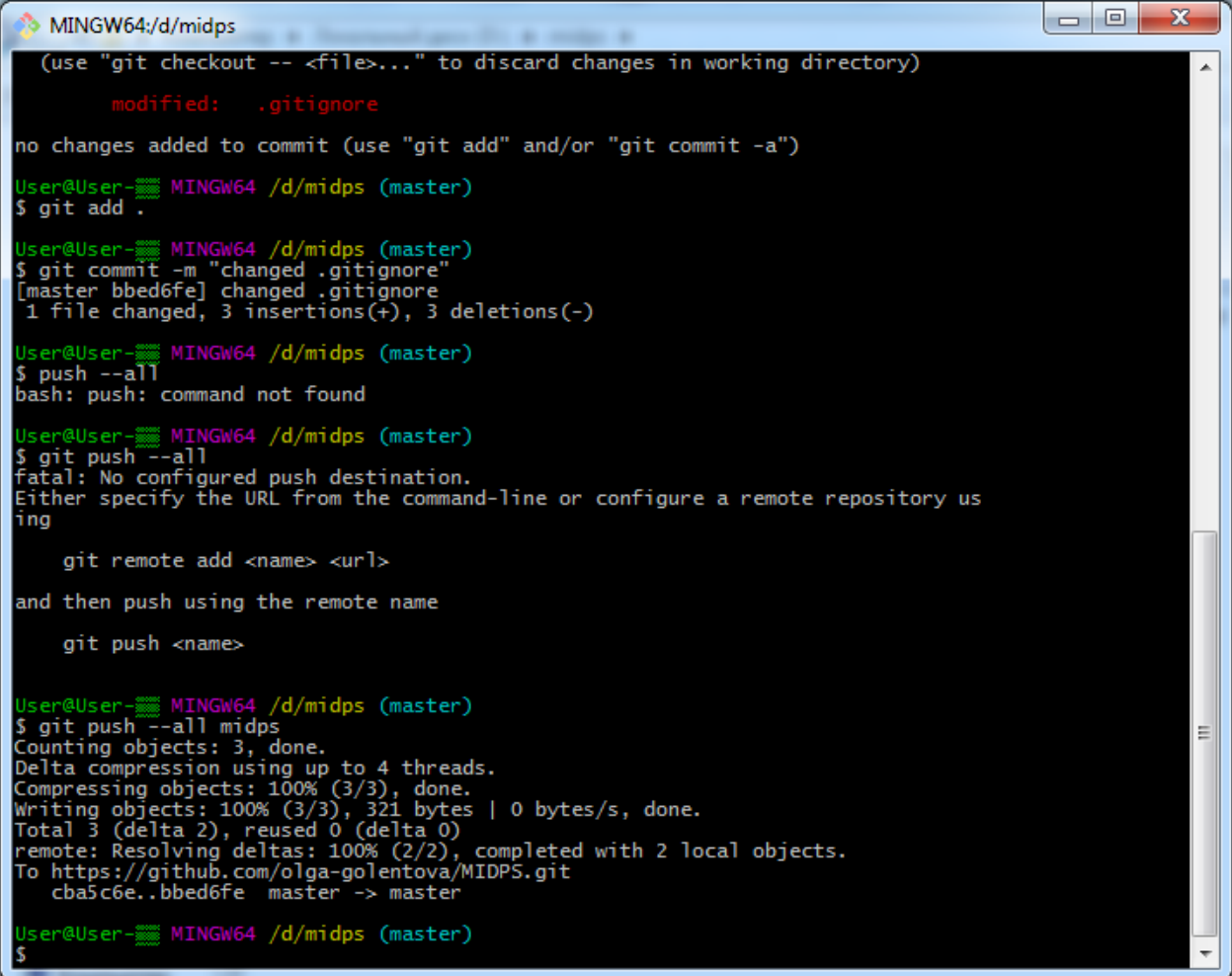
git push <name>

User@User-MINGW64 /d/midps (master)
$ git push --all midps
Counting objects: 11, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (10/10), done.
Writing objects: 100% (11/11), 7.27 KiB | 0 bytes/s, done.
Total 11 (delta 5), reused 0 (delta 0)
remote: Resolving deltas: 100% (5/5), completed with 2 local objects.
To https://github.com/olga-golentova/MIDPS.git
515f2ff..cba5c6e master -> master
* [new branch] first -> first
* [new branch] second -> second

User@User-MINGW64 /d/midps (master)
$
```

Создан файл .gitignore, в котором прописывается игнорируемые файлы. К таким файлам обычно относятся автоматически генерируемые файлы (различные логи, результаты сборки программ и

т.п.).



```
(use "git checkout -- <file>..." to discard changes in working directory)

        modified:   .gitignore

no changes added to commit (use "git add" and/or "git commit -a")
User@User-MINGW64 /d/midps (master)
$ git add .
User@User-MINGW64 /d/midps (master)
$ git commit -m "changed .gitignore"
[master bbed6fe] changed .gitignore
1 file changed, 3 insertions(+), 3 deletions(-)
User@User-MINGW64 /d/midps (master)
$ push --all
bash: push: command not found
User@User-MINGW64 /d/midps (master)
$ git push --all
fatal: No configured push destination.
Either specify the URL from the command-line or configure a remote repository using

    git remote add <name> <url>

and then push using the remote name

    git push <name>

User@User-MINGW64 /d/midps (master)
$ git push --all midps
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 321 bytes | 0 bytes/s, done.
Total 3 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/olga-golentova/MIDPS.git
   cba5c6e..bbed6fe  master -> master
User@User-MINGW64 /d/midps (master)
$
```

Вывод:

В процессе выполнения данной лабораторной работы были изучены основные принципы работы с сервисом Git . Освоены основные команды, проведена работа с ветвями. В ходе лабораторной работы возникали ошибки, с коими удалось успешно справиться при помощи официальной документации Git. Git- один из важнейших инструментов разработчика , а тем более – группы разработчиков, позволяющий вести совместную работу над проектом, хранение множества версий проекта и самое главное – возврат к более ранним версиям, отслеживание изменений и пользователей, которыми были совершены изменения.