

**МИНОБРНАУКИ РОССИИ**  
**федеральное государственное бюджетное образовательное учреждение**  
**высшего образования**  
**«Московский государственный технологический университет «СТАНКИН»**  
**(ФГБОУ ВО «МГТУ «СТАНКИН»)**

---

Институт цифровых  
интеллектуальных систем

Кафедра  
компьютерных систем управления

Дисциплина «Основы системного программного обеспечения»

**Отчет по лабораторной работе № 1**

**Выполнил**  
**студент гр. АДБ-20-6:**

\_\_\_\_\_

(дата)

\_\_\_\_\_

(подпись)

**Солдатов А.Ю.**

**Проверил**  
**к.т.н., доцент**

\_\_\_\_\_

(дата)

\_\_\_\_\_

(подпись)

**Ковалев И.А.**

**Москва 2023 г.**

## Содержание

1. Цель работы:.....	3
2. Основные задачи работы: .....	3
3. Описание и выполнение тестовых (если присутствуют) и индивидуальных заданий .....	3

## 1. Цель работы:

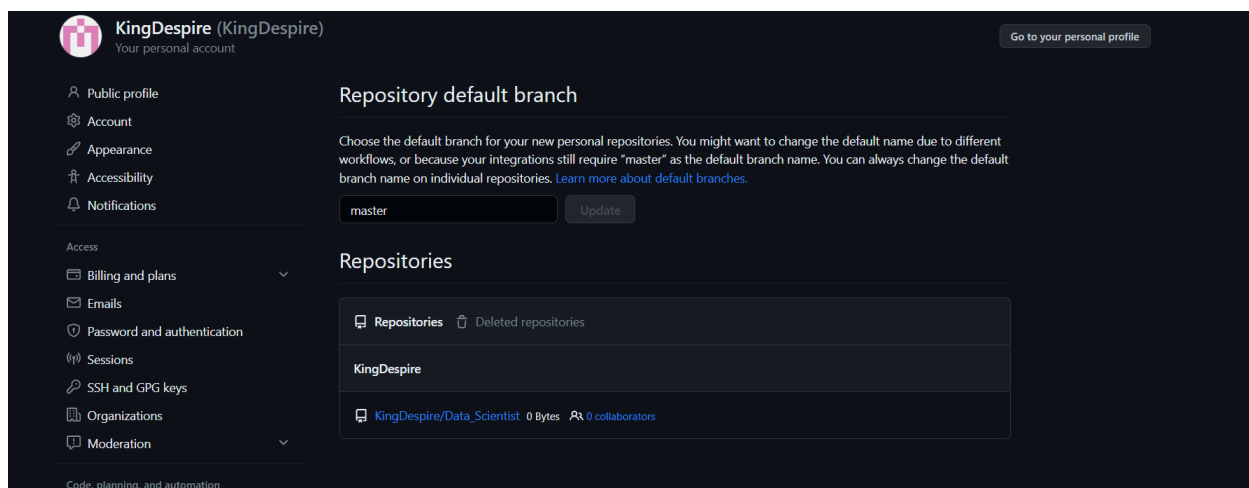
Работа с системами контроля версий на примере Git Hub

## 2. Основные задачи работы:

- Создать учетную запись на github.com
- Создать локальный репозиторий
- Зафиксировать изменения в области заготовленных файлов
- Переслать локальный коммит на сервер
- Сделать слияние веток
- Просмотреть изменения и разрешение конфликтов
- Удалить ветку на сервере и вернуть к предыдущему состоянию
- Исправить коммит
- Совместная работа с git

## 3. Описание и выполнение тестовых (если присутствуют) и индивидуальных заданий

Поскольку самый первый пункт-создание учётной записи на git'e, а у меня она имеется, войдём в учётную запись github. Затем изменим название главной ветки на master.



Создадим новый репозиторий соответственно номеру лабораторной работы.

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner \*



KingDespire

Repository name \*

lr1

✓ lr1 is available.

Great repository names are short and memorable. Need inspiration? How about **ideal-spork**?

Description (optional)



**Public**

Anyone on the internet can see this repository. You choose who can commit.



**Private**

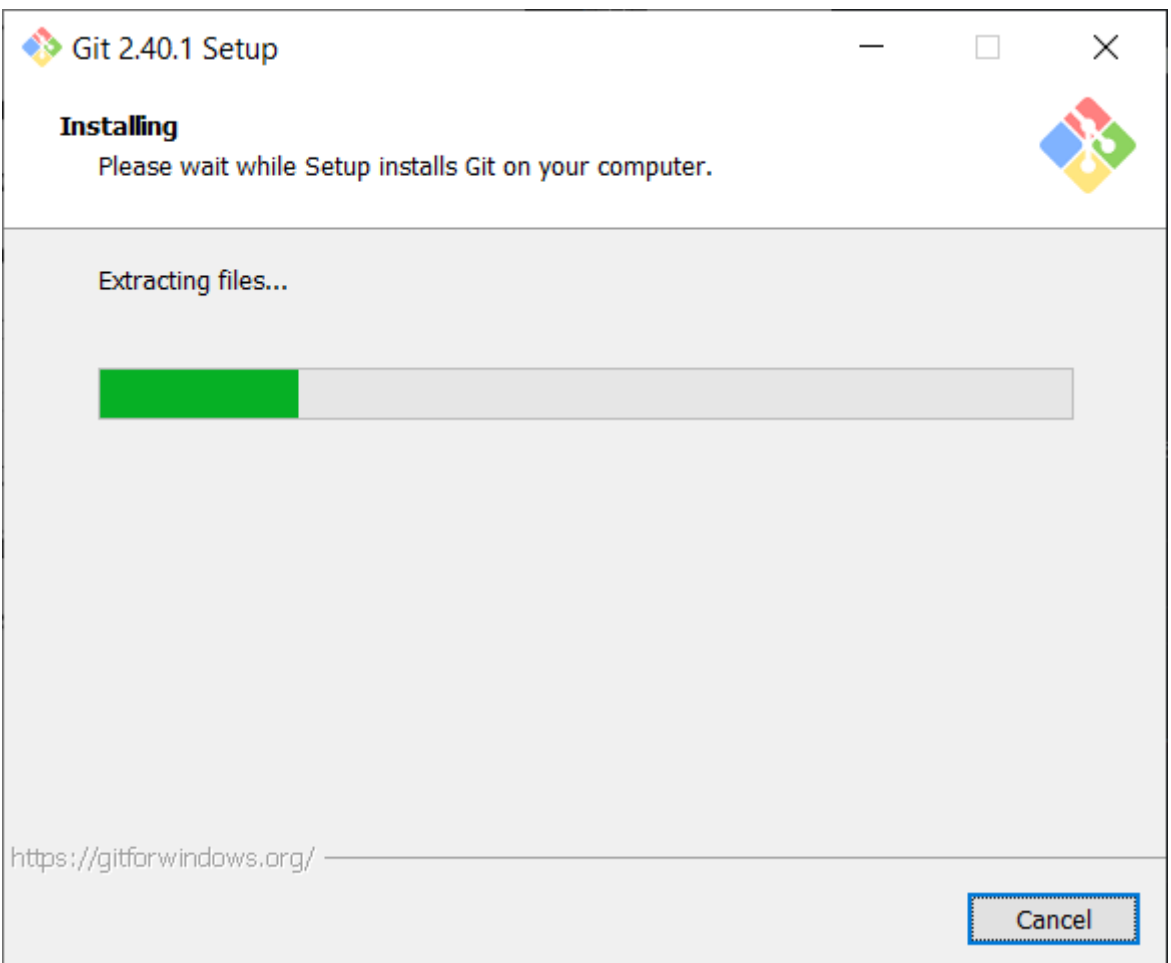
You choose who can see and commit to this repository.

Initialize this repository with:



Add a README file

Установим git на ПК



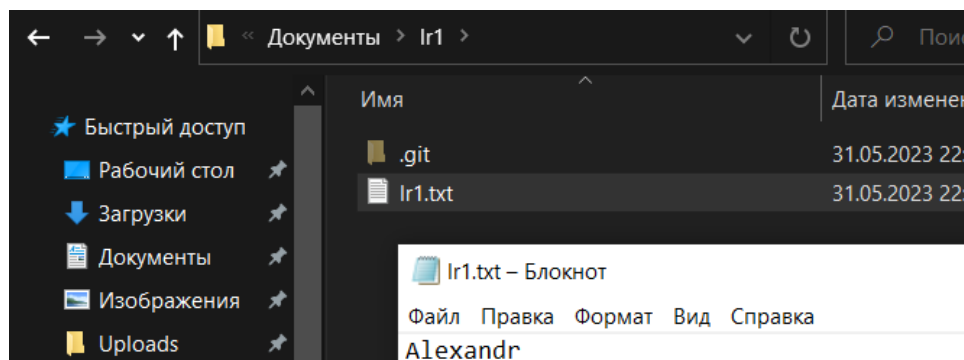
Создадим локальный каталог. Назвали его lr1. Работать будем в powershell. При помощи команды cd перейдём в наш каталог и выполним настройки имени и почты.

```
PS C:\Users\Alexandr\Documents> cd lr1
PS C:\Users\Alexandr\Documents\lr1> git config --global user.name KingDes
pire
PS C:\Users\Alexandr\Documents\lr1> git config --global user.email soldat
ov392@bk.ru
```

Инициализируем эту папку как git репозиторий.

```
PS C:\Users\Alexandr\Documents\lr1> git init
Initialized empty Git repository in C:/Users/Alexandr/Documents/lr1/.git/
```

Создадим в папке текстовый файл и пропишем в него имя



Введем команду git status. Получим информацию о наличии нового файла. После этого объявим его специальным образом. После добавления файла в репозиторий, проверим его статус.

```
PS C:\Users\Alexandr\Documents\lr1> git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  lr1.txt

nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\Alexandr\Documents\lr1> git add .
PS C:\Users\Alexandr\Documents\lr1> git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
  new file:   lr1.txt
```

Зафиксируем изменения

```
PS C:\Users\Alexandr\Documents\lr1> git commit -m "first commit"
[master (root-commit) 96ed7fd] first commit
1 file changed, 1 insertion(+)
create mode 100644 lr1.txt
PS C:\Users\Alexandr\Documents\lr1> git status
On branch master
nothing to commit, working tree clean
```

Отправим коммит на сервер

```
PS C:\Users\Alexandr\Documents\lr1> git remote add origin https://github.com/KingDespire/lr1
PS C:\Users\Alexandr\Documents\lr1> git remote -v
origin https://github.com/KingDespire/lr1 (fetch)
origin https://github.com/KingDespire/lr1 (push)
```

```
PS C:\Users\Alexandr\Documents\lr1> git pull origin master --allow-unrelated-histories
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 588 bytes | 53.00 KiB/s, done.
From https://github.com/KingDespire/lr1
* branch master -> FETCH_HEAD
* [new branch] master -> origin/master
Merge made by the 'ort' strategy.
 README.md | 1 +
1 file changed, 1 insertion(+)
create mode 100644 README.md
```

Запросим изменения с сервера

В нашем случае у нас есть файл и на сервере и локальном репозитории, поэтому нам вначале надо получить все данные с нашего репозитория, а потом уже их заливать на сервер нашими локальными данными.

```
PS C:\Users\Alexandr\Documents\lr1> git pull origin master
From https://github.com/KingDespire/lr1
* branch master -> FETCH_HEAD
Already up to date.
PS C:\Users\Alexandr\Documents\lr1> _
```

Перешлём локальный коммит на сервер

Теперь отправим коммит на сервер, команда, предназначенная для этого — push. Она принимает два параметра: имя удаленного репозитория (мы назвали наш origin) и ветку, в которую необходимо внести изменения (master — это ветка по умолчанию для всех репозиториях). git push origin master.

```
PS C:\Users\Alexandr\Documents\lr1> git push origin master
info: please complete authentication in your browser...
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (5/5), 509 bytes | 509.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/KingDespire/lr1
c44b785..3db7322 master -> master
PS C:\Users\Alexandr\Documents\lr1>
```

Посмотрим наши изменения с использованием команды git log Видны коммиты по изменениям в файле Readme.md и в файле с именем (first commit)

```

PS C:\Users\Alexandr\Documents\lr1> git log
commit 3db7322a84461d9b6b71000a5308817dae950df5 (HEAD -> master, origin/master)
Merge: 96ed7fd c44b785
Author: KingDespire <soldatov392@bk.ru>
Date:   Wed May 31 22:32:18 2023 +0300

    Merge branch 'master' of https://github.com/KingDespire/lr1





commit 96ed7fde518d4df6fc8ae9b1610795d0a3e0875e
Author: KingDespire <soldatov392@bk.ru>
Date:   Wed May 31 22:25:33 2023 +0300

    first commit

commit c44b7854165e1d1e769cca20cfbb64fa3ad711ab
Author: KingDespire <103002280+KingDespire@users.noreply.github.com>
Date:   Wed May 31 22:05:39 2023 +0300

    Initial commit
PS C:\Users\Alexandr\Documents\lr1>

```

	KingDespire Merge branch 'master' of <a href="https://github.com/KingDespire/lr1">https://github.com/KingDespire/lr1</a>	3db7322 9 minutes ago	 3 commits
	README.md	Initial commit	36 minutes ago
	lr1.txt	first commit	16 minutes ago

### Создание новой ветки

Во время разработки новой функциональности считается хорошей практикой работать с копией оригинального проекта, которую называют веткой. Ветви имеют свою собственную историю и изолированные друг от друга изменения до тех пор, пока вы не решаете слить изменения вместе. Это происходит по набору причин:

- Уже рабочая, стабильная версия кода сохраняется.
- Различные новые функции могут разрабатываться параллельно разными программистами.
- Разработчики могут работать с собственными ветками без риска, что кодовая база поменяется из-за чужих изменений.
- В случае сомнений, различные реализации одной и той же идеи могут быть разработаны в разных ветках и затем сравниваться.

Обычно в проекте создается минимум 2 ветки (branch) – master и ветка для разработки. Ветка master создается сам, а дополнительные ветки создаются по желанию пользователя (локально), заливаются они в удаленный репозиторий уже с учетом доступных прав.

Создадим новую ветку second, затем переключимся на неё.

```

PS C:\Users\Alexandr\Documents\lr1> git branch second
PS C:\Users\Alexandr\Documents\lr1> git branch
* master
  second
PS C:\Users\Alexandr\Documents\lr1> git checkout second
Switched to branch 'second'
PS C:\Users\Alexandr\Documents\lr1> git branch
  master
* second
PS C:\Users\Alexandr\Documents\lr1> _

```

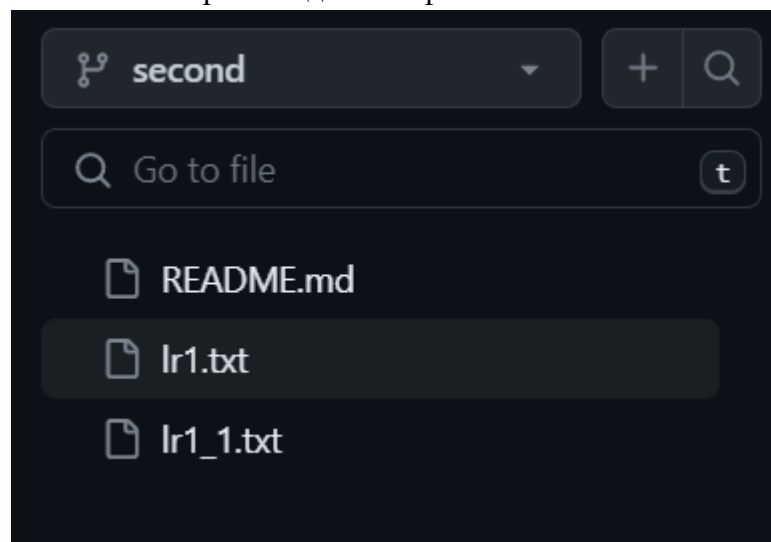
Создадим в ветке second новый файл и напишем в нём фамилию, добавим в область подготовленных файлов, закоммитим и отправим на сервер.

```

PS C:\Users\Alexandr\Documents\lr1> git push origin second
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'second' on GitHub by visiting:
remote:   https://github.com/KingDespire/lr1/pull/new/second
remote:
To https://github.com/KingDespire/lr1
 * [new branch]      second -> second

```

Посмотрим созданный файл в ветке second.



Переключимся на ветку master. Затем выполним слияние веток master с веткой second, а после удалим ветку second.

```

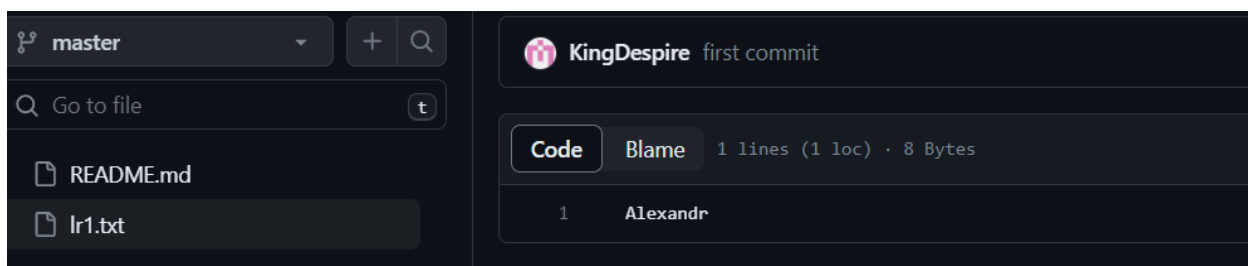
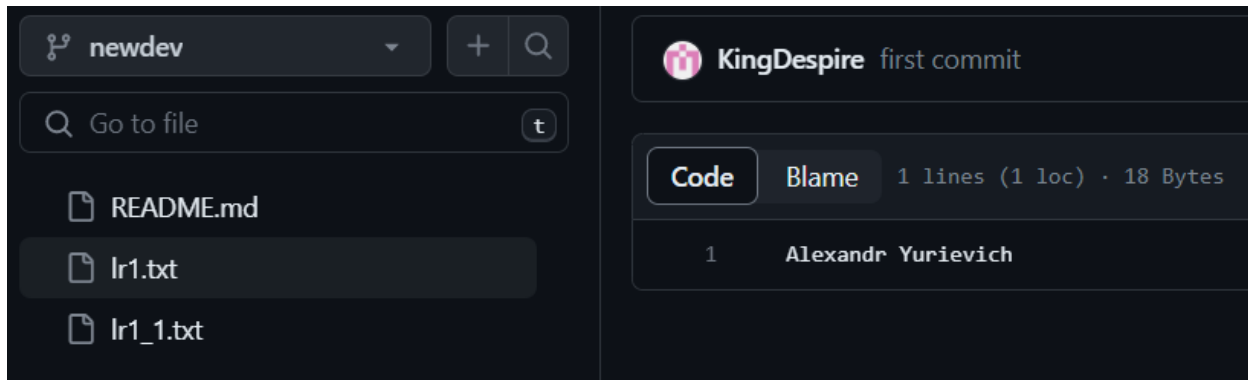
PS C:\Users\Alexandr\Documents\lr1> git checkout master
Switched to branch 'master'
PS C:\Users\Alexandr\Documents\lr1> git merge second
Updating 3db7322..ef349bd
Fast-forward
 lr1_1.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 lr1_1.txt
PS C:\Users\Alexandr\Documents\lr1> git branch -d second
Deleted branch second (was ef349bd).
PS C:\Users\Alexandr\Documents\lr1> _

```



Создадим ветку с названием newdev, переключимся на ней, добавим в файл с именем и отчеством, зафиксируем изменения, закомитим изменения.

```
PS C:\Users\Alexandr\Documents\lr1> git branch newdev
PS C:\Users\Alexandr\Documents\lr1> git checkout newdev
Switched to branch 'newdev'
PS C:\Users\Alexandr\Documents\lr1> git add .
PS C:\Users\Alexandr\Documents\lr1> git commit -m "first commit"
[newdev 1f75b0b] first commit
1 file changed, 1 insertion(+), 1 deletion(-)
PS C:\Users\Alexandr\Documents\lr1> git push origin newdev
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 318 bytes | 318.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'newdev' on GitHub by visiting:
remote: https://github.com/KingDespire/lr1/pull/new/newdev
remote:
To https://github.com/KingDespire/lr1
 * [new branch]      newdev -> newdev
```



Объединим ветки

```
PS C:\Users\Alexandr\Documents\lr1> git merge newdev
Updating ef349bd..1f75b0b
Fast-forward
 lr1.txt | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)
PS C:\Users\Alexandr\Documents\lr1>
```

Добавим буквы к отчету в master, закомитим и отправим на сервер

```

PS C:\Users\Alexandr\Documents\lr1> git add .
PS C:\Users\Alexandr\Documents\lr1> git commit -m "first commit"
[master b398960] first commit
 1 file changed, 1 insertion(+), 1 deletion(-)
PS C:\Users\Alexandr\Documents\lr1> git push origin master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 322 bytes | 322.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/KingDespire/lr1
   3db7322..b398960  master -> master
PS C:\Users\Alexandr\Documents\lr1> _

```

Уберем буквы из отчества в newdev, закомитим и отправим на сервер.

```

PS C:\Users\Alexandr\Documents\lr1> git checkout newdev
Switched to branch 'newdev'
PS C:\Users\Alexandr\Documents\lr1> git add .
PS C:\Users\Alexandr\Documents\lr1> git commit -m "first commit"
[newdev 0347931] first commit
 1 file changed, 1 insertion(+), 1 deletion(-)
PS C:\Users\Alexandr\Documents\lr1> git push origin master
Everything up-to-date
PS C:\Users\Alexandr\Documents\lr1> git push origin newdev
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 314 bytes | 314.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/KingDespire/lr1
   1f75b0b..0347931  newdev -> newdev

```

Объединим ветки

```

PS C:\Users\Alexandr\Documents\lr1> git checkout master
Switched to branch 'master'
PS C:\Users\Alexandr\Documents\lr1> git merge newdev
Auto-merging lr1.txt
CONFLICT (content): Merge conflict in lr1.txt
Automatic merge failed; fix conflicts and then commit the result.
PS C:\Users\Alexandr\Documents\lr1> _

```

Разрешим конфликт. Наберём команду git diff для просмотра изменений. Внесём изменения, примем решения и отправим исправленный файл на сервер.

```

PS C:\Users\Alexandr\Documents\lr1> git checkout master
Switched to branch 'master'
PS C:\Users\Alexandr\Documents\lr1> git merge newdev
Auto-merging lr1.txt
CONFLICT (content): Merge conflict in lr1.txt
Automatic merge failed; fix conflicts and then commit the result
PS C:\Users\Alexandr\Documents\lr1> git diff
diff --cc lr1.txt
index 13f02ce,fc3b3ff..0000000
--- a/lr1.txt
+++ b/lr1.txt
@@@ -1,1 -1,1 +1,5 @@@
- Alexandr Yurievich123
- Alexandr Yuriev
++<<<<<< HEAD
++Alexandr Yurievich123
++=====
++Alexandr Yuriev
++>>>>>> newdev

```

```

PS C:\Users\Alexandr\Documents\lr1> git push origin master
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 346 bytes | 346.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/KingDespire/lr1
   b398960..11c032d master -> master
PS C:\Users\Alexandr\Documents\lr1>

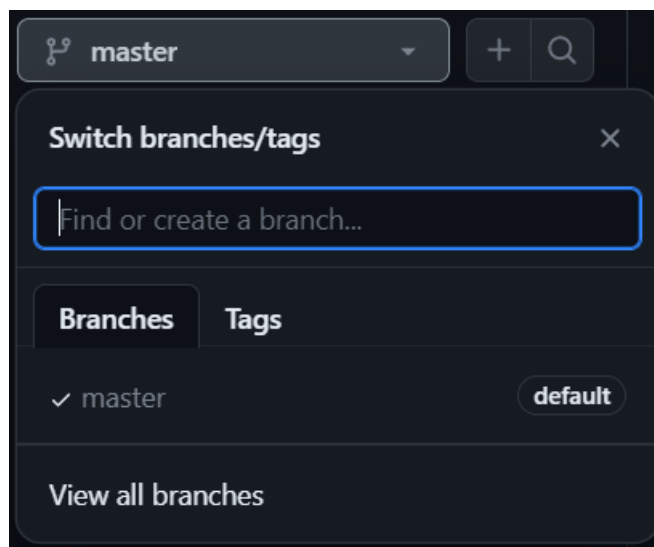
```

### Удаление веток на сервере

```

PS C:\Users\Alexandr\Documents\lr1> git push origin --delete second
To https://github.com/KingDespire/lr1
- [deleted]          second
PS C:\Users\Alexandr\Documents\lr1> git push origin --delete newdev
To https://github.com/KingDespire/lr1
- [deleted]          newdev

```



### Возврат к предыдущему состоянию

Посмотрим все комиты при помощи git log. Выберем любой коммит, на который хотим откатиться и воспользуемся командой checkout

```
PS C:\Users\Alexandr\Documents\lr1> git checkout 1f75b0b1ce1410bd775974683646249ba4779e5f
Note: switching to '1f75b0b1ce1410bd775974683646249ba4779e5f'.

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 switching back to a branch.

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

    git switch -c <new-branch-name>

Or undo this operation with:

    git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at 1f75b0b first commit
```

```
PS C:\Users\Alexandr\Documents\lr1> git branch
* (HEAD detached at 1f75b0b)
  master
```

### Исправление коммита

```
PS C:\Users\Alexandr\Documents\lr1> git add .
PS C:\Users\Alexandr\Documents\lr1> git status
HEAD detached at 1f75b0b
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   lr1.txt

PS C:\Users\Alexandr\Documents\lr1> git reset HEAD
Unstaged changes after reset:
M       lr1.txt
PS C:\Users\Alexandr\Documents\lr1> git status
HEAD detached at 1f75b0b
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:   lr1.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

### Изменение комментария

```
PS C:\Users\Alexandr\Documents\lr1> git commit --amend
[detached HEAD 8091863] lab test commit
Date: Wed May 31 23:04:45 2023 +0300
1 file changed, 1 insertion(+), 1 deletion(-)
PS C:\Users\Alexandr\Documents\lr1> _
```

### Отправка только нужных файлов на сервер

```
PS C:\Users\Alexandr\Documents\lr1> git status
HEAD detached from 1f75b0b
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:   .gitignore.txt

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

no changes added to commit (use "git add" and/or "git commit -a")
PS C:\Users\Alexandr\Documents\lr1>
```

### Совместная работа с git

```
PS C:\Users\Alexandr\Documents\project> git clone https://github.com/Diisample/Lab1
Cloning into 'Lab1'...
remote: Enumerating objects: 33, done.
remote: Counting objects: 100% (33/33), done.
remote: Compressing objects: 100% (16/16), done.
remote: Total 33 (delta 9), reused 24 (delta 6), pack-reused 0
Receiving objects: 100% (33/33), 1.54 MiB | 896.00 KiB/s, done.
Resolving deltas: 100% (9/9), done.
```